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

February 2019

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
February 2019

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

Introduction

1.1 On Artificial Intelligence (AI) and Computational Linguistics

In 1950 Alan Turing in a seminal paper (Turing 1950) published in Mind was asking if "machines can do what we (as thinking entities) can do?" He questioned what intelligence was and whether it could be manifested in machine actions indistinguishable from human actions.

He proposed the famous *Imitation Game* also known as the *Turing test* in which a machine would have to exhibit intelligent behaviour equivalent or indistinguishable from that of a human. The test was set up by stating the following rules. The machine (player A) and a human (player B) are engaged in a written *natural language* conversation with a human judge (player C) who has to decide whether each conversation partner is human or a machine. The goal of players A and B is to convince the judge (player C) that they are human.

This game underpins the question whether "a computer, communicating over a teleprinter, (can) fool a person into believing it is human?", moreover, whether it can exhibit (or even appear to exhibit) human(-like) cognitive capacities (Harnad 1992). Essential parts of such cognitive capacities and intelligent behaviour that the machine needs to exhibit are of course the linguistic competences of comprehension (or "understanding") and generation of "appropriate" responses (for a given input from the judge C). The *Artificial Intelligence* (AI) field was born from dwelling on Turing's questions. The term was coined by McCarthy for the first time in 1955 referring to the "science and engineering of making intelligent machines" (McCarthy et al. 2006).

The general target is to program machines to do with language what humans do. Various fields of research contribute to this goal. Linguistics, amongst others, contributes with theoretical frameworks systematizing and accounting for language in terms of morphology, phonology, syntax, semantics, discourse or grammar in general. In computer science increasingly more efficient algorithms and machine learning techniques are developed. Computational linguistics provides methods of encoding linguistically motivated tasks in terms of formal data structures and computational goals. In addition, specific algorithms and heuristics operating within reasonable amounts of time with satisfiable levels of accuracy are tailored to accomplish those linguistically motivated tasks.

Computational Linguistics (CL) was mentioned in the 1950s in the context of automatic translation (Hutchins 1999) of Russian text into English and developed before the field of Artificial Intelligence proper. Only a few years later CL became a sub-domain of AI as an interdisciplinary field dedicated to developing algorithms and computer software for intelligent processing of text (leaving the very hard questions of intelligence and human cognition aside). Besides *machine translation* CL incorporates a broader range of tasks such as *speech synthesis and recognition, text tagging, syntactic and semantic parsing, text generation, document summarisation, information extraction* and others.

This thesis contributes to the field of CL and more specifically it is an advancement in *Natural Language Parsing* (NLP), one of the central CL tasks informally defined as the process of transforming a sentence into (rich) machine readable syntactic and semantic structure(s). Developing a program to automatically analyse text in terms of such structures by involving computer science and artificial intelligence techniques is a task that has been pursued for several decades and still continues to be a major challenge today. This is especially so when the target is *broad language coverage* and even more when the desired analysis goes beyond simple syntactic structures and towards richer functional and/or semantic descriptions useful in the latter stages of *Natural Language Understanding* (NLU). The current contribution aims at a reliable modular method for parsing unrestricted English text into a feature rich constituency structure using Systemic Functional Grammars (SFGs).

In computational linguistics, broad coverage natural language components now exist for several levels of linguistic abstraction, ranging from tagging and stemming, through syntactic analyses to semantic specifications. In general, the higher the degree of abstraction, the less accurate the coverage becomes and, the richer the linguistic description, the slower the parsing process is performed.

Such working components are already widely used to enable humans to explore and exploit large quantities of textual data for purposes that vary from the most theoretical, such as understanding how language works or the relation between form and meaning, to very pragmatic purposes such as developing systems with natural language interfaces, machine translation, document summarising, information extraction and question answering systems to name just a few. Nevertheless there is still a long way to go through before machines excel in these narrowly scoped tasks and even longer before machines start using language in the ways human do.

1.2 Living in a technologically ubiquitous world

The human language has become a versatile highly nuanced form of communication that carries a wealth of meaning which by far transcends the words alone. When it comes to *human-machine* interaction this highly articulated communication form is deemed impractical. So far humans had to learn to interact with computers and do it in a formal, strict and rigorous manner via graphical user interfaces, command line terminals and programming languages. Advancements in *Natural Language Processing* (NLP) are a game changer in this domain. NLP starts to unlocks the information locked in the human speech and make it available for processing to computers. NLP becomes an important technology in bridging the gap between natural data and digital structured data.

In a world such as ours, where technology is ubiquitous and pervasive in almost all aspects of life, NLP becomes of great value and importance regardless of whether it materializes as a spell-checker, an intuitive recommender system, spam filters, (not so) clever machine translators, voice controlled cars, or intelligent assistants such as Siri, Alexa or Google Now.

Every time an assistant such Siri or Alexa is asked for directions to the nearest Peruvian restaurant, how to cook Romanian beef stew or what is the dictionary definition for the word “germane”, a complex chain of operations is activated that allows ‘her’ to understand the question, search for the information you are looking for and respond in a human understandable language. Such tasks are possible only in the past few years thanks to advances in NLP. Until now we have been interacting with computers in a language they understand rather than us. The next challenge is to develop a technology that enables computers to interact with us in a language we understand rather then they.

1.3 NLP for business

NLP opens new and quite dramatic horizons for businesses. Navigating with limited resources stormy markets of competitors, customers and regulators and finding an optimal answer/action to a business question is not a trivial task. In this section I present a few example application areas and use them to discuss tasks that need to be accomplished for NLP in such contexts. These examples underline the ever growing need for NLP putting into perspective the need of ever deeper and richer linguistic analysis across a broad range of domains and applications.

Markets are influenced by the information exchange and being able to process massive amounts of text and extract meaning can help asses the status of an industry and play an essential role in crafting a strategy or a tactical action. Relevant NLP tasks for gathering market intelligence are *named entity recognition* (NER), *event extraction* and *sentence classification*. With these tasks alone one can build a database about companies, people, governments, places, events together with positive or negative statements about them and run versatile analytics to audit the state of affairs.

Compliance with governmental, European or international regulations is a big issue for large corporations. One question for addressing this problem is whether a product is a liability or not and if yes then in which way. Pharma companies for example, once a drug has been released for clinical trials, need to process the unstructured clinical narratives or patient's reports about their health and gather information on the side effects. The NLP tasks needed for this applications are primarily *NER* to extract names of drugs, patients and pharma companies and *relation detection* used to identify the context in which the side effect is mentioned. NER task help transforming a sentence such as "Valium makes me sleepy" to "(drug) makes me (symptom)" and relation detection will apply patterns such as "I felt (symptom) after taking (drug)" to detect the presence of side effects.

Many customers, before buying a product, check online reviews about the company and the product regardless of whether it is pizza or a smartphone. Popular sources for such inquiry are blogs, forums, reviews, social media, reports, news, company websites, etc. All of these contain a plethora of precious information that stays trapped in unstructured human generated text. This information if unlocked can play a great deal in company's reputation management and decisions for necessary actions to improve it. The NLP tasks sufficient to address this business required are *sentiment analysis* to identify attitude, judgement, emotions and intent of the speaker, and *co-reference resolution* which connects mentions of things to their pronominal reference in the following or preceding text. These tasks alone can extract the positive and negative

attitudes from the sentence “The pizza was amazing but the waiter was awful!” and connect it to the following sentence “I love when it is topped with my favourite artichoke”, disambiguating the sentence so that it is clear that it is about pizza and not the waiter and so discover a topping preference.

NLP is heavily used in customer service in order to figure out what a customer means not just what she says. Interaction of companies with their customers contain many hints pointing towards their dissatisfaction and interaction itself is often one of the causes. Companies record, transcribe and analyse large numbers of call recordings for extended insights. They deploy chat bots for increased responsiveness by providing immediate answers to simple needs and also decrease the load on the help desk staff. NLP tasks that are essential in addressing some of the customer service needs are *speech recognition* that converts speech audio signal into text and *question answering* which is a complex task of recognising the human language question, extract the meaning, searching relevant information in a knowledge base and generate an ineligible answer. Advances in deep learning allow nowadays to skip the need for searching in a knowledge base by learning from large corpora of question-answer pairs complex interrelations.

The above cases underline the increased need in NLP whereas the variation and ever increasing complexity of tasks reveal the need in deeper and richer semantic and pragmatic analysis across a broad range of domains and applications. Any analysis of text beyond the formal aspects such as morphology, lexis and syntax inevitably lead to a functional paradigm of some sort which can be applied not only at the clause level but at the discourse as a whole. This makes the text also an artefact with relation to the socio-cultural context where it occurs. At the moment mainly shallow or limited functional analysis have been realised and more research still needs to be done in this area.

1.4 The linguistic framework

The present work is conducted under the premise that a theory of language is important and worth adopting. It is possible, in NLP, to reach considerable results even without adoption of such a framework. This is demonstrated by the latest advancements in (deep) machine learning. In current work the Systemic Functional (SF) theory of language is adopted because of its versatility to account for the complexity and phenomenological diversity of human language providing descriptions along multiple semiotic dimensions. Further I explaining why a theory is valuable and emphasize the strengths of SFL.

Any meaningful description or analysis involving language implies some theory of about its essential nature and how it works. A linguistic theory includes also goals of linguistics, assumptions about which methods are appropriate to approach those goals and assumptions about the relation between theory, description and applications (Fawcett 2000: 3).

In his seminal paper "Categories of the theory of grammar" (Halliday 1961a), Halliday lays the foundations of *Systemic Functional Linguistic* (SFL) following the works of his British teacher J. R. Firth, inspired by Louis Hjelmslev (Hjelmslev 1953) from the Copenhagen School of linguistics and by European linguists from the Prague Linguistic Circle. Halliday's paper constitutes a response to the need for a *general theory of language* that would be holistic enough to guide empirical research in the broad discipline of linguistic science:

> ... the need for a *general* theory of description, as opposed to a *universal* scheme of descriptive categories, has long been apparent, if often unformulated, in the description of all languages (Halliday 1957: 54; emphasis in original) ... If we consider general linguistics to be the body of theory, which guides and controls the procedures of the various branches of linguistic science, then any linguistic study, historical or descriptive, particular or comparative, draws on and contributes to the principles of general linguistics (Halliday 1957: 55)

Embracing the *organon model* formulated by Bühler (1934), Halliday refers to the language functions as metafunctions or lines of meaning that offer a trinocular perspective on language through *ideational*, *interpersonal* and *textual* metafunctions. In SFL, language is first of all an interactive action serving to enact social relations under the umbrella of the *interpersonal metafunction*. Then it is a medium to express the embodied human experience of inner (mental) and outer (perceived material) worlds via the *ideational metafunction*. Finally the two weave together into a coherent discourse flow whose mechanisms are characterised through the *textual metafunction*.

SFL regards language as a social semiotic system where any act of communication is regarded as a conflation of *linguistic choices* available in a particular language. Choices are organised on a paradigmatic rather than structural axis and represented as *system networks*. Moreover, in the SFL perspective language has evolved to serve particular *functions* influencing their the structure and organisation of the language. However, their organisation around the paradigmatic dimension leads to a significantly different functional organisation than those found in several other frameworks which as Butler (2003a,b) has extensively addressed. Also, making the paradigmatic organization of

language a primary focus of linguistic description decreased the importance of the formal structural descriptions which from this perspective appear as realisation of (abstract) features.

A linguistic description is then provided at various levels of granularity, that in SFL are is called *delicacy*. Just as the resolution of a digital photo defines the clarity and the amount of detail in the picture, in the same way delicacy refers to the how fine- or coarse-grained distinctions are made in the description of the language.

There is no distinction, in SFL tradition, between lexicon and grammar. And to emphasize this fact, the term *lexico-grammar* is used which means the combination of grammar and lexis into a unitary body (see Section **??**).A deeper description of the SFL theory of language is provided below in Chapter **??**.

Until today, two major Systemic Functional Grammars (SFG) have been developed: the *Sydney Grammar* (Halliday & Matthiessen 2013) and the *Cardiff Grammar* (Fawcett 2008). The latter, as Fawcett himself regards it, is an extension and a simplification of the Sydney Grammar (Fawcett 2008: xviii). Each of the two grammars has advantages and shortcomings (presented in Chapter **??**) which I will discuss from the perspective of theoretical soundness and suitability to the goals of the current project.

Both the Cardiff and Sydney grammars have been used as language models in natural language generation projects within the broader contexts of social interaction. Some researchers (Kasper 1988; O'Donoghue 1991; O'Donnell 1993; Souter 1996; Day 2007) consequently attempted to reuse the grammars for the purpose of syntactic parsing. I come back to these works in more detail in Section **??**.

To sum up, in this thesis I adopt the Systemic Functional Linguistic (SFL) framework because of its versatility to account for the complexity and phenomenological diversity of human language providing descriptions along *multiple semiotic dimensions* i.e. paradigmatic, syntagmatic, meta-functional, stratification and instantiation dimensions (Halliday 2003b) and at different *delicacy levels* of the *lexico-grammatical cline* (Halliday 2002; Hasan 2014). To what degree it is possible and what are the benefits of such descriptions still remains to be explored as we do not know yet how much of the SFL descriptive potential needs to be employed in practice in order to achieve useful results or solve problems as those exemplified in Section 1.3. These concepts and other elements of the SFL theory are addressed below in Chapter **??**.

1.5 An example of Systemic Functional analysis

To provide a better intuition on the current work, this section describes an analysis of a simple sentence in Example 1. It will guide you starting from a traditional "school grammar" concepts down to an SFL description of the sentence. SFL provides us with a variety of functions and features serving to express text meaning from several perspectives. Another source of the descriptive breadth is achieved through a practice of feature systematisation as mutually exclusive choices. The feature analysis provided here is partial and restricted to only two constituents (the clause and it's Subject) as this suffices to provide the reader with an intuition of what to expect from an SFL analysis.

(1) He gave the cake away.

