FORMAL MODELS IN NATURAL LANGUAGE PROCESSING

Eva Zámečníková, Petr Horáček
Doctoral Degree Programme (3), FIT BUT
E-mail: {xzamec10 | xhorac06}@stud.fit.vutbr.cz

Supervised by: Alexander Meduna
E-mail: meduna@fit.vutbr.cz

Abstract: There are many models describing the natural languages. The main interest of most of them is to capture the syntax and semantics of the languages. The motivation was to find a way to transform natural languages in between and also automatize machine translation. The main purpose of this paper is to introduce a pair of frameworks widely used in today's natural language processing and to present in which aspect they are useful. These frameworks are dependency and phrase structure grammars. More detailed description of them will be given and in the end these two approaches will be compared.

Keywords: NLP, FLT, phrase structure grammar, HPSG, dependency grammar

1 INTRODUCTION

Processing of natural languages (NLP) is a field of theoretical informatics and linguistics and is concerned with the interactions between computers and human (natural) languages. It is defined as a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts (which means any language) at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications (according to [1]). The most frequent applications utilizing NLP include the following: Information Retrieval, Information Extraction (IE), Question-Answering, Summarization, Machine Translation and more others.

The history goes back to the the late 1940s when was an effort to understand and formally describe the syntax of natural languages. But the first step forward was the publishing of book called *Syntactic Structures*, by Noam Chomsky, introducing the idea of generative grammar [2]. Due to the developments of the syntactic theory of language and parsing algorithms, people believed that fully automatic high quality translation systems would be able to produce results on par with human translators. But that moment has not come yet.

Firstly NLP was in interest of artificial inteligence (AI), but then split into two separate disciplines. The first is known as a set of formalism with are generally called formal language theory (FLT). Today's FLT focuses mainly on theoretical studies of formal models and their properties. And the second, NLP, studies many other aspects of natural languages besides their syntax (such as morphology or semantics). This discipline is focused mainly on practical applications (tasks as speech recognition and synthesis).

Recent research has increasingly focused on unsupervised and semi-supervised learning algorithms. Today's machine translation we can see in practise – eg. Google translator, or web pages translation. There are also some statistical methods which are in practise mostly used for syntax checking in grammar correctness.

The outline of this paper will be as follows. After the introduction to problematic the definitions of two NLP frameworks will be proposed which are widely used in practise and theirs advantages and disadvantages will be stated. These methodologies are: *dependency grammars* and the *phrase stucture grammars*. We will discuss the *core points* of these theories. In the end there will be the comparison of these two approaches.

This paper assumes that reader is familiar with basic theory of formal languages. For more information see [3], [4].

2 DEPENDENCY GRAMMARS

It is the name for whole group (or better *framework*) of syntactic theories such as *Algebraic syntax*, *Operator Grammar*, *Functional Generative Description*, *Lexicase Grammar*, *Meaning-Text Theory* and others. For more information about particular theoretical model follow [4].

The first concept of dependency grammar (DG) was published in the late 50's and lately has largely developed as a form for syntactic representation.

2.1 Dependency

The basic assumptions behind the notion of dependency are summarized in the following sentences from the seminal work of Tesnière which is usually taken as the starting point of the modern theoretical tradition of dependency grammars (taken from [2]):
The sentence is an organized whole; its constituent parts are the words. Every word that functions as part of a sentence is no longer isolated as in the dictionary: the mind perceives connections between the word and its neighbours; the totality of these connections forms the scaffolding of the sentence. The structural connections establish relations of dependency among the words. Each such connection in principle links a superior term and an inferior term. The superior term receives the name governor; the inferior term receives the name dependent.

So the fundamental notion of *dependency* is based on the idea that the syntactic structure of a sentence consists of *binary asymmetrical relations* between the words of the sentence (see [5]). In dependency grammar, one word is the *head* of a sentence, and all other words are either a *dependent* of that word or of some other word which is connected to the headword through a sequence of dependencies. Dependencies are usually represented by curved arrows (as can be seen in figure 1). Terms of dependencies are very useful eg. in statistical methods of natural laguage processing. Statistical approach also considers probabilities or weights of particular rules [6].

2.2 Definition

Words in dependency relation are marked as (the names differs) *parent* and *child* (or according Tesnière *governor* and *dependent*) and arrow which connects these words usually leads from children to parent.

Notation If w is child and v is its parent, we write $w \rightarrow v$ If there is a *path* from w to v, we write $w \rightarrow^* v$ (*transitive closure*).

2.3 Properties of DG

- *Single head* – each word has one and only one parent (except for the root node).
- *Connected* – all words form a connected graph.
- *Acyclic* – if $w_i \rightarrow w_j$, $w_j \rightarrow^* w_i$ never holds. The graph does not contain cycles. Note that w_i denotes i-th word in sentence.

