A link grammar for Vietnamese

Nguyen Thi Thu Huong[1], Pham Nguyen Quang Anh[2]
[1] School of Information and Communication Technology, Hanoi University of Science Technology
1 Dai Co Viet, Hanoi, Vietnam, Tel 84 4 38696121, Email: huongnt@soict.hut.edu.vn
[2] NaisCorp joint stock IT Corporation, Room 101, Building B1, Nguyen Khanh Toan, Hanoi, Vietnam,
Tel 84 1234870864, Email:pnquanganh1986@gmail.com

Abstract: Link grammar is a model of formal grammar developed by Daniel Sleator and David Temperley[1]. A link grammar consists of a set of words, each of which has a linking requirement. A sequence of words is a correct sentence if there is a way to draw arcs (links) among the words to satisfy three conditions: planarity, connectivity, satisfaction. The formalism of link grammars has been used in parsing English, German, Russian, Turkish, Chinese and other languages. Several translation systems have been built to translate from English to German, Indonesian Bahasa, etc. To the extent of our knowledge, we have not seen any research on link parser for Vietnamese yet.

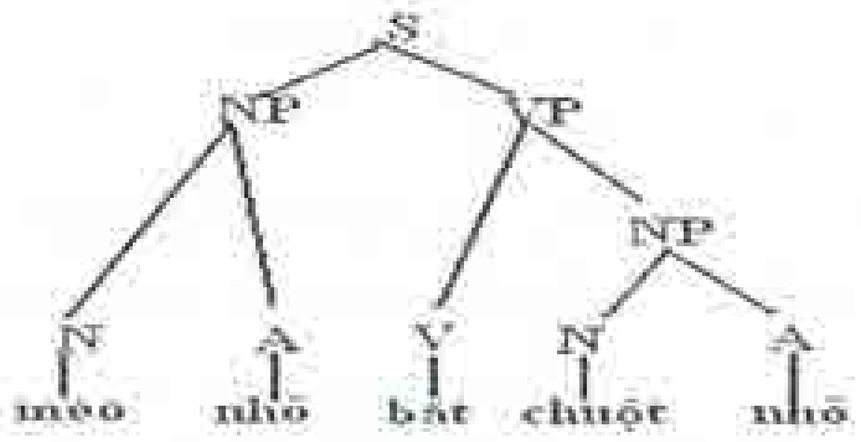
After studying the Vietnamese syntax from the computational perspective, we develop a link grammar for Vietnamese. We also present a link parser for Vietnamese. The parser has acceptable result for simple, complex sentences and compound sentences. It would be much better if we could improve the word segmenter.

Keywords: Link Grammar, Parser, Natural Language Processing, Vietnamese Syntax.

I. INTRODUCTION

As well known, a classical method of describing the syntactic structure of natural sentences is phrase-structure tree (PS- tree).

A PS-tree of a natural-language expression shows which items of word forms or phrases combine with which other items to form tight units of a higher order. A sentence is cut into (generally two) major constituents, each of which is subsequently cut in its turn, etc. Symbols like NP, VP, N, PP, etc. appear in PS-trees as labels on nodes. Most of nodes do not correspond to actual word forms of a sentence. Figure 1 demonstrates a PS-tree for a Vietnamese sentence "Mèo nhỏ bắt chuột nhỏ" (small cats catch small mice).



(small cats catch small mice)

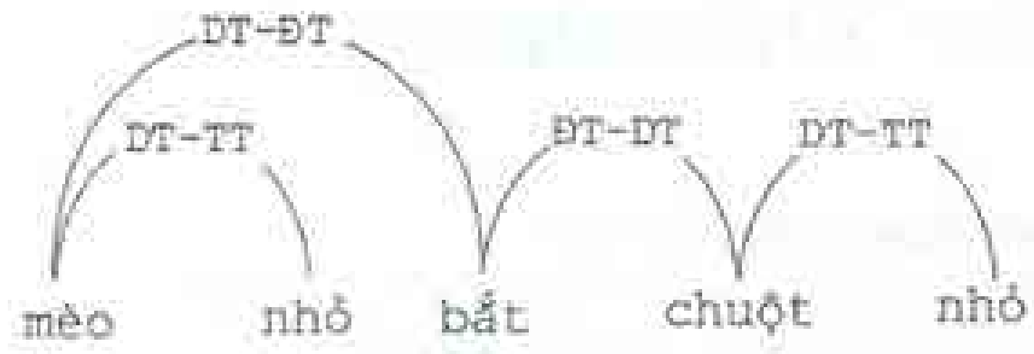
Figure 1. A Phrase Structure Tree

This approach does not allow a natural representation of relations between word forms. For rule based machine translation, if the PS approach is used in parsing, it is very difficult to build the destination sentence when the structures of sentences of the source language and the destination language are different. To cover all grammar phenomena of a language, the set of context free rules is much larger than the set of link formulas.

A link grammar, which is considered closer to human intuition, offers impressive characterizations.

A link parser shows which item is related to which other item and in what way. The output of a link parser reveals the structure of an expression in terms of links between its actual elements.

In a PS tree, properties of an item cannot properly represent its actual syntactic role in a text. But in link grammar model, syntactic roles are indicated explicitly by special labels. Figure 2 shows a linkage of sentence "mèo nhỏ bắt chuột nhỏ" mentioned above.



(small cats catch small mice)

Figure 2. Output of a link parser for sentence "mèo nhỏ bắt chuột nhỏ"

Comments of figure 2:

ĐT-ĐT is the typical connector between a noun and a verb (like connector subject – verb in English)

ĐT-TT is the connector between a noun and an adjective. In Vietnamese there is no verb between a noun and an adjective.

ĐT-ĐT is the typical connector between a verb and a noun (like connector verb-object in English)

According to D.Sleator and D.Temperley [1], a link grammar consists of a set of words, each of which has a linking requirement. A sequence of words is a correct sentence if there is a way to draw arcs (links) among the words to satisfy the following conditions:

Planarity: The links do not cross.

Connectivity: The links suffice to connect all the words of the sequence together.

Satisfaction: The links satisfy the linking requirements of each word in the sequence.