School grammar teaches us how to perform a syntactic analysis of a sentence. So let's consider Example 1 in order to perform one. First we would assign a *part of speech* (verb, noun, adjective etc.) to each word, then we would focus on clustering words into constituents guided by the intuitive question "which words go together as a group".

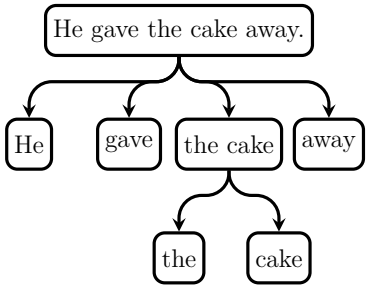

Fig. 1.1 Constituency diagram for Example 1

Figure 1.1 depicts a constituency division of the simple clause in Example 1.The nodes represent grammatical constituents and the edges stand for the *structure-substructure composition*. Next we can move on to assign constituent class and a grammatical function. Here the sentence is formed of a single clause which has four constituting functional parts: a subject providing information who is it about, a predicate indicating the action performed by the subject, a complement denoting what was in scope of the action and an adjunct describing the manner of the action. Each of these functional parts is filled correspondingly by a pronoun, a verb, a nominal group and an adverb. This analysis is depicted in Figure 1.2 as a constituency tree where the nodes carry classes and are given functions in parent units. The nods have been split

into three sections for clarity purposes. The first section is filled with text fragments, the second (in blue) with unit classes and the third (in red) with unit functions.

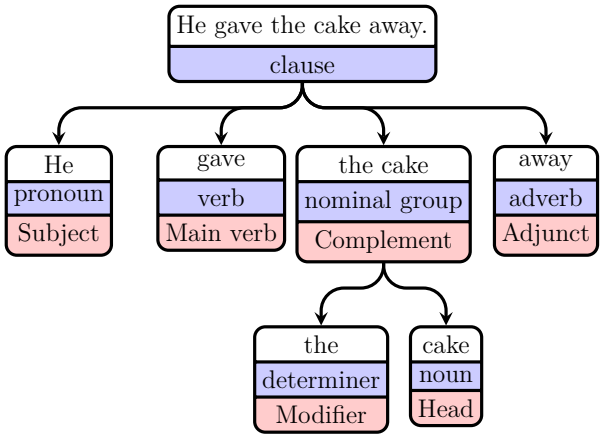

Fig. 1.2 Constituency analysis of Example 1 with unit classes and grammatical functions

Next each constituent can be assigned a set of relevant linguistic features. For example The subject "He" is a pronoun that has features known in traditional grammar: *singular*, *masculine*, and *3rd person*. These features are well differentiated in traditional grammar. For example *singular* means *non-plural*, *masculine* means *non-feminine* and *3rd person* means *non-1st* and *non-2nd*. These are closed classes meaning that there is no *4th person* or that there is no *neutral* grammatical gender in English as other languages have. These features can be systematised (see Figure 1.3) as three systems of mutually exclusive choices that can be assigned to pronominal units. Note that the gender is enabled for 3rd person singular pronouns which can be expressed as is the figure below representing a *system network* which I will explain in Chapter **??**.

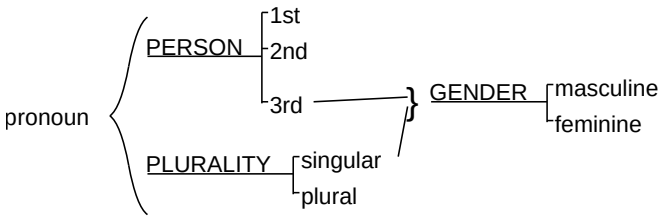

Fig. 1.3 The systematisation of three pronominal features in traditional grammar

In SFG the pronouns are systematised in the system network of Person from *Introduction to Functional Grammar* (Halliday & Matthiessen 2013: 366) that is depicted in Figure 1.4. The (red) rectangles from the figure represent the selections that are applicable to the Subject constituent "He" in example above. These selections are the result of traversing a system network deciding at each step which branch to

follow and advance to the next system if one is available. A simplified traversal for selecting a pronominal referent can be described as follows. The choices are performed based on realisation rules but for now I postpone describing them focusing mainly on the traversal process. So, first the deciding agent shall choose in the PERSON system whether the referent participates in the interaction or not (follow in Figure 1.4). In our example the referent is does not so *non-interactant* choice is made and the agent proceeds towards the next system further distinguishing the type of *non-interactant*. It can be singular or plural and four our case *one-referent* is chosen. Further, the deciding agent needs to make a distinction between referents on the consciousness axis. For our example she chooses the *conscious* feature. And finally conscious things need to be distinguished by gender, which in this example is masculine and therefore *male* sex type is chosen. This path of choices uniquely identifies the pronoun "He" in a system network which also defines the boundaries of all choice possibilities.

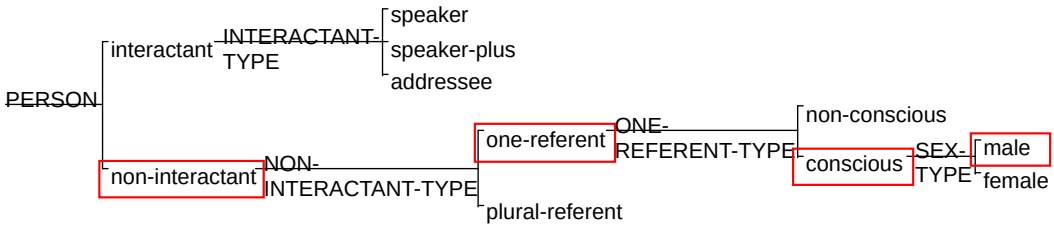

Fig. 1.4 The selections in Person system network from Halliday & Matthiessen (2013: 366) for pronoun "He"

Lets take now the clause constituent that is the root of the constituency tree (see Figure 1.2) and see how SFL features can be applied to it. If in traditional grammar the clause is usually ascribed relatively few features, e.g. as having *passive voice*, *positive polarity* and *simple past tense*; in terms of SFL grammar the corresponding features are many more i.e. *major, positive, active, effective, receptive, agentive, free, finite, temporal, past, non-progressive, non-perfect, declarative, indicative, mood-non-assessed, comment-non-assessed*. Figure 1.5 depicts the selections applicable to clause constituent in Example 1 from Mood system network that is an adaptation of the Mood network proposed in Halliday & Matthiessen (2013: 162). These selections represent, in SFL, choices made by the speaker when generating the utterance in a similar manner as explained for the pronominal referent above. Organisation of the linguistic features is system networks is one of the main things that distinguishes SFL from other linguistic traditions. I will introduce system networks, how they are structured and how they function in Chapter **??** that follows below.

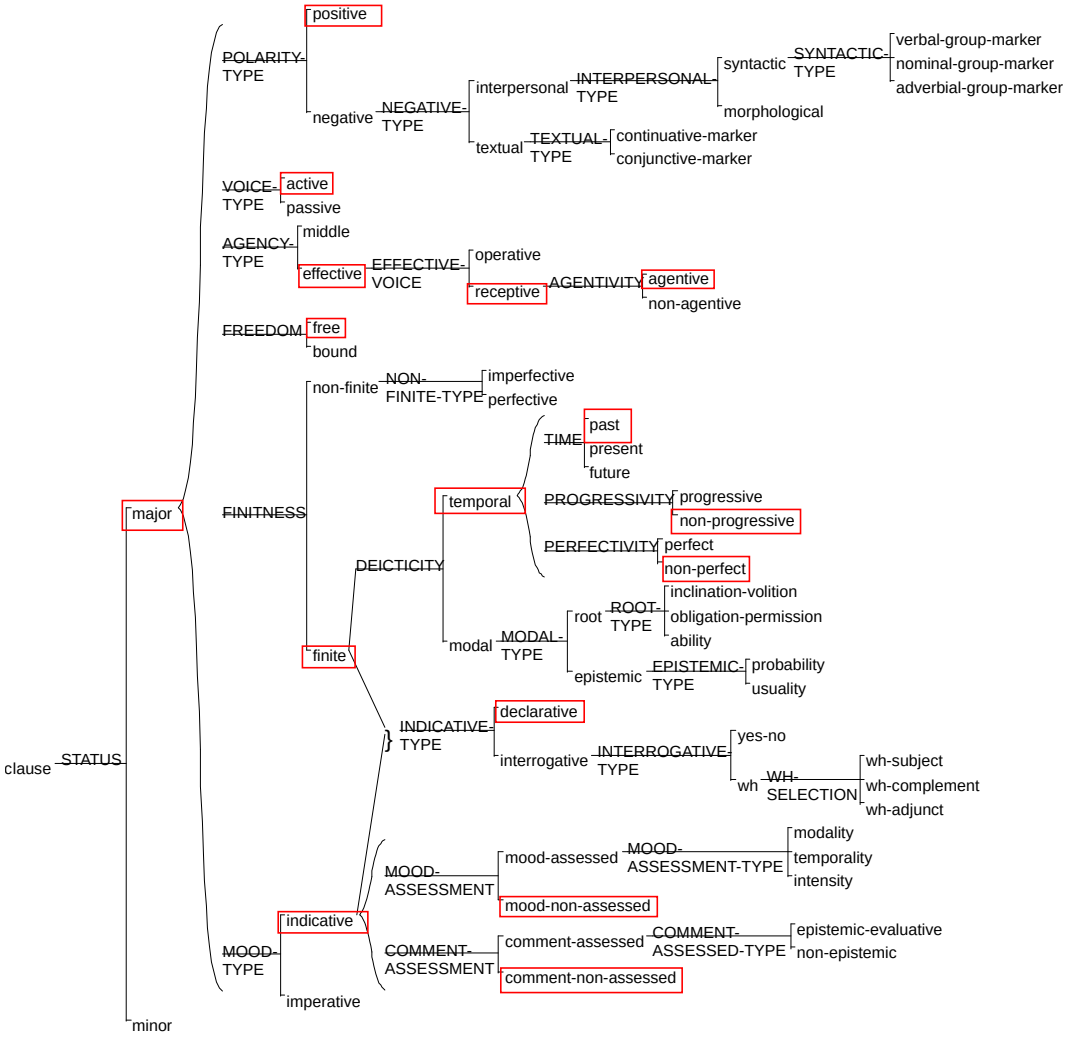

Fig. 1.5 The feature selections in the Mood system network for clause constituent in Example 1

So far we have seen constituents assigned syntactic functions such as Subject, Complement, Adjunct etc. In SFL, they are elements of the interpersonal metafunction that will be explained in Chapter **??**. SFL covers a wide range of linguistic features and functions depending on the kind of meaning it aims at describing. For example another view on the same clause can be provided from a perspective that in SFL is called experiential and corresponds to what in traditional linguistics is known as semantics. It is systematised, in SFL, as Transitivity system network which aims at providing domain independent *semantic frames* called *process configurations*. They describe semantic actions and relationships, along with *semantic roles* ascribed to their *participants*. These semantic frames generally are "governed" by verbs and more specifically each verb meaning has a dedicated semantic frame.

The clause in Example 1 corresponds to a Possessive semantic frame where "He" is the Agent and Carrier while "the cake" is the Affected and Possessed thing. Example 2 provides these annotations. These configurations and participant roles correspond to the Transitivity system network proposed by Neale (2002) which I will introduce in Chapter **??**.

(2) [$_{Agent-Carrier}$ He] gave [$_{Affected-Possessed}$ the cake] away.

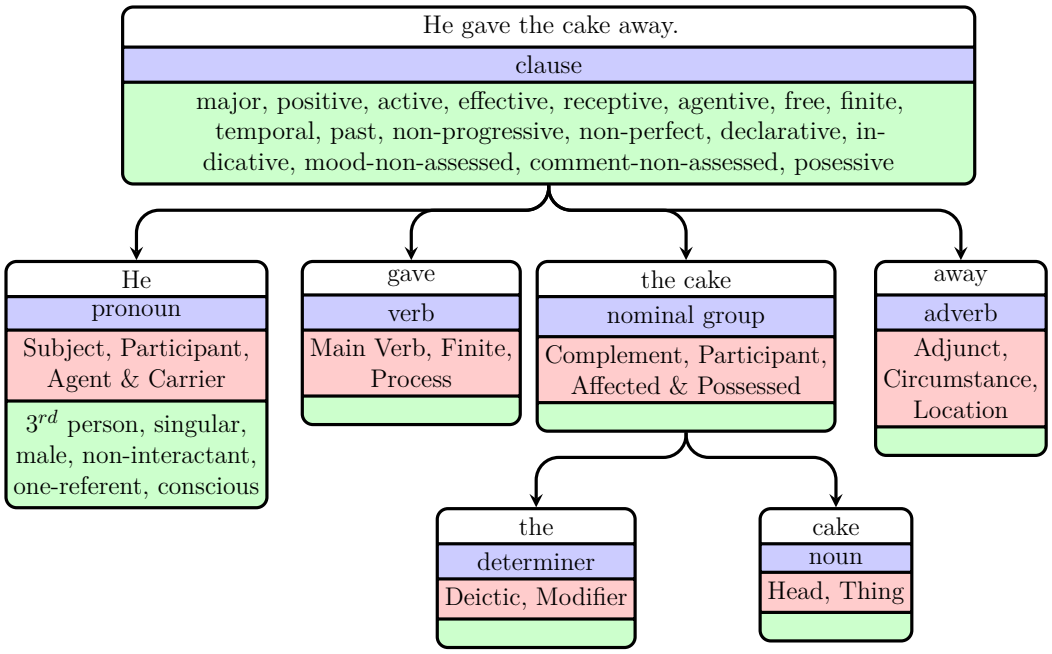

Fig. 1.6 Representation of Example 1 as feature rich constituency tree

There are more functions and features that can be assigned to the constituents in the Example 1 but this is sufficient for the current purposes of introduction. Figure 1.6 summarises everything discussed above into a partially filled constituency tree. The constituents that were not discussed are assigned only a few functions. The last (green) section of every node in the constituent tree is filled with a limited set of grammatical features selected from system networks. In practice the feature set is much richer than those shown in the nodes in Figure 1.6; the restriction aims simply to avoid an over-crowded example and simplify the exposition. Important to underline here is the systemic functional anchoring of the features into system networks that is an SFL practice.

