SEMANTIC PARSING OF SIMPLE SENTENCES IN UNIFICATION-BASED VIETNAMESE GRAMMAR

Dang Tuan Nguyen, Khoa Dang Nguyen, Ha Thanh Le

Faculty of Computer Science, University of Information Technology,
Vietnam National University – Ho Chi Minh City,
Ho Chi Minh City, Vietnam

{ntdang, ndkhoa, ltha}@nlke-group.net

ABSTRACT

In this research, we would like to build an initial model for semantic parsing of simple Vietnamese sentences. With a semantic parsing model like that, we can analyse simple Vietnamese sentences to determine their semantic structures that are represented in a form that was defined by our point of view. So, we try to solve two tasks: first, building an our taxonomy of Vietnamese nouns, then we use it to define the feature structures of nouns and verbs; second, to build a Unification-Based Vietnamese Grammar we define the syntactic and semantic unification rules for the Vietnamese phrases, clauses and sentences based on the Unification-Based Grammar. This Vietnamese grammar has been used to build a semantic parser for single Vietnamese sentences. This semantic parser has been experienced and the experiment results get precision and recall all over 84%.

KEYWORDS

Parsing, Semantics, Unification-Based Grammar, Taxonomy of nouns

1. INTRODUCTION

In general, parsing approaches based on Unification-Based Grammars (UBG) [1] can determine which sentence is syntactically and semantically correct. Practically, this research aims to build and implement a UBG based semantic parsing model for simple sentences in Vietnamese language.

The sentence in Example 1 is the case that a Vietnamese sentence has two correct syntactic parses but only one of them could be accepted practically.

Example 1: "Báo ăn thịt người gieo rắc kinh hoàng tại Nepal." [2]
(Translation in English: "Man-eating panther sowed terror in Nepal.")

The correct parsing of the sentence in Example 1 is introduced in Figure 1. In this figure, the main verb is "gieo rắc" (English: "to sow"), the subject of this verb is "báo ăn thịt người" (English: "man-eating panther"), and the object of this verb is "kinh hoàng" (English: "terror").

DOI : 10.5121/ijnlc.2013.2407

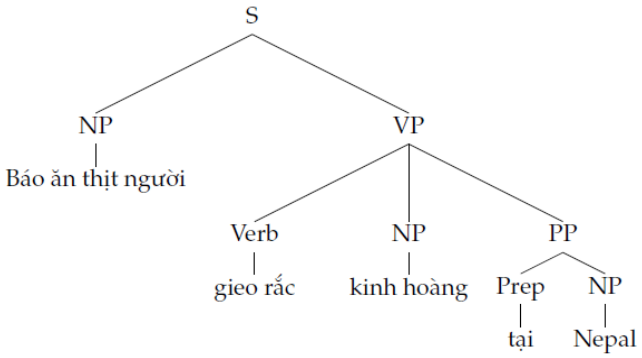

Figure 1. Syntactic parse of Vietnamese sentence in Example 1

But, when we define a CFG (Context-Free Grammar) that is used for syntactic parsing, there is a semantic mistake when the computer chooses "ăn" (English: "eat") as the main verb of the sentence, the subject of this verb is "báo" (English: "panther"), and the object of this verb is "thịt người gieo rắc kinh hoàng" (meaningless).

