Chapter 1

Introduction

1.1 On artificial intelligence 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.

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 intelligible 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. Yet there is still much work to be done before the technology is capable to pertorm such complex levels of automatic analisys.

1.4 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 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. This explanation is extended further in this section emphasizing SFL strengths.

Any meaningful description or analysis involving language implies some theory 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. Thus, 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 *syntagmatic* (structural) axis and represented as *system networks*. Moreover, in the SFL perspective language has evolved to serve particular *functions* influencing their 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

units. In the figure, nodes 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.

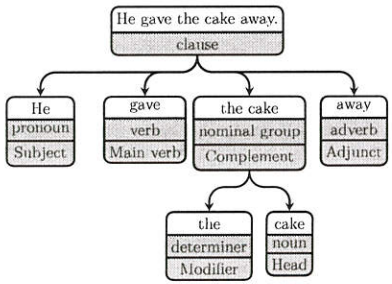

Fig. 1.2 Constituency analysis of Example 1 with unit classes and grammatical functions

Next each constituent can be assigned a set of relevant linguistic features. For example the subject "He" is a pronoun whose features, well defined in the traditional grammar, are: *singular*, *masculine*, and *3rd person*. 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 1.3 below. This representation constitutes what in SFL is called a *system network* and will be formally introduce in Chapter 3.

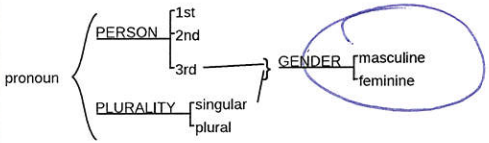

Fig. 1.3 The systematisation of three pronominal features in traditional grammar

In SFG the pronouns are systematised in the system network of Person from *Introduction to Functional Grammar* (Halliday & Matthiessen 2013: 366) that has a different structure as depicted in Figure 1.4. This systematisation reflects a semiotic

perspective where language is placed into an interactive context. The (red) rectangles from the figure represent 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 in case one is available.

From the perspective of an agent generating the utterance in Example 1, to produce the pronoun "he" in the subject position it has to make a few choices in the system network. This process is called system network *traversal*. A simplified traversal for selecting the needed pronominal referent can be described as follows. For now, to make it simpler, the explanation on how the decissions are made is ommited focusing mainly on the traversal process itself. So, first the deciding agent chooses in the PERSON system whether the referent participates in the interaction or not (see Figure 1.4). In our example the referent does not participate so the *non-interactant* feature is selected and we proceeds towards the next system further distinguishing the type of *non-interactant*. It can be plural or, as in our case, singular leading to *one-referent* feature. Next, the referent needs to be differentiated on the consciousness axis which, in our example, is a *conscious* thing. And finally conscious referents 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, just like the one in Figure 1.3, the boundaries of all choice possibilities.

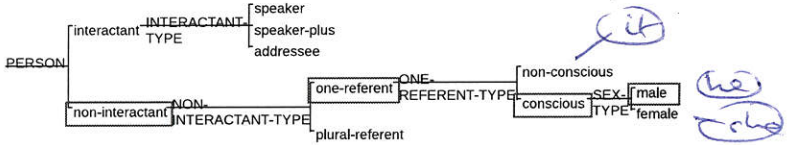

Fig. 1.4 The selections in Person system network from Halliday & Matthiessen (2013: 366) for pronoun "He"

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

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 Challenges of parsing with SFGs

In this sections I describe the main challenges for using Systemic Functional Grammars (SFG) in computational contexts in general and parsing in particular. The first and the main challenge in parsing with SFGs is that of computational complexity. This problem stems partially from the manner grammars are structured and partially from the fact that paradigmatic description have received most of the attention in SFL at the expense of the syntagmatic one. The second challenge is parsing with features that depart from directly observable grammatical variations towards increasingly abstract semantic features. Next follows a detailed description of the main problems starting with the imbalance between paradigmatic and syntagmatic accounts in SFL. Then the computational aspects are bough into the picture into as a comparison between the natural language generation and parsing tasks. Finally the problem of parsing abstract features is described drawing parallels to *semantic role labelling* (SRL) task well-defined in the mainstream computational linguistics (Carreras & Màrquez 2005).

1.6.1 Syntagmatic descriptions in SFL

Since it was established, SFL 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 3, 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 3.2.10) 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 Computational complexity appears in parsing

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 (Bateman & Matthiessen 1988: 139). The transfer between SFL and computation typically involved a delay between the theoretical formulation and the computational instantiation of that formulation (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.

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 accurately 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

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).

1.6.3 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 comparable to the task of (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* (PTDB) (Neale 2002).

Such databases provide with domain independent *semantic frames* (Fillmore 1985), know in SFL as *configurations* or *figures* e.g. Action, Cognition, Perception, Possession etc. They describe semantic actions and relationships between *participants* each playing a distinct *semantic role* within the frame e.g. Agent, Carrier, Possessed, Phenomena etc. For instance the perception frame contains *Perceiver* and *Phenomenon* roles annotated in Example 3.

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

The challenge in this work is to implement a semantic parsing process for Transitivity systemic network employing PTDB as the lexical-semantic resource.

1.6.4 Covert elements

Besides the challenge of identifying configurations and their participants in text, the problem with semantic features goes one step further. Sometimes the participant

roles correspond to constituents that are displaced or not realised in the text, called *covert*(Fawcett 2008: 115.135.194) or *null elements*(Chomsky 1981, 1982, 1986). This increases the challenge of identifying frames and assigning roles correctly and next is explained why.

For a frame to be considered correctly realised in text, at least its mandatory roles must be filled by constituent units. This requirement constitutes a minimal semantic completeness constraint. This can be demonstrated by erasing parts of the text in Example 3. 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? This shows that configurations need to satisfy the minimal semantic completeness condition when realised in text. Conversely, one of the fundamental assumptions in this work is that the input text is well-formed and the completeness condition is satisfied.

(4) glanced at her new watch

(5) Jaqueline glanced

Consider now Example 6 consisting of a sentence that has three non-auxiliary verbs: seem, worry and arrive. According to the Cardiff grammar (introduced in Chapter 3) it corresponds to three clauses *embedded* into each other. Table 1.2 provides the constituency analysis in Cardiff of Example 6.

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

<table>
<tr><td>She</td><td>seemed</td><td colspan="7">to worry about missing the river 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 of Example 6

Table 1.3 provides the participant role configurations (i.e. the semantic frames) these verb meanings bring about. Usually the first role corresponds to the Subject function and the second role is filled by a Complement unit.

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

usable in real world applications. This conclusion 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. A detailed account of the current state of the art in parsing with SFGs is provided in Chapter 2. 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. Each had to accept limitations either in grammar or language size and eventually used simpler syntactic trees as a starting point.