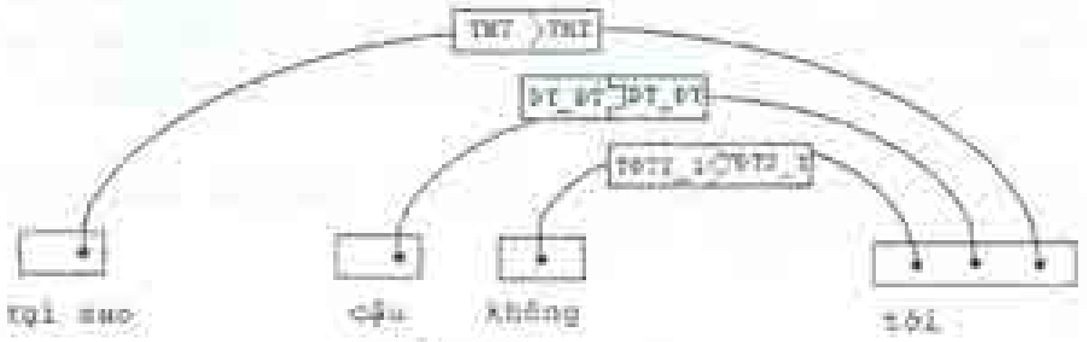
The linking requirement for each word in the dictionary is expressed as a formula involving the operators & and *or*, parentheses, and connector names. The + or - suffix on a connector name indicates the direction (relative to the word being defined) in which the matching connector (if any) must lie. The & of two formulas is satisfied by satisfying both the formulas. The *or* of two formulas requires that exactly one of its formulas be satisfied.

For example, in sentence, **"Tại sao cậu không tới?"** *(Why do not you come?)* the verb **"tới"**(*come*) has three left linking requirements, that are "THT-"(the connector of an interrogative word and the predicative), "ĐT_ĐT-" (similar to the subject link of other languages), and "TĐT2_1-"(the connector of a negative adverb and a verb or an adjective). On the other hand, **"Tại sao"** *(Why)* links to word **"tới"** *(come)* with the THT+ link, **"cậu"** *(you)* links to word **"tới"** *(come)* with the ĐT_ĐT link. **"không"** *(don't)* links to word **"tới"** *(come)* with the TĐT2_1+ link. Table 1 encodes the linking requirements of the sentence:

Table 1. Linking requirements of each word of sentence "Tại sao cậu không tới?"

words	formula
tại sao	THT+
cậu	ĐT_ĐT+
không	TĐT2_1+
tới	(THT- or ()) & (ĐT_ĐT-) & (TĐT2_1- or())

The diagram in Figure 3 below shows how the linking requirements are satisfied in sentence "Tại sao cậu không tới?"



(Why do not you come?)

Figure 3. Linking requirements are satisfied in the sentence "Tại sao cậu không tới"

D.Sleator and D.Temperley [1] also introduced an efficient way to express rules of a link grammar, namely disjunctive form. Each word of a language associates with a set of disjuncts.

A disjunct is a set of connector names that constitutes an instance of satisfying the requirements of a word. A disjunct has two parts: the left list and the right list. The left list consists of the connectors with "–" sign (left linking requirements), whereas the right list consist of the connectors with the "+" sign (right linking requirements).

A disjunct is denoted as: $((L_1, L_2, L_3 \dots L_x)(R_y, R_{y-1}, R_{y-2} \dots R_1))$. L_i denotes a left linking requirement and Rj denotes a right linking requirement. Either "x" or "y" can be zero. In previous formula, the word connected to current word with "L_1" link is closer than to the word with "L_2" link. Similarly, the word connected to current word with "R_y" link is closer than to the word with "R_{y-1}" link.

A formula can be translated into a set of disjuncts by enumerating all the ways that the formula can be satisfied. The following disjuncts can be constructed from formula *(THT- or ()) & DT_ĐT- & (TĐT2_1- or TĐT2+)*:

((THT, DT_ĐT,TĐT2_1)(TĐT2))

((DT_ĐT,TĐT2)(TĐT2))

In the reverse direction, to translate a set of disjuncts into a formula, all the disjuncts should be combined with the "or" operand.

English parsers have been developed based on a link grammar for English [1]. The latest parser for English was released in February 2011. It used a dictionary of about 60000 word forms and 1899 rules for 12 parts of speech.

Besides English, link parsers were built for other languages. A project of building Chinese link parser has been developed since 2004 [22]. Chinese hass problems similar to Vietnamese: tense, voice and some other grammatical categories are expressed by words, not by morphological transformation.

Link parsers are applied to many fields. Using Link Parser to extract information from texts is very popular in recent years. In [15] a link parser was used to parse biological documents and parse trees were saved in a database. Molecular events were extracted from that database. In [17] information of protein interaction from unstructured texts were extracted. In [18], a method for generating a Discourse Representation Structure (DRS) semantic representation from the output of a link grammar parser was presented. [21] proposed an extended model of Link Grammar: Categorical Link Grammar to generate knowledge piece in an emergency decision-making support system.

Link grammars show directly the connections of words in graphs. That model can be useful for a direct translation from syntax to semantics. Some machine translation system based on link grammar model have been developed for Asian Language, for example, N. Zamin et al. [19] have developed a translation system from English to Indonesian Bahasa using Annotated Disjunct (ADJ) model. Although it was a rule-based system, it could be applied effectively to less-resourced languages in South East Asia where large corpora are not available. [16] also used the link grammar model to build a machine translation system from English to Indian ethnic languages such as Panini, Kannada.

II. VIETNAMESE PARSING

Vietnamese language is affected by both Chinese and French: use an extended Latin alphabet but whitespaces are not used to identify the word boundaries.

Vietnamese words are categorized into the following parts of speech: noun, verb, adjective, pronoun, numeral, adverb, conjunction, preposition, determiner. Not like English where each sentence must contain at least one verb, in Vietnamese sentences, the main part of a predicative may be a verb or an adjective, for example, in sentence "**Lan lau sàn nhà**" *(Lan cleans the floor)*, the main part of predicative is the verb **"lau"** *(clean)* while in the sentence **"Cô ấy đẹp"** *(She is beautiful)*, the subject is **"cô ấy"** *(she)* and the main part of predicative is the adjective (without verb) **"đẹp"** *(beautiful)*.

