Abstract

Background/Objectives: Dependency parsing is one of the remaining problems that attracts researchers’ interest because of its contribution to natural language processing applications. This paper presents a dependency parser and provides an effective sentiment analysis model for Vietnamese. Methods/Statistical Analysis: Up to now, there has been some remarkable work on dependency parsing for Vietnamese, of which the approach of transforming the constituents into a dependency grammar has been proven to give the best performance. In this study, we proposed another way to parse Vietnamese sentences to create dependency trees based on the incremental algorithm by Covington and Graph Unification Programming. We have also combined a dependency parser, sentiment dictionary, domain ontology, and syntactic dependency rules for solving the sentiment analysis problem. Findings: We found that the experiments using simple sentences collected from Vietnamese reviews provided better results than with other methods. Application/Improvements: The positive results, which came from an experimental period, will provide us with a basis from which to expand grammar rules to build a parser for more complicated Vietnamese sentences and to integrate it into many cognitive tasks, such as sentiment analysis.

Keywords: Dependency Parsing, Dependency Syntactic Analysis, Opinion Mining, Sentiment Analysis, Vietnamese Parser

1. Introduction

Dependency grammar describes the syntactic structure of a sentence by linking words and representing their relations. Each link has a head and a dependent. The dependent is the modifier or the complement and the head determines the attribute of the dependent. The head is required while the dependent is optional in a sentence. The head and dependent pair can be represented graphically in many ways. The grammar rules will identify the head and its dependent(s).

In this work, we implement the dependency parser for Vietnamese, based on the incremental algorithm by Covington and GULP (Graph Unification Logic Programming). We only deal with simple sentences (without handling a subordinate and coordinate clause) that are some hotel service reviews. This study is our first step toward a deeper understanding of the analysis of dependency grammar and its adoption in Vietnamese language processing.

The rest of this paper is organized as follows. Section 2 presents related works. Section 3 describes the dependency relations of parts of speech in Vietnamese. In Section 4, the Vietnamese dependency parser is explained in more detail. Section 5 discusses the experiments. In Section 6, we introduce our application. Finally, we conclude the report and discuss future work.

2. Related Works

Concerning dependency parsing for Vietnamese, recently, there are having three remarkable publications displayed by Hong et al., Thi et al., and Dat et al. Hong et al. adopted lexicalized tree-adjoining grammars (LTAG) for Vietnamese dependency parsing. From their experiments,
they achieved approximately 73.21% accuracy with 441 sentences of 30-words or less. Thi et al.\(^4\) and Dat et al.\(^5\) utilized the same approach for constituent-to-dependency transformation. Among them, the work of Dat et al. gave higher results, with the best accuracy at 76.21%. These authors also provided a Vietnamese dependency treebank named VnDT\(^6\) with 10,200 sentences.

On the use of the graph unification logic programming (GULP) framework for Vietnamese language processing, lately, Trung Tran et al.\(^7\) have described some grammatical characteristics of the noun and pronoun “nó”.

In\(^8\), M. Kamayani and A. Purwarianti described dependency grammar for Indonesian based on Stanford Dependency Label. The implementation of dependency parsing used the incremental algorithm by Covington and GULP for handling feature structures. The authors identified 30 non-clausal rules and the system successfully parsed simple Indonesian sentences.

3. The dependency relations of the parts of speech in Vietnamese

3.1 The Dependency Relation between a Noun and Verb

Example 1. Lan đi du lịch. (Lan goes traveling.)

In this type of simple sentence, a noun is the subject receiving the effect from the verb. Therefore, the main part in the above sentence is the verb; the noun depends on the verb and complements the verb following it.

In addition, the noun also complements the verb preceding it. In Example 1, “du lịch” complements the verb “đi”. Figure 1 shows the dependency tree of the sentence “Lan đi du lịch”.

3.2 The Dependency Relation between a Noun and Adjective

Example 2. Nơi đây đẹp như khách sạn cao cấp. (Here looks like a luxury hotel.)

In the phrase “khách sạn cao cấp”, the adjective “cao cấp” complements “khách sạn”, because it is the noun phrase. The phrase “nơi đây đẹp” is a sentence in a fact, so the noun “nơi đây” complements the adjective “đẹp”. Meanwhile, “đẹp” is the predicate. As we mentioned, there are two sorts of relations between a noun and adjective: 1) an adjective complements a noun in the case of a noun phrase and 2) a noun complements an adjective in the case that the adjective is the predicate of the sentence. Fig.2 shows the dependency tree of the sentence “Nơi đây đẹp như khách sạn cao cấp”.