Next I describe what opportunities and limitations exist in automatically generating rich SFL analyses as until now it has not been possible to use these detailed analysis in computational contexts. This makes them unavailable for corpus work, for training data

in machine learning and other end-user application scenarios provided as motivation in the Sections 1.2 above.

1.6 The problem of parsing with SFGs

In this sections I describe the main challenges for using Systemic Functional Grammars (SFG) in computational contexts and parsing in particular. In short the first and main challenge in parsing with SFGs is that of computational complexity. Partially this problem stems from the manner grammars are structured and the fact that paradigmatic descriptions have received most of the attention at the expense of the syntagmatic ones. The second challenge is parsing with features that depart from directly observable grammatical variations towards increasingly abstract semantic features. Addressing this problem requires answers to a couple of other issues: the availability of a lexical-semantic database and a resolution mechanism for *covert constituents*. As we will see motivated below, the latter are useful constructs in providing a solution even though some linguistic theories reject the mere existence of such things. Next I describe in detail the main problems. I will start with unbalance between paradigmatic and syntagmatic accounts in SFL, then bring the computational aspects into the picture comparing the natural language generation and parson problems, after which I turn towards the problem of parsing with more abstract features and draw parallels to *semantic role labelling* problem in mainstream linguistics. Where appropriate, I will also suggesting how potential solutions may look like which will be further presented in Section 1.8.

1.6.1 The lack of suitable syntagmatic descriptions in SFG

SFL since it was established has been primarily concerned with the paradigmatic axis of language. Accounts of the syntagmatic axis of language, such as the syntactic structure, have been put in the background. Within SFL, as we will see in Chapter **??**, structure is a syntagmatic ordering in language capturing regularities and patterns which can be paraphrased as *what goes together with what*. It has been placed on the theoretical map and defined in terms of *rank*, *unit*, *class* and *function*, but afterwards it received minimal attention.

Most of the descriptive work, in SFL, is carried paradigmatically via *system networks* (Definition **??**) describing *what could go instead of what* (Halliday & Matthiessen 2013: 22). Having the focus set on the paradigmatic organisation in language is in fact the

feature that sets SFL apart from other approaches to study language. This has led to progress in accounting how language works at all strata but little was said about language constituency. And this can be considered "unsolved" within SFL accounts leaving a "gap in what must be one the central areas of any characterisation of language" (Bateman 2008: 25).

If we attend SFL literature, however, the syntagmatic dimension is implicit and present everywhere in the SFL literature, which makes the above claims sound little surprising. For instance all example analyses in the *Introduction to Functional Grammar* (Halliday & Matthiessen 2013) are predominantly syntagmatic. Moreover, Robin Fawcett for decades promotes the motto *no system network without realisation statements* (Fawcett 1988b: 9) which means that every paradigmatic description must be accompanied by precise rules how it is syntagmatically realised in text. Yet, despite these inducements, the situation could not have been more different. Bateman (2008) presents in detail why there is a severe imbalance between syntagmatic and paradigmatic axes in SFL, how it came to be this way and how it is especially damaging to the task of automatic text analysis, yet quite beneficial for the text generation task.

1.6.2 Parsing is asymmetric to generation: the computational complexity issue

O'Donnell & Bateman (2005) offer a detailed description to the long history of SFL being applied in computational contexts yielding with productive outcomes on language theorising, description and processing. The transfer between SFL and computation typically involved a delay between the theoretical formulation and the computational instantiation of that formulation (Bateman & Matthiessen 1988: 139) (Matthiessen & Bateman 1991: 19). The theoretically formulated ideas contain hidden pitfalls that are revealed only upon explicit formulations required in computation (Bateman 2008: 27).

The active exchange between SFL theory and computation has been almost entirely oriented towards automatic *natural language generation*. Such systems take abstract semantic specifications as input and use grammars to produce grammatically correct and well connected texts. One of the grammars successfully used in generation tasks is the Nigel grammar developed within Penman generation project (Mann 1983a). The efficiency in generation tasks is, in part, due to decomposition of language along the paradigmatic axis using functionally motivated sets of choices between functionally motivated alternatives (McDonald 1980). The Nigel grammar contains 767 grammatical systems defined over 1381 grammatical features which Bateman evaluates as "a very

large computational grammar by current standards, although nowadays by no means the broadest when considered in terms of raw grammatical coverage" (Bateman 2008: 29).

The computational processes driving natural language generation relied heavily on the notion of *search*. A well defined search problem is defined in terms of a precise description of the search space which then helps a navigation process effectively to find solutions. The paradigmatic organization of the *lexicogrammar* as system networks assumed within SFL turns out to organise the search space for possible grammatical units appropriate for expressing communicative goals in generation in almost ideal manner (Bateman 2008: 28).

Automatic analysis or *parsing* can be seen as a reverse problem of finding appropriate analysis within a search space of possible solutions. That is to identify, as accurate as possible, the meaning systematised in the grammar, of a given natural language sentence. As seen in Section 1.5 above, an account of the sentence meaning would have to provide two things. First a description in terms of a formal structure of the sentence revealing the constituents plus their syntactic relations to each other. And second, a description in terms of a complete set of features (detailed to the extent that grammar permits) applicable to each constituent of structure. If, in the generation process, the abstract semantic specifications are increasingly materialised through choice making by traversing the system network towards finally generated text (see example in Section 1.5), then, in the parsing process, the reverse is the case. The process starts from a given sentence aiming to derive/search the feature choices in the system network afferent to each of the constituents. But if the paradigmatically organised lexicogrammatical resource is effective for generation it turns out, as we will see next, to be by far unsuitable for the analysis task because of the *size problem*. Halliday himself mentions this problem when he asks *how big is a grammar?*.

> Given any system network it should in principle be possible to count the number of alternatives shown to be available. In practice, it is quite difficult to calculate the number of different selection expressions that are generated by a network of any considerable complexity (Halliday 1996: 10).

The issue is that of handling a combinatorial space which emerges from the way connections and (cross-)classifications are organised in a system network. In addition to that, the orientation of systemic grammars towards choice means that a typical grammar includes many disjunctions, which leads to the problem of search complexity. Also the abstract nature of systemic features leads to a structural richness that adds

logical complexity to the task (O'Donnell 1993). So estimating the size of the grammar would in fact mean estimating the potential number of feature combinations. For example, a hypothetical network of 40 systems the "size of the grammar it generates lies somewhere between 41 and 2^{40} (which is somewhere around 10^{12})" (Bateman 2008: 28). However it is not easy to calculate where would the upper limit of a grammar fall even when the configuration of relations of a particular system network is known.

For the generation task the issue of size is not a problem at all as the number of choice points is actually rather small. Such a paradigmatic organisation is, in fact, a concise and efficient way to express the linguistic choices where the possible feature selections are relevant only when they are enabled by prior paradigmatic choices and it is only those alternatives that need to be considered (Halliday 1996: 12–13). This property of gradual exposure of choices characterises the traversal of the system networks, in generation process, which starts from the root and gradually advances towards more delicate features down to a leaf.

In the analysis task, the paradigmatic context of choice, that helps navigation during the generation process is no longer available. It is not known any longer which features of a systemic network are relevant and which are not. This leads to a radical asymmetry between the two tasks. That is: in generation, the simple traversal of the network finds only the compatible choices because that is what the network leads to; whereas in analysis it is not evident in advance which path to follow therefore the task is to explore the entire search space in order to discover which features apply to the text. This means that any path is potentially relevant and shall be passed and needs to be checked leading to evaluation of the system network as a whole. There is then no way to restrict the search space as in the case of generation (Bateman 2008: 29).

One of the grammars successfully used in generation tasks is the Nigel grammar, described above, which is a large grammar by modern standards. To parse with such a grammar would means exploring a search space of approximately 3×10^{18} feature combinations. A more detailed break down the complexity by rank or primary class as provided in Table 1.1 below.

1.6.3 The challenge of parsing with semantic features

Another difficulty in parsing with SFGs lies in the fact that, as the analysis moves away from directly observable grammatical variations towards more abstract semantic variations, the difficulty of generating an accurate account increases drastically. The Transitivity system network for example consists of such semantic features and it is

rank or primary class	*size*
adverbial-group	18
words	253
quantity-group	356
prepositional-phrase	744
adjectival-group	1045
nominal-group	$>2 \times 10^9$
clause	$>3 \times 10^{18}$

Table 1.1 Size of major components of the Nigel grammar expressed in terms of the number of selection expressions generated (Bateman 2008: 35)

comparable to what is called in computational linguistics (shallow) *semantic parsing* or *Semantic Role Labelling* (SRL) (Carreras & Màrquez 2005).

The main challenge of SRL, well explained in (Gildea & Jurafsky 2002: 245–250), remain the same since Winograd (1972): *moving away from the domain specific, hand-crafted semantic specifications towards domain independent and robust set of semantic specifications*. This goal was undertaken in several projects to build large broad-scope lexico-semantic databases such as WordNet (Fellbaum & Miller 1998), FrameNet (Baker et al. 1998; Johnson & Fillmore 2000; Fillmore et al. 2003) and VerbNet (Schuler 2005; Kipper et al. 2008). A similar database exists for Transitivity system network as described in Fawcett (forthcoming) called Process Type Database (Neale 2002).

Such databases provides with domain independent *semantic frames* (Fillmore 1985), know in SFL as *configurations* or *figures*, which describe semantic actions and relationships, along with *semantic roles* ascribed to their *participants*. The semantic frames generally are governed by verbs and more specifically each verb meaning has a dedicated semantic frame. For instance the perception frame contains *Perceiver* and *Phenomenon* roles as can be seen in Example 3.

(3) [$_{Agent-Perceiver}$ Jaqueline] glanced [$_{Phenomenon}$ at her new watch].

The tendency is to identify frames that are generic enough to cover classes of verb meanings (for example Action, Cognition, Perception, Possession frames) and the same applies to participant roles where the tendency is to reuse roles across semantic frames (for example agent role from Action frame is reused in Perception or Possession frames, or Phenomenon is reused in Cognition and Perception frames).

1.6.4 The issue of covert elements

Besides the challenge of identifying configurations and their participants in text, the problem with semantic features goes one step further. Sometimes the semantic roles correspond to constituents that are displaced or not realised in the text called *covert* or *null elements*. This increases the challenge of identifying and assigning them correctly. Next I show how (non-)realisation in text of the semantic roles impacts possibility to interpret that text. Then I will show how frames may still be valid even when an element is realised or displaced which will bring us to the core of the problem. This is followed by a brief description of the approach taken in the current work to solve it.

For a frame to be considered correctly realised in text it needs to fill at least the mandatory roles. Let's imagine that a part of the text in Example 3 is erased. If we take the Agent-Perceiver away as in Example 4 the texts is perceived as incomplete because it is not possible to interpret its meaning. It leaves us with the questions *Who* glanced at her new watch? Similarly, if we delete the Phenomenon like in Example 5, we are unable to resolve the meaning of the text without first answering the question *what* or *who* did Jaqueline glance at?

(4) glanced at her new watch

(5) Jaqueline glanced

Consider now Example 6. It is a sentence that has three non-auxiliary verbs: seem, worry and arrive. According to the Cardiff grammar, which will be introduced in Chapter **??**, this corresponds to three clauses *embedded* into each other. A constituency analysis is provided in in Table 1.2.

(6) She seemed to worry about missing the river boat.

<table>
<tr><td>She</td><td>seemed</td><td>to</td><td>worry</td><td>about</td><td>missing</td><td>the</td><td>river</td><td>boat.</td></tr>
<tr><td colspan="9">clause</td></tr>
<tr><td>Subject</td><td>Main Verb</td><td colspan="7">Complement</td></tr>
<tr><td></td><td></td><td colspan="7">clause</td></tr>
<tr><td></td><td></td><td>Infinitive Element</td><td>Main Verb</td><td colspan="5">Complement</td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="5">clause</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Binder</td><td>Main Verb</td><td colspan="3">Complement</td></tr>
</table>

Table 1.2 SF constituency analysis in Cardiff grammar style

The participant role configurations (or the semantic frames) these verbs bring about are provided in the Table 1.3. For the sake of this example the first role corresponds to the Subject constituent and the second to the Complement constituent. This way

the verb meaning *seem*$_1$ corresponds to an Attributive configuration that distributes Carrier and Attribute roles to the Subject "She" and the Complement "to worry about missing the river boat". In the case of *worry about*$_1$ and *miss*$_1$ the first roles provided by Cardiff grammar are *compound* (i.e. composed of two simple ones) while the second ones are simple. So, in the example above, the verb meaning *worry about*$_1$ distributes the Phenomenon to the Complement "about missing the river boat" and the Agent & Cognizant role to an empty Subject that is said to be *non-realised*, *covert* or *null element*. A similar situation is for *miss*$_1$ that assigns an Affected & Carrier role to the empty Subject and the Possessed role to the Complement "the river boat".

Verb meaning	**Semantic configuration**	**Participant role distribution**
seem$_1$	Attributive	Carrier + Attribute
worry about$_1$	Two Role Cognition	Agent & Cognizant + Phenomenon
miss$_1$	Possessive	Affected & Carrier + Possessed (thing)

Table 1.3 Semantic role configurations according to Neale (2002); Fawcett (forthcoming)

Those unrealised Subjects in the embedded clauses are recoverable from the immediate syntactic context (no need for discourse) and correspond, in this case, to the Subject in the higher clause. This is easy to see in Examples 7 and 8 therefore we can just mark the places of the null Subjects in the embedded clause in order to be able to assign the semantic labels (otherwise the frame cannot be assigned to the constituents). Notice also an index i to highlight that the null elements correspond to the higher clause Subject "She".

(7) *She* worried about missing the river boat.

(8) *She* missed the river boat.

(9) She$_i$ seemed [*null-Subject*$_i$ to worry [about *null-Subject*$_i$ missing the river boat]].