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, is to establish a cross-theoretical and inter-grammatical links and to enable integration of the existing methods, resources and solutions in order to maximise reuse of positive outcomes.

The process developed in this thesis follows a pipeline architecture (see Section 1.7.4) 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.

1.7.1 On theoretical compatibility and reuse

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 some theoretical framework may face common or different challenges to those implementing another theoretical frameworks. The converse can be said of the solutions. When a solution is achieved using one framework it becomes potentially reusable in other ones provided a degree of adaptation. Thus the successes and achievements in any school of thought can be regarded as valuable for other ones to

the degree cross theoretical links and correspondences can be established. In this thesis reusing components that have been shown to work and yield "good enough results" is a strong pragmatic motivation in the present work which brings us to Research question 1.

Research question 1 (Reuse positive results). To what extent resources and techniques from other areas of computational linguistics can be reused for the SFL parsing and how?

In this thesis three linguistic frameworks are employed, namely the *Systemic Functional Linguistics*, *Dependency Grammar* and *Governance & Binding Theory*. SFL has already been motivated as target analysis framework in Section 1.4. The other two frameworks are employed because the accomplishments in those domains carry answers to above stated problems.

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.

Research question 2 (Compatibility of DG and SFG). 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?

The dependency parse structures provide information about functional dependencies between words and grants direct access to the predicate-argument relations. This information is sufficient to supplement the missing syntagmatic account in SFL. In addition, it provides some functional information that helps to reduce the complexity of system network traversing. This hypothesis, formulated as Research question 2, is investigated at the theoretical level in Chapter 5 and then empirically evaluated in Chapter 10 based on Stanford Dependencies parser version 3.5 (Marneffe & Manning 2008b,a; Marneffe et al. 2014).

the latter 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 multiple parents. In both cases the tree structure is broken. The graph construct allows a wide range of structural configurations including the trees. This comes with the cost of higher computation power thus subject for optimisations in the future work.

Most of the graph patterns in this work 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.

Research question 5 (Coverage of syntactic patterns). What degree of systemic delicacy can be reached using syntactic patterns alone without any lexical-semantic resources?

The pattern may comprise solely of syntactic specifications or it can also carry lexical-semantic descriptions. The two main system networks targeted in this work are MOOD and TRANSITIVITY (described in Chapter 4). One hypothesis formulated in Research question 5 is that the MOOD network is composed of syntactically identifiable features thus the graph patterns need to involve no more than unit classes and functions available in the constituency structure.

Research question 6 (PTDB suitability). How suitable is Process Type Database as a resource for SFL Transitivity parsing?

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 generation of such patterns. The appropriateness of PTDB for these tasks is inquitred by Research question 6 and addressed in Chapters 4 and 9. In the next section is presented an overview of how these processes fit together in a unitary pasing process.

1.7.4 Parsimonious Vole architecture

The current thesis is accompanied by a software implementation called the Parsimonious Vole parser. It is programmed with Python language and is available as open source

distribution[1]. It takes the text of an English sentence as input and outputs the rich systemic functional constituency structure. This section explains the implemented parsing process architecture.

The parser follows the pipeline architecture depicted in Figure 1.8. Three types of boxes are used here: (a) the red rounded rectangles in the middle represent parsing steps, (b) the green trapezoid boxes represent input and output data and (c) the orange double framed trapezoid boxes represent external resources involved in the parsing process e.g. system network, graph paterns, lexical-semantic databases etc.

The parsing steps linearly flow from one to the next via green trapezoid boxes on the left-hand side of the diagram. It means that the output data of a step constitutes the input for the next one. On the right-hand side are positioned double edged orange trapezoids representing fixed resources needed by some operations. For example, the *graph normalization* step takes a set of graph patterns that serve as normalisation rules indicating how to update the input.

Two green vertical arrows are provided on the far right of the diagram delimiting parsing phases: *Graph Building* (spanning the first three process steps) that accomplishes construction of the constituency backbone (motivated in Section 1.7.2 above) and the second phase *Graph Enrichment* (spanning the last three process steps) flashes out the backbone with features (motivated in Section 1.7.3).

The parsing process starts with from an English text which is sent to Stanford Dependency parser (Chen & Manning 2014) version 3.5[2] to produce a Dependency parse graph of that text. The output is a sequence of dependency graphs corresponding to sentences delimited by punctuation marks.

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, dependency graphs produced by the Stanford parser are *Corrected and Normalised* against a collection of known errors and a set of normalisation rules using pattern matching techniques.

Afterwards, the normalised 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 constructed in parallel guided by a *Rule Table*. This table contains the mapping of structural context fragments from the dependency grammar (i.e. node type, edge type, combinations of the two etc.) to constituency

[1] https://bitbucket.org/lps/parsimonious-vole
[2] https://nlp.stanford.edu/software/nndep.html

Chapter 2

An overview of selected works on parsing with SFG

There have been various attempts to parsing with SFGs. This chapter covers the most significant attempts to parse with a Systemic Functional Grammar. The first attempt was made by Winograd (Winograd 1972) which was more than a parser. it was an interactive a natural language understanding system for manipulating geometric objects in a virtual world.

Starting from early 1980s onwards, Kay, Kasper, O'Donell and Bateman tried to parse with Nigel Grammar (Matthiessen 1985), a large and complex natural language generation (NLG) grammar for English used in Penman generation project. Other attempts by O'Donoghue (1991), Weerasinghe (1994), Souter (1996), Day (2007) aim for corpus based probability driven parsing within the framework of COMMUNAL project starting from late 1980s.

In a very different style, Hounibal (2004); Hounibal & Curran (2007) constructed a system to convert Penn Treebank into a corresponding SFGBank. This managed to provide a good conversion from phrase structure trees into systemic functional constituency trees covering sentence Mood and Theme structure. No systemic feature selections have been assigned to any of the constituents. Also not Transitivity account was provided in their attempts. The current work is following a similar approach of converting a parse structures and in addition providing a set of feature selections from system networks. Next are presented a few selected attempts to parse with SFGs.

2.0.1 Winograd's SHRDLU

SHRDLU is an interactive program for understanding (if limited) natural language written by Terry Winograd at MIT between 1968-1970. It carried a simple dialogue about a world of geometric objects in a virtual world. The human could ask the system to manipulate objects of different colours and shapes and the ask questions about what has been done or the new state of the world.

It is recognised as a landmark in natural language understanding demonstrating that a connection with artificial intelligence is possible if not solved. However, his success was not due to the use of SFG syntax but rather due to small sizes of every system component to achieve a fully functional dialogue system. Not only it was parsing the input but it was developing an interpretation of it, reason about it and generate appropriate natural language response.