3.3 The Dependency Relation between a Verb and an Adjective

Example 3. Khách sạn không được dọn dẹp sạch sẽ. (The hotel is not really clean.)

In Vietnamese, some verbs combine with adjectives to form predicates, as shown in Example 3. This sort of phrasal verb usually comprises action verbs combined with most sorts of adjectives and lets us know the speaker’s evaluation of the subject “khách sạn” concerning the action “đon dẹp”. Adjectives may precede verbs or follow them. Whether preceding or following verbs, adjectives always complete them. Figure 3 shows the dependency tree of the sentence “Khách sạn không được dọn dẹp sạch sẽ”.

3.4 The Dependency Relation between Verbs

Example 4. Khách sạn cần được sửa chữa. (The hotel needs to be repaired.)

![Figure 1. Dependency tree of the sentence “Lan đi du lịch”.](image1)

![Figure 2. Dependency tree of the sentence “Nơi đây đẹp như khách sạn cao cấp”](image2)

![Figure 3. Dependency tree of the sentence “Khách sạn không được dọn dẹp sạch sẽ”](image3)
The predicate here is the combination of three verbs ("cần", "được", and "sửa chữa"). In this case, we have a combination of a non-independent verb and independent verbs. The non-independent verb ("cần") is the main element in this phrase and conveys only ability and expectation, but not any specific content. Therefore, it needs a verb to complete it. In this case, following verbs usually use completed verbs that precede them. Figure 4 shows the dependency tree of the sentence "Khách sạn cần được sửa chữa".

3.5 Conjunction

Example 5. Nhân viên vui vẻ và thân thiện. (The staff is funny and friendly.)

The example displays a conjunctive structure using the conjunction "và". Therefore, all the words in the conjunctive structure have the same dependency relations, apart from the conjunction. For example, "vui vẻ" and "thân thiện" have similar functions. Their relationship is shown by the conjunction "và". To describe this relationship, [6] defines the coordinate and conjunction relations. We may focus on "và" and depend on "vui vẻ," according to the coordinate relation and "thân thiện" depending on "và", according to the conjunction relation. Fig.2 shows the dependency tree of the sentence "Nhân viên vui vẻ và thân thiện".

4. Vietnamese dependency parser

Similar to M. Kamayani and A. Purwarianti [8], we conducted the implementation of the dependency parser based on Covington’s algorithm with some modifications by using GULP to define dependency rules for Vietnamese. In our system, the dependency labels are called after VnDT Vietnamese dependency treebank [6] and Stanford dependency label [9].

4.1 Dependency Relations

There are about 33 grammatical relations in the VnDT Vietnamese dependency tree bank. The dependencies are all binary relations: a grammatical relation holds between a head and a dependent. The dependency definitions make use of Penn Treebank POS tags and phrasal labels. They consist of linear relations and sub-clause relations. For this work, we only deal with linear relations and use 14 dependency labels.

4.2 Covington’s algorithm

Despite the usefulness of Covington’s method of free word order dependency parsing at the algorithm level [1], we must eliminate some of his limitations, for instance, his treating word order as totally free. In order to achieve this goal, enriching the grammar with feature structures using GULP [2] is the best way, as it will facilitate the implementation of unification-based features in Prolog.

Feature structures and feature values are the key elements allowing unification-based grammar represent information of features, such as meaning, part of speech, number, case, subcategorizations and etc. Only the features that are related to the word need instantiating in our system.

4.3 Lexicon

Lexical entries have the form: Word (Form, Category([GULP feature structures])).

Where:

- The Form is the word, as it actually occurs in the input.
- The Category is the syntactic category (noun, verb, adjective...)
- GULP feature structures such as:
  - Glossary is a label that identifies the word in the output.
  - Grammatical features: Subject (noun), Object (noun)...
Dependency Syntactic Analysis for Opinion Mining

Example 6:
- word(khách_sạn, n(sem~hotel..gram~subj)).
- word(khách_sạn, n(sem~hotel..gram~obj)). % (In Vietnamese, all nouns or pronouns can be an object as well as a subject in sentence)
- word(trông, v(sem~look)).
- word(tốt, adj(sem~good)).
- word(không, r(sem~not)).
- word(dé, e(sem~to)).
- word(thì, c_dep(sem~is)).