Now that the places of covert constituents are explicitly marked and the recoverable constituent coindexed we can see the distribution of semantic roles realised for this sentence in the Table 1.4 below.

In language there are many cases where constituents are empty but recoverable from the immediate vicinity relying in most cases on syntactic means and in a few cases additional lexical-semantic resources are required. In SFL, Fawcett describes these elements in the context of Cardiff grammar (Fawcett 2008: 115,135,194) but provides

She$_i$	*seemed*	$\emptyset_i$	*to*	*worry*	*about*	$\emptyset_i$	*missing*	*the*	*river*	*boat.*
Attributive configuration										
Agent		Attribute								
		Two role cognition configuration								
		Agent & Cognizant			Phenomenon					
					Possessive configuration					
						Affected & Carrier		Possessed		

Table 1.4 Transitivity analysis in Cardiff grammar style (Neale 2002; Fawcett forthcoming)

no means to recover them. Some mechanisms of detecting and resolving the empty constituents are captured in the Government and Binding Theory (GBT) developed in (Chomsky 1981, 1982, 1986) and based on phrase structure grammar. GBT explains how some constituents can *move* from one place to another, where are the places of *non-overt constituents* and what constituents do they refer to i.e. what are their *antecedents*. Such accounts of empty elements are missing from any SFG grammar yet they are useful in determining the correct distribution of participant roles to the clause constituents. Translating the mechanisms from GBT into SFG could contribute to decreasing the complexity of the parsing problem mentioned above.

1.6.5 The problem summary

This section has shown some of the issues related to parsing with SFG. In summary, first, the parsing task cannot be treated as a reversible generation task because the methods that have been shown to work for generation are not usable for parsing as such due to a high computational complexity. Second, the parsing task, regardless of the grammar, should first and foremost account for the sentence structure on the syntagmatic axis and only afterwards for the (semantic) features selected on the paradigmatic axis. Such syntagmatic account in SFL is insufficient for the parsing task. Third, syntagmatic account alone does not provide enough clues for assignment of semantic features and require a lexical-semantic account within the grammar or as external semantic databases. Moreover it can be aided by identification of places where covert constituents are said to exist. Identifying such the null elements is not the only method of assigning semantic features and some approaches do without them but having access to such information is considered valuable in the present thesis.

Regarding the problem of computational complexity explained above, how could the large search space of grammars such as Nigel be restricted to a reasonable size and how can be compensated the lack of proper syntagmatic description in SFGs? The first part of the question has already been addressed in O'Donnell (1993) in at least what would a possible solution look like. The lack for an answer to the second

part and probably for other hidden reasons the results of parsing with SFGs so far are not usable in real world applications. This is drawn from the past attempts such as Kasper (1988), Kay (1985), O'Donoghue (1991), O'Donnell (1993) and Day (2007), to mention just a few, none of which managed to parse broad coverage English with full SFG without aid of some sort. Each had to accept limitations either in grammar or language size and eventually used simpler syntactic trees as a starting point of the parsing process. A detailed account of the current state of the art in parsing with SFGs is provided in Chapter **??**.

Therefore to address parts of the above problems I attempt a different approach. Some linguistic frameworks, other than SFL, have been shown to work well in computational contexts solving problems similar to the ones identified above. For the purposes of this thesis I selected Dependency Grammar (DG) and GBT. And instead of attempting to find novel solutions within the SFL framework, an alternative approach, I argue in the next section, would be to establish a cross-theoretical and inter-grammatical links and to enable integration of the ready solutions.

1.7 Theoretical motivation - on theoretical compatibility and reuse

This thesis employs three linguistic frameworks namely the *Systemic Functional Linguistics*, *Dependency Grammar* and *Governance & Binding Theory*. SFL has already been motivated as target analysis framework in Section 1.4 which is in detail introduced in Chapter **??**. The other two frameworks are employed because some of the accomplishments in those domains carry answers to above stated problems. The goal is to maximise positive properties and enable reusing the results.

In the past decades much significant progress has been made in natural language parsing framed in one or another linguistic theory each adopting a distinct perspective and set of assumptions about language. The theoretical layout and the available resources influence directly what is implemented into the parser and each implementation approach encounters challenges that may or may not be common to other approaches in the same or other theories.

Parsers implementing one theoretical framework may face common or different challenges to those implementing other frameworks. The converse can be said of the solutions. When a solution is achieved using one framework it is potentially reusable in other ones. The successes and achievements in any school of thought can be regarded as valuable cross theoretical results to the degree links and correspondences can be

established. Therefore reusing components that have been shown to work and yield "good enough results" is a strong pragmatic motivation in the present work.

In the past decade *Dependency Grammar* (Tesniere 2015) has become quite popular in natural language processing world favoured in many projects and systems. The grammatical lightness and the modern algorithms implemented into dependency parsers such as Stanford Dependency Parser (Marneffe et al. 2006), MaltParser (Nivre 2006), MSTParser (McDonald et al. 2006) and Enju (Miyao & Tsujii 2005) are increasingly efficient and highly accurate. Among the variety of dependency parsing algorithms, a special contribution bring the *machine learning* methods such as those described in McDonald et al. (2005); McDonald & Pereira (2006); Carreras (2007); Zhang & Nivre (2011); Pei et al. (2015) to name just a few.

As the dependency parse structures provide information about functional dependencies between words and grants direct access to the predicate-argument relations and can be used off the shelf for real world applications. This information alone makes the dependency grammar a suitable candidate to supplement the syntagmatic account missing in SFGs and provide some functional hooks for reducing complexity in parsing with SFGs. One of the goals in this work is to investigate to which degree the dependency grammar is structurally and functionally compatible with SFGs to undergo a cross theoretic transformation. This hypothesis is investigated at the theoretical level in Chapter **??** and then indirectly evaluated empirically in Chapter **??** based on Stanford Dependencies parser version 3.5 (Marneffe & Manning 2008b,a; Marneffe et al. 2014).

The problem of accounting for the *null elements*, mentioned above, is not addressed either in SFL or in Dependency Grammar. It is, however, addressed in detail in the Government and Binding Theory (GBT) (Chomsky 1981; Haegeman 1991) which is one of Chomsky's Transformational Grammars (Chomsky 1957). One other goal in this thesis is to investigate to which degree GBT accounts of null elements can be reused as DG or SFG structures to undergo a cross-theoretic transformation enabling those accounts in DG or SFG contexts. Chapter **??** introduces GBT and investigate this hypothesis providing some of the cross-theoretic and inter-grammatical links to Dependency and SFL grammars that as we will see in Chapter **??** benefits the Transitivity analysis.

1.8 Thesis goals and scope

This thesis aims at a modular method for parsing unrestricted English text into a Systemic Functional constituency structure using fragments of Systemic Functional Grammar (SFG) and dependency parse trees.

As will be described in Chapter **??**, some parsing approaches use a syntactic backbone which is then flashed out with an SFG description. Others use a reduced set or a single layer of SFG representation; and the third group use an annotated corpus as the source of a probabilistic grammar. Regardless of approach, each limits the SFG in one way or another, balancing the depth of description with language coverage: that is either *deep description but a restricted language* or *shallow description but broad language coverage* is attempted. The current thesis tilts towards the latter: while keeping the language coverage as broad as possible the aim is to provide, in the parse result, as many systemic features as possible.

The process developed in this thesis can be viewed as a pipeline architecture (see Section 1.8.3) comprising of two major phases: the *structure creation* and the *structure enrichment*. The structure creation phase aims to account for the syntagmatic dimension of language.

The structure enrichment phase aims at discovering and assigning systemic features (accounting for the paradigmatic dimension of language) afferent to each of the nodes constituting the structure. In this phase, two kinds of feature enrichments can be distinguished by the kinds of clues used for feature identification. The first kind of clues are syntagmatic (constituency tree, unit class, unit function, linear order, position) and can be detected using, what I call, the *structural patterns* while the second kind of clues are lexical-semantic requires lexical-semantic and potentially more kinds of resources in addition to the structural patterns.

1.8.1 Towards the syntagmatic account

The problem in using SFGs for parsing, as we have seen in Section 1.6 above, manifests when the grammar is instantiated computationally with a primary focus on paradigmatic organisation (prevalent in SFL) at the cost of syntagmatics which leads to the first difficulty that needs to be addressed: discovering from a sequence of words what possible groups are combinable into grammatical groups, phrases or clauses. This is a task of bridging a sequence of words as input and the grammatical description of how they can combine to form a (syntactic) constituency tree structure (known in SFL as *syntagmatic organizations* which will be addressed in Section **??**).

This challenge will be addressed by filling the gap of the syntagmatic account within the SFL grammar directly. This involves, first, providing information about which grammatical functions operate at each rank, second, which grammatical functions can be filled by which classes of units and, third, providing relative and absolute description of the element order for each unit class. This information in the grammar can guide the process of building the constituency structure.

Alternatively the problem of structure construction can be outsourced as parsing with other grammars. This is done in the works of Kasper Kasper (1988) and Honnibal (2004); Honnibal & Curran (2007) and is knows in SFL literature as *parsing with a syntactic backbone*. In this case, the problem changes into creating a transformation mechanism to obtain the SFL constituency structure rather than build it from scratch.

This thesis addresses the problem of building the constituency structure by the latter approach: parsing the text with Stanford Dependencies parser version 3.5 (Marneffe & Manning 2008b,a; Marneffe et al. 2014) and then transforming the parse result into SFG constituency tree. The transformation mechanisms from one grammar into the other one requires also a theoretical discussion in terms of what is being transformed. Such account of linguistic primitives or configurations of primitives in the source grammar corresponds to SFG primitives is provided in Chapter **??**.

The SFG constituency structures is generated through a process that involves: traversing the source (dependency) parse tree and, at each traversal step, executing a constructive operation on a parallel tree following a predefined rule set of operations and mapping relations. The detailed description of the structure generation process is provided in the Chapter **??**.

1.8.2 Towards the paradigmatic account

Once the constituency structure is in place it can inform the following feature derivation process. The configurations of units of specific classes and carrying grammatical functions can operate as "hooks" on system network to guide the traversal in the same way the paradigmatic context available in the generation process. Such configurations resemble the SFG *realization rules* which, in the generation process, instantiate the (abstract) features into text.

The system network fragment in Figure 1.7 contains the realisation rules positioned in rectangles below a few features. These realisation rules indicate what shall be reflected in the structure when a feature is selected (discussed already in Section 1.6). The converse is also true: if structure contains a certain pattern then it is a (potential) manifestation of a given feature.

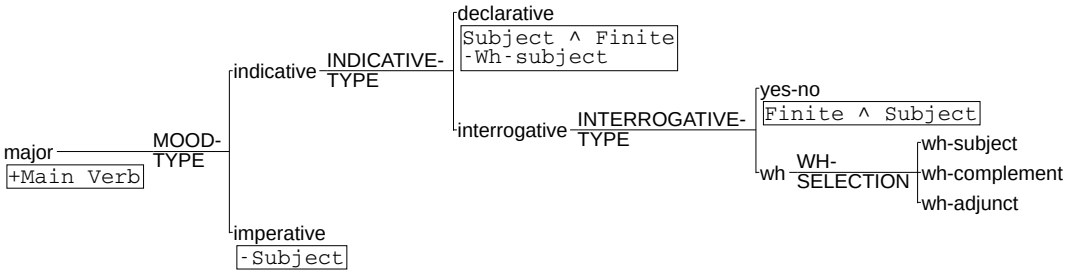

Fig. 1.7 A fragment of mood system from Halliday & Matthiessen (2013: 366)

For example the structure of a *major* clause needs to have a predicate or Main Verb element realised. In the parsing process, testing whether there is a unit functioning as Main Verb below in the clause node suffices to assign the *major* feature to that clause. Next, if the clause has no unit functioning as Subject then it shall be assigned *imperative* feature otherwise the *indicative* one. Further the INDICATIVE-TYPE system is enabled. Here the test is whether a Subject node is positioned in front of the Finite node and whether the Subject contain the preposition "who". This sort of queries on the structure can be formulated as *structure patterns* (see Section **??**) and associated to features in the system network in the same manner the realisation rules are.

The pattern recognition plays an essential role in current parsing method for fleshing out the constituent backbone with systemic selections. In the parsing process the structural patterns are tested whether they *match* (see Section **??**) anywhere in the constituency structure and if so then the matched nodes are enriched with the features proved in the pattern (described in Section **??**).

The structural patterns in this work are expressed as *graph patterns* (described in Section **??**). Note that I employ graph and not tree patterns because the tree patterns are too restrictive for the purpose of the current work. While most of the time they are hierarchically structured as a tree there are few patterns that involve sibling connections or nodes with more than one parent. In both cases the tree structure is broken. The graph approach allows a wide range of structural configurations including the trees.

The enrichment stage of the parsing process comprises of a series of graph pattern matching operations that in case of success leads to the enrichment of the constituency structure with the features given in the graph pattern. This mechanism is described in detail in the Section **??**.

In this work most of the graph patterns have been manually created. Because this is laborious exercise only a few system networks have been covered in the implementation

of the parser. Nonetheless they suffice for deriving some conclusions regarding the parsing approach. The future work may investigate how can graph patterns be generated automatically from the realisation rules of large grammars such as Nigel grammar.

The two main system networks targeted in this are MOOD and TRANSITIVITY (both briefly described in Chapter **??**). The MOOD network is composed of features which can be identified through graph patterns involving only the unit classes and functions provided in the constituency structure (described in Chapter **??**). The TRANSITIVITY network requires a lexical-semantic database in order to derive graph patterns. This work employs the Process Type Database (PTDB) (Neale 2002) to aid enrichment with TRANSITIVITY features described in Chapter **??**.

1.8.3 Parsimonious Vole architecture

The current thesis is accompanied by a software implementation called the Parsimonious Vole parser. It is written in Python programming language and is available as open source distribution[1] . This section provides an overview to the construction process.