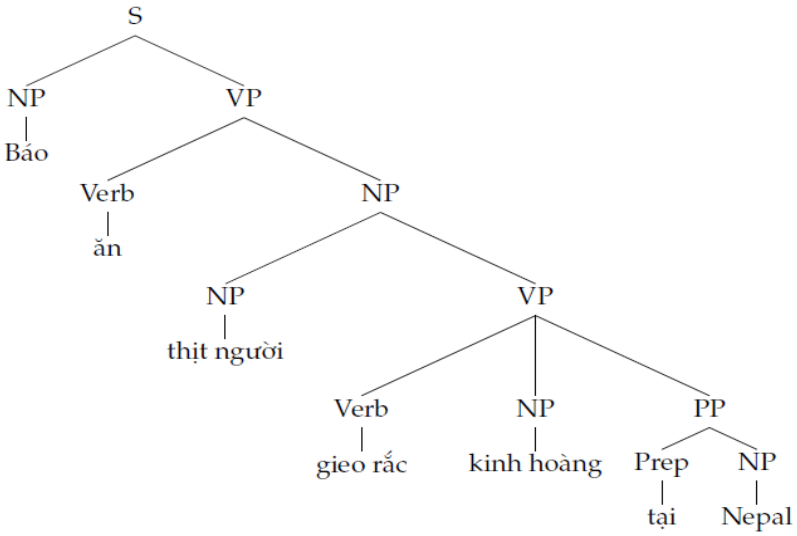

Figure 2. A semantically mistaken parse of Vietnamese sentence in Example 1

The syntactic parse in Figure 2 is correct in syntax but it's not semantically acceptable: "thịt người" (English: "human flesh") can't "gieo rắc kinh hoàng" (English: "sow terror").

There's a question here that how to implement the parsing model based on UBG to exactly analyse syntactic and semantic of simple Vietnamese sentences? Obviously, it will depend on the approach which is used to define the UBG. The defined rules in UBG are used to solve the syntactic and semantic unifications of verb and its arguments: these rules are based on the specific definition of feature structures of verbs and nouns, and methods that describe values of all of these semantic features. To describe the nominal feature structures, we constructed taxonomy of

Vietnamese nouns. In addition, this taxonomy of Vietnamese nouns is also used to resolve some problems when combining semantics between nouns in a noun phrase.

2. TAXONOMY OF VIETNAMESE NOUNS

The taxonomy of Vietnamese nouns is used to solve two questions: the syntactic and semantic unifications of verb and its arguments, and the semantic combination between the nouns in a noun phrase. Based on the linguistic theory of W. L. Chafe [3], we define a taxonomy composing groups of Vietnamese nouns which are organized in the Table 1.

Table 1. Taxonomy of Vietnamese nouns

Danh từ (Noun)				
Vật chất (Material)	Trừu tượng (Abstraction)	Tổ chức (Organization)	Địa điểm (Location)	Đơn vị (Unit)

The taxonomy of substantial nouns is presented in Table 2.

Table 2. Taxonomy of substantial nouns

Danh từ vật chất (Substantial nouns)	Hữu sinh (Biotic)	Thực vật (Vegetal)	
		Động vật (Animals)	Người (Person)
			Thú vật (Mammal)
			Bộ phận hữu sinh (Parts of biotic)
	Vô sinh (Non-biotic)	Đồ vật (Things)	
		Chất (Chemical element)	Rắn (Solids)
			Lỏng (Liquids)
			Khí (Gas)
		Phương tiện giao thông (Transport)	
		Công trình (Building)	

The taxonomy of abstract nouns is presented in Table 3.

Table 3. Taxonomy of abstract nouns

Danh từ trừu tượng (Abstract nouns)	Sự kiện (Event)	
	Hiện tượng (Phenomenon)	Hiện tượng tự nhiên (Natural phenomenon)
		Hiện tượng sinh lý (Physiological phenomenon)

	Phần mềm (Software)	
	Giác quan (Senses)	Cảm xúc (Emotions)
	Văn hóa (Culture)	
	Thuộc tính (Properties)	Tính cách (Personality)
		Tính chất (Nature)
	Công nghệ (Technology)	
	Giáo dục (Education)	Ngành học (Study)
		Bậc học (Educational level)
	Năng lượng (Energy)	

The taxonomy of other kinds of nouns is presented in Table 4.

Table 4. Taxonomy of other nouns

Tổ chức (Organization)	Địa điểm (Location)	Đơn vị (Units)	
Quốc gia (Country)	Địa danh (Place name)	Tiền tệ (Currency)	Nhiệt độ (Temperature)

3. DEFINITION OF FEATURE STRUCTURES

3.1. Feature structure of Vietnamese nouns

Table 5 presents the feature structure of nouns that we defined. In this feature structure of nouns, the features "Tiềm năng" ("Potent") and "Duy nhất" ("Unique") are used as W. L. Chafe proposed in [3], [4].