4.4 Dependency Rules

Dependency rules specify what dependents can be connected to certain heads. The form is check_dh (Dependent, Head), where ‘dependent’ and ‘head’ are feature lists like those in the lexicon.

Example 7. Khách sạn trông tốt. (The hotel seems good.)

To parse this sentence, our system uses the two rules below:

check_dh([NN,_,_,n,gram~subj],[N,_,_,v,_]) :- !, NN < N. (1)
check_dh([NN,_,_,a,_], [N,_,_,v,_]) :- !, NN > N. (2)

(1) means N_{khách_sạn} precedes V_{trông} and N is the subject. In this case, V is the head and N is its dependent.

(2) means V_{trông} precedes A_{tốt}. V would be the head and A its dependent.

Example 8. Tôi cũng mua một resort ở biển. (I have also bought a beach resort.)

To parse this sentence, our system uses the six rules below:

check_dh([NN,_,_,r,_], [N,_,_,v,_]) :- !, NN < N. (3)
check_dh([NN,_,n,gram~subj],[N,_,_,v,_]) :- !, NN < N. (4)
check_dh([NN,_,n,gram~obj],[N,_,_,v,_]) :- !, NN > N. (5)
check_dh([NN,_,m,_], [N,_,_,n,_]) :- !, NN < N. (6)
check_dh([N,_,e,_], [NN,_,e,_]) :- !, N is NN + 1. (7)
check_dh([N,_,n,gram~obj],[NN,_,e,_]) :- !, N is NN + 1. (8)

(3) means R_{cùng} precedes V_{mua} and V is the head, R is the dependent.

(4) means N_{(tô)} precedes V_{mua}, N is the subject. In this case, V would have as the head, N as the dependent.

(5) means V_{mua} precedes N_{(resort)} N is the object. In this case, V is the head, N is the dependent.

(6) means N_{(resort)} precedes M_{(o)} N is the object. In this case, N is the head, M is the dependent.

(7) means N_{(resort)} is preceded adjacent E_{(o)} N is the object. In this case, N is the head, E is the dependent.

(8) means E_{(o)} is preceded adjacent N_{(biển)} N is the object. In this case, E is the head, N is the dependent.

In our system, in order to process simple sentences within the context of customer reviews for hotel services, we define 21 dependency rules, as shown in Table 1.

Table 1. Our proposed dependency rules

| Word pairs* | Rules | Label** |
|-------------|-------|---------|
| R – V | V: head, R: dependent, R precedes V | adv |
| N_{ad} – V | V: head, N: dependent, N precedes V | sub |
| V_{1} - V_{2} | V_{1}: head, V_{2}: dependent, V_{1} is preceded adjacent V_{2} | vmod |
| N – P | N: head, P: dependent, N precedes P | det |
| M – N | N: head, M: dependent, M is preceded adjacent N | det |
| N_{ad} – N_{ad} | N_{2}: head, N_{1}: dependent, N_{1} is preceded adjacent N_{2} | nmod |
| R – N | N: head, R: dependent, R is preceded adjacent N | pob |
| V – N_{ad} | V: head, N: dependent, V precedes N | dob |
| V – N_{ad} | V: head, N: dependent, V precedes V | root, sub |
| V – A | V: head, A: dependent | vmod |
| N_{ad} – A | A: head, N: dependent, A precedes N | sub, nmod |
| A – C | A: head, A: dependent, A is preceded adjacent C | amod |
| N – E | N: head, E: dependent, N is preceded adjacent E | coord |
| E – N | E: head, N: dependent, E precedes N | prp |
| A – C | A: head, A: dependent, C precedes A | conj |
| V – C | V: head, C: dependent, V precedes C | coord |
| C – V | C: head, V: dependent, C precedes V | conj/prp, vmod |
| V – P | V: head, P: dependent, V precedes P | loc |
| V – E | V: head, E: dependent, V precedes E | loc |

