Parsimonious Vole

A Systemic Functional Parser for English

Universität Bremen

Eugeniu Costetchi

Supervisor: Prof. John Bateman

Advisor: Dr. Eric Ras

Faculty 10: Linguistics and Literary Studies
University of Bremen

This dissertation is submitted for the degree of
Doctor of Philosophy

May 2018

I would like to dedicate this thesis to my loving parents ...

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Eugeniu Costetchi
May 2018

Acknowledgements

And I would like to acknowledge ...

Abstract

This is where you write your abstract ...

Table of contents

List of figures

List of tables

List of definitions

Chapter 1

The dependency grammar

The Stanford dependency analysis of a given text constitutes the input for the algorithm developed in the current work. It provides the foundation to build the syntactic backbone used adopted here. This chapter offers an overview of the grammar and the parser developed at the Stanford university. In the last part of the chapter is discussed the cross theoretical connection between the dependency and systemic functional grammars.

1.1 Origins of the dependency theory

For the first time a complete linguistic theory based on the dependency concept was elaborated by the French linguist Lucien Tesniere in his seminal work *"Elements de syntaxe strusturale"* published in 1959 after his death. He devoted much effort to argue for the adequacy of *dependency* as the organizational principle underlying numerous phenomena and in fact attempting to demonstrate the universality of his syntactic analysis method for human languages. In doing so he introduced a series of concepts and ideas among which the *verb centrality*, *stratification*, *language typology*, *nuclei*, *valency*, *metataxis*, *junction* and *transfer* are the most important ones which I introduce following the connections.

> The sentence is an *organized set*, the constituent elements of which are the words. Each word in a sentence is not isolated as it is in the dictionary. The mind perceives *connections* between a word and its neighbours. The totality of these connections forms the scaffold of the sentence. These connections are not indicated by anything. But it is absolutely crucial that

> they be perceived by the mind; without them the sentence would not be intelligible. (Tesniere 2015: 3)

Tesniere holds the view that the connection, what is know today as *dependencies*, are the foundations of the *structural syntax* known as *dependency grammar* today. According to him "to construct a sentence is to breathe life into an amorphous mass of words, establishing a set of connections between them. Conversely, understanding a sentence involves seizing upon the set of connections that unite the various words" (Tesniere 2015: 4). He introduces the hierarchy of connections as follows.

> Structural connections establish *dependency* relations between words. In principle, each connection unites a superior term and an inferior term. The superior term is called the *governor*, and the inferior term the *subordinate*. We say that the subordinate depends on the governor and that the governor governs the subordinate. [. . .] A word can be both subordinate to a superior word and governor of an inferior word. [. . .] The set of words of a sentence constitutes a veritable *hierarchy*. (Tesniere 2015: 5–6)

Introduction of hierarchy and governor-subordinate dependencies permeates to define now what is a *node* and the *stemma* resembling what is now known as *dependency tree* (although the stemmas do not include labels on the tree edges).

> [. . .] In principle, a subordinate can only depend on a sole governor. A governor, in contrast, can govern multiple subordinates [. . .] Every governor that governs one or more subordinates forms what we call a node. [. . .] it follows that *each subordinate shares the fate of its governor*. (Tesniere 2015: 6)

speaks
|
Alfred

Fig. 1.1 Stemma for "Alfred speaks"

This asymmetry of connection permits construction of a tree-like structure. The diagram of the two word sentence "Alfred speaks" is provided in the Figure 1.1. The word "speaks" is the governor of the word "Alfred". The connection is depicted bu the vertical line connecting the two. But to make it complete it is important to decide on the root node.

> The node formed by the governor that governs all the subordinates of a sentence is the *node of nodes*, or the central node. It is at the centre of the sentence and ensures its structural unity by tying the diverse elements into a single bundle. It can be identified with a sentence. *The node of nodes is generally verbal* [...] (Tesniere 2015: 7)

The fundamental insight presented above about the nature of the syntactic structure concerns the grouping of words at the clause level. Tesniere rejects the subject-predicate formation that was the de facto syntactic understanding of his time. He argued that this division is belongs to Aristotelian logic and is not associated to linguistics. Instead of the subject-predicate division Tesniere positions the verb at the root of the clause structure making the subject and the object subordinated seedlings. Figure 1.2 depicts the clause structure "Alfred speaks slowly" where both the subject and the object are subordinated to the central verb speaks.

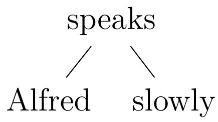


Fig. 1.2 Stemma for "Alfred speaks slowly"

Tesniere is among pioneer linguists recognising that the language is organised at different levels thus advocating a *stratified model of language*. He recognises the to dimensional syntactic representation and the one dimensional chain of spoken language.

> *speaking* a language involves transforming structural order to linear order, and conversely, *understanding* a language involves transforming linear order to structural order. The fundamental principle of transforming structural order to linear order involves changing the connections of structural order into the sequences of linear order. This transformation occurs in such a manner that the elements connected in structural order become immediate neighbours in the spoken chain (Tesniere 2015: 12).

In the structural realm Tesniere goes even deeper and describes the separation between syntax and semantics. To argue for that, he uses an example similar to the famous Chomskian *colourless green ideas sleep furiously* (Chomsky 1957b) (that occurred three years after Tesniere's death). He employed the sentence *the vertebral silence antagonizes the lawful sail*.

> Syntax is distinct from morphology, and it is no less distinct from semantics. The structure of a sentence is one thing, and the idea that it expresses and that constitutes its meaning is another. It is therefore necessary to distinguish between the structural plane and the semantic plane. [...] The structural plane and the semantic plane are therefore entirely independent of each other from a theoretic point of view. The best proof is that a sentence can be semantically absurd and at the same time syntactically perfectly correct. (Tesniere 2015: 33).

Tesniere distinguishes between *nodes* and *nuclei*. Initially he defines the node in a way that resembles the phrase or a constituent but after that he changes his mind.

> we define a *node* as a set consisting of a governor and all of the subordinates that are directly or indirectly dependent on the governor and that the governor in a sense links together into a bundle. (Tesniere 2015: 6).

Latter in the book, he uses the term node to mean merely a vertex and even redefines it saying that "The node is nothing more than a geometric point whereas the nucleus is a collection of multiple points ... " (Tesniere 2015: 39). It is perhaps the inconsistent use of the terminology that lead to the assumption that the dependency grammar does not recognises phrases (i.e. that is the complete subtree of a vertex). In fact he defines nucleus as playing the role of both a semantic and syntactic unit.

> We define the nucleus as the set which joins together, in addition to the structural node itself, all the other elements for which the node is the structural support, starting with the semantic elements. (Tesniere 2015: 38).

A notable contribution to the field of syntax is the concept of *valency*. It is the notion used in other linguistic schools as *transitivity* to express combinatorial properties of verbs and other lexical items. Inspired from natural sciences, Tesniere compares the relationship between verbs and the so called *actants* (a.k.a. *arguments*) to atom's bonds.