The general word order in Vietnamese is Subject-Verb-Object (SVO). But several other orders are accepted.

There are many approaches to Vietnamese parsing, almost based on the Context Free Grammar model. At present, model of Lexical Probabilistic Context Free

Grammar (LPCFG) is very popular. Because LPCFG is a statistical-based model, parsing quality highly depends on the size of its training corpus. Unfortunately, corpora of Vietnamese are not large enough to cover all the characteristics of Vietnamese.

[1] proved that any language generated by a Context Free grammar is accepted by a link grammars and vice versa.

Since link grammars are completely lexicalized, results of a link parser may represent some semantic information, it would be useful in translation systems. For instance, using ADJ [19], the problem of word disambiguation may be solved.

Vietnamese is suitable for link parsing because the order of words is significant. Forms of Vietnamese words are never changed, so the morphologic parser is not required. However, instead of having prefix or suffix, to express tense, form, comparative, plural, some words are added (**"đã", "từng"** to express past tense, **"sẽ"** for future tense)

III. BUILDING A LINK GRAMMAR FOR VIETNAMESE

We have built a link grammar of more than one hundred definitions that capture many phenomena of the Vietnamese grammar. It handles: noun-verb agreement, questions, adjective, complex, different types of nouns and many other things. In this section, we introduce rules for typical parts of Vietnamese speech.

A. Noun phrase

Noun phrase is constructed from 3 parts: the central noun, preceding and succeeding auxiliary elements.

The third preceding auxiliary element (e.g. **"cái"** *(unit)*) stands immediately before central noun. This element has a connector TDT3+ and nouns have a connector TDT3-.

For example,

bàn *(table)*, **ghế** *(chair)*: {TDT3-}

cái *(unit)*: TDT3+

The second preceding element has a connector TDT2+ and nouns has a connector TDT2-.

We got:

những, các (the words stand for the plural in Vietnamese), **mỗi** (each) , **mọi** (all), **từng (every)**: TDT2+

Numerals come before the central noun and have a connector ST_DT+.

một *(one)*, **hai** *(two)*, **ba** *(three)*, **bốn** *(four)*: ST_DT+

The first preceding element connects to the central noun via a connector named TDT1+.

tất cả, tất thảy, toàn bộ, toàn thể *(all)*: TDT1+

Combining these features discussed above, we have the following rules for nouns:

bàn *(table)*, **ghế** *(chair)*: {TDT3-} & {TDT2- or ST_DT-} & {TDT1-}

Consider the order of the elements. The third element is closest to the central noun; hence, TDT3- is the farest to the left. In contrast, the TDT1- is the last one to the right because the first element comes before all other elements. Noun phrase may not have the second preceding element and numerals at same time, thus we have operand or between two connectors TDT2- and ST_DT-. Also note that all these connectors are optional.

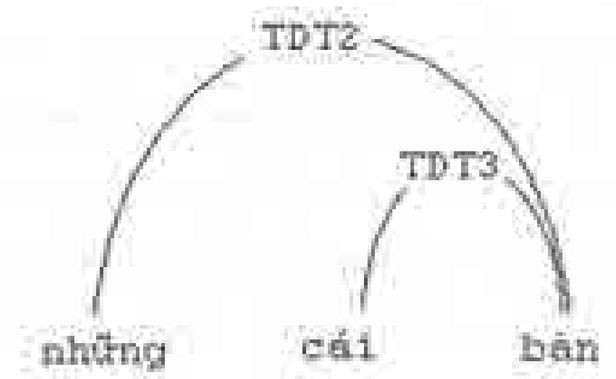
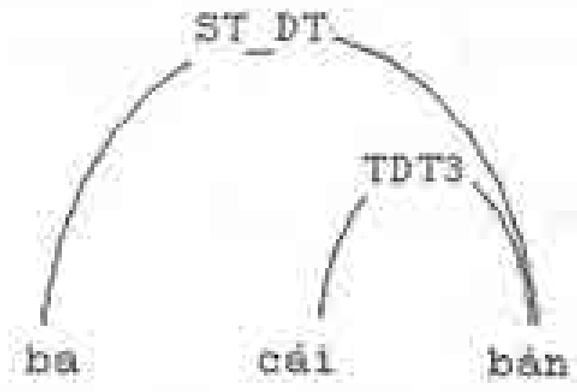


Figure 4. Linkage in phrase "những cái bàn"

With the above rules, the following phrases are recognized:

những cái bàn *(tables)*, **từng cái ghế** *(every chair)*

ba cái bàn (three tables), **hai cái ghế** (two chairs)



(three tables)

Figure 5. Linkage in phrase "ba cái bàn"

tất cả những cái bàn *(all the tables)*

toàn bộ mọi cái ghế *(all the chairs)*

tất cả ba cái bàn (*three tables as a whole*)

The first succeeding auxiliary element is a noun, supplying meaning to the central noun. This element has a connector SDT1+ and the central nouns have the corresponding connector SDT1-. Once the central noun has the connector SDT1-, it may not connect to any preceding or succeeding elements. Hence, we got these rules for noun phrase:

bàn *(table)*, **ghế (chair)**, **giường** *(bed)*, **lò xo** *(spring)* : SDT1- or ({TDT3-} & {TDT2- or ST_DT-} & {TDT1-} & {SDT1+})

These rules get rid of the wrong cases where other preceding and succeeding elements are considered supplying meaning to this first succeeding element (like the example above, **"cái"**(unit) cannot be linked to **"lò xo"**(spring)).

(the spring bed)

Figure 6. Links in phrase " cái giường lò xo"

The third succeeding element (**"bằng"***(made of)*, **"về"***(about)*) plays a role as a link between the central noun and another noun. This element connects to the central noun by a SDT3- and to the other noun by a GT_DT+, which is a connector between a preposition and the following noun (we will mention this connector later).

bằng *(made of)*, **về** *(about)*: SDT3- & GT_DT+

Like the third succeeding element, the fourth one (**"của","ở"** *(of- possessive case))* has a connector GT_DT+ and a connector SDT4- linking to the central noun.