Table 5. Feature structure of Vietnamese nouns

Feature	Value	Function
SEM	A feature structure	"SEM" is a common feature for all kinds of parts of speech. Its value is the word's semantic structure that is represented in our defined form.
TYPE	Its values is extracted from our taxonomy of Vietnamese nouns	This feature contains information about noun's types. A noun can refer to many types in the taxonomy of Vietnamese nouns.
ATTR	Include three features: TIEM_NANG, DUY_NHAT, DANH_TU_RIENG	"Tiềm năng" (Potent), "Duy nhất" (Unique), "Danh từ riêng" (Proper noun) features are grouped into ATTR feature

TIEM_NANG	True or False	These features are used for creating constraints between words.
DUY_NHAT	True or False	
DANH_TU_RIENG	True or False	
SUBNOUN	Feature structure of nouns	This feature is used to create constraints between this noun and the its following noun
PRENOUN	Feature structure of nouns	This feature is used to create constraints between this noun and the its leading noun

Example 2: Feature structure of noun "Ấn Độ" (English: "India").

$$\text{Ấn Độ:}\begin{bmatrix}\text{SEM} & \begin{bmatrix}\text{NOUN} & \text{'ấn\_độ'}\end{bmatrix} \\ \text{TYPE} & \begin{Bmatrix}to\_chuc \\ dia\_danh\end{Bmatrix} \\ \text{ATTR} & \begin{bmatrix}\text{DANH\_TU\_RIENG} & \text{true} \\ \text{DUY\_NHAT} & \text{true} \\ \text{TIEM\_NANG} & \text{true}\end{bmatrix} \\ \text{SUBNOUN} & \text{null}\end{bmatrix}$$

Figure 3: Feature structure of noun "Ấn Độ"

In the feature structure in Figure 3, "sem" contains semantic information of "Ấn Độ" and its semantic structure is represented by a feature structure which has a feature named "noun" and its value is "ấn_độ". We have "type" feature that contains information about what the noun refers to. In the example, "Ấn Độ" can be a "to_chuc" (English: "organization") or a "dia_danh" (English: "place"). And "attr" is a group of feature-value pairs: danh từ riêng (English: "proper noun"), duy nhất (English: "unique"), tính tiềm năng (English: "potent"). "Ấn Độ" is a proper noun, is unique and is potent. Finally, "subnoun" contains information for creating contrains between "Ấn Độ" and its following noun.

3.2. Feature structure of Vietnamese verbs

We define the feature structure of verbs as presented in Table 6.

Table 6: Feature structure of verbs

Feature	Value	Function
SEM	A feature structure	(The same as noun's "sem" feature)
ARGNUM	1, 2 or 3	Contains information about how many arguments that the verb has
SUBJECT	Feature structure of nouns	Contains information about the verb's subject. Like "subnoun" feature of noun, by unification, we can check if a noun is suitable to be the verb's subject or not
OBJECT	Feature structure of nouns	With two-argument verb, this feature contains information about the verb's object

OBJECT1		With three-argument verb, this feature contains information about the verb's first object
OBJECT2		With three-argument verb, this feature contains information about the verb's second object
PREP	Feature structure of prepositions	With three-argument verb, this feature contains information about the preposition used with the verb

Example 3: The feature structure of verb "phát hiện" (English: "to discover").