> The verb may therefore be compared to a sort of atom, susceptible to attracting a greater or lesser number of actants, according to the number of bonds the verb has available to keep them as dependents. The number of bonds a verb has constitutes what we call the verb's *valency* (Tesniere 2015: 241).

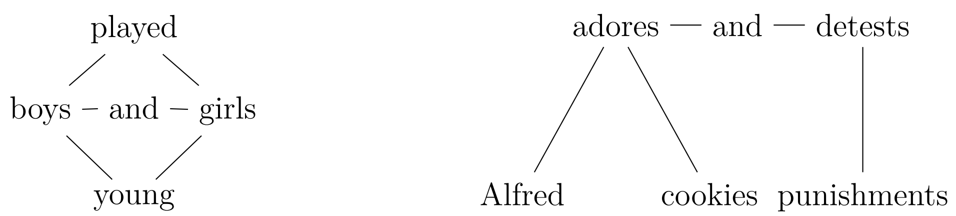
Atoms are not the only metaphor he uses and next I present another one regarding the *verbal node* that is especially important for showing the syntax-semantics interplay.

> The verbal node, found at the centre of the majority of European languages, is a theatrical performance. Like a drama, it obligatorily involves a *process* and most often *actors* and *circumstances*. [...] Transferred from the theatre to structural syntax, the process, the actors, and the circumstances become respectively the *verb*, the *actants*, and the *circumstants* (Tesniere 2015: 97).

Comparison of the verb to an atom seems to emphasize connection to the syntactic aspect of valency while comparing it to a theatrical performance seems to emphasize the semantic properties of valency. Therefore his theory of valency has semantic and syntactic properties. He believed that the first actant is the agent of the action, identified as the subject in traditional grammar, and the second actant is the one that bears the action, identified as the syntactic object. Tesniere regards both of them as complements to complete the governor verb making, in this sense, the subject indistinguishable from other complements.

There are some phenomena that are deemed quite problematic, namely they are the *coordination* or *apposition*. They constitute a challenge because they are not governor-subordinate relations but are rather orthogonal relations among siblings. Tesniere analyses the coordination, or as he calls it *junction*, as a phenomena used in language to express (semantic) content efficiently.

He viewed the junction as fundamentally different from the subordination and represented it with horizontal lines. Subordination is a principle of organization on the vertical axis whereas the coordination (i.e. junction) on the horizontal axis. Figure 1.3 depicts two example representations for the sentence "Young boys and girls played" and "Alfred adores cookies and detests punishments".



(a) Young boys and girls played (b) Alfred adores cookies and detests punishments

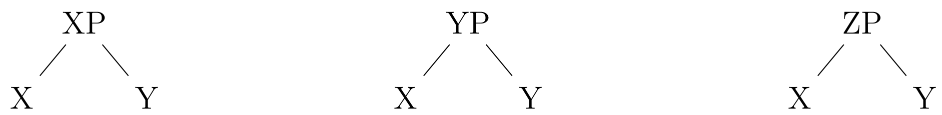
Fig. 1.3 Sample stemmas with *junction* representation

A big part of the Tesniere's *Elements* (Tesniere 1959) is dedicated to the theory of *transfer*. It describes the phenomena when one class of a syntactic unit occupies a position usually devoted to another one. In SFL it is called the grammatical metaphor defined in **??**. For example the noun can be transferred to an adjective by preposition "of", as for example *a linguist of France* where the source *France* is transferred to target *of France* which modify *linguist* that is typically an adjectival function. Transfer is a tool that explains how for example a clause can be embedded into another one or how a verb can be subordinate to another one.

Tesniere splits the words into *function words* or *translatives* (i.e. prepositions, conjunctions, auxiliary verbs and articles) and four basic categories of *content words* (i.e. verbs (I), nouns (O), adverbs (E) and adjectives (A)). The former are empty of content marker transfer of content words from one syntactic category to another one. That is, allowing one word to occupy a position that is generally associated with a word of another category.

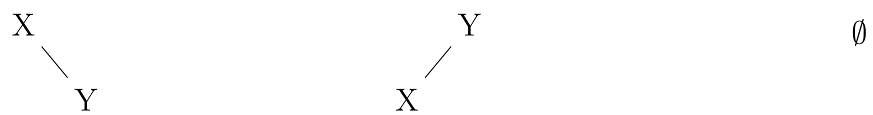
One distinguishing trait of the transfer is that the words transferred from source to target category continue to behave as the source category with respect to their dependants and as source category to its governor.

The transfer theory is controversial for the translators of the Elements. They write (Tesniere 2015: liv-lx) that while the transfer schema can not be interpreted in terms of pure dependency it is debatable whether it can be interpreted in terms of constituency. The main distinction is in the number of nodes that one assumes to be in the syntactic structure i.e. whether there are intermediary virtual nodes.



(a) Headed endocentric (b) Headed endocentric (c) Non-headed exocentric

Fig. 1.4 Constituency structure



(a) Headed endocentric (b) Headed endocentric (c) Non-headed exocentric

Fig. 1.5 Dependency structure

Figure 1.4 shows how a sequence of two elements X and Y can be represented in terms of constituency where the Figure 1.4a and 1.4b represent that one element governs the other called *endocentric* structures and in Figure 1.4c a non-headed structure called *exocentric*. Dependency structure depicted below in Figure 1.5, in contrast, cannot represent non-headed structures. Hence there is no correspondent dependency representation to Figure 1.4c in Figure 1.5c.

Bringing back the discussion on the number of nodes, the constituency structure requires three nodes each time whereas dependency structure only two. In this sense the transfer schemas provided by Tesniere in his Elements (Tesniere 1959) resembles constituency structure more than dependency structure simply because it assumes more nodes than words.

1.2 Evolution into the modern dependency theory

Nowadays the dependency theory differs from the original one presented by Tesniere. At the time the original text was written there was no such distinction as dependency and constituency structures and that Tesniere's Elements (Tesniere 1959) in fact contains descriptions of and references to what may nowadays be considered constituency. Next I present which of the initial ideas did not take hold, were not addressed or merely assumed and instead have evolved into the modern dependency theory of grammar.

1.2.1 Definition of dependency

Tesniere's definition of dependency is not satisfiable. His mentalist approach that "the mind perceives connections between the word and it's neighbours"(Tesniere 2015: 3) makes it impossible to falsify his choices hence leaving no means to validate one choice over the other ones.

One way to define dependency relations and structure is by employing the constituency concept. There are efforts by (Bloomfield 1933; Hockett 1958; Harris 1951) in constituency grammar to identify constituents using tests that shed light on which segments should hold together as phrases or whether they should be considered constituents at all. One needs to decide within each constituent which word it is being headed by which means deciding which word controls the distribution of that constituent (Bloomfield 1933; Zwicky 1985) . A word y depends of a word x if and only if y heads the a phrase which is an immediate constituent of the phrase headed by x (Lecerf 1961).