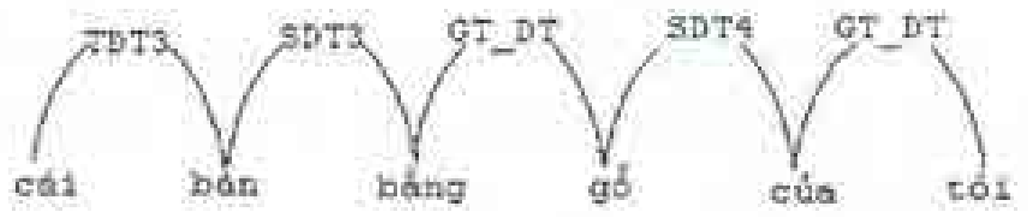
của, ở *(of- possessive case)*: SDT4- & GT_DT+

Entries in the sixth group of succeeding elements have the connector SDT6-

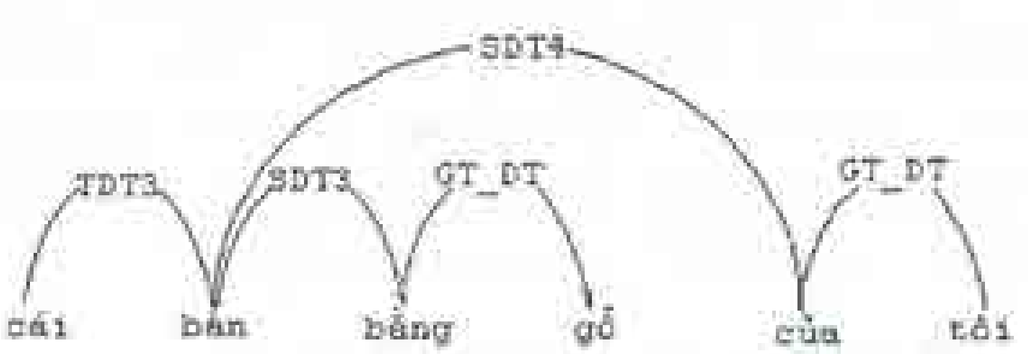
ấy, đấy *(such)*, **kia** *(that)*, **này** *(this)*: SDT6-

The central noun will have connectors SDT3+, SDT4+, SDT6+ and GT_DT-.

Bàn *(table)*, **ghế** *(chair)*, **giường** *(bed)* , **lò xo** *(spring)*: SDT1- or ({TDT3-} & {TDT2- or ST_DT-} & {TDT1-} & {SDT1+} & {SDT3+} & {SDT4+} & {SDT6+} & {GT_DT-})



(my wooden table)



(the table made of my wood)

Figure 7. Linkage in phrase "cái bàn bằng gỗ của tôi"

Consider the case in figure 7 where the noun phrase has both the third and fourth preceding element:

We have ambiguties here; in the first case **"của"***(of)* is connected to **"gỗ"***(wooden)*, while in the second case it is connected to **"bàn"***(table)*.

B. Verb phrase

Verb phrases have the six preceding groups.

The first group has the connector TĐT4+:

cũng *(also)*, **còn** *(still)* , **vẫn** *(yet)*, **cứ** *(keep on)*: TĐT4+

The second group describes the continuousness of the action:

đã, đang, sẽ, vừa, mới, sắp, sắp sửa (words for past, present, future tense in Vietnamese): TĐT3+

(I extremely expect)

Figure 8. Links in phrase " tôi mong chờ lắm"

The fifth group, which shows the negation to the action, is linked to the central noun by a connector named TĐT2_1+.

không, chẳng, chưa *(not)*: TĐT2_1+

There are several verbs which may connect to the entries in the fourth group (**"rất"**, **"hơi"**, **"khá"** *(very, rather)*,...). This group has the connector TĐT2_2+. Note that the central verb cannot have both TĐT2_1- and TĐT2_2- at the same time.

rất, hơi, khá *(very, rather)*: TĐT2_2+

Entries in the third group describes the frequency of the action., they have connector TĐT1+.

thường *(usual)*, **hay** *(frequent)*: TĐT1+

The sixth group supplies the meaning of recommend or demand to the verb. When the verb combines with this group, it cannot connect with any other auxiliary elements. Hence, verb has a connector TĐT5- combined with other connectors by operand "or".

đừng, chớ *(do not)*: TĐT5+

Rules for verbs:

đi *(go)*, **đứng** *(stand)*, **chạy** *(run)*, **làm** *(do)*: ({TĐT1-} & {TĐT2_1- or TĐT2_2-} & {TĐT3-} & {TĐT4-}) or {TĐT5-}

Words like **"lắm"**, **"quá"***(very)* go behind some verbs which are able to connect to such words.

lắm, quá *(very)*: SĐT-

These verbs have rules:

mong *(expect)* , **mong chờ** *(wait)*, **tiếc, hối tiếc** *(regret)*, **sợ, lo lắng** *(worry)*, **nhớ** *(miss)*, **mơ** *(dream)*, **ước** *(wish)*, **quên** *(forget)*: {TĐT2_1- or TĐT2_2-} & {TĐT3-} & {TĐT4-} & {SĐT+}

C. Adjective phrase

Preceding and succeeding elements in adjective phrases are nearly the same as those in verb phrases except adjectives cannot come with **"đừng"**, **"chớ"***(do not)*. Therefore, these auxiliary elements of verb phrase are supplied with corresponding connectors to adjectives:

cũng *(also)*, **còn** *(still)*, **vẫn** *(yet)*, **cứ** *(keep on)*: TĐT4+ or TTT4+

đã, đang, sẽ, vừa, mới, sắp, sắp sửa *(words for past, present, future tense in Vietnamese)*: TĐT3+ or TTT3+

không, chẳng, chưa *(not)*: TĐT2_1+ or TTT2_1+

rất*(very)*, **hơi, khá** *(rather)*: TĐT2_2+ or TTT2_2+

thường *(usual)*, **hay***(frequent)* : TĐT1+ or TTT1+

lắm, quá *(much)* : SĐT- or STT-

The following rules have been built for adjectives:

tốt *(good)*, **đẹp** *(beautiful)*, **đỏ** *(red)*, **xanh** *(green)*: {TTT1-} & {TTT2_1- or TTT2_2-} & {TTT3-} & {TTT4-} & {STT+}

Here, connector TTT1 links with a frequency adverb ("thường", "hay"...), TTT2_2, STT links with an adverb of degree on the left ("rất", "hơi"...) or on the right (lắm, quá), TTT2_1 links with a negative adverb ("không", "chẳng"...)