The parser follows the pipeline architecture depicted in Figure 1.8 where, starting from an input text, a rich systemic functional constituency structure is progressively built. Figure 1.8 provides three types of boxes: the rounded rectangles represent the parsing steps, the green trapezoid boxes represent input and output data while the orange double framed trapezoid boxes represent additional resources involved in the parsing step. The parsing steps linearly flow from one to the next via green trapezoid boxes on the left-hand side, which represent input-output data in between the steps. On the right-hand side are positioned double edged orange trapezoids representing fixed resources needed by some operations. For example, the *constituency graph creation* step takes a normalised dependency graph for input and produces a constituency graph as output.

On the right-hand side a series of green vertical arrows are provided naming phases in parsing process (spanning one ore more process steps) where the first one, *Graph Building* (spanning the first three process steps), accomplish construction of the constituency backbone (corresponding to the syntagmatic account described in Section 1.8.1 above) and the second phase *Graph Enrichment* (spanning the last three process steps) flashes out the backbone with features (described in Section 1.8.2).

One important feature of this implementation is its heavy reliance on graph pattern matching and other operations using graph patterns described in Chapter **??**.

[1] https://bitbucket.org/lps/parsimonious-vole

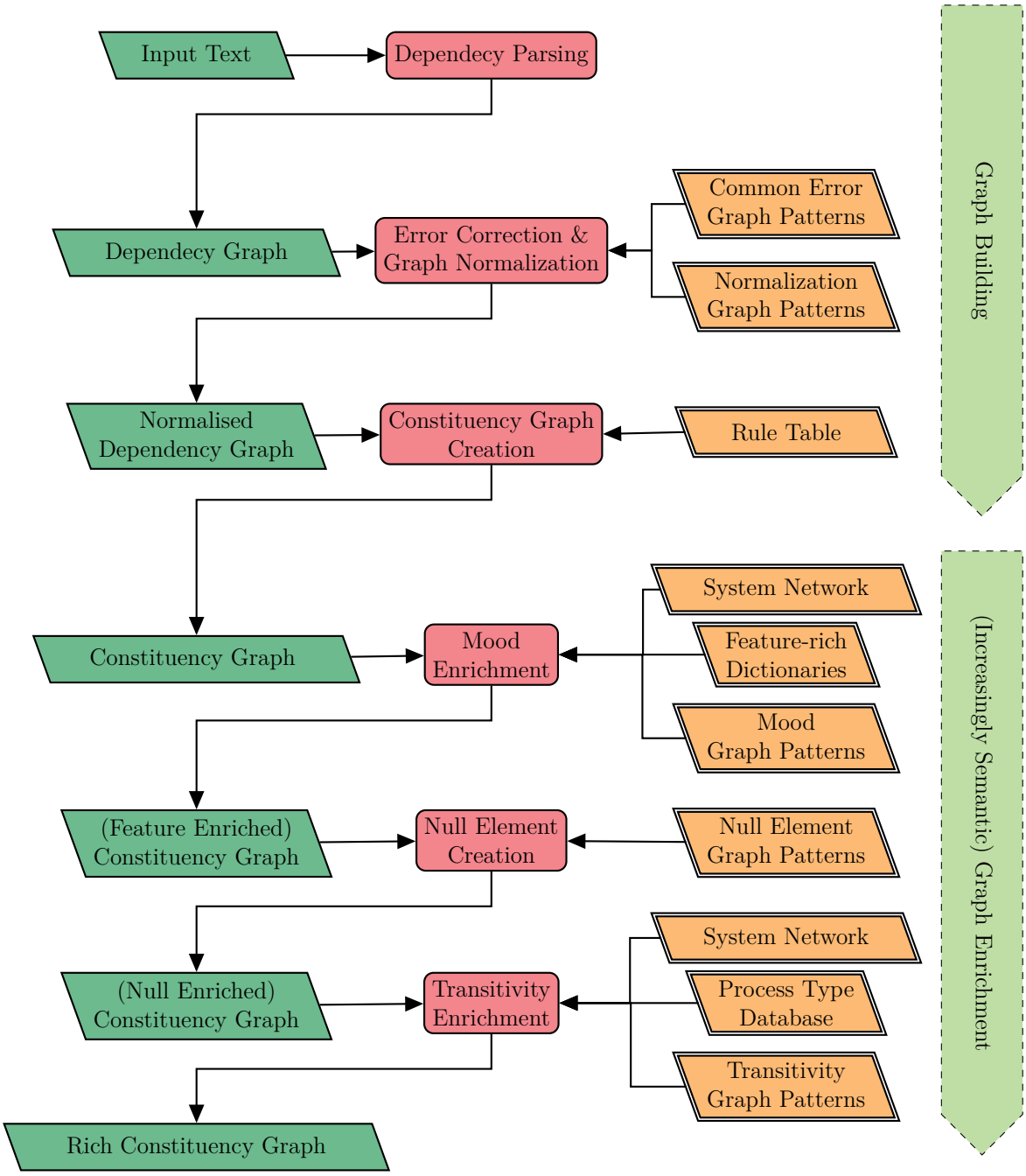

Fig. 1.8 The parsing process pipeline

The parsing process starts with an input English text and ends with production of a Rich Constituency Graph. Input Text is first parsed with the Stanford Dependency parser (Chen & Manning 2014) version 3.5[2] producing a Dependency Graph.

[2] https://nlp.stanford.edu/software/nndep.html

The dependency graphs often contain errors. Some of these errors are predictable and so easy to identify and correct. Also, some linguistic phenomena are treated in a slightly different manner than that proposed in the current thesis. Therefore the dependency graph produced by the Stanford parser is *Corrected and Normalised* using graph pattern matching against collections of known errors and a set of normalization rules encoded as graph patterns.

Once normalised the dependency graph is ready to guide the *building process* of the systemic functional constituency graph. Through a traversal of the dependency graph the constituency graph is parallelly constructed guided by a mapping *Rule Table*. The mappings indicate what operation to perform on the newly emerging graph given the visited dependency node and its incoming and outgoing relations. Even if it a distinct structure, the constituency graph is, in a way, a transformation of the dependency graph. It constitutes the syntactic backbone on which the subsequent enrichment phases are performed.

Next follows the phase where each constituent node of the syntactic backbone is *enriched* with features, some of which are of a *syntactic* and others of a *semantic* nature. In between these enrichment phases there is an additional construction process adding where needed *empty constituents* that play an important role in semantic enrichment. The enrichment steps use additional resources such as *system networks*, *feature rich lexicons*, *graph patterns* and a *semantic database*. The *null element creation* process also needs a collection of graph patterns for identifying where and what kind of null elements occur as motivated in Section 1.6 and explained in detail in Chapter **??**. The final result of the process is a *Rich Constituency Graph* of the original text comprising a substantial set of systemic feature selections associated with constituting units of structure.

The the detailed parser implementation choices and developed algorithms are presented in Chapters **??** and **??**.

1.8.4 Research questions and contributions

This thesis addresses the following questions:

- To what extent techniques from other areas of computational linguistics can be reused for the SFL parsing?
- To what degree the syntactic structures of the Dependency Grammar and Systemic Functional Grammar are compatible to undergo a transformation from one into the other?

- Is Stanford dependency grammar suitable as a syntactic backbone for Systemic Functional Grammar parsing?
- Can the Process Type Database be used as a resource for SFG Transitivity parsing?
- How can Government and Binding Theory be used for detecting places of null elements in the context of SFL constituency structure?

Also it brings the following theoretical and practical contributions:

- A set of theoretical principles and generalizations establishing cross-theory links between Dependency Grammar and SFL.
- A set of mapping rules between Stanford Dependency v3.5 to SFG constituency structure.
- A parallel graph construction process for creating SF syntactic backbone.
- A flexible and expressive method to represent systemic features as graph patterns and a strategy for choice propagation in the systemic networks.
- A set of pattern graphs covering Mood, Transitivity and a few other small system networks.
- A clean machine readable version of the PTDB.
- A method to transform PTDB into Transitivity graph patterns.
- Several principles for detecting null elements in the sentence which were translated from the Government and Binding Theory (GBT) into Dependency and SFL terms.
- Implementation of the translated GBT principles as graph patterns corresponding usable to identify the null elements in a sentence.
- A small test corpus to evaluate the Parsimonious Vole parser.

1.9 Overview of the thesis

Chapter 1 has provided an introductions to the work described in this thesis. It has indicated the are to which it seeks to contribute, and described the motivation of work from an applied and a theoretical perspective. In Chapter **??** a list of selected works on parsing with SFG is presented and briefly discussed.

Chapter **??** provides and overview of the SFL theoretical foundations. There are two outstanding traditions in SFL each providing a theory of grammar. First is developed in Sydney by Halliday, Matthiessen, Hassan, Martin, Rose and others. The second is developed in Cardiff by Fawcett, Tucker, Tench and others. I present both schools in first two sections of the chapter and then, in the third section, I provide a comparative critical discussion on both theories of grammar motivating relaxation of the rank scale, approach to structure formation, unit classes and few other concepts relevant to current work.

In the following chapter I provide a description of the grammar implemented in the Parsimonious Vole which. It is a selection of unit classes from both Sydney and Cardiff Grammars following the theoretical motivation from the previous chapter. Here is also presented a selection of two system networks: MOOD and TRANSITIVITY that were selected to demonstrate how the current parsing method works. The former system network is tightly linked to the syntagmatic variations in the structure whereas the latter describes ideational choices of the semantic structures and, thus, is farther from the surface variations. In order to integrate this system network I use a lexical database of verb meanings called Process Type Database Neale (2002).

Chapter **??** introduces the Dependency Grammar 1959, starting with it's origins and foundations, evolution into its modern form, its applications in computational contexts particularly highlighting the Stanford grammatical model and parser. The usage of dependency grammar and dependency parse graphs is motivated in 1.8.1 as the primary input into the current parsing pipeline for creating the constituency structure. The last part of the chapter provides a cross theoretical bridge from the dependency grammar towards the systemic functional grammar integrated into the Parsimonious Vole implementation.

Next chapter starts with an introduction of Government and Binding Theory (GBT) explaining where the empty constituents occur in sentences. These constituents were motivated in 1.6 and are a part of solution for parsing with TRANSITIVITY system network. The second section of the chapter provides an inventory of different null elements and the last section provides, just like in the previous chapter, a cross theoret-

ical overview, this time from GBT phrase parse structures into Stanford dependency grammar.

Chapter **??** provides the building blocs of the algorithms of this thesis. It makes the transition from the linguistic theoretic presentations towards the computer science foundations introducing necessary typed sets, feature structures and graphs. These concepts are employed in the chapters that follow to represent linguistic constructs described in the previous chapters. An important role, in the current work, play the pattern graphs and the operations enabled by using them presented in Sections **??** – **??**. The system networks, are defined in a simplified form corresponding to how they are currently used followed by a brief discussion.

The first phase of the parsing pipeline (see Figure 1.8) concerning the constituency graph building is entirely covered by the Chapter **??**. It presents how the input dependency graphs are first corrected and normalised and the how they are rewritten, using a custom algorithm, into constituency graphs.

The second phase of the pipeline (see Figure 1.8) concerning the enrichment of the constituency graph with increasingly more semantic features is described in the Chapter **??**. It addresses two main system networks, that of MOOD and TRANSITIVITY introduced in Chapter **??**. The MOOD features are close to syntactic variation of text and can be addressed via graph patterns alone in the first part of the chapter. The TRANSITIVITY features are semantic in nature and require additional lexical-semantic resources from which graph patterns are generated first and then applied to enrich the constituency graph.

Chapter **??** describes how the Parsimonious Vole parser was evaluated with a small evaluation set created for this parser. The chapter describes the evaluation settings and the results for syntactic and semantic parsing. Chapter 2 concludes this work by providing a thesis summary overview, indications for practical applications of this work and future directions to follow.

Chapter 2

Conclusions

The aim of this work is to design a reliable method for English text parsing with Systemic Functional Grammars. To achieve this goal I have designed a pipeline which, starting from a dependency parse of a sentence, generates the SFL-like constituency structure serving as a syntactic backbone and then enriches it with various grammatical features.

In this process a primary milestone the first steps is the creation of constituency structure. Chapter **??** describes the essential theoretical foundations of two SFL schools, namely Sydney and Cardiff schools, and provides a critical analysis of the two to reconcile on the diverging points on rank scale, unit classes, the constituency structure, treatment of coordination, grammatical unit structure, clause boundaries, etc. and state the position adopted in this work.

In order to create the constituency structure from the dependency structure there needs to be a mechanism in place providing a theoretical and a practical mapping between the two. The theoretical account on the dependency grammar and how it is related to SFL is described in Chapter **??**. The practical aspects and concrete algorithms are described in Chapter **??** together with the mapping rules used in the process.

To make clear what are the basic ingredients and how the algorithms are cooked, Chapter **??** introduces all the data structures and operations on them. These structures are defined from a computer scientific point of view emulating the needed SFL concepts. These range from a few graph types, simple attribute-value dictionaries and ordered lists with logical operators. In addition to that, a set of specific graph operations have been defined to perform pattern matching and system network traversals.

Once the constituency structure is created, the second milestone is to enrich it with systemic features. Many features can be associated to or derived from the dependency

and constituency graph fragments. Therefore graph pattern matching is a cornerstone operation used for inserting new or missing units and adding features to existing ones. I describe these operations in detail in the second part of **??**. Then in Chapters **??** and **??** I show how these operations are being used for enrichment of the syntactic backbone with systemic features.

The more precisely graph pattern is defined the less instances it will be matched to and thus decreasing the number of errors and increasing the accuracy. The semantic enrichment is performed via spotting instances of semantic graph patterns. It is often the case that the patterns, in their canonical form, list all the participants of a semantic configuration but in practice, instances of such configurations may miss a participant or two. If applied in their canonical form the patterns will not identify with such the instance. One solution would be to reduce the specificity of the patterns, which leads to a high increase in erroneous applications or populate where possible the covert participants to yield successful matches. It is the second approach that was implemented in this work. To identify and create the covert participants I turned to Government and Binding theory. Two more contributions I bring in this thesis is the theoretical mapping from GBT into dependency structures covered in Chapter **??** and then a concrete implementation described in Chapter **??**.