Another way to define dependencies, avoiding constituency, is by using combinations of two words as proposed by Garde (1977) and Mel'čuk (1988). To discern which governs the other one needs to determine which determined the distribution of the two together. This way the governor is the word that determines the environment in which the two together can appear (Tesniere 2015: lxi). In fact the word notion is not necessary to define dependency, it can be abstracted away to the notion of syntactic units. As soon as two units combine one can posit dependency between them whereby the dependency structure is the set of dependencies between the most granular syntactic units (Gerdes & Kahane 2013).

In addition Tesniere did not make distinctions between the dependency types. As discussed in the previous section, he had noticed that there is a difference between syntactic and semantic dependencies and that the former generally corresponds to the latter but not as a strict rule and even some other times the correspondence is in the opposite direction e.g. "the stone frees" vs. "the frozen stone". The dependency based semantic representations have been around since '60s named *semantic networks* (Žolkovskij & Mel'čuk 1967; Mel'čuk 1988) and *conceptual graphs* (Schank 1969; Sowa 1976).

1.2.2 Grammatical function

In the modern linguistics the notion of grammatical functions e.g. subject, object, determiner etc. are attached to the notion of syntactic dependency. They are in fact an essential account in the modern dependency-based approaches because they are the only way to distinguish between various roles the dependents play in relation to their governors. The grammatical functions attached to the dependency relations are primitives of the dependency grammars. This is not the case for Chomskian phrase structure constituency where the functions are derived from the structural configurations. Nevertheless in latter constituency models such as *Lexical Functional Grammars* (Bresnan 2000) and *Head-Driven Phrase Structure* (Pollard & Sag 1994) have introduces the grammatical functions as grammatical primitives.

The grammatical functions were not important in Tesniere's theory. He mentioned only, in the context of valency theory, three *actant functions* called *first*, *second* and *third* the other verb dependants being *circumstantial*. Most dependecy grammars assume dozens of functions to offer a fine-grained syntactic characterization of language based on distinguishable syntactic properties. This way two elements have the same grammatical function if and only if they have the same *markers, order* (linear position), *agreement properties* and *distribution*. Several grammatical function sets have been developed in the fields of formal dependency grammars, parsers and tree-banks. The most important ones for English Language are the ones of Mel'čuk & Pertsov (1986), of Johnson & Fillmore (2000) and of Marneffe & Manning (2008a,b). In this work is employed the latter as it is part of the Stanford dependency parser described latter in this Chapter.

1.2.3 Projectivity

Central to how the word order is accounted for in dependency grammar is *projectivity*. It is not present in the Elements but it is the basis for identifying *long-distance dependencies* also known as *discontinuities* or *gapping*. The concept is introduced by Lecerf (1961) following publication of the Elements (Tesniere 1959). It is defined in terms of crossing lines when drawing dependency trees where the ones without containing crossing lines are called *projective* and the ones with crossing lines are called *non-projective* i.e. violating projection principle.

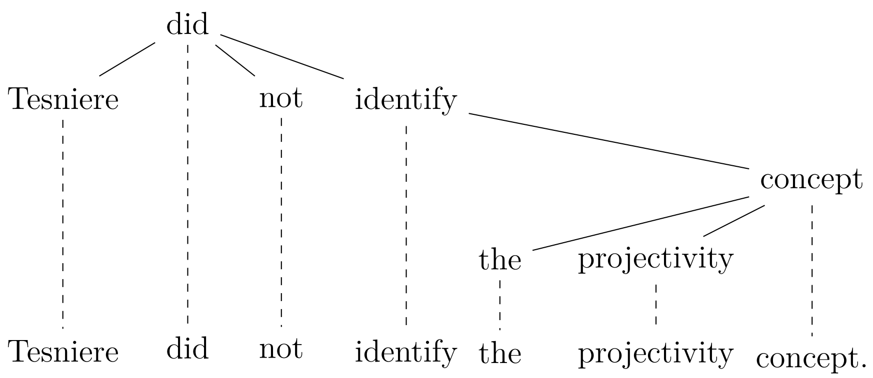


Fig. 1.6 Projective tree

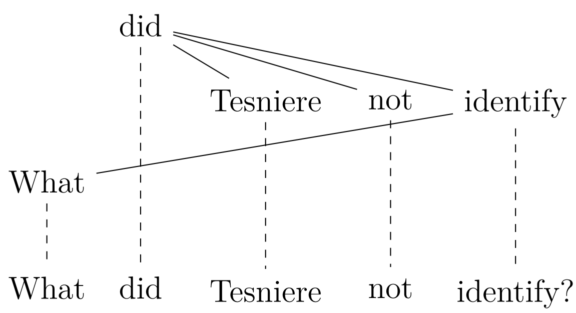
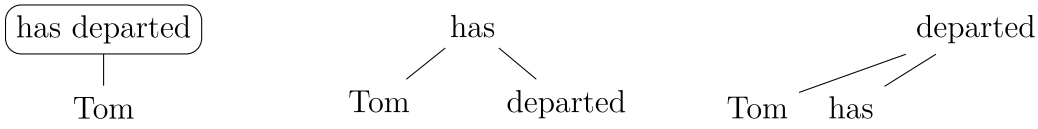


Fig. 1.7 Non-projective tree

To illustrate this principle consider Figure 1.6 where there are no crossing lines whereas Figure 1.7 contains projectivity violation because the word "what" is connected to it's governor "identify" crossing three dashed projection lines. Linguistic phenomena involving non-projecting are: wh-fronting, topicalization, scrambling, and extrapolation.

1.2.4 Function words

Tesniere's transfer theory, despite it's insightfulness, has little if any at all application in modern dependency grammar. The main reason is the implications it has on the hierarchical structure because it does not provide the *translatives* (prepositions, auxiliary verbs, sub-ordinators and conjunctions) with autonomy but a kind of secondary status and thus cannot be constitutive of a nucleus. The issue is reduced to the hierarchical status of such translatives whether they gain node status or not.



(a) (b) (c)

Fig. 1.8 Possible analysis representation for "Tom has departed"

Figure 1.8 represents three possible ways to analyse the word has in "Tom has departed". In Figure 1.8a is represented the original approach Tesniere proposed using transfer schema where the word "has" is enclosed within the full verb node "departed". The two together are granted the status of a dissociated nucleus which means that neither alone can form a nucleus. In contrast the Figure 1.8b and 1.8c the auxiliary has is granted autonomy and corresponds to the modern analysis varying from one model to the other.

As we will see in the next Section, the Stanford dependency schema (Marneffe & Manning 2008a,b) adopts the content words as governors of the function words. This corresponds to the representation in Figure 1.8c. Moreover it provides a collapsed schema where the function words are suffixed to the grammatical functions. For example in "Bob and Jacob" there is a "conj" dependency relation between Jacob and Bob and a "cc" relation between "and" and "Bob". In the collapsed form the relation becomes "conj:and" between between Jacob and Bob integrating the conjunction into the relation name. This is the case for prepositions and conjunctives whereas auxiliary verbs remain nodes in the collapsed form.