D. Connections between a noun phrase and a verb phrase

In the simple sentence structure S + V, in which S is a noun phrase and V is a verb phrase, the central noun and the central verb are linked together. Nouns have another connector DT_ĐT+ and verbs have a connector DT_ĐT-.

Nouns may come after verbs playing the role of the object of the action. Hence, nouns and verbs have connectors ĐT_DT+ and ĐT_DT-, respectively. Some verbs (**"khóc"***(cry)*, **"cười"***(smile, laugh)*, **"ngủ"***(sleep)*,...) which do not require objects will not have this connector.

Nouns and verbs may be connected together indirectly in the form: V + preposition + O. In this case, we add connector ĐT_GT+ to verbs and ĐT_GT- to prepositions. Prepositions may be linked to nouns via the connector GT_DT mentioned above.

So far, we got these rules for nouns, verbs and prepositions:

Nouns, pronouns:

tôi *(I)*, **bạn** *(you)*, **bàn** *(table)*, **ghế** *(chair)*, **giường** *(bed)*, **lò xo** *(spring)*: SDT1- or ({TDT3-} & {TDT2- or ST_DT-} & {TDT1-}& {SDT1+} & {SDT3+} & {SDT4+} & {SDT6+} & {GT_DT- or ĐT_DT- or DT_ĐT+})

Verbs:

học *(learn)*, **làm** *(do)*, **ăn** *(eat)*, **đọc** *(read)*, **viết** *(write)*, **nghe** *(listen)*, **gặp** *(meet)*, **thấy** *(see)*: (({TĐT1-} & {TĐT2_1- or TĐT2_2-} & {TĐT3-} & {TĐT4-}) or {TĐT5-}) & {ĐT_ĐT-} & {ĐT_DT+} & {ĐT_GT+}

Prepositions:

bằng *(made of)*, **về** *(about)* : (SDT3- or ĐT_GT-) & GT_DT+

***ở** (in)*, **dưới** *(under)*, **trước** *(in front of, before)*, **sau** *(behind, after)* : (ĐT_GT- or SDT4-) & GT_DT+

từ (from), **vào** (to, into), **theo** (according to) , **bởi** (because), **cùng** (together) ĐT_GT- & GT_DT+

E. Connections between a noun phrase and a adjective phrase

In Vietnamese, adjectives may go after nouns to form a structure like S + V. Nouns and pronouns have another connector called DT_TT+ and adjectives have a connector DT_TT-.

Tôi *(I)*, **bạn** *(you)*, **bàn** *(table)*, **ghế** *(chair)*, **giường** *(bed)*, **lò xo** *(spring)*: SDT1- or ({TDT3-} & {TDT2- or ST_DT-} & {TDT1-}& {SDT1+} & {SDT3+} & {SDT4+} & {SDT6+} & {GT_DT- or ĐT_DT- or DT_ĐT+ or DT_TT+})

tốt *(good)*, **đẹp** *(beautiful)*, **đỏ** *(red)*, **xanh** *(green)*: {TTT1-} & {TTT2_1- or TTT2_2-} & {TTT3-} & {TTT4-} & {DT_TT-} & {STT+}

F. Connections between a verb phrase and a adjective phrase

Adjectives will play the role of adverbs when they stand after the verbs. Verbs have ĐT_TT+ connector and adjectives (adverbs) have ĐT_TT- connector.

G. Complex Sentences

In Vietnamese, complex sentences includes several clauses, connecting with conjunctions, such as **"bởi vì"** *(because)*, **"tuy nhiên"** *(however)*, **"nhưng"** *(but)* etc. or without conjunction (commas, parentheses).Vietnamese conjunctions have connector CL+ connecting with succeeding clause and connector EV- connecting with the predicative of preceding

clause. Parsing the complex sentences, of which clauses connecting with commas may lead to an ambiguity, because comma has many grammatical functions. Only in case **bởi vì** stands at the beginning of the sentence, and two clauses are connected with a comma, connector PH+ for comma is added

The following rules have been built for conjunctions:

tuy nhiên *(however)*, **nhưng** *(but)*, **nên** *(so)*, **cho nên** *(hence)*, **bởi vì** *(since)*, **vì** *(because)*: CL+ & (EV- or QHT-)

vì *(since)*, **vì** *(because)*,**dù** *(although)*: CL+ & {PH+} & (EV- or (CO+ or QHT+))

Figure 9 shows the result of parsing a sentence with comma separated two clauses:

In [1] the phenomenon of "and" conjunction has been introduced. Authors also have some problems with the equivalent conjunction **"và"** in Vietnamese. The application of solution of [1] with English "and" conjunction to parse sentences with "và" conjunction has given similar results.

(Because you write very quickly, your handwriting is very bad)

Figure 9. Result of parsing "bởi vì cậu viết nhanh quá, chữ của cậu rất xấu"