In the last part of the thesis I describe the empirical evaluation I conducted in order to test the parser accuracy on various features. To conduct this evaluation I created together with Ela Oren a corpus using blog articles of OCD patients covering the Mood system and another corpus was provided to my by Anke Schultz covering the Transitivity system. The results show very good performance (0.6 – 0.9 F1) on Mood features slightly decreasing as the delicacy of the features increases. On Transitivity features, the results are expectedly less precise (0.4 – 0.8 F1) and constitute a good baseline for future improvements.

As discussed in the Section **??** further investigation shall be conducted to determine the error types, shortcomings in the corpus and the parser. Since for both syntactic and semantic annotations there is only a single author annotation available, the results shall be considered indicative and by no means representative for the parser performance. Nevertheless they can already be considered as a good feedback for looking into certain areas of the grammar with considerably low performance in order identify the potential problems.

2.1 Practical applications

A wide variety of tasks in natural language processing such as document classification, topic detection, sentiment analysis, word sense disambiguation don't need parsing. These are tasks can achieve high performance and accuracy with no linguistic feature or with shallow ones such as as lemmas or part of speech by using powerful statical or machine learning techniques. What these tasks have in common is that they generally train on a large corpus and then operate again on large input text to finally yield a prediction for a single feature that they have been trained for. Consider for example the existing methods for sentiment analysis: they will provide a value between -1 to 1 estimating the sentiment polarity for a text that can be anything from one word to a whole page.

Conversely, there are tasks where extracting from text (usually short) as much knowledge as possible is crucial for the task success. Consider a dialogue system: where deep understanding is essential for a meaningful, engaging and close to natural interaction with a human subject. It is no longer enough to assign a few shallow features to the input text, but a deep understanding for planning a proper response. Or consider the case of information extraction or relationship mining tasks when knowledge is extracted at the sub-sentential level thus the deeper linguistic understanding is possible the better.

Current parser is useful to achieve the latter set of tasks. The rich constituency parses can be an essential ingredient for further goals such as anaphora resolution, clausal taxis analysis, rhetoric relation parsing, speech act detection, discourse model generation, knowledge extraction and other ones.

All these tasks are needed for creating an intelligent interactive agent for various domains such as call centers, ticketing agencies, intelligent cars and houses, personal companions or assistants and many other.

In marketing research, understanding the clients needs is one of the primary tasks. Mining intelligence from the unstructured data sources such as forums, customer reviews, social media posts is particularly difficult task. In such cases the more features are available in the analysis the better. With the help of statistical methods feature correlations, predictive models and interpretations can be conveyed for task at hand such as satisfaction level, requirement or complaint discovery, etc.

2.2 Impact on future research

Pattern graphs and the matching methods developed in this work can be applied for expressing many more grammatic features than the ones presented in this work. They can serve as language for systematizing grammatical realizations especially that the realization statements play a vital role in SG grammars. The graph matching method itself can virtually be applied to any other languages than English. So similar parsers can be implemented for other languages and and respectively grammars.

Linguists study various language properties, to do so they need to hand annotate large amounts of text to come up with conclusive statements or formulate hypothesizes. Provided the parser with a target set of feature coverage, the scale at which text analysis is performed can be uplifted orders of magnitude helping linguists come with statistically significant and grounded claims in much shorter time. Parsimonious Vole could play the role of such a text annotator helping the research on text genre, field and tenor.

This section describes improvements of the project that are desirable or at least worth considering along with major improvements that arouse in the process of theoretical development and parser implementation.

2.2.1 Verbal group again: from syntactically towards semantically sound analysis

The *one main verb per clause* principle of the Cardiff school that I adopted in this thesis (briefly discussed in Section **??**) provides a basis for simple and reliable syntactic structures. The alternative is adopting the concept of verbal group, simple or complex, as proposed by the Sydney school in (Halliday & Matthiessen 2013: p.396–418, 567–592), which provides a richer semantically motivated description. However, analysis with verbal group complex is potentially complex one and subject to ambiguities.

Ants	*keep*	*biting*	*me*
Subject	Finite	Predicator	complement
Actor	Process: Material		Goal/Medium
	Verbal group complex expansion, elaborative, time-phase, durative $\alpha \longrightarrow = \beta$		

Table 2.1 Sydney sample analysis of a clause with a *verbal group complex*

Ants	*keep*	-	*biting*	*me*
Subject	Finite/Main Verb	Complement		
Agent	Process: Influential	Phenomena		
		Subject(null)	Main Verb	Complement
		Agent	Process: Action	Affected

Table 2.2 Cardiff sample analysis of a clause *embedded* into another

Check the sample analyses in Table 2.1 and 2.2. The two-clause analysis proposed by Cardiff school can be quite intuitively transformed into a single experiential structure with the top clause expressing a set of aspectual features of the process in the lower (embedded) clause just like in the Sydney analysis in Table 2.1.

The class of *influential* processes proposed in the Cardiff transitivity system was introduced to handle expressions of process aspects through other lexical verbs. I consider it as a class of pseudo-processes with a set of well defined and useful syntactic functions but with poor semantic foundation. The analysis with influential process types reminds me to an unstable chemical substance that, in a chain of reactions, is an intermediary step towards some more stable substance. Similarly, I propose merging the two clauses towards a more meaningful analysis, such as the one suggested by Sydney grammar.

Generalization 2.2.1 (Merging of influential clauses)**.** When the top clause has an influential process and the lower (embedded) one has any of the other processes, then the lower one shall be enriched with aspectual features that can be derived from the top one.

This rule of thumb is described in Generalization 2.2.1. Of course, this raises a set of problems that are worth investigating. Firstly, one should investigate the connections and mappings between the influential process system network described in Cardiff grammar and the system of verbal group complex described in Sydney grammar (Halliday & Matthiessen 2013: p.589). Secondly, one should investigate how this merger impacts the syntactic structure.

The benefits of such a merger leads to an increased comprehensiveness, not only of the transitivity analysis – demonstrated by the examples in Tables 2.1 and 2.2 – but also of the modal assessment that includes modality, as demonstrated by the Examples 10 and 11.

(10) *I think* I've been pushed forward; *I don't really know*, (Halliday & Matthiessen 2013: p.183)

(11) *I believe* Sheridan once said you would've made an excellent pope. (Halliday & Matthiessen 2013: p.182)

Examples 10 and 11 represent cases when the modal assessment of the lower clause is carried on by the higher one. In both examples, the higher clause can be replaced by the modal verb *maybe* or the adverb *perhaps*.

2.2.2 Nominal, Quality, Quantity and other groups of Cardiff grammar: from syntactically towards semantically sound analysis

Cardiff unit classes are semantically motivated as compared to more syntactic ones in Sydney grammar. This has been presented in Sections **??** and discussed in **??**.

For instance, Nominal class structure proposed in Cardiff grammar (discussed in Section **??**), uses elements that are more semantic in nature (e.g. various types of determiners: representational, quantifying, typic, partitive etc.) than the syntactic one offered in Sydney grammar (e.g. only deictic determiner). To do this shift we need to think of two problems: (a) how to detect the semantic head of the nominal units and (b) how to craft (if none exists) a lexical-semantic resources to help determining potential functions (structural elements) for each lexical item in the nominal group. In my view building lexical-semantic resources asked at point (b) bears actually a solution for point (a) as well.

I need to stress that some existing lexical resources such as WordNet (Miller 1995) and/or FrameNet(Baker et al. 1998) could and most likely are suitable for fulfilling the needs at point (b) but the solution is not straight forward and further adaptations need to be done for the context of SFL.

The same holds for Adverbial and Adjectival groups (discussed in Section **??**) which, in Cardiff grammar, are split into the Quality and Quantity groups. The existent lexical resources such as such as WordNet (Miller 1995) and/or FrameNet(Baker et al. 1998) combined with the delicate classification proposed by Tucker (1997) (and other research must exist on adverbial groups of which I am not aware at the moment) can yield positive results in parsing with Cardiff unit classes.

Just like in the case of verb groups discussed in previous section, moving towards semantically motivated unit classes, as proposed in Cardiff grammar, would greatly benefit applications requiring deeper natural language understanding.

2.2.3 Taxis analysis and potential for discourse relation detection

Currently Parsimonious Vole parser implements a simple taxis analysis technique based on patterns represented as regular expressions.

In the Appendix is listed a database of clause taxis patterns according to systematization in IFG 3 (Halliday & Matthiessen 2004). Each relation type has a set of patterns ascribed to it which represent clause order and presence or absence of explicit lexical markers or clause features.

Then, in taxis analysis process, each pair of adjacent clauses in the sentence is tested for compliance with every pattern in the database. The matches represent potential manifestation of the corresponding relation.

Currently this part of the parser has not been tested and it remains a highly desirable future work. Further improvements and developments can be performed based on incremental testing and corrections of the taxis pattern database.

This work can be extended to handle relations between sentences taking on a discourse level analysis which is perfectly in line with the Rhetorical Structure Theory (RST) (Mann & Thompson 1988; Mann et al. 1992).

To increase the accuracy of taxis analysis, I believe the following additional elements shall be included into the pattern representation: Transitivity configurations including process type and participant roles, co-references resolved between clauses/sentences and Textual metafunction analysis in terms of Theme/Rheme and eventually New/Given.

2.2.4 Towards speech act analysis

As Robin Fawcett explains (Fawcett 2011), Halliday's approach to MOOD analysis differs from that of Transitivity in the way that the former is not "pushed forward towards semantics" as the latter is. Having a semantically systematised MOOD system would take the interpersonal text analysis into a realm compatible with Speech Act Theory proposed by Austin (1975) or its latter advancements such as the one of Searle (1969) which, in mainstream linguistics, are placed under the umbrella of pragmatics.

Halliday proposes a simple system of speech functions (Halliday & Matthiessen 2013: p.136) which Fawcett develops into a quite delicate system network (Fawcett 2011). It is worth exploring ways to implement Fawcett's latest developments and because the two are not conflicting but complementing each other, one could use Hallidayan MOOD system as a foundation, especially that it has already been implemented and described in the current work.

2.2.5 Process Types and Participant Roles

The PTDB (Neale 2002) is the first lexical-semantic resource for Cardiff grammar Transitivity system. Its usability in the original form doesn't go beyond that of a resource to be consulted by linguists in the process of manual analysis. It was rich in human understandable comments and remarks but not formal enough to be usable by computers. In the scope of current work the PTDB has been cleaned and brought into a machine readable form but this is far from it's potential as a lexical-grammatical resource for semantic parsing.

In the mainstream computational linguistics, there exist several other lexical-semantic resources used for Semantic Role Labelling (SRL) such as FrameNet (Baker et al. 1998), VerbNet (Kipper et al. 2008). Mapping or combining PTDB with these resources into a new one would yield benefits for both sides combining strengths of each and covering their shortcomings.

Combining PTDB with VerbNet for example, would be my first choice for the following reasons. PTDB is well semantically systematised according to Cardiff Transitivity system however it lacks any links to syntactic manifestations. VerbNet, on the other hand contains an excellent mapping to the syntactic patterns in which each verb occur, each with associated semantic representation of participant roles and some first order predicates. However, the systematization of frames and participant roles could benefit from a more robust basis of categorisation. Also the lexical coverage of VerbNet is wider than that of PTDB.

Turning towards resources like FrameNet and WordNet could bring other benefits. For example FrameNet has a set of annotated examples for every frame which, after transformation into Transitivity system, could be used as a training corpus for machine learning algorithms. Another potential benefit would be generating semantic constrains (for example in terms of WordNet (Miller 1995) synsets or GUM (Bateman et al. 1995, 2010) classes) for every participant role in the system.

PTDB can benefit from mappings with GUM ontology which formalises the experiential model of Sydney school. First by increasing delicacy (at at the moment it covers only three top levels of the system) and second by importing constraints on process types and participant roles from Nigel grammar (Matthiessen 1985). To achieve this, one would have to first map Cardiff and Sydney Transitivity systems and second extract lexical entries from Nigel grammar along with adjacent systemic selections.

2.2.6 Reasoning with systemic networks

Systemic networks are a powerful instrument to represent paradigmatic dimension of language. Besides hierarchies they can include constraints on which selections can actually go together or a more complex set of non hierarchical selection interdependencies. Moreover systemic choices can be also accompanied by the realization rules very useful for generation purpose but they could potentially be used in parsing as well.

In current work system networks are used solely for representation purposes and what would be highly desirable is to enable reasoning capabilities for constraint checking on systemic selections and on syntactic and semantic constituency. For example one could as whether a certain set of features are compatible with each other, or provided a systemic network and several feature selections what would be the whole set of system choices, or being in a particular point in the system network what are the possible next steps towards more delicate systemic choices, or for a particular choice or set of choices what should be present or absent in the constituency structure of the text and so on. All these questions could potentially be resolved by a systemic reasoner.

Martin Kay is the first to attempt formalization of systemics that would become known as Functional Unification Grammar (FUG) (Kay 1985). This formalization caught on popularity in other linguistic domains such as HPSG, Lexical Functional Grammars and Types Feature Structures. One could look at what has been done and adapt the or build a new reasoning system for systemic networks.

With the same goal in mind, one could also look at existing reasoners for different logics and attempt an axiomatization of the systemic networks; and more specifically one could do that in Prolog language or with description logics (DL) as there is a rich set of tools and resources available in the context of Semantic Web.

2.2.7 Creation of richly annotated corpus with all metafunction: interpersonal, experiential and textual

In order to evaluate a parser, a gold standard annotation corpus is essential. The bigger the corpus, covering various the text genres, the more reliable are the evaluation results. A corpus can as well be the source of grammar or distribution probabilities for structure element and potential filling units as is explored by Day (2007), Souter (1996) and other scholars in Cardiff. Moreover such a corpus can also constitute the training data set for a machine learning algorithm for parsing.

A corpus of syntactically annotated texts with Cardiff grammar already exists but, from personal communication with Prof. Robin Fawcett, it is not yet been released to

public because it is considered still incomplete. Even so this corpus covers only the constituency structures and what I would additionally find very useful, would be a set of systemic features of the constituting units covering a full SFG analysis in terms of experiential, interpersonal and textual metafunctions; and not only the unit class and the element it fills.