1.3 Dependency grammar in automated text processing

Tesniere had no intention in providing a computational theory of grammar and he was neither aware that ideas he was proposing have such potential. Shortly after his death, inspired by Chomsky's Syntactic Structure (Chomsky 1957b), Hays (1960, 1964) makes the first attempts to formalise the dependency grammar with intention to apply it to automated text processing. A year latter his colleague Gaifman (1965) proofs that the *dependency grammar* formalism proposed by Hays is equivalent to Chomsky's *context free grammar* and to *categorial grammars* proposed by Bar-Hillel (1953).

Outshined by Chomskyan grammars, the serious developments in parsing with dependency grammars did not come into being until mid '90s. First efficient parser with the dependency-based model called *Link Grammar* was created by Sleator & Temperley (1995) and ten year latter the dependency parsing gain in popularity yielding remarkable results such as the MaltParser (Nivre 2006; Nivre et al. 2007b), MATE parser (Bohnet 2010) and early Stanford parser (Marneffe et al. 2006) that was generating the dependency trees from phrase structure trees. A summary of dependency parsing techniques is provided by Kübler et al. (2009).

In parallel to parsers, large annotated corpses and treebanks have been developed for parser training and testing and suitable as well for theoretical applications. A treebank is a collection of records consisting of natural language sentences associated with corresponding syntax tree (using a specific grammatical model) and optionally additional annotations such as part of speech tags, named entities, and other annotations. The first treebank was Penn Treebank (Santorini 1990; Marcus et al. 1993) which is a constituency-base treebank. A well known dependency treebank is the Prague Dependency Treebank (Hajic et al. 2001; Böhmová et al. 2003) originally created for Czech but now containing English as well. Recently started an initiative to create a Universal Dependency model (Nivre 2015) and correspondingly with extended efforts was also created a multilingual treebank applying the scheme (Nivre et al. 2016) which continues growing today.

Before arriving to the broadly accepted Universal Dependency, early dependency grammars were quite dispersed. The schemes more often were developed in the context of corpus annotation. An early work (Carroll et al. 1998) towards unification was within the Grammar Evaluation Interest Group (Harrison et al. 1991) also know as *PARSEVAL* initiative that was originally destined for constituency parsers. Carroll et al. (1999) proposed an application independent corpus annotations scheme (see Figure 1.9) specifying the syntactic dependency which holds between each head and its dependent(s) that took into account language phenomena in English, Italian, French and German.

In early 2000 the existing treebanks were still inadequate for evaluating the predicate-argument structure of English clauses. To address this problem, PARC 700 treebank

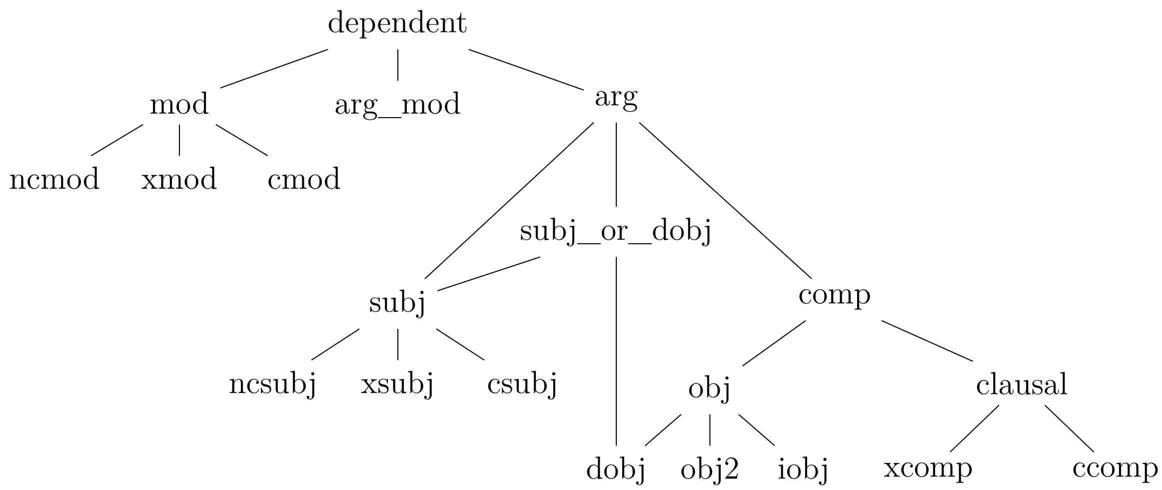


Fig. 1.9 The grammatical relations (GR) hierarchy from Carroll et al. (1999)

(King & Crouch 2003) was created by randomly extracting 700 sentences from Penn treebank, parsed with a Lexical Functional Grammar (LFG), converted into dependency relations and manually corrected by human validators. This scheme has played role in creation of Stanford dependency model that I describe in detail latter.

One advantage of dependency representations is that they can be encoded in a tabular format such as CoNLL (Nivre et al. 2007a) which now is adopted as the standard representation. It is employed in a recurring open competition called "CoNLL shared task" launched for improving and innovating the dependency parsing methods. The most notable are the ones from 2006 on dependency parsing (Buchholz & Marsi 2006) followed in 2007 that included a track for multilingual and one for domain specific dependency parsing. Fast forward to 2017 (Zeman et al. 2017) the task was for parsing from raw text (as previous ones were lemmatised and annotated with part of speech) into universal dependency.

1.4 The Stanford dependency model

The functional dependency descriptions is precisely the aspect which makes possible the beneficial link between the Stanford Dependency Grammar and the Systemic Functional structures targeted in the current thesis.

Stanford parser is one of the leaders in the domain of dependency parsing. Since 2006 (Marneffe et al. 2006) for ten years Stanford parser implemented the Stanford dependency model for English (and a few other languages). Then in 2015 Nivre et al. (2016) proposes the language independent Universal Dependency scheme. In this section I present the Stanford dependency model (prior to Universal Dependency) that is used in the current parser.

The design of the Stanford dependency set (Marneffe et al. 2006; Marneffe & Manning 2008a; Marneffe et al. 2014; Silveira et al. 2014) bears a strong intellectual debt to the framework of Lexical Functional Grammars (Bresnan 2000) from which many relations were adopted. Marneffe et al. (2006) departs from the relation typology described in (Carroll et al. 1999) which was employed in PAREVAL initiative (Harrison et al. 1991) and from the grammatical relations of PARC 700 (King & Crouch 2003) scheme following a style of Lexical Functional Grammar. Marneffe arranges the grammatical relations into a hierarchy rooted in a generic relation *dependent*. This is then classified into a more fine-grained set of relations thet may hold between a head and its dependent following the set of principles (Marneffe & Manning 2008b) stipulated in Generalization 1.4.1.