Winograd combined the parsing and interpretation processes such that the semantic interpreter was actually guiding the parsing process. The knowledge of syntax was encoded in the procedures of interpretation program. He also implemented an ingenious backtracking mechanism where the the program does not simply go back, like other parsers, to try the next possible combination choice but actually takes a decision on what shall be tried next.

Having data embedded into the program procedures. as Winograd did, makes it non-scalable for example in accommodation of larger grammars and knowledge bodies and unmaintainable on the long term as it becomes increasingly difficult to make changes (Weerasinghe 1994).

2.1 Kasper

Bob Kasper in 1985 being involved in Penman generation project embarked on the mission of testing if the Nigel grammar, then the largest available generation grammar, was suitable for natural language parsing. Being familiar with Functional Unification Grammar (FUG), a formalism developed by Kay and tested in parsing (Kay 1985) which caught on popularity in computational linguistics regardless of Kay's dissatisfaction with results, Kasper decided to re-represent Nigel grammar into FUG.

Faced with tremendous computational complexity, Kasper (1988) decided to manually create the phrase-structure of the sentences with hand-written rules which were mapped onto a parallel systemic tree structure. Kasper in 1988 was the first one to parse with a context-free backbone. He first parsed each sentence with a Phrase Structure Grammar (PSG), typical to Chomsky's Generative Transformational Linguistics

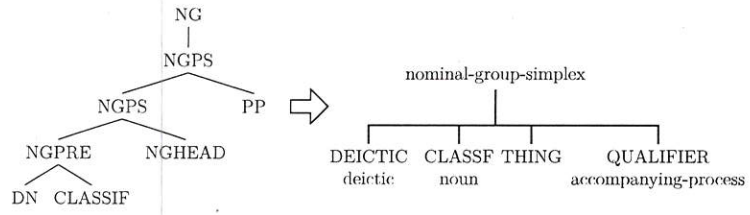

Fig. 2.1 Transformation from phrase structure into systemic constituency structure. Rule example from O'Donnell & Bateman (2005).

Chomsky (1957a). He created a set of rules for mapping the phrase structure (PS) into a parallel systemic tree like the one depicted in Figure 2.1. When all possible systemic tree were created they were further enriched using information from Nigel Grammar (Matthiessen 1985).

Once the context-free phrase-structure was created using bottom-up chart parser it was further enriched from the FUG representation of Nigel grammar. This approach to parsing is called *parsing with a context-free backbone* as phrase-structure is conveyed as simplistic skeletal analysis, fleshed out by the detail rich systemic functional grammar.

Even though Kasper's system is represents the first attempt to parse with full Hallidayan grammar, it's importance is lowered, as O'Donnell & Bateman (2005) point out, by the reliance on phrase structure grammar.

2.2 O'Donnell

Since 1990, Mick O'Donnell experimented with several parsers for small Systemic grammars, but found difficulty when scaling up to larger grammars. While working in EAD project, funded by Fujitsu, he recompiled a subset of Nigel grammar into two resources: the set of possible function bundles allowed by the grammar (along with the bundles preselections) and a resource detailing which functions can follow a particular function (O'Donnell 1993, 1994).

This parser was operating without a syntactic backbone directly from a reasonable scale SFG. However when scaled to the whole Nigel grammar the system became very slow because of the sheer size of the grammar and its inherent complexity introduced by multiple parallel classifications and functional combinations - a problem well described by Bateman (2008). Then O'Donnell wrote his own grammar of Mood that was more suitable for the parsing process and less complex than the recompiled Nigel.

In 2001, while working in a Belgian company O'Donnell came to conclusion that dependency grammars are very efficient for parsing. Together with two colleagues, he developed a simplified systemic grammar where elements were connected through a single function hence avoiding (functional) conflation. Also the ordering of elements was specified relative to the head rather than relative to each other.

More recently, O'Donnell in CAM Corpus Tool embedded a systemic chart parser (O'Donnell 2005) with a reduced systemic formalism. He classifies his parser as a left to right and bottom up with a custom lexicon where verbs are attributed features similar to Hallidayan process types and nouns a unique semantic category like thing-noun, event-noun, location-noun etc.

Because of previously reported complexity problems (O'Donnell 1993) with systemic grammars, the grammatical formalism is reduced to a singular functional layer of Mood-based syntactic structure (Subject, Predicate, Object etc.) ignoring the Transitivity (Actor/Goal, Sensor/Phenomenon etc.) and Textual (Theme/Rheme) analyses. O'Donnell deals away with the conflation except for the verbal group system network. He also employs a slot based ordering where elements do not relate to each other but rather to the group head only simplifying the number of rules and calculation complexity.

In his paper (O'Donnell 2005) does not provide a parser evaluation so its accuracy is still unknown today. The lexicon that was created is claimed to deal with word semantic classes but it is strongly syntactically based assigning a single sense to nouns and verbs ignoring the peculiar aspect of language polysemy. Moreover it is not very clear the framework within which the semantic classes have been generated.

2.3 O'Donoghue

O'Donoghue proposes a corpus based approach to parsing using *Vertical Strips* (O'Donoghue 1991). They are defined as a vertical path of nodes in a parse tree starting from the root down to the lexical items but not including those. He extracted the set of vertical strips from a corpus called Prototype Grammar Corpus together with their frequencies and probability of occurrence. This approach differ from the traditional one with respect to the kind of generalization it is concerned and specifically, the traditional approached are oriented towards horizontal order while the vertical strip approach is concerned with vertical order in the parse tree.

To solve the order problem O'Donoghue uses a set of probabilistic collocation rules extracted from the same corpus indicating which strips can follow a particular strip. He

Chapter 3

Systemic functional theory of grammar

Any description of language requires a theory that provides the frame, scope and necessary concepts. Having a solid theory of grammar contributes to explaining what language is and how it works. It also frames how language is ought to be analysed by either human or machines.

In his seminal paper Halliday (1961b) addresses the ardent need of the time for a general theory of language and partially answers the proposal for a universal theory of language. He sets out what was known at the time as Scale and Category Grammar. In such a model *units* are set up to account for pieces of language which carry grammatical patterns. They are seen as arranged on a hierarchical *rank* scale of words, groups and clauses. These and other foundational concepts are covered in the first part of this Chapter.

There are two variants of Systemic Functional Grammars: the *Sydney Grammar* started in 1961 by Halliday (2002) and the *Cardiff Grammar* proposed by Fawcett (2008) which is a simplification and an extension of the Sydney Grammar. To understand the underlying common motives and how they are different we shall start looking at their theories of grammar. They also have quite different historical developments.