IV. VIETNAMESE LINK PARSER

A. The dictionary

A dictionary of about one hundred rules and over than one thousand items has been built. In the dictionary, Vietnamese words are categorized into subgroup of each part of speech, each subgroup has some link formulas. Here are the images of two main parts in the dictionary (see Figures 10 and 11)

```
1: (SDT1- & (SDT1+)) or ((TDT3-) & (TDT2- or
ST_DT-) & (TDT1-) & (SDT1+) & (SDT2+) & (SDT3+) &
(SDT4+) & (SDT6+) & ((MA+) & SDT5+) & (DT_LA+ or
GT_DT- or DT_DT- or DT_DT+ or DT_TT+ or DT_THI+ or
DT_LA+ or LA_DT-)) & (SS_DT- or DT_SS+))

2: (SDT1- & (SDT1+)) or ((TDT1-) & (SDT1+) &
(@SDT2+) & (SDT3+) & (SDT4+) & ((MA+) & SDT5+) &
(SDT6+) & (SS_DT- or DT_LA+) & (GT_DT- or DT_DT-
or DT_DT+ or DT_TT+ or DT_THI+) & (SS_DT- or
DT_SS+))

3: (SDT1- & (SDT1+)) or ((SDT1+) & (@SDT2+) &
(SDT4+) & (((MA+) & SDT5+) & (SS_DT- or DT_LA+)) &
(((MA+) & SDT5+) & (GT_DT- or DT_DT- or DT_DT+ or
@DT_TT+ or DT_THI+)) & (SS_DT- or DT_SS+))

4: TDT3+

5: TDT2+

6: TDT2+ or TDT2_1+ or TTT2_1+ or TST2_1+

7: TDT1+

8: (((TST2_1- or TST2_2-) & ((TST4-) & TST3-) &
(TST4-)) & (TST-)) & ST_DT+

9: TST+

10: (SDT3- or DT_GT- or TT_GT-) & GT_DT+
```

Figure 10. Rules in the dictionary

```
# Trước danh từ 3
cái | chiếc | tấm | bộ | con | ngôi | sự : 4
# Trước danh từ 2
những | các | mỗi | mọi : 5
từng : 6
# Trước danh từ 1
toàn thể | tất thảy | tất cả | toàn bộ | bất kỳ : 7
```

Figure 11. Groups of words corresponding above rule

We used built-in data structure Hash Map of Java to represent our dictionary. This is a hash table storing elements in key/value pairs. Keys are words in our dictionary, values are rules corresponding to the words. Keys are hashed so that there are no two pairs with the same key. If we put two rules of a same word to the hash map, the second rule will replace the first one. Besides HashMap, Java has a same data structure, HashTree. In HashTree, elements are stored in a tree. Our dictionary is not really big, therefore HashMap and HashTree are efficient to present the

dictionary. There is no significant difference in performance when using these data structures. We prefer HashMap to HashTree because we do not need to visit the keys in sorted order.

B. Pre-processing

Because a Vietnamese word may include several morphemes separated by spaces, word segmentation is required before parsing. The system uses JvnSegmenter, a word segmenter built by authors of [7]. The result of the word segmenter has a remarkable affection to the result of our parser.

Another important tool is the POS-tagger. Since forced parts of speech may cause errors in parsing, atpresent, instead of using a POS-tagger, we let the parser recognize grammatical role of each word through links.

C. Pruning performance

As mentioned above, we use the algorithm proposed by [1] to parse Vietnamese sentences with rules of Vietnamese grammar. Disjuncts are built by browsing rules of the dictionary. List of disjuncts for each word is stored in a link list.

Compare to English, number of disjuncts in Vietnamese is quite high, because there are so many words that have link requirements with nouns, pronouns and verbs, for instance: words express verb tense: **đã, đã từng, từng** (words for the past tense), sẽ, chuẩn bị (words for the future tense), **đang, hiện** (words for the present tense), words express plural/singular **những, nhiều, một**, etc.

We have applied both pruning methods[1]: pruning and power pruning afterward. Both processes (pruning and power pruning) terminate when a pass changes no nearest- word field and deletes no disjunct. With the collection sentences in our sample set, we have not seen any sentence needing more than five passes. In our practical experiments, it was shown that using these techniques makes our program more efficient. The sizes of the problem and therefore the running time are significantly reduced.

The following figure shows the number of disjuncts after each iteration of pruning when the sentence **"chúng tôi muốn giành các danh hiệu"** (*we want to get titles*) was analyzed.

[chúng tôi] [muốn] [giành] [các] [danh hiệu]

Power

	59138	44010	131330	1	59138
Left to Right:	59138	128	1536	2	4096
Right to Left:	650	12	24	2	4096
Left to Right:	650	12	24	5	4096
Power Prune					
Left to Right:	6	9	20	1	5
Right to Left:	6	9	20	1	5
Left to Right:	6	9	20	2	5

Figure 12. Number of disjuncts after each iteration of pruning sentence "chúng tôi muốn giành các danh hiệu"

As mentioned above, number of initial disjuncts was quite high, from one to hundreds thousands disjuncts, for instance, 59138 for each noun and 131330 for each verb. After two pruning phrases, this number became only five to twenty. Table 2 displays percentage reduction rate of number of disjuncts, gained in our test.

D. Disambiguation

Following the model of link grammar, a sentence is grammatical correct if it satisfies the conditions mentioned in I of this paper. A sentence may have many parsing results. The number of results for compound, complex and compound complex sentences may increase strongly.

To solve the problem of disambiguation with the word "**và**" (and) and the following commas, firstly we use the method shown in [1] with fat connectors (F). A fat connector is an ordered sentence of ordinary connectors. A priority of 0, 1, 2 is given to each connector, too. Normally, all ordinary connectors are given 0, the fat connectors on words are given 1, and the fat connectors on "và" are given 2(may be 1). In order for two connectors to match, they must match according to the normal criteria, and their priorities must also be compatible: 0 is compatible with 0, 1 is

compatible with 2, and 2 is compatible with 1. No other priorities are compatible.

Table 2. Percentage reduction rate of number of disjuncts

	Noun, pronoun	Normal verb	Verbs: "nói"(say), "kể"(tell), etc.	Adjective	Words express tenses ("đã", "đang", "sẽ")
Number of disjuncts	59138	131330	311042	153216	3
Reduction rate after pruning (%)	98.6	99.7	99.9	99.6	66
Reduction rate after power pruning (%)	40.2	15.7	29.1	74.2	0

So, to parse the sentence **"Tôi đã gặp anh và chị của cậu"** (*I met your brother and sister*), we got the correct result:

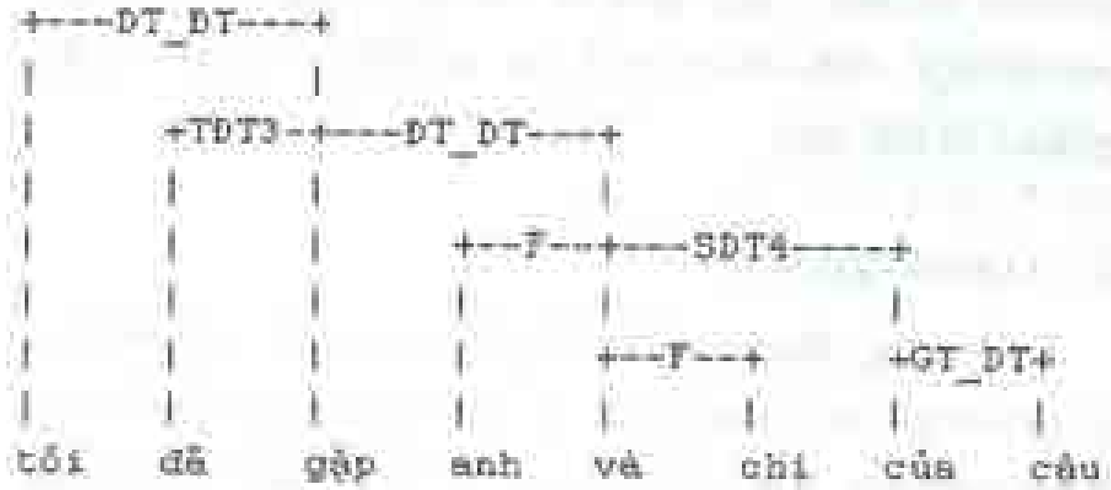
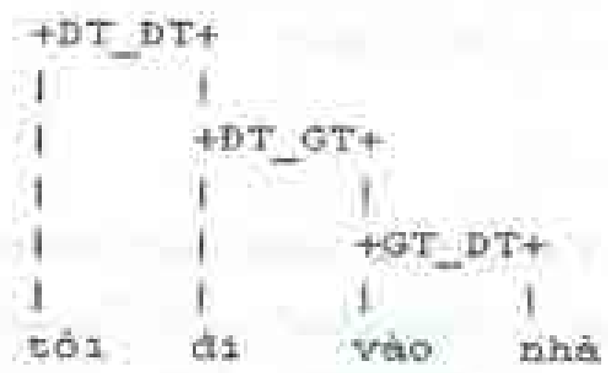


Figure 13. Parsing result of "tôi đã gặp anh và chị của cậu"

Complex compound sentences, like **"Vì trời mưa rất to và gió rất mạnh nên tôi nghỉ"** (*Because it rained very heavily and the wind was very strong, I was absent*) may have many results. To parse such sentences, relationships between clauses must be recognized and each clause must be parsed separately. It requires a more complex model.

(I go home)

Figure 14. Incorrect result of parsing "tôi đi vào nhà"



(I go home)

Figure 15. Correct result of parsing "tôi đi vào nhà"

For simple sentences, a statistical approach to calculate probability of links is applied. For example, with sentence **"tôi đi vào nhà""**, the correct result gets a higher probability than the incorrect one because link GT_ĐT appears more frequently than link ĐT_ĐT.

E. Experimental results

The following pictures (Figure 16, 17, 18) show the result of parsing two simple sentences and one compound sentence. The first simple sentence includes a comparative, and the second simple sentence uses a special verb of Vietnamese **"là"** (similar to the verb *to be*). The complex sentence use conjunctions **"bởi vì"** *(because)*, **"nên"***(so)*

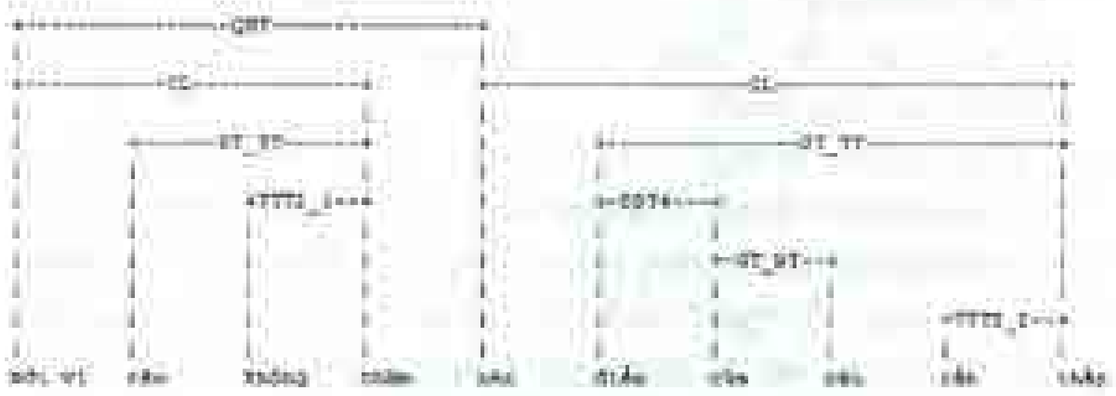
(He looks more like a teacherthan an engineer)

Figure 16. Result of parsing "anh ấy giống giáo viên hơn là kỹ sư"

We collected sentences from papers on the Internet concerning the literature, sports, news. Before testing, we removed all the punctuations from the sentences and removed the incorrect sentences from word segmenter. Sentences with commas are out of our scope. The total of sentences is 100.

(Yesterday's banquet was a great success)

Figure 17. Result of parsing
"bữa tiệc hôm qua là một thành công lớn"



(Because you were not diligent, you marks are low)

Figure 18. Result of parsing
"bởi vì cậu không chăm nên điểm của cậu rất thấp"

The following table 3 shows the results of our test.

Table 3. Result of tests on sample set

Number of sentences	100
Average number of words in each sentence	7.46
Number of correct sentences	72
Percentage of correct parse	72%

In [6], sentences were broken up into smaller ones, the average number of words is 4.53. In our test, we still keep complex sentences (without comma) and try to parse them as a whole.