Generalization 1.4.1 (Design principles for Stanford dependency set)**.**

1. Everything is represented uniformly as binary relation pairs of words.
2. Relations should be semantically contentful and useful to NLP applications.
3. Where possible, relations should use the notions of traditional grammar (Quirk et al. 1985) for easier comprehension by users.
4. To deal with text complexities underspecified relations should be available.
5. When possible content words shall be connected directly, not indirectly mediated by function words (prepositions, conjunctions, auxiliaries, etc.).

When motivating the approach to schema development, Marneffe et al. (2006) insists on practical rather than theoretical concerns proposing that structural configurations be defined as grammatical roles (to be read as grammatical functions)(Marneffe et al. 2006). In the Chomsky tradition Chomsky (1981) the grammatical relations are defined structurally as configurations of phrase structure. Other theories such as Lexical-Functional Grammar reject the adequacy of such an approach (Bresnan 2000) and advocate a functional representation for syntax at the atomic level. Following the latter approach, she insists that information about functional dependencies between words is very important and shall be explicitly available in the dependency tree.

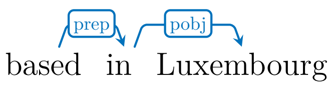
The advantage of explicit relations is that the predicate-argument relations are readily available as edge labels in the dependency structure and can be used off the shelf for real world applications which was an important goal in the schema design.

The grammar had to be suitable for parsing within the context of syntactic pattern learning (Snow et al. 2005), relation extraction, machine translation, question answering and inference rule discovering (Lin & Pantel 2001), domain specific parsing (Clegg & Shepherd 2007), and others. The complete set of dependency relations is Appendix ??.

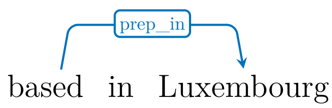
1.5 Stanford Parser

The Stanford Dependency Parser generates four types of dependency representations. It produces parse trees with *basic dependencies*, *collapsed dependencies* and *collapsed dependencies with propagation of conjunct* that are not necessarily a tree structure and finally the *collapsed dependencies that preserve a tree structure*. The variant employed in the current work is the collapsed dependencies with propagation of conjunct. This structure concerns preposition, conjunction and relative clause referent nodes, and is generated by a series of transformations after the initial basic dependency parse is ready.

For example, consider fragment "based in Luxembourg". In basic dependency representation, such as is shown in Figure 1.10a, the function words are governing the content words and thus there is a preposition (prep) edge from "based" to a dependent preposition "in" from from which continues a preposition object edge (pobj) to "Luxembourg". In collapsed dependency representation the relation sequences of the type "prep-pobj" are replaced by a direct edge between the two content words labelled with "prep" function concatenated with the intermediary preposition as can be seen in Figure 1.10b. There is a single relation between "based" and "Luxembourg" labelled "prep_in". Similar transformations are done for conjunctions.



(a) Basic (uncollapsed) preposition dependency

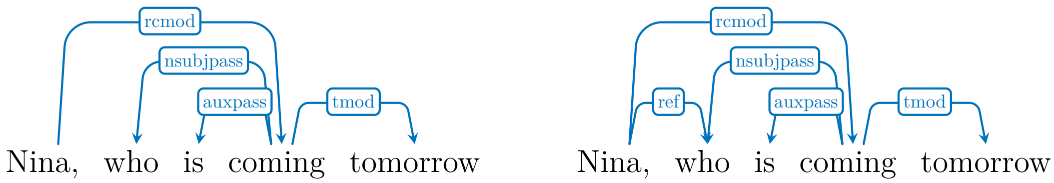


(b) Collapsed preposition dependency

Fig. 1.10 Function words in Stanford dependency model

Besides collapsing prepositions and conjunctions the dependency structure is further processed to introduce more relations even if they break the tree structure. This is the reason why often in this thesis the references ar to dependency graphs and not trees. In fact the fundamental assumption here is that the dependecy structures are graphs with a root node. I further develop this aspect in Chapter ??.

The relative clauses is such a case where the tree structure is broken. Consider Figure 1.11a where the relative clause is introduced by a relative clause modifier relation (rcmod) from the noun "Nina" to the main verb of the relative clause "coming". The clause contains an interrogative pronoun "who" functioning as passive subject (subjpass) and which anaphorically resolves to the clause governor "Nina". This sort of information about the antecedent of the relative clause is also introduced in the collapsed dependency representation. And thus, as depicted in Figure 1.11b, a new referent relation is added connecting "Nina" to the subordinate subject "who" of the relative clause.



(a) Basic (uncollapsed) relative clause (b) Collapsed relative clause

Fig. 1.11 Relative clause in Stanford dependency model

Relations like *ref* introduce cycles but add valuable information useful in various stages of further processing. However, ensuring a tree structure in important for the CG creation stage because it is based on a top down traversal. This is taken care of in the preprocessing stage described in the next stage.

1.6 Penn part-of-speech tag set

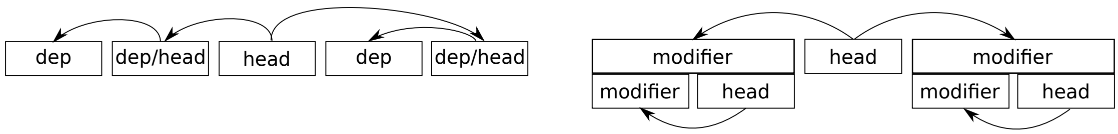
Stanford dependency parser starts creation of the parse structure process from the list of tokens annotated with Penn part-op-speech tags. Embedded into the dependency graph, these tags are the part of the syntactic context from which SFG constituency graph is built.

1.7 Cross theoretical bridge from DG to SFL

The concept of dependency between pairs of words is long acknowledged in linguistic communities. In traditional terms dependencies are treated as *hypotactic expansions* (see Definition **??**) of word classes (or parts of speech) where the expanded word acts as *heads* and expanding ones as *dependent* establishing parent-daughter structural relations illustrated in Figure 1.12a.

In SFL community the concept of dependency is less salient than the foundational role it plays in the Dependency Grammars. Dependencies are regarded as orthogonal relations between sibling elements of a unit (Figure 1.12b) and link the *heads* to their *modifies* in by Hallidayan *logical structure*(Halliday & Matthiessen 2013).

Figure 1.12 illustrates side by side the parent-daughter and sibling dependency relations. In Figure 1.12a dependency are the only relations between the units of structure whereas in Figure 1.12b there are multiple levels (ranks) of units and the dependency relations are relevant only between siblings at the same level within the structure of an unit. SFL regards dependency relations holding only between elements of a unit whereas the relations that connect the units of lower and higher rank are *constituency relations*. Yet when we look at the two structures they resemble in a way each other and next I show how.