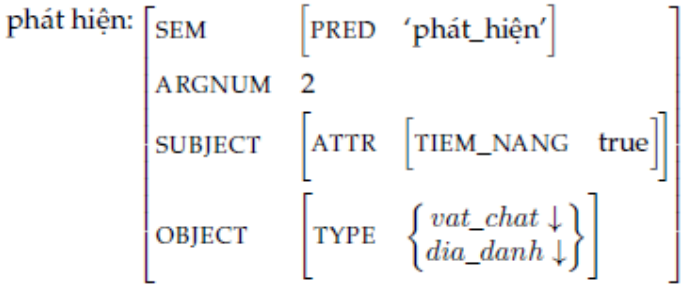

Figure 4: The feature structure of verb "phát hiện"

In Figure 4, "phát hiện" is a two-argument verb. "Ai đó phát hiện cái gì đó" (English: "Someone discovers something"). The subject of "phát hiện" must be potent, and the object must refer to "vật chất" or "địa danh".

4. UNIFICATION RULES OF PHASES, CLAUSES AND SENTENCES

In this section, we present our syntactic and semantic unification rules that we defined on GULP (Graph Unification Logic Programming) [5], [6] for semantic parsing simple Vietnamese sentences. We also applied a few linguistic theories from [7], and [8]. Our grammar is written to cover Vietnamese sentences in our corpus, composing 500 titles of scientific news collected from VnExpress online journal [2].

4.1. Unification rules of Vietnamese noun phrase

4.1.1. Simple noun phrase

Rule 1:

np_head(F) --> noun(F).

Rule 1 is applied in the case that noun phrase consists of only one noun, and that noun is also the centre of noun phrase. So that, the feature structure of the noun becomes the noun phrase's feature structure.

Rule 2:

```
np_head(F) --> noun(F1), num(F2),
        {
                F1 = sem~subnum~X,
                F2 = sem~X,
                F = F1
        }.
```

Rule 2 is applied in the case that noun phrase consists of one noun and a number follows after the noun.

Rule 3:

```
np_head(F) --> noun(Noun1F), ap(Adj1F),
        {
                Noun1F = X,
                Noun1F = sem~Y,
                Adj1F = object~X..sem~Y,
                F = Noun1F
        }.
```

Rule 3 is applied in the case that noun has one or several nouns follow after it.

Rule 4:

```
np_head(F) --> noun(F1), verb(F2),
        {
                F2 = sem~Y..argnum~1..subject~X,
                F1 = X,
                F1 = sem~subpred~Y,
                F = F1,
                F = attr~isclause~true,
                Temp = pred~sẽ,
                Y \= Temp
        }.
```

Rule 4 is applied in the case that a verb follows after the noun.

4.1.2. Complex noun phrase

Rule 5:

```
np(F) --> np_head(F).
```

Rule 5 is applied in the case that complex noun phrase consists of only one simple noun phrase.

Rule 6:

```
np(F) --> np_head(F1), pp(F2),
        {
                F1 = X,
                F2 = subject~X..sem~Y..modifier~np~true,
                F1 = sem~subprep~Y
                F = F1
        }.
```

Rule 6 is applied in the case that complex noun phrase consists of a simple noun phrase and a prepositional phrase.

Rule 7:

```
np(F) --> np_head(F1), np(F2),
        {
                F1 = subnoun~X,
                F2 = X,
                F2 = sem~Y,
                F1 = sem~subnoun~Y,
                F = F1
        }.
```

Rule 7 is applied in the case that complex noun phrase has a simple noun phrase as its centre, and one or more noun phrases follow after that are its complement.

Rule 8:

```
np(F) --> num_head(F1), np(F2),
        {
                F1 = sem~X,
                F2 = sem~X,
                F = F2
        }.
```

Rule 8 is applied when there are one or more numbers before the noun phrase, and the noun phrase is also the centre of the complex noun phrase.

4.2. Unification rules of Vietnamese verb phrase

Rule 9:

```
vp_head(F) --> verb(F).
```

Rule 10:

```
vp_head(F) --> adverb_head(F1), verb(F2),
        {
                F1 = tense~X..negative~Y..sem~Z,
                F2 = tense~X..negative~Y..sem~adverb~Z,
                F = F2
        }.
```

Rule 9 and 10 are about the adverb and the main verb component. We must have a verb in these rules and the adverb is optional. And we can have one or several adverbs.