Table 4. Number of sentences in each paper

No	Title	Number of sentences
1	Fabregas thề khiến MU phải đời ngày đăng quang	29
2	Triều Tiên hủy mọi thỏa thuận với Hàn Quốc	10
3	Kaka không đến Chelsea cùng Ancelotti	16
4	Loài người có thể 'tuyệt chủng'	13
5	Khi con mồi phục kích kẻ đi săn	22
6	Những con khỉ phát sáng đầu tiên trên thế giới	21
	Total	111

Table 5. Parsing results of above papers

Paper No	Number of incorrect sentences		Number of correct sentences
	by word segmenter	by link parser	
1	13(44.8%)	10 (34.5%)	6 (20.7%)
2	5 (50%)	3 (30%)	2 (20%)
3	7 (43.8%)	6 (37.5%)	3 (18.7%)
4	6 (46.2%)	5 (38.5%)	2 (15.3%)
5	11 (50%)	5 (22.7%)	6 (27.3%)
6	8(38.1%)	6(28.6%)	7(33.3%)
Total	50(45%)	35 (31.5%)	26(23.4%)

With several papers selected randomly from site vnexpress.net, the parser worked less effectively.

Most of incorrect sentences given by the link parser were complex compound sentences with at least two independent clauses and one or more subordinate clauses. This motivates us to the development of a new model to parse complex compound sentences.

V. Conclusion

We have created a link grammar for Vietnamese. Based on such grammar, the link parser is applicable to parse simple sentences. It also parses compound sentences with two clauses. Sentences which were not able to recognize or were recognized falsely are:

- Too long sentences consisting of many clauses or too complicated links between phrases.

- Sentences using a grammatical phenomenon that have not been handled yet.

References

[1] Daniel D.K. Sleator, Davy Temperley. Parsing English with Link Grammar. CMU-CS-91-96, October 1991.

[2] Nguyen Chi Hoa. A Practical Grammar of Vietnamese (in Vietnamese). Hanoi National University Publishing House 2004.

[3] Dennis Grinberg, John Lafferty, Daniel Sleator. A robust parsing algorithm for Link Grammars. CMU-CS-95-125, August 1995.

[4] John Lafferty, Daniel Sleator, Davy Temperley. Grammatical Trigrams: A Probabilistic Model of Link Grammar. AAAI Technical Report FS-92-04, 1992.

[5] Filip Ginter, Sampo Pyysalo, Jorma Boberg, Tapio Salakoski. Regular Approximation of Link Grammar. FinTAL 2006, LNAI 4139, pp. 564–575, 2006.

[6] Ozlem Istek. A Link Grammar for Turkish. MSc.Thesis August, 2006.

[7] Cam Tu Nguyen, Trung Kien Nguyen, Xuan Hieu Phan, Le Minh Nguyen, and Quang Thuy Ha: Vietnamese Word Segmentation with CRFs and SVMs: An Investigation, The 20th Pacific Asia Conference on Language, Information, and Computation (PACLIC), 1st-3rd November, 2006, Wuhan, China.

[8] John Lafferty, Andrew McCallum, Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.

[9] Gerold Schneider. A Linguistic Comparison of Constituency, Dependency and Link Grammar. July, 1998.

[10] Sandra Kubler. Learning a Lexicalized Grammar for German. In D.M.W. Powers (ed.) NemLaP3/ CoNLL98: New Methods in Language Processing and Computational Natural Language Learning, ACL, pp 11-18.

[11] Commission of Social Sciences, Vietnam. Grammar of Vietnamese (In Vietnamese), Social Science Publishing House, Hanoi, 1983.

[12] Diep Quang Ban General Grammar of Vietnamese (In Vietnamese) Higher and Vocational Education Publishing House, Hanoi, 1989.

[13] Nguyen Thien Nam. Intermediate Vietnamese (Bilingual: Vietnamese- English,) Education Publishing house, 1998.

[14] Mel'čuk .Dependency Syntax: Theory and Practice, Albany, N.Y.: The SUNY Press (1988)

[15] J¨org Hakenberg, Ill´es Solt, Domonkos Tikk, Luis Tari.

[16] Akshar Bharati, Dipti Misra Sharma, Sukhada Adapting Link Grammar Parser (LGP) to Paninian Framework .National Seminar on Computer Science and its Applications in Traditional Shastras (CSATS'09).

[17] Rania A. Abul Seoud, Nahed H. Solouma, Abou-Bakr M. Youssef, Yasser M. Kadah Extraction of Protein Interaction Information from Unstructured Text Using a Link Grammar Parser. Proceeding of Cairo International Biomedical Engineering conference 2006.

[18] Marshall and E. Safar, "Extraction of Semantic Representations from Syntactic CMU Link Grammar linkages.", Recent Advances in Natural Language Processing (RANLP), Tzigov Chark Bulgaria, Sept 2001, pp154-159.

[19] N. Zamin and B., Baharudin and T.B., Adji Applying link grammar formalism in the development of English-Indonesian machine translation system. In: 9th Int. Conf. Artificial Intelligence and Symbolic Computation, AISC 2008 - 15th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning, Calculemus 2008 - 7th Int. Conf. Mathematical Knowledge Management, MKM 2008, 28 July 2008 through 1 August 2008, Birmingham.

[20] Norshuhani Zamin, Information Extraction Using Link Grammar,csie, vol. 5, pp.149-153, 2009 WRI World Congress on Computer Science and Information Engineering, 2009.

[21] Qingquan Wang, Lili Rong, Kai Yu, Toward a Categorical Link Grammar for Knowledge Piece Generation in Emergency Decision-Making Support, ncm, vol. 2, pp.732-735, 2008 Fourth International Conference on Networked Computing and Advanced Information Management, 2008.

[22] Chih-yu Chao, Parsing Chinese "de" (的) Particle with Link Grammar ,11-722 Grammar Formalisms Term Project, MIT,2004.

AUTHORS' BIOGRAPHIES

Nguyen Thi Thu Huong was born in 1964. She graduated from Hanoi University of Science and Technology, Vietnam with M.Sc. in Information Technology. She is currently a lecturer of Department of Computer Science, School of Information and Communication Technology,Hanoi University of Science and Technology, Vietnam. Her special fields of interest are Computational Theory, Compiler and Natural Language Processing.

Pham Nguyen Quang Anh was born in Hanoi , Vietnam in 1986. He finished master program from Hanoi University of Science and Technology. His employment experience included NaisCorp IT Corporation, Hanoi, Vietnam. His fields of interest are Data Mining and Natural Language Processing.