Sydney and Cardiff grammars have been formalised to the point where they could be computationally applied to natural language generation. They have been implemented in PENMAN (Mann 1983b; Penman Project 1989) and respectively COMMUNAL projects (Fawcett 1990). Both versions of SF grammars have been used predominantly for English and implementations of for other languages are also available. The major component of PENMAN is a computer model of Halliday's SF grammar described by Mann & Matthiessen (February 1983), Matthiessen & Bateman (1991), Matthiessen

1995) and others. COMMUNAL is the computer implementation of Cardiff grammar described by Fawcett (1988a), Fawcett (1993) and others.

This chapter first sets out the basic organisational dimensions for each of the theories and then discusses comparatively Halliday's (Halliday 2002) and Fawcett's (Fawcett 2000) versions of SFL.

3.1 A word on wording

Before going into deeper discussion I first make terminological clarifications on the terms: grammar, grammatics, syntax, semantics and lexicogrammar. I start with definitions adopted in "mainstream" generative linguistics and then present how the same terms are discussed in systemic functional linguistics.

Radford, a generative linguist, in the "Minimalist Introduction to Syntax" (1997), starts with a description of grammar as a field of study, which, in his words, is traditionally subdivided into two inter-related areas of study: syntax and morphology.

Definition 3.1.1 (Morphology (Radford)). Morphology is the study of how words are formed out of smaller units (traditionally called morphemes) (Radford 1997: 1).

Definition 3.1.2 (Syntax (Radford)). Syntax is the study of how words can be combined together to form phrases and sentences. (Radford 1997: 1)

Halliday, in the context of *rank* scale discussion (see Definition 3.2.1 and 3.2.2), refers to the traditional meaning of syntax as the *grammar above the word* and to morphology as *grammar below the word* (Halliday 2002: 51). Such a distinction, he states, has no theoretical status and is deemed as unnecessary distinction. Halliday adopts this position to motivate the architecture of grammar he was developing and is inherited from his precursor, Firth, as he puts it:

> ...the distinction between morphology and syntax is no longer useful or convenient in descriptive linguistics. (Firth 1957: 14)

Radford adds that, traditionally, grammar is not only concerned with the principles governing formation of words, phrases and sentences but also with principles governing their interpretation. Therefore *structural aspects of meaning* are said to be also a part of grammar.

Definition 3.1.3 (Grammar (Radford)). [Grammar is] the study of the principles which govern the formation and interpretation of words, phrases and sentences. (Radford 1997: 1)

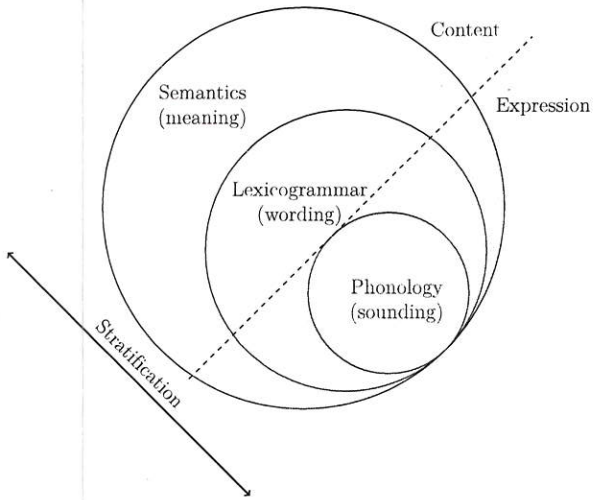

Fig. 3.1 The levels of abstraction along the realisation axis

The SFL model defines language as a resource organised into three strata: phonology (sounding), lexicogrammar (wording) and semantics (meaning). Each is defined according to its level of abstraction on the realisation axis. The realisation axis is divided into two planes: the expression and the content planes. Although debate about the precise division continues, for current purpose it is sufficient to see the first stratum (i.e. phonology/morphology) belongs to the *expression plane* and the last two (lexicogrammar and semantics) belong to the *content plane*. In this context, the formal grammar could be localised entirely within the expression plane, including the phonology/morphology, syntax, lexicon while formal semantics, stripped of any explanations in terms of the meaning potential, belongs in the content plane.

3.2 Sydney theory of grammar

I start introducing the terms of SFL theory with the Sydney grammar as this is in accordance with the historical development originating with Halliday (2002) defining the categories of the theory for grammar. He proposes four fundamental categories:

unit, *structure*, *class* and *system*. Each of these categories is logically derivable from and related to the other ones in a way that they mutually define each other. These categories relate to each other on three scales of abstraction: *rank*, *exponence*, *delicacy*. Halliday also uses three scale types: *hierarchy*, *taxonomy* and *cline*.

Definition 3.2.1 (Hierarchy). Hierarchy [is] a system of terms related along a single dimension which involves some sort of logical precedence. (Halliday 2002: 42).

Definition 3.2.2 (Taxonomy). Taxonomy [is] a type of hierarchy with two characteristics:

1. the relation between terms and the immediately following and preceding one is constant
2. the degree is significant and is defined by the place in the order of a term relative to following and preceding terms. (Halliday 2002: 42)

Definition 3.2.3 (Cline). Cline [is] a hierarchy that instead of being made of a number of discrete terms, is a continuum carrying potentially infinite gradations. (Halliday 2002: 42).

The concept of cline may not necessarily originate in SFL but it is used quite extensively in the domain literature. Next I define and introduce each category of *grammatics* and the related concepts that constitute the theoretical foundation for the Sydney Theory of grammar.

3.2.1 Unit

Language is a patterned activity of meaningful organization. The patterned organization of substance (*graphic* or *phonic*) along a linear progression is called *syntagmatic order* (or simply *order*).

Definition 3.2.4 (Unit). The unit is a grammatical category that accounts for the stretches that carry grammatical patterns (Halliday 2002: 42). The units carry a fundamental *class* distinction and should be fully identifiable in description (Halliday 2002: 45).

Generalization 3.2.1 (Constituency principles). The five principles of constituency in lexicogrammar are:

1. There is a scale or rank in the grammar of every language. That of English (typical of many) can be represented as: clause, group/phrase, word, morpheme.

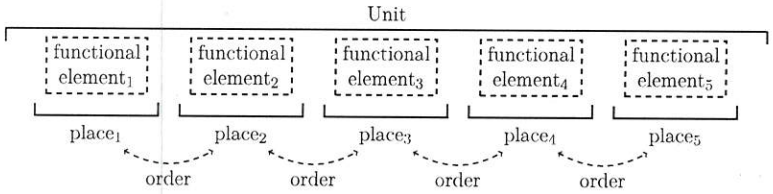

Fig. 3.2 The graphic representation of (unit) structure

is depicted in Figure 3.2. The unit structure is referred in linguistic terminology as *constituency* (whose principles are enumerated in Generalization 3.2.1). In the unit structure, the elements resemble an array of empty slots that are *filled* by other units or lexical items.