Rule 11:

```
vp(Features) --> vp_head(VerbFeatures),
                {
                        VerbFeatures = argnum~1,
                        Features = VerbFeatures
                }.
```

In the simplest case, verb phrase consists of one main verb. In this case, we need to check if the verb has only one argument or not? We did it by checking the argnum feature of the verb.

Rule 12:

vp(Features) --> vp_head(VerbFeatures), adj(AdjF), {...}.
vp(Features) --> vp_head(VerbFeatures), adj(AdjF), pp(PpF), {...}.

Rule 13:

vp(Features) --> vp_head(VerbFeatures), np(NpFeatures), {...}.

Rule 14:

vp(Features) --> vp_head(VerbFeatures), pp(PpFeatures), {...}.

Rule 15:

vp(Features) --> vp_head(VerbFeatures), np(NpFeatures), pp(PpFeatures), {...}.

Rule 16:

vp(Features) --> vp_head(VpF), np(NpF1), prep(PpF), np(NpF2), {...}.

Rule 17:

vp(Features) --> vp_head(VpF), vp(F2), {...}.

Other unification rules for verb phrase are defined in rule 12, 13, 14, 15, 16, and 17. Rule 17 is used in the case that there are two or more verbs follow after each other. In this case, we choose the first verb as the main verb of the verb phrase. Following verbs are complement of the main verb.

4.3. Unification rules of Vietnamese prepositional phrase

Generally, the prepositional phrase consists of a preposition and a noun phrase after it. In this paper, we will show a few features that we used for linking prepositional phrase with noun phrase or verb phrase.

4.3.1. Modifier feature

The special feature "modifier" is checked when combining a noun phrase and a prepositional phrase. There are a few prepositions in Vietnamese that only is a noun's complement or is a verb's complement.

Example 4: "Phi thuyền Mỹ ra khỏi hệ mặt trời." [2]
(Translation in English: "USA Spacecraft leaves the Solar System")

prep(Features) --> [khỏi],
{ Features =
sem~prep~khỏi..
modifier~np~false
}.

In Example 4, we have marked that "khỏi" (English: "out") is not a noun's complement.

4.3.2. Prepositional phrase being noun phrase's complement

In our corpus, if a noun phrase consists of a verb in it, when combining this noun phrase with prepositional phrase, there will be many mistakes in syntactic parsing.

Example 5: "Đá từ sao Hỏa rơi xuống địa cầu" [2]
(Translation in English: Stone from the Mars fall into the Earth)

The most accurate parsing for the sentence in Example 5 is "đá từ sao Hỏa" (English: "stone from the Mars") is a noun phrase. In that noun phrase, "đá" (English: "stone") is the centre of noun phrase. And prepositional phrase "từ sao Hỏa" (English: "from the Mars") is the noun's complement. We have, "đá" (English: "stone") is the subject for the verb "rơi" (English: "to fall").

But the system can make a mistake that it considers "sao Hỏa rơi xuống địa cầu" (*meaningless*) is a clause, and the subject of the verb "rơi" (English: "to fall") would be "sao Hỏa" (English: "the Mars"). To avoid this mistake in parsing, we have a special feature named "isclause" and set its value to true for each noun phrase consists a verb in it. When combining preposition and noun phrase to make a prepositional phrase, we must check the value of "isclause" feature of the noun phrase. If its value is false, the combining is allowed.

4.4. Unification rules of Vietnamese clause

Rule 18:

```
clause(Features) --> np(NpFeatures), vp(VpFeatures),
                {
                        VpFeatures = subject~X..sem~(arg1~Y),
                        NpFeatures = X,
                        NpFeatures = sem~Y,
                        Features = VpFeatures
                }.
```

Rule 18 defines a complete clause including a noun phrase and a verb phrase.