- *Projective* – if $w_i \to w_j$, then for all w_k, where $i < k < j$, either $w_k \to^* w_i$ or $w_k \to^* w_j$ holds. A projective tree does not contain any crossing between dependencies (as can be seen in 1).

  Some dependency formalisms allow *non-projectivity* (where the dependencies can be crossed). Broadly speaking, we can say that whereas most practical systems for dependency parsing do assume projectivity, most dependency-based linguistic theories do not [7].

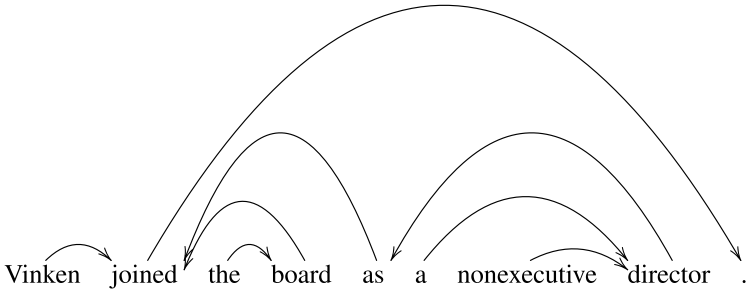

Figure 1: Projective Dependency Tree

3 PHRASE STRUCTURE GRAMMARS

Phrase structure grammars (PSG) were introduced by Chomsky in 1957 [8] and they were defined by phrase structure rules (or rewrite rules). Instead of the dependency relation used in dependency grammars, phrase structure grammars are based on *constituency* relation. They are not used in many practical applications, but their basic principles inspired some much more successful models (such as Head-driven PSG).

This formalism uses *phrase structure trees* to describe the structure of sentences. This approach is quite new in comparison with the the approach using dependencies. It involves phrase structure rules, which yield trees with labels added to indicate the syntactic category of each constituent (as Noun Phrase, Verb Phrase, etc.). The resulting tree is seen to recapitulate the process by which a sentence is generated by the rules of grammar: a group of elements forms a *constituent* whenever they have been introduced by the application of the same rule.

3.1 DEFINITION

A *phrase structure grammar* (PSG) G is a quadruple $G = (N,T,P,S)$, where N is a finite set of *nonterminals*, T is a finite set of *terminals*, $N \cap T = \emptyset$, $P \subseteq (N \cup T)^* N (N \cup T)^* \times (N \cup T)^*$ is a finite relation – we call each $(x,y) \in P$ a *rule* (or *production*) and usually write it as $x \to y, S \in N$ is the *start symbol*.

Generalized PSG (GPSG) were created as an attempt to show that it is possible to describe natural languages in a context-free framework, without using transformations. Apart from context-free rules, GPSG includes *features* and *metarules*. Head-driven PSG representations also use feature structures (*signs*), often written as attribute-value-matrixes (AVMs), to represent grammar principles, grammar rules and lexical entries. A *constituent* is licensed if it is described by a feature structure and this feature structure conforms to each grammatical principle. When the constituent is phrasal, it also has to conform to a grammar rule and when it is lexical, it has to conform to a lexical entry [9].

3.2 HEAD-DRIVEN PSG (HPSG)

HPSG is seen as a later development of GPSG, but it is worth noting that HPSG has had influences from a number of linguistic theories. For example, HPSG categories are more complex than those in

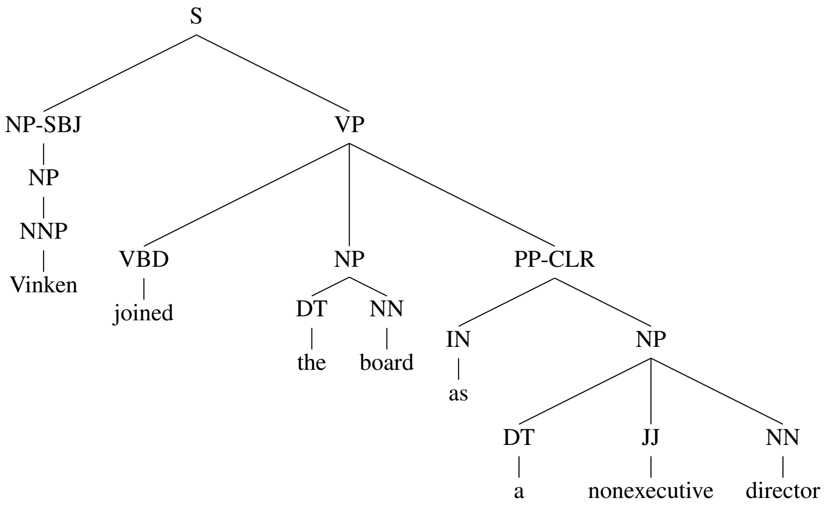

Figure 2: PSG Derivation Tree (Adapted from Penn Treebank)

GPSG and HPSG makes more specific claims about universals and variation. HPSG is more suitable for computer implementations, often used in practice in NLP.