For example to account for the English clause structure four elements are needed: *subject*, *predicator*, *complement* and *adjunct*. They yield the distinct symbols, so that S, P, C, A is the inventory of elements. They then can be arranged in various orders falling in particular places, say SPC, SAPA, ASPCC etc. The places of elements are important with respect to the structure of the whole unit but also with respect to the relative ordering between these elements. For example S always fronts P, C is fronted by P unless the clause realises a Wh-interrogative whereas A is quite free and can occur anywhere in the unit structure.

3.2.3 Class

To one place in the structure corresponds one occurrence of the unit next below. This means that there will be a certain grouping of members identified by the functional element they take in the structure. Patterning such groupings leads to emergence of *classes* of units.

In the clause structure example, elements in the unit are occupied by units of lower rank and of a particular class. The relation between the element and the class is mutually determined. In each of these elements is placed a lower rank unit and of an expected class. For instance in the S position can be placed a *noun*, *nominal group*, *pronoun* or another *clause* (that will be a down-ranking situation defined above).

Definition 3.2.8 (Class). The class is that grouping of members of a given unit which is defined by the operation (i.e. functional element) in the structure of the unit next above (Halliday 2002: 49).

Halliday defines class (Definition 3.2.8) as likeness of the same rank *phenomena* to occur together in the structure. He adopts a top-down approach stating that the class of a unit is determined by the *function* (Definition 3.2.13) it plays in the unit above and not by its internal structure of elements. In SFG the structure of each class is well accounted in terms of syntactic variation recognizing six unit classes: *clause*, *nominal*, *verbal*, *adverbial* and *conjunction* groups and *prepositional phrase*. The Sydney unit structure model is briefly summarised in the Appendix 11.3.

Halliday identifies the concept of *grammatical metaphor* defined in 3.2.9 and it plays an important role in the SFG as a whole for accounting for the versatility of natural language. It typically found in adult language where one type of process are expressed in the grammar of another.

Definition 3.2.9 (Grammatical metaphor). Grammatical metaphor involves the substitution of one grammatical class or structure for another, often resulting in a more compressed expression.

(13) The fifth day saw them at the summit.

(14) On the fifth day they arrived at the summit.

(15) Guarantee limited to refund of purchase price of goods.

(16) we guarantee only to refund the price for which the goods were purchased.

For example 13 and 15 are instances of grammatical metaphor whereas the 14 and 16 are their non metaphorical counterparts. In Examples 13 and 14 the temporal circumstance of an action expressed through a prepositional phrase becomes the nominal actor of a perception process. Children speech is largely free of such kind of metaphors, in fact this is the main distinctions between the two.

3.2.4 System

As described above, structure is a syntagmatic ordering in language capturing regularities and patterns which can be paraphrased as *what goes together with what*. However in SFG most of the descriptive work is carried not syntagmatically but paradigmatically via *system networks* (Definition 3.2.10) describing *what could go instead of what* (Halliday & Matthiessen 2013: 22). Note that the paradigmatic-syntagmatic axes date back to the works of Saussure (1959 [1915]). Both are important for completing a linguistic description. Here lies one of the main differences between SFL and other approaches which is is taking the paradigmatic path whereas many others take the syntagmatic path to language representing it as an inventory of structures. The structure of course

And in relation to the previous section, the class stands in the relation of exponence to an element of primary structure of the unit next above. This breakdown gives a system of classes that constitute choices implied by the nature of the class (Halliday 2002: 41).

3.2.5 Functions and metafunction

Above, when talking about structure, I described a unit as being composed of elements accounted in terms of *functions* and places taken by the lower (constituting) units or lexical items.

Definition 3.2.13 (Function). The functional categories or functions provide an interpretation of grammatical structure in terms of the overall meaning potential of the language (Halliday & Matthiessen 2013: 76).

Most constituents of clause structure, however, have more than one function, which is called a *conflation of elements*. For example in the sentence "Bill gave Dolly a rose", "Bill" is the Actor doing the act of giving but also the Subject of the sentence. So we say that Actor and Subject functions are conflated in the constituent "Bill". This is the concept of *metafunction* or *strand of meaning* comes into the picture. The Subject function is said to belong to the *interpersonal metafunction*, while the Actor function belongs to the *experiential metafunction*.

Halliday identifies three fundamental dimensions of structure in the clause, each meaning: *experiential*, *interpersonal* and *textual*. He refers to them as *metafunctions* and they account for the functions that language units take on in communication. Table 3.2 presents the metafunctions and their reflexes in grammar as proposed by Halliday & Matthiessen (2013: 85).

Metafunction	Definition(kind of meaning)	Corresponding status in clause	Favored type of structure
experiential	construing a model of experience	clause as representation	segmental (based on constituency)
interpresonal	enacting social relationship	clause as exchange	prosodic
textual	creating relevance to context	clause as message	culminative
logical	constructing logical relations	complexes (taxis & logico-semantic type)	iterative

Table 3.2 Metafunctions and their reflexes in the grammar

Across the rank scale, with respect to structure and metafunctions, Halliday formulates the general principle of *exhaustiveness* (Generalization 3.2.3) saying that clause constituents have at least one and may have multiple functions in different strands of meaning; however this does not mean that it must have a function in each of them. For example interpersonal Adjuncts such as "perhaps" or textual Adjuncts such as "however" play no role in the clause as representation.

Generalization 3.2.3 (Exhaustiveness principle). Everything in the wording has some function at every rank but not everything has a function in every dimension of structure (Halliday 2002; Halliday & Matthiessen 2013).

This principle implicitly relates to the property of language meaning that there is nothing meaningless and thus every piece of language must be explained and accounted for in the lexicogrammar. Also this principle implies that each metafunction has its own structure or that text is analysed through a multi-structural approach.

At the very top of the rank scale, clauses form complex structures. Halliday employs systematically the concepts of *taxis* and *logico-semantic relations* to account for inter-clausal relations.

Definition 3.2.14 (Taxis). *Taxis* represents the degree of interdependency between units systematically arranged in a linear sequence where *parataxis* means equal and *hypotaxis* means unequal status of units forming a *nexus* or a *unit complex* together.

The concept of taxis which is very useful at describing unit relations not only at the group and clause ranks but all the way down to smallest linguistic unit such as

[...and] are to be identified by the elements of their internal structure (Fawcett 2000: 195).

For English Fawcett proposes four main kinds of semantic entities: situations, things, qualities (of both situations and things) and quantities. Each of these semantic units corresponds to five major classes of syntactic units: *clause*, *nominal group*, *prepositional group*, *quality group* and *quantity group*. In addition he recognises two more minor classes be. the *genitive cluster* and the *proper name cluster* (Fawcett 2000: 193–194).