Rule 19:

```
clause_miss_vp2(Features)--> np(F1), vp_miss_arg2(F2),
        {
                F1 = X,
                F1 = sem~Y,
                F2 = subject~X..sem~arg1~Y,
                Features = F2
        }.

vp_miss_arg2(Features) --> vp_head(VerbFeatures),
        {
                VerbFeatures = argnum~2,
                Features = VerbFeatures
        }.
```

Rule 19 is used for clause without the second argument. For example, “ong mật biết” (English: “bee knows”). “Biết” (English: “know”) is a verb with two arguments, but our above sentence has only a subject and doesn’t have an object.

Rule 20:

clause_miss_np_vp2(Features) --> vp_miss_arg2(Features).

Rule 20 is similar to the rule 19. However, this clause doesn’t have subject and the object of verb.

Rule 21:

clause_miss_np(Features) --> vp(Features).

Rule 21 is used for clause without subject.

4.5. Unification rules of Vietnamese sentences

Rule 22:

s(Features) --> clause(Features).

This rule is used for the sentence that consists of a clause.

Rule 23:

```
s(F) --> clause_miss_vp2(F1), clause(F2),
        {
                F1 = object~X..sem~arg2~S1,
                F2 = subject~X..sem~S1,
                F=F1
        }.
```

This rule is used in the case that the sentence consists of a clause without object and another clause follows after it.

Rule 24:

```
s(F) --> clause_miss_np_vp2(F1), clause(F3),
        {
                F1 = object~X..sem~arg2~S1,
                F3 = sem~S1..subject~X,
                F=F1
        }.
```

This rule is used in the case that the sentence consists of a clause without subject – object and another clause follows after it.

Rule 25:

s(F) --> clause_miss_np_vp2(F1), clause_miss_np_vp2(F2), clause(F3),

```
        {
                F1 = object~X..sem~arg2~S1,
                F2 = object~X..sem~S1,
                F2 = sem~arg2~S2,
                F3 = sem~S2..subject~X,
                F=F1
        }.
```

This rule is used in the case that the sentence consists of two clauses without subject – object and a clause at the end.

Rule 26:

```
s(F) --> clause_miss_vp2(F1), clause_miss_np_vp2(F2), clause(F3),
        {
                F1 = object~X..sem~arg2~S1,
                F2 = object~X..sem~S1,
                F2 = sem~arg2~S2,
                F3 = sem~S2..subject~X,
                F=F1
        }.
```

This rule is used in the case that the sentence consists of a clause without object with a clause without subject – object and a clause at the end.

Rule 27:

```
s(Features) --> clause_miss_np(F1), clause_miss_np_vp2(F2), clause(F3),
        {
                F1 = sem~subclause~X,
                F2 = sem~X..object~Y,
                F3 = subject~Y..sem~X2,
                F2 = sem~agr2~X2,
                Features = F1
        }.
```

This rule is used in the case that the sentence consists of a clause without subject with a clause without subject – object and a clause at the end.

The common between rules 23, 24, 25, 26, 27 is the last component of them is a clause. This clause's function is like a noun phrase and the centre of the noun phrase is the subject of this clause. So, in front of this clause is a clause without the second argument of verb phrase.

Rule 28:

```
s(F) --> np_clause(F1), clause_miss_np(F2),
        {
                F2=subject~X..sem~arg1~Y1,
                F1=subject~X..sem~Y1,
                F=F2
        }.
```

In rule 28, the first clause does a function like a noun phrase (named np_clause). Np_clause consists of a noun phrase and a verb phrase like normal clause. But its semantic presentation likes a noun phrase's semantic presentation. The np_clause acts as the subject for the clause without

subject follows after. The semantic of sentence (its predicate) depends on the clause without subject.

Rule 29:

```
s2(Features) --> np(Features).
```

Rule 29 is used in the case that the sentence has only a noun phrase.

Rule 30:

```
s2(Features) --> np(F1), clause_miss_np(F2),
        {
                (F1 = sem~noun~hành_trình; F1 = sem~noun~nguyên_nhân),
                 F1 = sem~subpred~X,
                 F2 = sem~X,
                 Features = F1
        }.
```

Rule 30 is used in the case that the sentence consists of a noun phrase and a verb phrase.