*: R: adverb, V: verb, N: noun, P: pronoun, M: numeral, A: adjective, C: conjunction, E: preposition.
**: The dependency labels are called after VnDT Vietnamese dependency Treebank [6].
### Table 2. Some experimental results

| Sentences                        | Our system                                                                 | VNDP system                                                                 | Explanation                                                                 |
|----------------------------------|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Khách sạn này cần được nâng cấp (This hotel needs to be upgraded) | try([khách_sạn,n,sem~hotel..gram~subj].nmod(提升,提升)) | root(提升)                      | The fact that VNDP system understands the parts of speech of “được” as adverb (R) leads to a wrong parsing result. |
| Phòng ốc khách sạn này đẹp (This hotel’s rooms are nice) | try([phòng_ốc,khách_sạn,n,sem~room..gram~subj].nmod(漂亮,漂亮)) | root(漂亮)                      | In the case of understanding “này” as a word depending on “khách sạn” instead of “phòng ốc”, VNDP system also leads to a wrong parsing result. |
| Phong cảnh đẹp (The view is beautiful) | try([phong_cảnh,dep]).nmod(漂亮,漂亮)) | root(phong_cảnh)                 | Similar to above cases, the understanding “phong cảnh đẹp” as a noun phrase instead a sentence and “đẹp” to be dependent on “phong cảnh” makes the results obtained from VNDP system wrong. |
| Thật ấn tượng bởi vẻ đẹp của resort (What an impressive view of resort) | try([thật,sự,ấn_tượng,bởi,vẻ_đẹp,của,resort]).nmod(漂亮,漂亮)) | root(漂亮)                      |                                                                                                                                         |
| Cảm thấy được thoải mái và thư giãn (Feel relaxed and comfortable) | try([cảm_thấy,duyệt,thoái_mại,và,thư_gian]).nmod(舒服,舒服)) | root(舒服)                      | The fact that VNDP system understands the parts of speech of “duyệt” as adverb (R) and also gives the same wrong result as in the first case. |
7. Conclusion

In this research, we have carried out the experiment of building the dependency parser for simple Vietnamese sentences basing on Covington’s algorithm and GULP. The 21 dependency grammar rules and 14 dependency labels are described in total. The positive results, which came from an experimental period, will provide us with a basis from which to expand grammar rules to build a parser for more complicated Vietnamese sentences. With the dependency parser, we can deal with many cognitive tasks, such as sentiment analysis/opinion mining, etc.

8. Acknowledgment

This work is supported by the research project TNCS-2015-KHMT-39 funded by Ho Chi Minh City University of Technology, Vietnam National University Ho Chi Minh City (VNU-HCM). We would like to give special thanks for Dr. Tran Van Tieng, who has contributed to the comments for linguistics in this article.

9. References

1. Covington M. A Fundamental Algorithm for Dependency Parsing. Proceedings of the 39th Annual ACM Southeast Conference, 2009. p. 95–102.
2. Covington MA. GULP 4: An Extension of Prolog for Unification Based Grammar. USA: Artificial Intelligence Center, The University of Georgia, Research Rep. AI-1994-06. 2007.
3. Hong PL, Nguyen TMH, Roussanaly A. Vietnamese Parsing with an Automatically Extracted Tree-Adjoining Grammar. Proceedings of the 9th IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future. 2012. p. 1–6.
4. Thi LN, My LH, Viet HN, Minh HNT, Hong P. “Building a Treebank for Vietnamese Dependency Parsing. Proceedings of the 10th IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future. 2013. p. 147–51.
5. Nguyen DQ, Nguyen DQ, Pham SB, Nguyen P-T, Nguyen ML. From Treebank Conversion to Automatic Dependency Parsing for Vietnamese. Proceedings of 19th International Conference on Application of Natural Language to Information Systems, NLDB’14, Springer LNCS. 2014. p. 196–207.
6. Dat Quoc Nguyen, Dai Quoc Nguyen, Dang Duc Pham, and Son Bao Pham. 2014. “RDRPOSTagger: A Ripple Down
Rules-based Part-Of-Speech Tagger,” In Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 17–20.

7. Tran T, Nguyen DT. The Solution for Resolving Inter-Sentential Anaphoric Pronoun “nó” in Vietnamese Paragraphs Composing 3 to 5 Simple Sentences. International Journal of Advanced Science and Technology. 2014; 65:95–112.

8. Kamayani M, Purwarianti A. Dependency Parsing for Indonesian. International Conference on Electrical Engineering and Informatics, ICEEI. 2011. p. 1–5.

9. de Marneffe M-C, Manning CD. Stanford dependencies manual. Technical report, Stanford University. 2008.