(a) The parent-daughter relations in DG (b) The head-modifier relations in SFG

Fig. 1.12 The dependency relations in DG and SFG

In a nutshell, the parent-daughter dependency relations in dependency grammar unpack into multiple function in systemic functional grammar and specifically it is the head-modifier indirect relation, unit-element componence relation and the head of unit a "representativeness" function.

This difference implies that, when translated to a constituency unit (described in Section **??**), the dependency unit, stands for both a unit and that unit's head element. In other words a DG node corresponds to two functional elements at different rank scales. For example the root verb in dependency graph corresponds to the clause node and the lexical item which fills the Main Verb of the clause. By analogy, the head noun of a Nominal Group anchors the entire unit (as a group) and fills the head element of the group.

text	**some**	**very**	**small**	**wooden**	**ones**
units	**Nominal Group**				
elements	*Quantifying Determiner*	*Modifier*		*Modifier*	*Head*
units		**Quality Group**			
elements		*Degree Tamperer*	*Apex*		

Table 1.1 SF analysis of Example **??** (reproduced from Table **??**)

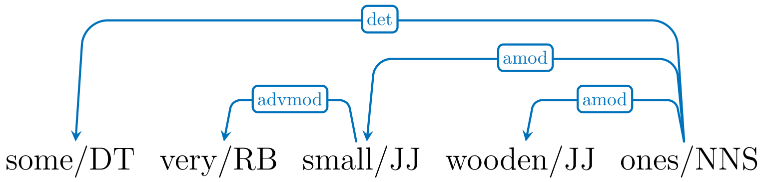


Fig. 1.13 Dependency analysis for Table 1.1

Figure 1.13 and the Table 1.1 represent the analysis of a nominal group from Example **??** ("some small very small wooden ones") in Cardiff grammar and Stanford dependency grammar exhibiting a contrast of the two structures. Consider the dependency relation "det" a link between the noun "ones" and the determiner "some". When translated into SF variant the dependency relation stands within Nominal Group between the Head element (filled by word "ones") and the Quantifying Determiner element (filled by the word "some"). By definition all elements in a unit are equal in the structure so the Head and Quantifying Determiner are siblings. So the items (words) filling those elements are sibling. How is then the dependency relation established?

In SFL there is the concept of Head and Modifier. There is no direct relationship between them but it is said that what the Modifier modifies is the Head. The relation is not a direct one, the Modifier and Head stand for two different kinds of meaning and what the Modifier modifies is not the Head per se but the referent denoted by the head (and thus construed by the entire unit). It is precisely this modification of the head that is called a (sibling) dependency relation and is seldom mentioned in SFL literature because it is considered implicit and recoverable from the SF constituency structure.

The Head also is the element that anchors the entire unit and plays a constitutive role. In this sense the word "ones" realizes not only the Head function (sided with Determiner "some") but also anchors the entire unit. The relation between the group and it's elements is one of *componence* (Definition **??**) described in Section **??**. Yet in the role of unit anchor we cannot say that there is a componence relation between "one" and "some" because it is merely a proxy to the referent rather than the entire unit. So in this role "one" can be said to be standing in a parent-daughter dependency relation to "some" incorporating the filling and componence relations.

I just showed how the dependency relation in dependency structure (Figure 1.12a) can be unpacked into two relations in systemic functional structure (Figure 1.12b): sibling dependency considered an indirect relation between Head and Modifier (Logical Metafunction) and parent-daughter dependency between unit anchor and the compounding elements, relation which resembles unit componence but is not.

Lets look at a second example of two relations “advmod” from “small” to “very” and “amod” from “ones” to “small”. The interesting case here is the item “small” which is the Head of the Quality Group, it anchors the meaning of the whole group and the Quality group fills the Modifier element within the Nominal group. What is not covered in previous example is that the Apex “small” not only is a representative of the entire group but it also is a *representative filler* of the Modifier element within Nominal Group. Using the similar translation mechanism as above, this means that, the incoming dependency needs to be unpacked into three levels: the element within the current group (Modifier), the unit class that element is filling (Quality Group) and finally the head of the filler group (Apex). In fact, to be absolutely correct there is one more level. The elements of a unit are expounded by lexical items, so fourth relation to unpack is the expounding of the Apex by “small” word.

In this section I laid the theoretical principle for transforming the dependency structure into systemic functional structure. In practice to achieve this level of unpacking the algorithm requires a bottom up and a top down contextualization in terms of elements of structure within a unit and realised sequence of textual units. This implied that the unpacking needs two traversals, a bottom-up and a top one. More on that and the exact algorithm for the translation is provided in Section **??**.

Next follows the chapter on Governance and Binding Theory needed to account for the unrealised, covert (Null) Elements in the syntactic structure. It is also an opportunity to perform a similar theoretical translation exercise (as in this section) from one theory of grammar into another.

References

Bar-Hillel, Yehoshua. 1953. A quasi-arithmetical notation for syntactic description. *Language* 29. 47–58. Reprinted in Y. Bar-Hillel. (1964). *Language and Information: Selected Essays on their Theory and Application*, Addison-Wesley 1964, 61–74.

Bateman, John A. 2008. Systemic-Functional Linguistics and the Notion of Linguistic Structure: Unanswered Questions, New Possibilities. In Jonathan J. Webster (ed.), *Meaning in context: Implementing intelligent applications of language studies*, 24–58. Continuum.

Bloomfield, Leonard. 1933. *Language*. New York and London: Henry Holt and Co. and Allen and Unwin Ltd.

Böhmová, Alena, Jan Hajič, Eva Hajičová & Barbora Hladká. 2003. *The prague dependency treebank* 103–127. Dordrecht: Springer Netherlands. doi:10.1007/978-94-010-0201-1_7. <https://doi.org/10.1007/978-94-010-0201-1_7>.

Bohnet, Bernd. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd international conference on computational linguistics* COLING '10, 89–97. Stroudsburg, PA, USA: Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=1873781.1873792>.

Bresnan, Joan. 2000. *Lexical-functional syntax*. Wiley.

Buchholz, Sabine & Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning* CoNLL-X '06, 149–164. Stroudsburg, PA, USA: Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=1596276.1596305>.

Butler, Christopher. 1985. *Systemic linguistics: Theory and applications.* Batsford Academic and Educational.

Carroll, John, Ted Briscoe & Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of the international conference on language resources and evaluation*, 447–454.

Carroll, John, Guido Minnen & Ted Briscoe. 1999. Corpus Annotation for Parser Evaluation. In *Proceedings of the eacl workshop on linguistically interpreted corpora (linc)* June, 7. Association for Computational Linguistics. <http://arxiv.org/abs/cs/9907013>.

Chomsky, Noam. 1957a. *Syntactic Structures*. Mouton & Co.

Chomsky, Noam. 1957b. *Syntactic structures*. The Hague: Mouton.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Foris.

Clegg, Andrew B & Adrian J Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC bioinformatics* 8(1). 24.