5. EXPERIMENTS

The Unification-Based Vietnamese Grammar with syntactic and semantic rules is defined on GULP [5], [6] in order to build a semantic parser.

The parser can be used to reduce incorrect parsing that is not semantically acceptable in practice. By example, in the sentence "Báo ăn thịt người gieo rắc kinh hoàng" in Example 1, the subject of verb "gieo rắc" will be described by its tiem_nang (potent) feature through the definition of the feature structure of the verb "gieo rắc":

```
verb(Features) --> [gieo, rắc],
        { Features =
                sem~(pred~gieo_rắc)..
                argnum~2..
                subject~attr~tiem_nang~true..
                object~type~X
        },
        {isa(X, cam_xuc)}.
```

In the feature structure of verb "gieo rắc", the feature tiem_nang (potent) of the subject must have the value true. The feature structure of the noun "thịt" is defined as follows:

```
noun(Features) --> [thịt],
        {Features =
         sem~noun~(thịt)..
         type~bo_phan_huu_sinh..
         attr~(
                danh_tu_rieng~false..
                duy_nhat~false..
                tiem_nang~false)
        }.
```

When combining the noun "thịt" and the verb "gieo rắc", the system will try to unify the value of "subject" feature of "gieo rắc" and the feature structure of "thịt". The unification will not be successful because "thịt" is not potent. And, the system will not return the semantically mistaken parse in Figure 2.
For evaluating the semantic parser, we used five testing datasets to perform experiments. These testing datasets are built from our corpus mentioned above which has 500 scientific news titles from VnExpress [2] online journal.

All syntactic and semantic parses are manually evaluated. Table 7 presents experiment results.

Table 7: Results of experiments

Testing datasets	Number of words in lexicon	Precision	Recall
100 sentences	317	89,31	89,37
200 sentences	480	89,02	88,99
300 sentences	661	87,73	87,74
400 sentences	895	86,16	86,20
500 sentences	1135	84,96	84,97

In general, the precisions and recalls of experiments are over 84% on four testing datasets.

6. CONCLUSION

In this paper, we focus on solving this below problems of semantic parsing for simple Vietnamese sentences: build an own taxonomy of Vietnamese nouns, define the noun and verb feature structures based on using this nominal taxonomy, and define the syntactic and semantic unification rules based on Unification-Based Grammar [1] for phrases, clauses and sentences in Vietnamese language.
The results of experiments are very appreciable on the testing datasets; however, we observe that precisions and recalls decrease a little when the number of testing sentences increases, which is an exciting challenge for our future works. For better result, we found that we need to do more research on building the taxonomy of Vietnamese nouns and the feature structures of nouns and verbs.

REFERENCES

[1] Stuart M. Shieber (1986). *An Introduction to Unification-Based Approaches to Grammar*. Microtome Publishing, Brookline, Massachusetts.
[2] VnExpress website. Available at: http://vnexpress.net/.
[3] Wallace L. Chafe (1998). Ý nghĩa và cấu trúc của ngôn ngữ (Nguyễn Văn Lai Trans.). Nxb Giáo dục.
[4] Walter A. Cook, SJ (1989). *Case Grammar Theory*. Washington, DC: Georgetown University Press.
[5] Micheal A. Covington (1994). *GULP 3.1: An Extension of Prolog for Unification-Based Grammar.* Artificial Intelligence Center, University of Georgia.
[6] GULP (Graph Unification Logic Programming). Available at: *http://www.ai.uga.edu/mc/gulp/.*
[7] Nguyễn Tài Cẩn (2004). *Ngữ pháp tiếng Việt: tiếng, từ ghép, đoản ngữ*. Đại học Quốc gia Hà Nội.
[8] Cao Xuân Hạo (2001), *Tiếng Việt mấy vấn đề ngữ âm, ngữ pháp, ngữ nghĩa*. Nxb Giáo dục.