Fawcett's classification is based on the idea that the syntactic and semantic units are mutually determined and supported by grammatical patterns. However those patterns lie beyond the syntactic variations of the grammar and so blend into lexical semantics.

In Sydney theory the class is determined by the function it plays in the unit above. By contrast, in Cardiff theory. the class of unit is determined based on its internal structure i.e. by its *elements of structure* (and not by the function it plays in the parent unit).

3.3.2 Element of structure

The terms *element* and *structure* have roughly the same meaning as defined in Sydney theory of grammar (defined in Section 3.2) but with two additional stipulations presented below.

Definition 3.3.2 (Element of Structure). Elements of structure are immediate components of classes of units and are defined in terms of their *function* in expressing meaning and not in terms of their absolute or relative position in the unit. (Fawcett 2000: 213–214).

The definition above leads as a consequences to two important properties of elements formulated as follows.

Generalization 3.3.1 (Element functional uniqueness). Every element in a given class of unit serves a function in that unit different from the function of the sibling elements (Fawcett 2000: 214).

Even if for example. different types of *modifiers* in English nominal group seem to have very slight differences in functions. they are still there.

Generalization 3.3.2 (Element descriptive uniqueness). Every element in every class of unit will be different from every element in every other class of unit (Fawcett 2000: 214).

Thus the terms of modifier and head shall not be used for more than one class of unit. In English grammar the head and modifier are used for nominal group only. And in other groups the elements of structure may seem similar to modifier and head, they still receive different names such as *apex* and *temperer* in the quality group.

The elements (of structure) are functional slots which define the internal structure of a unit but still they are *located* in *places*. One more category that intervenes between element and unit is the concept of *place* which become essential for the generative versions of grammar.

There are two ways to approach place definition. The first is to treat places as positions of elements relative to each other (usually previous). This leads to the need for an *anchor* or a *pivotal element* which may not always be present/realised.

The second is to treat places as a linear sequence of locations at which elements may be located, identified by numbers "place 1", "place 2" etc. This place assignment approach is absolute within the unit structure and makes elements independent of each other. This approach has been used in COMMUNAL (Fawcett 1990) and PENMAN (Mann 1983b) projects.

3.3.3 Item

Definition 3.3.3 (Item). The item is a lexical manifestation of meaning outside syntax corresponding to both words (in the traditional sense), morphemes and either intonation or punctuation (depending whether the text is spoken or written). (Fawcett 2000: 226–232).

Items correspond to the leaves of syntactic trees and constitute the raw *phonetic* or *graphic* manifestation of language. The collection of items of a language is generally referred to as *lexis*.

Since items and units are of different natures, the relationship between an element and a (lexical) item must be different from that to a unit. We say that items *expound* elements and not that they *fill* elements as units do.

Definition 3.3.4 (Exponence (restricted)). Exponence is the relation by which an element of structure is realised by a (lexical) item (Fawcett 2000: 254).

Definition 3.3.8 (Reiteration). Reiteration is the relation between successive occurrences of the same item expounding the same element of structure (Fawcett 2000: 271).

Reiteration (Example 19) often is used to create the effect of emphasis. Like coordination, reiteration is a relation between entities that fill the same element of the unit structure which is problematic in my opinion and I further discuss it in Section 3.4.6.

Filling also makes possible the embedding relation which Fawcett treats as a general principle in contrast to more specific Definition 3.2.5 from Sydney model.

Definition 3.3.9 (Embedding (generic)). Embedding is the relation that occurs when a unit fills (directly or indirectly) an element of the same class of units; that is when a unit of the same class occurs (immediately) above it in the tree structure (Fawcett 2000: 264).

(20) (To become an opera singer) takes years of training.

(21) The girl (whom he is talking to) is an opera singer.

In Example 20 we can see an occurrence of direct embedding where an infinite clause acts as the subject of another clause. In Example 21 the embedding is indirect as the relative clause is part of the nominal group which functions as the subject in the parent clause. In both cases we say that a lower clause is embedded (directly or indirectly) in higher or parent clause. I will further discuss this in the context of rank-scale concept in Section 3.4.1.

A situation converse to reiteration and coordination where a element is filled by more than one unit is known as *conflation*, where a unit can take more than one function within another.

Definition 3.3.10 (Conflation). Conflation is the relationship between two elements that are filled by the same unit having the meaning of "immediately after and fused with" and function as one element (Fawcett 2000: 249–250).

Conflation is useful in expressing multi-faceted nature of language when for example syntactic and semantic elements/functions are realised by the same unit. For example the Subject "the girl whom he is talking to" is also a *Carrier* while the Complement "an opera singer" is also an *Attributc*. Also conflation relations frequently occur between syntactic elements as well such as for example the *Main Verb* and *Operator* or *Operator* and *Auxiliary Verb*.

Cardiff Grammar in case of both coordination and embedding relations deals without *hypotaxis* and *parataxis* relations described in Sydney Grammar.

Note also that filling and componence are two complementary relations that occur in the syntactic tree down to the level when the analysis moves out of abstract syntactic categories to more concrete category of items via the relationship of exponence.

3.4 Critical discussion on both theories: consequences and decisions for parsing

The two sections above cover the definitions and fundamental concepts from each of the two systemic functional theories of grammar. The work in this thesis uses a mix of concepts from both theories and this section discusses in detail what is being adopted and why attempting a rather pragmatic reconciliation for the purposes of achieving a parsing system than a theoretical debate. Next I draw parallels and highlight correspondences between the Sydney and Cardiff theories of grammar and where alter and present my position on the matter.

3.4.1 Relaxing the rank scale

The *rank scale* proposed by Halliday (2002) became over time a highly controversial concept in linguistics. The discussion whether it is suitable for grammatical description or not still continues. The historic development of this debate is documented in some detail (Fawcett 2000: 309–338).

In this section I present a few cases that highlighting when the rank scale as defined by Sydney is too rigid. As a consequence for the purpose of thesis I drop the *rank scale constraints* as enunciated in Generalization 3.2.2. Also the *rankshift* operation, exceptionally employed to accommodate special cases, is overridden by a broad definition of *embedding* operation (Definition 3.3.9) treated as naturally occurring phenomena in language at all ranks. I do not entirely dismiss the concept of rank scale as proposed in Cardiff school as I still find it useful in classification of units.

(22) some very small wooden ones

Consider the nominal group 22. Here the modifying element, the Epithet "very small", is not a single word but a group (Halliday & Matthiessen 2013: 390–396). As the rank scale constraints mentioned above state that the group elements need to be filled by words. To account for this phenomena, Halliday introduces a *substructure* of

3.4.2 Approach to structure formation

The *unit* and *structure* are two out of the four fundamental categories in the systemic theories of grammar. The Sydney and Cardiff theories vary in their perspectives on *unit* and *structure* influencing how units are defined and identified.