Fawcett, R. 1988. Language Generation as Choice in Social Interaction. In Zock, M. & G. Sabah (eds.), *Advances in natural language generation*, vol. 2, 27–49. Pinter Publishers. (Paper presented at the First European Workshop of Natural Language Generation, Royaumont, 1987).

Fawcett, Robin. 2000. *A Theory of Syntax for Systemic Functional Linguistics*. John Benjamins Publishing Company paperback edn.

Fawcett, Robin P. 1990. The COMMUNAL project: two years old and going well. *Network: news, views and reviews in systemic lingustics and related areas* 13/14. 35–39.

Fawcett, Robin P. 1993. The architecture of the COMMUNAL project in NLG (and NLU). In *The Fourth European Workshop on Natural Language Generation*, Pisa.

Fawcett, Robin P. 2008. *Invitation to Systemic Functional Linguistics through the Cardiff Grammar*. Equinox Publishing Ltd.

Firth, J.R. 1957. A synopsis of linguistic theory 1930-1955. *Studies in linguistic analysis* 1–32. <http://www.bibsonomy.org/bibtex/25b0a766713221356e0a5b4cc2023b86a/glanebridge>.

Gaifman, Haim. 1965. Dependency systems and phrase-structure systems. *Information and Control* 8(3). 304 – 337. doi:https://doi.org/10.1016/S0019-9958(65)90232-9. <http://www.sciencedirect.com/science/article/pii/S0019995865902329>.

Garde, Paul. 1977. Ordre lineaire et dependance syntaxique. contribution a une typologie. In *Bulletin de la societe de linguistique de paris*, vol. 1 72, 1–26. Paris.

Gerdes, Kim & Sylvain Kahane. 2013. Defining dependencies (and constituents). *Frontiers in Artificial Intelligence and Applications* 258. 1–25.

Hajic, Jan, Eva Hajicová, Petr Pajas, Jarmila Panevová, Petr Sgall & Barbora Vidová-Hladká. 2001. Prague dependency treebank 1.0 (final production label). cdrom cat: Ldc2001t10. Tech. rep. ISBN 1-58563-212-0.

Halliday, Michael A. K. 1961. Categories of the theory of grammar. *Word* 17(3). 241–292. Reprinted in abbreviated form in Halliday (1976) edited by Gunther Kress, pp 52-72.

Halliday, Michael A. K. 1994. *An Introduction to Functional Grammar*. London: Edward Arnold 2nd edn.

Halliday, Michael A. K. 1997. Linguistics as metaphor, 3–27. Continuum.

Halliday, Michael A. K. 2003. Ideas about language. In Michael A. K. Halliday & Jonathan J. Webster (eds.), *On language and linguistics. Volume 3 of collected works of M.A. K. Halliday*, 490. New York: Continuum.

Halliday, Michael A.K. 2002. Categories of the theory of grammar. In Jonathan Webster (ed.), *On grammar (volume 1)*, 442. Continuum.

Halliday, Michael A.K. & Christian M.I.M. Matthiessen. 2013. *An Introduction to Functional Grammar (4^{th} Edition)*. Routledge 4th edn.

Harris, Z.S. 1951. *Methods in structural linguistics* Methods in Structural Linguistics. University of Chicago Press. <https://books.google.lu/books?id=a6nYjgEACAAJ>.

Harrison, Philip, Steven Abney, Ezra Black, Dan Flickinger, Claudia Gdaniec, Ralph Grishman, Donald Hindle, Robert Ingria, Mitch Marcus, Beatrice Santorini & Tomek Strzalkowski. 1991. Evaluating syntax performance of parser/grammars. In *Proceedings of the natural language processing systems evaluation workshop, berekely, ca, june 1991* Rome Laboratory Technical Report, RL-TR-91-362, .

Hasan, Ruqaiya. 2014. The grammarian's dream: lexis as most delicate grammar. In Jonathan Webster (ed.), *Describing language form and function*, vol. 5 Collected Works of Ruqaiya Hasan, chap. 6. Equinox Publishing Ltd.

Hays, David G. 1960. Grouping and dependency theories. *Proceedings of the National Symposium on Machine Translation* 2538. 258–266.

Hays, David G. 1964. Dependency theory: A formalism and some observations. *Language* 40(4). 511–525. <http://www.jstor.org/stable/411934>.

Hjelmslev, Louis. 1953. *Prolegomena to a theory of language*. Bloomington, Indiana: Indiana University Publications in Anthropology and Linguistics. Translated by Francis J. Whitfield.

Hockett, Charles F. 1958. *A course in modern linguistics*. New York: Macmillan.

Hudson, Richard. 2010. *An Introduction to Word Grammar*. Cambridge University Press.

Johnson, Christopher & Charles J. Fillmore. 2000. The framenet tagset for frame-semantic and syntactic coding of predicate-argument structure. In *Proceedings of the 1st north american chapter of the association for computational linguistics conference* NAACL 2000, 56–62. Stroudsburg, PA, USA: Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=974305.974313>.

King, Tracy Holloway & Richard Crouch. 2003. The PARC 700 Dependency Bank. In *4^{th} international workshop on linguistically interpreted corpora (linc03)*, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.15.3277>.

Kübler, S., R. McDonald & J. Nivre. 2009. *Dependency parsing* Online access: IEEE (Institute of Electrical and Electronics Engineers) IEEE Morgan & Claypool Synthesis eBooks Library. Morgan & Claypool. <https://books.google.lu/books?id=k3iiup7HB9UC>.

Kucera, Henry & W. Nelson Francis. 1968. Computational Analysis of Present-Day American English. *American Documentation* 19(4). 419. doi:10.2307/302397. <http://search.ebscohost.com/login.aspx?direct=true{&}db=bth{&}AN=16865479{&}login.asp{&}site=ehost-live>.

Lecerf, Yves. 1961. Une representation algebrique de la structure des phrases dans diverses langues naturelles. In *Compte rendu de l'academie des sciences de paris* 252, 232–234. Academie des Sciences.

Lemke, Jay L. 1993. Discourse, dynamics, and social change. *Cultural Dynamics* 6(1-2). 243–276.

Lin, Dekang & Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering* 7(4). 343–360.

Mann, William C. 1983. An overview of the PENMAN text generation system. In *Proceedings of the National Conference on Artificial Intelligence*, 261–265. AAAI. Also appears as USC/Information Sciences Institute, RR-83-114.

Mann, William C. & Christian M. I. M. Matthiessen. February 1983. A demonstration of the Nigel text generation computer program. In *Nigel: A Systemic Grammar for Text Generation*, USC/Information Sciences Institute, RR-83-105. This paper also appears in a volume of the *Advances in Discourse Processes Series*, R. Freedle (ed.): *Systemic Perspectives on Discourse: Volume I.* published by Ablex.

Marcus, Mitchell P, Beatrice Santorini & Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2). 313–330. doi:10.1162/coli.2010.36.1.36100. <http://portal.acm.org/citation.cfm?id=972470.972475>.