3.3 Sign feature.

An important concept in HPSG representations is the *sign* feature. The *sign* is a collection of information, including phonological, syntactic and semantic constraints and it is represented in AVMs. AVMs encode feature structures where each *attribute* (feature) has a type and is paired with a *value*. Signs receive the subtypes word or phrase depending on their phrasal status. These subtypes differ in that they conform to different constraints, but both contain attributes such as phonology (PHON) and syntax/semantics (SYNSEM). PHON has as its value a list of phonological descriptions [9].

4 COMPARISON OF DEPENDENCY GRAMMARS AND PHRASE STRUCTURE GRAMMARS

Dependency grammar is distinct from phrase structure grammar, as it lacks *phrasal nodes*. Phrase structure rules are commonly employed result in a view of sentence structure that is *constituency-based*. Thus grammars that employ phrase structure rules are constituency grammars as opposed to dependency grammars, which view sentence structure as dependency-based [9]. The constituency relation is a *one-to-one-or-more* correspondence. For every word in a sentence, there is at least one node in the syntactic structure that corresponds to that word. The dependency relation, in contrast, is a *one-to-one relation*; for every word in the sentence, there is exactly one node in the syntactic structure that corresponds to that word.

In recent years, DG also have become increasingly used in computational tasks, such as information extraction, machine translation, and efficient parsing. Among the purported advantages of dependency over phrase structure representations are conciseness, intuitive appeal, and closeness to semantic representations such as predicate-argument structures. On the more practical side, dependency representations are attractive due to the increasing availability of large corpora of dependency analyses, such as the Prague Dependency Treebank [2].

Phrase structure are considered to be more suitable for languages with fixed word order and clear constituency structures. Dependency representations, in contrast, may be found more adequate for languages which allow greater freedom of word order and in which linearisation is controlled more by pragmatic than by syntactic factors. This is eg. the case of Czech. In this aspect, DG are more robust and they are uniformly applicable to many languages.

From the view of ambiguity it is better to use dependency grammars. The lexical information is key for resolving ambiguities and disambiguation decisions are made directly in terms of word dependencies. There is no need to create large structures over the sentence as in the PSG [6].

There has been an effort to trasform these two methodologies. For example, in the paper [10] the authors show how HPSG can be simulated by a dependency grammar.

5 CONCLUSION

The main point of this paper was to discuss two NLP approaches and compare them. The differencies are discussed. Each of these frameworks has advantages, but they are distinct in their basic idea and it also depends on which purpose we want to use them.

ACKNOWLEDGEMENT

This work was partially supported by the research plan MSM0021630528, the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), and the BUT FIT grant FIT-S-11-2. It was also supported by the grant MSMT FRVS FR97/2011/G1. Results of this project are published on [11].

REFERENCES

[1] Khurana, P., Singh, V.: A Model For Human Cognition [online]. In *International Journal of Computing and Business Research, Volume 2 Issue 3*, 2011, <http://www.researchmanuscripts.com/PapersVol2N3/3.pdf>.

[2] Debusmann, R., Kuhlmann, M.: Dependency Grammar: Classification and Exploration, 2008.

[3] Meduna, A.: Automata and Languages: Theory and Applications. Springer, 2005, ISBN 1-85233-074-0, 892 p.

[4] Allen, J.: Natural Language Understanding. *The Benjamin/Cummings Publishing Company Inc.*, 2005.

[5] Nivre, J.: Dependency Grammar and Dependency Parsing [online], Växjö University, 2005. <http://w3.msi.vxu.se/ nivre/papers/05133.pdf>

[6] Manning, C. D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, 1999.

[7] Mel'čuk, I.: Dependency Syntax: Theory and Practice, State University of New York Press, 1988.

[8] Chomsky, N.: Syntactic Structures. The Hague/Paris: Mouton, 1957.

[9] Lima, A.: Introduction to LFG and HPSG [online]. University of Queensland, <http://emsah.uq.edu.au/linguistics/Working%20Papers/ananda_ling/HPSG_Introduction.htm>.

[10] Sylvain K. N., Sylvain K.: If HPSG were a dependency grammar ..., 1996.

[11] Horáček, P., Burgetová, I., Zámečníková, E.: Formal Models in Natural Language Processing [online] <http://www.fit.vutbr.cz/ ihoracekp/frvs2011>

[12] Lucien Tesnière: *Éléments de syntaxe structurale*, Editions Klincksieck, 1959