A small richly annotated set of text had been created in the scope of the current work for the purpose of evaluating the parser. However it is by far not enough to offer a reliable evaluation. Therefore it is highly desirable to create one.

To approach this task one could use a systemic functional annotation tool such as UAM Corpus Tool (O'Donnell 2008a,b) developed and still maintained by Mick O'Donnell or any other tool that supports segment annotation with systemic network tag set structure.

To aid this task one could bootstrap this task by converting other existing corpuses such as Penn Treebank. This task had been already explored by Honnibal in 2004; 2007.

2.2.8 The use of Markov Logics for pattern discovery

Markov Logic (Richardson & Domingos 2006; Domingos et al. 2010) is a probabilistic logic which applies ideas of Markov network to first order logic enabling uncertain inference. What is very interesting about this logics is that tools implementing it have learning capabilities not only of formulas weights but also of new logical clauses.

In current approach I am using graph patterns matching technique to generate a rich set of features for the constituent units. However creating those patterns is a tremendous effort.

Since, graph patterns can be expressed via first order functions and individuals, and assuming that there would already exist a richly annotated corpus, the Markov Logic instruments (for example Alchemy[1], Tuffy[2] and others) can be employed to inductively learn such patterns from the corpus.

This approach resembles the Vertical Strips (VS) of O'Donoghue (1991). The similarity is the probabilistic learning of patterns from the corpus. The difference is that VS patterns are syntactic segment chains from the root node down to tree leafs while with ML more complex patterns can be learned independently of their position in the syntactic tree. Moreover such patterns can be bound to specific feature set.

[1] http://alchemy.cs.washington.edu/
[2] http://i.stanford.edu/hazy/hazy/tuffy/

2.3 A final word

References

Austin, J L. 1975. *How to do things with words*, vol. 3 (Syntax and Semantics 1). Harvard University Press.

Baker, Collin F, Charles J Fillmore & John B Lowe. 1998. The Berkeley FrameNet Project. In Christian Boitet & Pete Whitelock (eds.), *Proceedings of the 36th annual meeting on association for computational linguistics*, vol. 1 ACL '98, 86–90. University of Montreal Association for Computational Linguistics. doi:10.3115/980845.980860. <http://portal.acm.org/citation.cfm?doid=980845.980860>.

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.

Bateman, John A, Renate Henschel & Fabio Rinaldi. 1995. The Generalized Upper Model . Tech. rep. GMD/IPSI. <http://www.fb10.uni-bremen.de/anglistik/langpro/webspace/jb/gum/gum-2.pdf>.

Bateman, John A, Joana Hois, Robert Ross & Thora Tenbrink. 2010. A linguistic ontology of space for natural language processing. *Artificial Intelligence* 174(14). 1027–1071. doi:10.1016/j.artint.2010.05.008. <http://linkinghub.elsevier.com/retrieve/pii/S0004370210000858>.

Bateman, John A. & Christian M. I. M. Matthiessen. 1988. Using a functional grammar as a tool for developing planning algorithms — an illustration drawn from nominal group planning. Tech. rep. Information Sciences Institute Marina del Rey, California. (Penman Development Note).

Bühler, Karl. 1934. *Sprachtheorie: die Darstellungsfunktion der Sprache*. Jena: Fischer.

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

Butler, Christopher S. 2003a. *Structure and function: A guide to three major structural-functional theories; Part 1: Approaches to the simplex clause.* Amsterdam and Philadelphia: John Benjamins.

Butler, Christopher S. 2003b. *Structure and function: A guide to three major structural-functional theories; Part 2: From clause to discourse and beyond.* Amsterdam and Philadelphia: John Benjamins.

Carreras, Xavier. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (emnlp-conll)*, .

Carreras, Xavier & Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the ninth conference on computational natural language learning* CONLL '05, 152–164. Stroudsburg, PA, USA: Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=1706543.1706571>.

Chen, Danqi & Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)*, 740–750.

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

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

Chomsky, Noam. 1982. *Some concepts and consequences of the theory of government and binding*, vol. 6. MIT press.

Chomsky, Noam. 1986. *Barriers*, vol. 13. MIT press.

Day, Michael David. 2007. *A Corpus-Consulting Probabilistic Approach to Parsing : the CCPX Parser and its Complementary Components*: Cardiff University dissertation.

Domingos, Pedro, Stanley Kok, Daniel Lowd & Hoifung Poon. 2010. Markov Logic. *Journal of computational biology a journal of computational molecular cell biology* 17(11). 1491–508. doi:10.1089/cmb.2010.0044. <http://www.ncbi.nlm.nih.gov/pubmed/21685052>.

Fawcett, R. 1988a. 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. 1988b. What makes a 'good' system network good? In James D. Benson & William S. Greaves (eds.), *Systemic functional approaches to discourse*, 1–28. Norwood, NJ: Ablex.

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.

Fawcett, Robin P. 2011. A semantic system network for MOOD in English (and some complementary system networks).

Fawcett, Robin P. forthcoming. How to Analyze Process and Participant Roles. In *The functional semantics handbook: Analyzing english at the level of meaning*, Continuum.

Fellbaum, Christiane & George Miller (eds.). 1998. *WordNet: An electronic lexical database*. The MIT Press.

Fillmore, Charles J. 1985. Frames and the semantics of understanding. *Quaderni di Semantica* 6(2). 222–254. <http://scholar.google.it/scholar?q=fillmore{&}hl=it{&}btnG=Cerca{&}lr={#}5>.

Fillmore, Charles J, Christopher R Johnson & Miriam RL Petruck. 2003. Background to framenet. *International journal of lexicography* 16(3). 235–250.

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

Gildea, Daniel & Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics* 28(3). 245–288.

Haegeman, Liliane. 1991. *Introduction to Government and Binding Theory*, vol. 2. Blackwell.

Halliday, Michael A. K. 1957. Some aspects of systematic description and comparison in grammatical analysis. In *Studies in Linguistic Analysis*, 54–67. Oxford: Blackwell.

Halliday, Michael A. K. 1961a. Categories of the theory of grammar. *Word* 17(3). 241–292.

Halliday, Michael A. K. 1961b. 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. 1996. On grammar and grammatics. In Ruqaiya Hasan, Carmel Cloran & David Butt (eds.), *Functional descriptions – theory in practice* Current Issues in Linguistic Theory, 1–38. Amsterdam: Benjamins.

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

Halliday, Michael A. K. 2003a. 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. 2003b. On the "architecture" of human language. In Jonathan Webster (ed.), *On language and linguistics*, vol. 3 Collected Works of M. A. K. Halliday, 1–32. Continuum.

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

Halliday, Michael A.K. & M.I.M. Matthiessen, Christian. 2004. *An introduction to functional grammar (3^{rd} Edition)*. Hodder Education.

Harnad, Stevan. 1992. The Turing Test Is Not A Trick: Turing Indistinguishability Is A Scientific Criterion. *SIGART Bulletin* 3(4). 9–10. <http://users.ecs.soton.ac.uk/harnad/Papers/Harnad/harnad92.turing.html>.

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.

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

Honnibal, Matthew. 2004. Converting the Penn Treebank to Systemic Functional Grammar. *Technology* 147–154.

Honnibal, Matthew & Jr James R Curran. 2007. Creating a systemic functional grammar corpus from the Penn treebank. *Proceedings of the Workshop on Deep ...* 89–96. doi:10.3115/1608912.1608927. <http://dl.acm.org/citation.cfm?id=1608927>.

Hutchins, W John. 1999. Retrospect and prospect in computer-based translation. In *Proceedings of mt summit vii "mt in the great translation era"* September, 30–44. AAMT.

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>.

Kasper, Robert. 1988. An Experimental Parser for Systemic Grammars. In *Proceedings of the 12^{th} International Conference on Computational Linguistics*, .

Kay, Martin. 1985. Parsing In Functional Unification Grammar. In D.Dowty, L. Karttunen & A. Zwicky (eds.), *Natural language parsing*, Cambridge University Press.

Kipper, Karin, Anna Korhonen, Neville Ryant & Martha Palmer. 2008. A large-scale classification of English verbs. *Language Resources And Evaluation* 42(1). 21–40. doi:10.1007/s10579-007-9048-2.

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>.

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

Mann, William C. 1983a. An Overview of the PENMAN Text Generation System. Tech. Rep. ISI/RR-83-114 USC/Information Sciences Institute Marina del Rey, CA.

Mann, William C. 1983b. 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.

Mann, William C., Christian M. I. M. Matthiessen & Sandra A. Thompson. 1992. Rhetorical Structure Theory and Text Analysis. In William C Mann & Sandra A Thompson (eds.), *Discourse description: Diverse linguistic analyses of a fund-raising text*, vol. 16 Pragmatics & Beyond New Series, 39–79. John Benjamins Publishing Company.

Mann, William C & Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text* 8(3). 243–281. doi:10.1515/text.1.1988.8.3.243.

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.

Matthiessen, M.I.M., Christian. 1985. The systemic framework in text generation: Nigel. In James Benson & Willian Greaves (eds.), *Systemic perspective on Discourse, Vol I*, 96–118. Ablex.

McCarthy, John, Marvin L. Minsky, Nathaniel Rochester & Claude E. Shannon. 2006. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI Magazine* 27(4). 12. doi:10.1609/aimag.v27i4.1904. <http://www.aaai.org/ojs/index.php/aimagazine/article/view/1904{%}5Cnhttp://www.mendeley.com/catalog/proposal-dartmouth-summer-research-project-artificial-intelligence-august-31-1955/{%}5Cnhttp://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.htmlhttp://>.

McDonald, David D. 1980. *Natural Language Production as a Process of Decision Making under Constraint*: MIT, Cambridge, Mass dissertation.

McDonald, Ryan, Koby Crammer & Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, 91–98. Association for Computational Linguistics.

McDonald, Ryan, Kevin Lerman & Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the tenth conference on computational natural language learning* CoNLL-X '06, 216–220. Stroudsburg, PA, USA: Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=1596276.1596317>.

McDonald, Ryan & Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *11th conference of the european chapter of the association for computational linguistics*, .

Miller, George A. 1995. WordNet: a lexical database for English.

Miyao, Yusuke & Jun'ichi Tsujii. 2005. Probabilistic disambiguation models for wide-coverage hpsg parsing. In *Proceedings of the 43rd annual meeting on association for computational linguistics* ACL '05, 83–90. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1219840.1219851. <https://doi.org/10.3115/1219840.1219851>.

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

Neale, Amy C. 2002. More Delicate TRANSITIVITY: Extending the PROCESS TYPE for English to include full semantic classifications. Tech. rep. Cardiff University.

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

O'Donnell, Michael. 1993. Reducing Complexity in Systemic Parser. In *Proceeedings of the third international workshop on parsing technologies*, .

O'Donnell, Michael J. & John A. Bateman. 2005. SFL in computational contexts: a contemporary history. In Ruqiaya Hasan, M.I.M. Matthiessen, Christian & Jonathan Webster (eds.), *Continuing discourse on language: A functional perspective*, vol. 1 Booth 1956, 343–382. Equinox Publishing Ltd.

O'Donnell, Mick. 2008a. Demonstration of the UAM CorpusTool for text and image annotation. In *Proceedings of the acl-08:hlt demo session* June, 13–16.

O'Donnell, Mick. 2008b. The UAM CorpusTool: Software for Corpus Annotation and Exploration. In Bretones Callejas & Carmen M. (eds.), *Applied linguistics now: Understanding language and mind*, vol. 00, 1433–1447. Universidad de Almería.

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

Pei, Wenzhe, Tao Ge & Baobao Chang. 2015. An effective neural network model for graph-based dependency parsing. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, vol. 1, 313–322.

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.

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.

Richardson, Matthew & P. Domingos. 2006. Markov logic networks. *Machine learning* 62(1-2). 107–136. doi:10.1007/s10994-006-5833-1.

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.

Schuler, Karin Kipper. 2005. Verbnet: A broad-coverage, comprehensive verb lexicon .

Searle, John R. 1969. *Speech Acts: An Essay in the Philosophy of Language*, vol. 0. Cambridge University Press. <http://books.google.com/books?id=t3{_}WhfknvF0C{&}pgis=1>.

Souter, David Clive. 1996. *A Corpus-Trained Parser for Systemic-Functional Syntax*: University of Leeds Phd. <http://etheses.whiterose.ac.uk/1268/>.

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.

Turing, Allan. 1950. Computing machinery and intelligence. *Mind* 59. 433–460.

Winograd, Terry. 1972. *Understanding natural language*. Orlando, FL, USA: Academic Press, Inc. <http://linkinghub.elsevier.com/retrieve/pii/0010028572900023>.

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

Zhang, Yue & Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies: short papers-volume 2*, 188–193. Association for Computational Linguistics.