Marneffe, Marie-Catherine, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre & Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the ninth international conference on language resources and evaluation (lrec-2014)(vol. 14)*, European Language Resources Association (ELRA). <http://www.lrec-conf.org/proceedings/lrec2014/pdf/1062{_}Paper.pdf>.

Marneffe, Marie-Catherine, Bill MacCartney & Christopher D Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Lrec 2006*, vol. 6 3, 449–454. Stanford University. <http://nlp.stanford.edu/manning/papers/LREC{_}2.pdf>.

Marneffe, Marie-Catherine & Christopher D. Manning. 2008a. Stanford typed dependencies manual. Tech. Rep. September Stanford University. <http://nlp.stanford.edu/downloads/dependencies{_}manual.pdf>.

Marneffe, Marie-Catherine & Christopher D. Manning. 2008b. The Stanford typed dependencies representation. *Coling 2008 Proceedings of the workshop on CrossFramework and CrossDomain Parser Evaluation CrossParser 08* 1(ii). 1–8. doi:10.3115/1608858.1608859. <http://portal.acm.org/citation.cfm?doid=1608858.1608859>.

Matthiessen, Christian M. I. M. 1995. *Lexicogrammatical cartography: English systems*. Tokyo, Taipei and Dallas: International Language Science Publishers.

Matthiessen, Christian M. I. M. & John A. Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. London and New York: Frances Pinter Publishers and St. Martin's Press.

Mel'čuk, Igor. 1988. Semantic description of lexical units in an Explanatory Combinatorial Dictionary: basic principles and heuristic criteria. *International Journal of Lexicography* 1. 165–188.

Mel'čuk, Igor A. & Nikolaj V. Pertsov. 1986. *Surface syntax of english*. John Benjamins Publishing. <http://www.jbe-platform.com/content/books/9789027279378>.

Moravcsik, Edith A. 2006. *An Introduction to Syntactic Theory*. Continuum paperback edn.

Nivre, Joakim. 2006. *Inductive dependency parsing (text, speech and language technology)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Nivre, Joakim. 2015. Towards a universal grammar for natural language processing. In *Computational Linguistics and Intelligent Text Processing*, 3–16. Springer.

Nivre, Joakim, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel & Deniz Yuret. 2007a. The conll 2007 shared task on dependency parsing. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (emnlp-conll)*, 915–932.

Nivre, Joakim, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov & Erwin Marsi. 2007b. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13(2). 95–135. doi:10.1017/S1351324906004505. <https://doi.org/10.1017/S1351324906004505>.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty & Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the tenth international conference on language resources and evaluation (lrec 2016)*, Paris, France: European Language Resources Association (ELRA).

O'Donoghue, Tim. 1991. The Vertical Strip Parser: A lazy approach to parsing. Tech. rep. School of Computer Studies, University of Leeds.

Penman Project. 1989. PENMAN documentation: the Primer, the User Guide, the Reference Manual, and the Nigel Manual. Tech. rep. USC/Information Sciences Institute Marina del Rey, California.

Pollard, Carl J. & Ivan A. Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech, Jan Svartvik & David Crystal. 1985. *A comprehensive grammar of the English language*, vol. 1 2. Longman. <http://www.amazon.com/dp/0582517346http://journals.cambridge.org/production/action/cjoGetFulltext?fulltextid=2545152>.

Radford, Andrew. 1997. *Syntax: A Minimalist Introduction*. Cambridge University Press.

Saitta, Lorenza & Jean-Daniel Zucker. 2013. *Abstraction in artificial intelligence and complex systems*. Springer-Verlag New York. doi:10.1007/978-1-4614-7052-6. <http://www.springer.com/la/book/9781461470519>.

Santorini, Beatrice. 1990. Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision). *University of Pennsylvania 3rd Revision 2nd Printing* 53(MS-CIS-90-47). 33. doi:10.1017/CBO9781107415324.004. <http://www.personal.psu.edu/faculty/x/x/xxl13/teaching/sp07/apling597e/resources/Tagset.pdf>.

Saussure, Ferdinand de. 1959 [1915]. *Course in General Linguistics*. New York / Toronto / London: McGraw-Hill and the Philosophical Library, Inc. Edited by Charles Bally and Albert Sechehaye, in collaboration with Albert Riedlinger; translated by Wade Baskin.

Schank, Roger. 1969. Conceptual dependency as a framework for linguistic analysis. *Linguistics* 7(49). 28–50.

Silveira, Natalia, Timothy Dozat, Marie-Catherine De Marneffe, Samuel Bowman, Miriam Connor, John Bauer & Chris Manning. 2014. A Gold Standard Dependency Corpus for English. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the ninth international conference on language resources and evaluation (lrec'14)*, European Language Resources Association (ELRA). <http://nlp.stanford.edu/pubs/USD{_}LREC14{_}paper{_}camera{_}ready.pdf>.

Sleator, Daniel Dominic & David Temperley. 1995. Parsing english with a link grammar. *CoRR* abs/cmp-lg/9508004. 93. <http://arxiv.org/abs/cmp-lg/9508004>.

Snow, Rion, Daniel Jurafsky & Andrew Y Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems*, 1297–1304.

Sowa, John F. 1976. Conceptual graphs for a data base interface. *IBM Journal of Research and Development* 20(4). 336–357. doi:10.1147/rd.204.0336. <http://dx.doi.org/10.1147/rd.204.0336>.

Taverniers, Miriam. 2011. The syntax-semantics interface in systemic functional grammar: Halliday's interpretation of the Hjelmslevian model of stratification. *Journal of Pragmatics* 43(4). 1100–1126. doi:10.1016/j.pragma.2010.09.003.

Tesniere, Lucien. 1959. *Elements de syntaxe structurale*. Paris: Klincksieck.

Tesniere, Lucien. 2015. *Elements of Structural Syntax*. John Benjamins Publishing Company translation by timothy osborne and sylvain kahane edn.

Tucker, Gordon H. 1997. A functional lexicogrammar of adjectives. *Functions of Language* 4(2). 215–250.

Tucker, Gordon H. 1998. *The Lexicogrammar of Adjectives: A Systemic Functional Approach to Lexis*. Bloomsbury.

Žolkovskij, Alexander K. & Igor A. Mel'čuk. 1967. O semantičeskom sinteze. *Problemy kibernetiki* 19(?). 177–238.

Zeman, Daniel, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis M. Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jaroslava Hlavácová, Václava Kettnerová, Zdenka Uresová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çagri Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali El-Kahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj & Josie Li. 2017. Conll 2017 shared task: multilingual parsing from raw text to universal dependencies. *Proceedings of the CoNLL Shared Task* 1–19.

Zhang, Niina Ning. 2010. *Coordination in syntax*. Cambridge University Press.

Zwicky, Arnold M. 1985. Heads. *Journal of Linguistics* 21. 1–30.