For Halliday, the *structure* (Definition 3.2.6) characterises each unit as a carrier of a pattern of a particular order of *elements*. The order is not necessarily a linear realisation sequence but a theoretical relation of relative or absolute placement. This perspective has been demonstrated to be useful in generation where unit placement emerges out of the realisation process.

The Cardiff School takes a bottom up approach and defines class in terms of its internal structure describing a relative or absolute order of elements. This sort of syntagmatic account is precisely what is deemed useful in parsing and is the one adopted in this thesis. In this work, as motivated in the Introduction, generation of the constituency structure is derived from the Stanford dependency parse trees. As it consists of words and relations between them the intuitive approach to form groups, clauses and complexes is by working them out bottom up. The method is to let the unit class emerge from recognition of constituent word classes and dependency relations between or sequence of already formed lower units. The exact mechanism how it is done I explain in Chapter 8. What is important to note here is the bottom up approach which is in line with Cardiff way of defining unit classes in contrast to top down approach of Sydney school.

3.4.3 Relation typology in the system networks

As system is expanded in delicacy to forms a systemic network of choices, choice of a feature in one system becomes the *entry condition* for choices in more delicate systems below. Halliday states that the relation on the systemic cline of delicacy is essentially of *sub-categorisation* (see Definition 3.2.11). In this subsection argue for occurrence of multiple kinds of inter-systemic relations. I also call them *activation relations* because in the traversal process from less to more delicate systems, when choices are made in the former then choice making is enabled or *activated* in the latter.

Next I present a distinction between two activation relations: the *sub-categorisation* and *choice enabling* that are of interest in the present thesis but this is by no means exhaustive distinction and more work is needed here.

Lets take as example the polarity system represented in figure 3.8. It contains two choices either positive or negative. An increase in delicacy can be seen as a taxonomic

"is a" relationship between features of higher systems and lower systems like in the case of POLARITY TYPE and NEGATIVE TYPE in figure 3.8 and in fact for the rest of the network. As a side note, the delicacy in a system network is akin to sub-classification relation, which was originally the intended one and the predominant one. In practice, however, a few kinds of abstraction relations can be encountered (e.g abstraction as information reduction, as approximation, as idealisation etc.) extensively treated by Saitta & Zucker (2013). This discussion however beyond the scope of the current work.

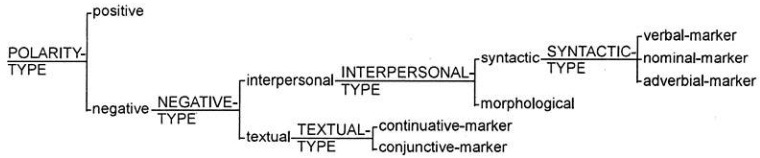

Fig. 3.8 System network of POLARITY

But the activation relation among systems in the cline of delicacy is not always taxonomic. Another relation is "enables selection of" without any sub-categorisation implied. As an example see the FINITENESS system in Figure 3.9 where in case that the finite option is selected then what this choice enables is not subtypes of finite but merely other systems that become available i.e. DEIXIS and INDICATIVE TYPE. The latter is there because selection of finite implies also selection of indicative feature in a sibling of FINITNES system, MOOD-TYPE comprised of options indicative and imperative.

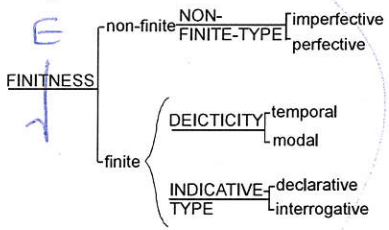

Fig. 3.9 A fraction of the finiteness system where increase of delicacy is not a "is a" relation

The distinction in the systemic relations is incorporated into the technical data structure definitions and traversal algorithms proposed in the Chapter 7.

Third, Penn POS set allows multiple tags per word, meaning that the annotators may be unsure of which one to choose in certain cases. There are 36 main POS and 12 other tags in the Penn tag set. A detailed description of the schema, the design principles and annotation guidelines are described in (Santorini 1990). Figure 3.10 depicts a classification summarising the Penn tag set.

3.4.5 Syntactic and semantic heads

In SFG the heads may be motivated by semantic or syntactic criteria (simply called here semantic or syntactic heads). In most cases they coincide but there are exceptions when they differ or even diverge. This topic is especially important in the discussions of the **nominal group** structure (continued in Section 4.1.3) on which Halliday & Matthiessen (2013) offers a thorough examination and Fawcett (2000) provides a more generic perspective.

In this discussion I show few examples when the syntactic and semantic heads diverge and argue my position on the group formation on two points. First, the class of the Head (in Sydney school) or pivotal element (in Cardiff school) is not always raised to establish the group class but the whole underlying structure determines the group class. Second, that syntactically motivated heads are easy to establish because they are based solely on the formal grounds whereas the semantic heads require an evaluation at the level of entire group, once one is established, employing additional lexical semantic resources. This can be a two step process but in the current implementation only the group structure on syntactic grounds is provided.

As mentioned before in Section 3.2.5, Sydney grammar fulfils the exhaustiveness principle (Generalization 3.2.3) through multiple parallel structures while Cardiff grammar through a single syntactic structure resembling a mixture of the former.

Let's briefly return to the Example 22 analysed with Sydney grammar in Table 3.3 and 3.4 that reflect the nominal group logical and experiential structures (Halliday & Matthiessen (2013: 391). When the Head (called here the *syntactic head* of the nominal group) coincides with the Thing (called here the *semantic head*) we say that they are conflated (Definition 3.3.10) and examples as this one may lead to the assumption that the Head, which is motivated by syntactic criteria, is also always the Thing which is motivated by the semantic criteria, but this is not so.

The logical structure is a Head-Modifier structure and "represents the generalised logical-semantic relations that are encoded in the natural language" (Halliday & Matthiessen 2013: 388). The experiential structure of the nominal group as a whole has the function of specifying the class of things, through the Thing element, and some

category of membership in this class, through the rest of the elements. In the nominal group there is always a Head but the Thing may be missing and so the Head element is conflated with either Epithet, Numerative, Classifier or Deictic instead.

(24) (Have) a cup of tea.

(25) The old shall pass first.

(26) I'll give you three.