SFL Syntactic Overview

.1 Cardiff Syntax

Elements found in all groups: Linker (&), Inferer (I), Starter (st), Ender (e)
Units: Sentence (Σ), Clause (Cl), Nominal Group (ngp), Prepositional Group (pgp), Quality Group (qlgp), Quantity Group (qtgp), Genitive Cluster (gencl)

.1.1 Clause

Relative Order of Elements in the Unit Structure:
& |B |L |F |A |C |O |S |O |N |A |I |X |M |Mex |C |A |V |E
Clause May fill: Σ (85%), C (7%), A (4%), Q (2%), f (0.5%), s, qtf, S, m, cv, po
Elements of the Clause: Adjunct (A), Binder (B), Complement (C), Formulaic Element (F), Infinitive Element (I), Let Element (L),Main Verb (M), Main Verb Extension (Mex), Negator (N), Operator (O), Subject (S), Vocative (V), Auxiliary Verb (A), X extension (Xex), Linker (&), Starter (St), Ender(E)

.1.2 Nominal Group

Possible Relative Order of Elements in the Unit Structure:
& |rd |v |pd |v |qd |v |sd |v |od |v |td |v |dd |m |h |q |e
Filling probabilities of the ngp: S (45%), C (32%), cv (15%), A (3%), m (2%), Mex, V, rd, pd, fd, qd, td, q, dt, po
Elements of the ngp: Representational determiner (rd), Selector (v), Partitive Determiner (pd), Fractionative Determiner (fd), Quantifying Determiner (qd), Superlative Determiner (sd), Ordinative Determiner (od), Qualifier-Introducing Determiner (qid), Typic Determiner (td), Deictic Determiner (dd), Modifier (m), Head (h), Qualifier (q)

.1.3 Prepositional Group

Possible Relative Order of Elements in the Unit Structure:
& |pt |p |cv |p |e
Filling Probabilities of the pgp: C (55%), a (30%), q (12%), s (2%) Mex, S, cv, f, qtf
Elements of the pgp: Preposition (p), Prepositional Temperer (pt), Completive (c)

.1.4 Quality Group

Possible Relative Order of Elements in the Unit Structure:
& |qld |qlq |et |dt |at |a |dt |s |f |s |e
Filling probabilities of the qgp: c (38%), m (36%), A (24%), sd (0.5%), Mex, Xex, od, q, dt, at, p, S
Elements of the qlgp: Quality Group Deictic (qld), Quality Group Quantifier (qlq), Emphasizing Temperer (et), Degree Temperer (dt), Adjunctival Temperer (at), Apex (a), Scope (s), Finisher (f)

.1.5 Quantity Group

Possible Relative Order of Elements in the Unit Structure:
ad |am |qtf |e **Filling probabilities of the qtgp:** qd (85%), A (8%), dt (6%), B, p, ad, fd, sd **Elements of the qtgp** Adjustor (ad), Amount (am), Quantity Finisher (qf)

.1.6 Genitive Cluster

Possible Relative Order of Elements in the Unit Structure:
& |po |g |o |e
Filling probabilities of the gencl: dd (99%), h, m, qld
Elements of the gencl: Possessor (po), Genitive Element (g), Own Element (o)

.2 Sydney Syntax

.2.1 Logical

Possible Relative Order of Elements in the Unit Structure:
Pre-Modifier |Head |Post-Modifier

.2.2 Textual

Possible Relative Order of Elements in the Clause Structure:
Theme |Rheme
New |Given |New

.2.3 Interactional

Possible Relative Order of Elements in the Clause Structure:
Residue |Mood |Residue |Mood tag
Adjunct |Complement |Finite |Subject |Finite |Adjunct |Predicator |Complement| Adjunct

.2.4 Experiential

Possible Relative Order of Elements in the Clause Structure:
Circumstance |Participant |Circumstance |Process| Participant |Circumstance
Possible Relative Order of Elements in the Nominal Group Structure:
Deictic |Numerative |Epithet | Classifier| Thing |Qualifier
Possible Relative Order of Elements in the Verbal Group Structure:
Finite |Marker |Auxiliary |Event
Possible Relative Order of Elements in the Adverbial and Preposition Group Structure: Modifier |Head |Post-Modifier
Possible Relative Order of Elements in the Prepositional Phrase Structure:
Predicator |Complement
Process |Range

.2.5 Taxis

Possible Relative Order of Elements in the Parataxis Structure:
Initiating |Continuing
Possible Relative Order of Elements in the Hypoataxis Structure:
Dependent |Dominant |Dependent

Stanford Dependency schema

The Stanford dependency relations as defined in Stanford typed dependencies manual (Marneffe & Manning 2008a)

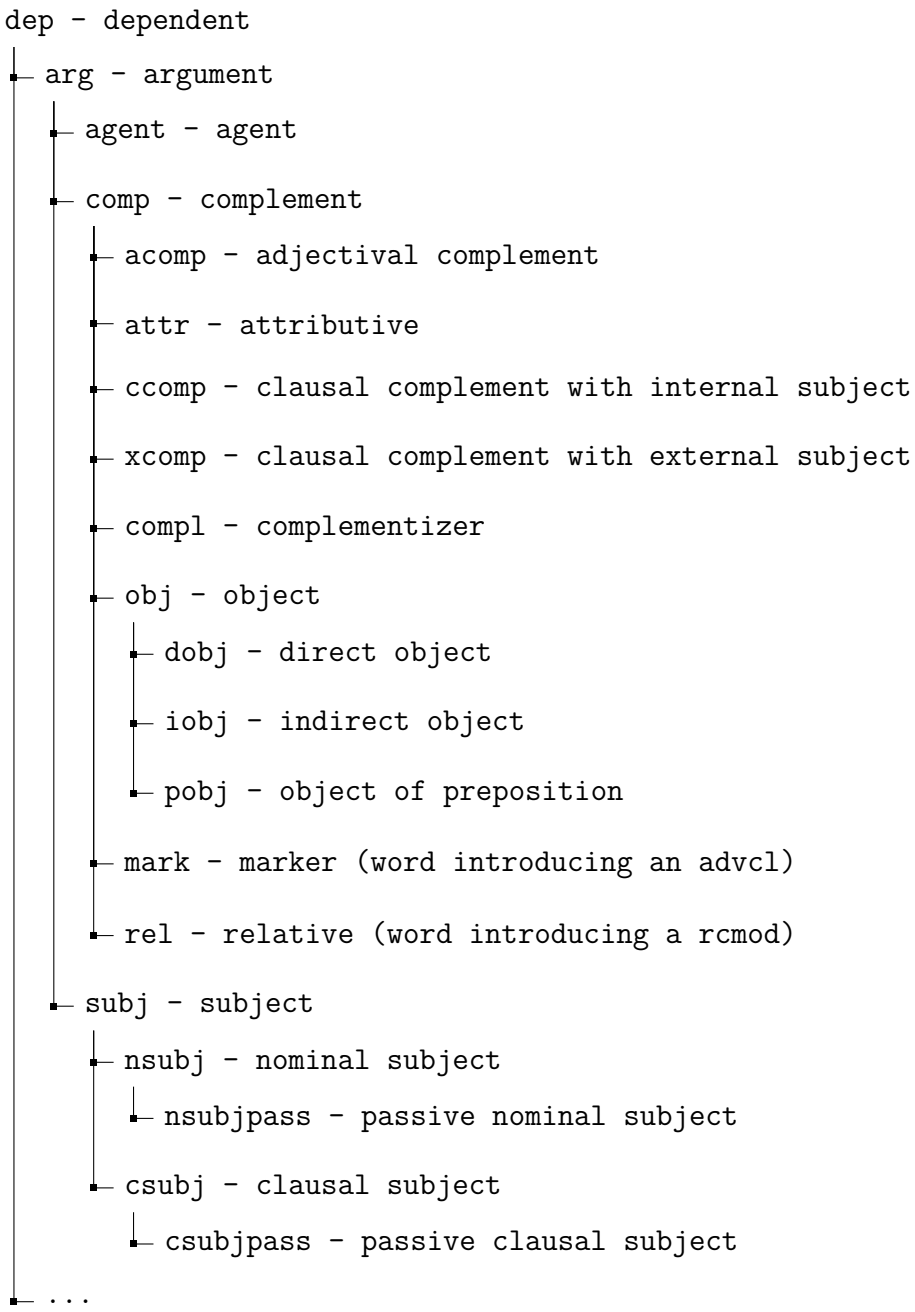

Fig. 1 The Stanford dependency scheme - part one

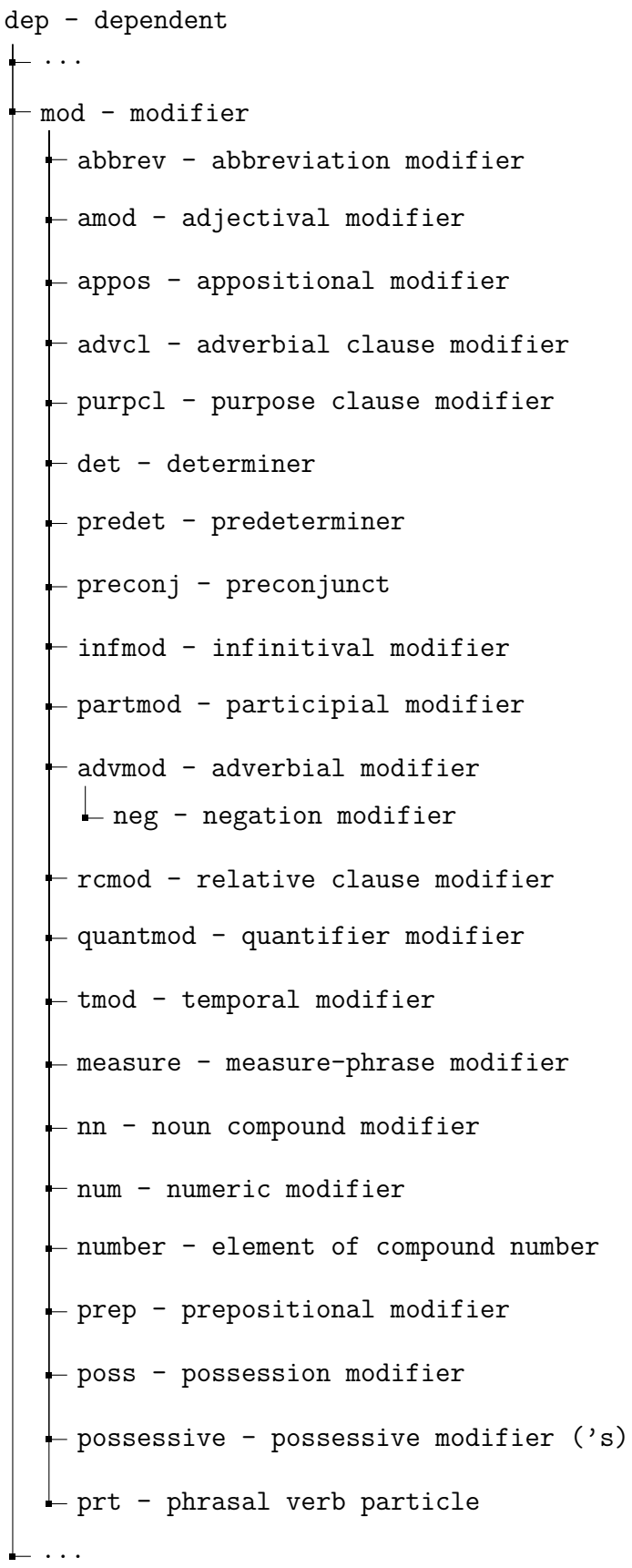

Fig. 2 The Stanford dependency scheme - part two

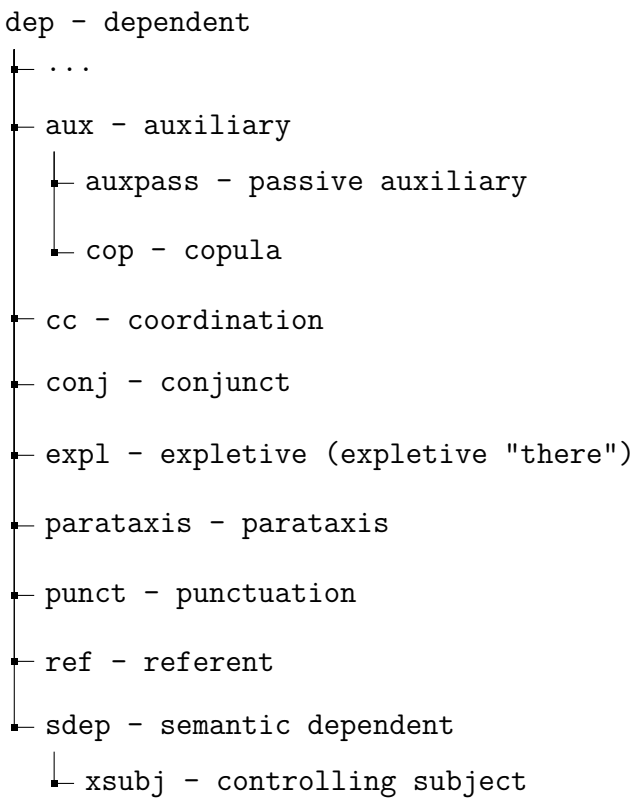

Fig. 3 The Stanford dependency scheme - part three

Penn treebank tag-set

Tag	**Description**	**Example**
CC	conjunction, coordinating	and, or, but
CD	cardinal number	five, three, 13%
DT	determiner	the, a, these
EX	existential there	there were six boys
FW	foreign word	mais
IN	conjunction, subordinating or preposition	of, on, before, unless
JJ	adjective	nice, easy
JJR	adjective, comparative	nicer, easier
JJS	adjective, superlative	nicest, easiest
LS	list item marker	
MD	verb, modal auxillary	may, should
NN	noun, singular or mass	tiger, chair, laughter
NNS	noun, plural	tigers, chairs, insects
NNP	noun, proper singular	Germany, God, Alice
NNPS	noun, proper plural	we met two Christmases ago
PDT	predeterminer	both his children
POS	possessive ending	's
PRP	pronoun, personal	me, you, it
PRP$	pronoun, possessive	my, your, our
RB	adverb	extremely, loudly, hard
RBR	adverb, comparative	better
RBS	adverb, superlative	best
RP	adverb, particle	about, off, up
SYM	symbol	%
TO	infinitival to	what to do?
UH	interjection	oh, oops, gosh
VB	verb, base form	think
VBZ	verb, 3rd person singular present	she thinks
VBP	verb, non-3rd person singular present	I think
VBD	verb, past tense	they thought
VBN	verb, past participle	a sunken ship
VBG	verb, gerund or present participle	thinking is fun
WDT	wh-determiner	which, whatever, whichever
WP	wh-pronoun, personal	what, who, whom
WP$	wh-pronoun, possessive	whose, whosever
WRB	wh-adverb	where, when
.	punctuation mark, sentence closer	.;?*
,	punctuation mark, comma	,
:	punctuation mark, colon	:
(	contextual separator, left paren	(
)	contextual separator, right paren	)

Table 3 Penn Treebank tag set