Consider Example 24 analysed with Sydney and Cardiff grammars in Table 3.7. In the Sydney Grammar the semantic and the syntactic heads differ. In the experiential analysis the semantic head is "tea" which functions as Thing, while in the logical analysis the syntactic head is "cup" which functions as Head. Cardiff Grammar does not offer multi-structural analysis and there is no Head/Thing distinction. The functional elements are already established based on semantic criteria and this is further discussed in Section 4.1.3.

		a	*cup*	*of*	tea
Sidney Grammar	experiential	Numerative			Thing
	interpersonal	Pre-Modifier	Head	Post-Modifier	
Cardiff Grammar		Quantifying Determiner		Selector	Head

Table 3.7 Analysis of Example 24 with Sydney and Cardiff grammars: diverging semantic and syntactic heads.

In the nominal group "The old" which is Subject in Example 25, the Head is adjective "old" and not a noun as would normally be expected. The noun modified by the adjective "old", also the *pivotal element* of the group defined in Section 3.3.2, is left covert and it should consequently be recoverable anaphorically or cataphorically from the context. We can insert a generic noun "one" to form a canonical noun group "the old one". In such cases when the pivotal noun is missing, the logical Head is conflated with other element, in this case the Epithet. The group class is not raised from the word class to quality group but is identified by internal structure of the whole group and in this case the presence of determiner signals a nominal class. Similarly, in Example 26, "three" in Sydney grammar is a nominal group where the Thing is missing and the Head has shifted left towards the Numeral. With examples as these ones, Fawcett argues that none of the constituting elements of the unit is mandatorily realised, even the so called pivotal element which is the group defining element. In Chapter 6 is provided an in depth description of the recovering mechanisms for covert nominal elements at the level of the clause.

Ike	*washed*	*his*	*shirt*	*and*	*his*	*jeans*
Subject	Main Verb	Complement				
		Deictic Determiner	Head	&	Deictic Determiner	Head

Table 3.10 Coordination analysis in Cardiff Grammar

one units. And this is a problem because if we do not account unit elements each in a unique place within the unit structure then we loose the capacity to order them. Therefore in this thesis I adopt the Sydney definition of structure (Definition 3.2.6) that constraints each element into a single place that is filled by another unit. Therefore the conjunction must be a nexus acting as a single unit filling a single element.

I argue for adoption of such unit type in order to ensure that maximum one unit can fill the place of an element. In the theory of grammar, only units are accounted for structure while the elements can only be filled by an unit (see Figure 3.2). Allowing multiple units to fill an element requires accounting at least for the *order* if not also for the relation between the filler units. The structure as it is described in theories of grammar by Halliday (Halliday 2002) and Fawcett (Fawcett 2000) is defined by the unit and not the element. There is no direct reference in the theory to the unit ordering. Instead, the order relation is accounted in the structure through the concept of place. A unit has a specific possible structure in terms of places of elements which hold absolute position in the unit structure or relative one to each other. Therefore if an element is filled by two units simultaneously it constitutes a violation of the above principle as the order of those units is not accounted for but it matters which is easy to show in the following examples.

(27) (Both my wife and her friend) arrived late.

(28) * (And her friend both my wife) arrived late.

(29) I want the front wall (either in blue or in green).

(30) * I want the front wall (or in green either in blue).

If the order would not have mattered then we could say that the conjunctions from the example 27 can be reformulated into 28 and the one from 29 into 30. But such reformulations are grammatically incorrect. Obviously the places do matter and they need to be accounted in the unit structure as one element per place with no more than a single unit filling it.

I turn now to the role and position of lexical items signalling the conjunction which I consider having no place in the structure of the conjuncted units but outside of them,

that way forming together a higher order unit, the *complex unit*. This is contrary to what is being described in Cardiff and Sydney grammars for different reasons.

Fawcett present the Linker elements (&) which are filled by conjunctions as parts of virtually any unit class placed in the first position of the unit. For example in the "or in green" the presence of "or" signals the presence at least of one more unit of the same nature and does not contribute to the meaning of the prepositional group but to the meaning outside the group requiring presence of a sibling. Even more, the lack of a sibling most of the time would constitute an ungrammatical formulation. The only potential objection here is for the perfectly acceptable cases of clauses/sentences starting with a conjunction such as "and" or "but". In those cases the conjunction plays a textual function and still invites the presence of a sibling clause/sentence preceding the current one to be resolved in a clause complex or discourse level.

Halliday omits to discuss in IFG (Halliday & Matthiessen 2013) the place of Linkers. He implicitly proposes the same as Fawcett through his examples of paratactic relations at various rank levels (Halliday & Matthiessen 2013: 422, 534, 564, 566) that the lexical items signalling conjunction are included in the units they precede in the logical structure but not the experiential one. The main insufficiency here is that the logical structure does not provide any meaningful elements or unit class but some sort of proto-elements that resemble rather places than any functions. In this sense I consider treatment of conjunctions insufficiently accounted in IFG.

So conjunctions and pre-conjunctions shall not be placed inside the conjuncted units because they do not contribute to their meaning. They shall be enclosed as Linkers into unit complexes. But if we adopt the unit complexing then we need to define a unit structure. Hence I propose the following generic structure for the *coordination unit*.

Pre-Linker	Initiating Conjunct	... Conjunct ...	Linker	Conjunct
1		+ 2 ... + n−1	+ n	

Table 3.11 Generic structure of the coordination unit

In Table 3.11 the first row presents a series of Conjuncts where the first one is initiating or the head and the rest are continuation Conjuncts of the former. In the first place there may be a Pre-Linker element such as "both" or "either" for example, but it is optional and in the place before the last one is located the Linker element that determines the type of coordination. On the second row I provide, for orientation purposes, the Sydney logical structure of a paratactic expansion applied to the coordination unit complex. Note that the Pre-Linker and the Linker elements are merged with the conjuncted units.

This section laid out how and why I treat the coordination phenomena in parsing. I adopt the unit complexing mechanism with taxis relations described in Sydney grammar in order to account for a new unit class, the *coordination unit*. I do that to ensure that each element of a unit is filled by no more than one other unit contrary to what Cardiff grammar proposes (see Definition 3.3.7). But taxis relations in Sydney grammars are represented via logical structures which is not rich enough to account for internal structure of the coordination unit. Therefore I also propose here a unit structure in terms of ordered functional elements just as for the rest of unit classes.

3.5 Concluding remarks

This chapter has introduced the fundamentals of systemic functional linguistics and presented a consideration of Sydney and Cardiff theories of grammar to the task of parsing.

First, in Section 3.2, I introduced the Sydney theory of grammar that originated SFL. Then, in Section 3.3, I introduce Cardiff theory of grammar. It builds on top of Sydney school but differs in several important ways from it. Finally, in Section 3.4, I conduct a critical discussion on the important aspects of both grammars such as unit, class, function, element, rank scale, unit heads, structure and other. This discussion settles my position on some elements of the theory of grammar necessary for implementation of Parsimonious Vole parser.