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

March 2018

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

Declaration

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

Eugeniu Costetchi
March 2018

Acknowledgements

And I would like to acknowledge ...

Abstract

This is where you write your abstract ...

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 On AI, Computational Linguistics, universe and everything

In 1950 Allan 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 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) which 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 generate human(-like) cognitive capacities (Stevan 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).

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 tendency 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 language in terms of for morphology, phonology, syntax, semantics, discourse or grammar in general. In computer science are developed increasingly more efficient algorithms and machine learning techniques. Computational linguistics provides ingenious methods of encoding linguistically motivated tasks in terms of formal data structures and computation goal. 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) mentioned in 1950 in the context of automatic translations (Hutchins 1999) of Russian text into English stared developing before the field of Artificial Intelligence. 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 somehow aside that up to now still need massive inputs on human mind from cognitive, psycho-linguistic and other related sciences). 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*, etc.

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 the text in terms of such structures by involving computer science and artificial intelligence techniques is a task pursued for several decades and still continues to be a major challenge today. This is especially so when the target is a *broad language coverage* (?) and even more when the desired analysis goes beyond simple syntactic structures towards richer functional and/or semantic descriptions useful in the latter stages of *Natural Language Understanding* (NLU).

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.

These 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 ones 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 and that is just to name a few.

These software programs originally were designed by and for the domain experts but over time the fruits of the technological advancement became available to millions of ordinary people. In a world as ours, where technology is so ubiquitous and pervasive into almost all aspects of our life, natural language processing and understanding becomes of great value and importance regardless whether it materializes as a spell-checker, (not so) clever machine translation, voice controlled car or phone and so on.

1.2 The goal of the thesis

This thesis aims at a reliable modular method for parsing unrestricted English text into a feature rich constituency structure using Systemic Functional Grammars (SFGs).

Before describing the parsing method, the following aspects need to be clarified first: the theoretical framework and its descriptive power, the depth and meaningfulness of the analysis, the computational complexity of the process and the level of accuracy and how it is measured. I address each of these aspects in the following chapters and before advancing any further I would like to illustrate through an example of what parsing is.

1.3 An example of Systemic Functional analysis

Traditional linguistic teaches us how to do the syntactic analysis of a sentence. So let's consider Example 1 in order to perform one. We need to focus on clustering words together into constituents guided by the intuitive rule "which word stands closest to another one" within the sentence. The word "He" is closest to "gave" and it seems that they go together especially that if we try to group "He" and "the" they do sense at all. Then, using sequential proximity criteria, "gave" must be related to "the" but they do not stick together at all it is rather "the" and "cake" that are a unity. So "the" has a stronger relation to "cake" than "gave". But actually "cake" seems connected to "gave" and not the other way around, so the direction of connection seems to matter just as much as its strength. So we see that the sequential proximity sometimes indicated

relatedness between words but often it does not. The strength and direction as well are crucial and tt is some sort of meaning making that governs the grouping process. This grouping into syntactic constituents can be expressed using bracketed notation as in Example 2.

(1) He gave the cake away.

(2) ((He) (gave) ((the) (cake)) (away) (.))

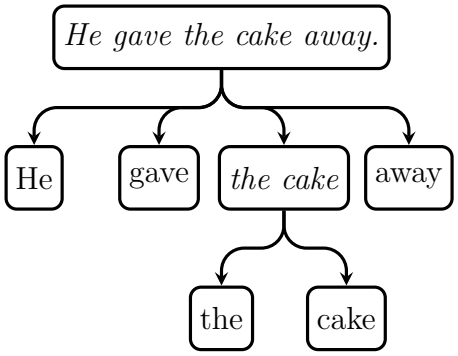

Fig. 1.1 Representation of the Example 2 as constituency tree

Figure 1.1 depicts the constituency division of the clause which is identical to the bracket notation in Example 2. The nodes represent grammatical constituents and the edges stand for the part-whole composition. This constitutes the simplest structure that could be expected from a parser without any specification of constituent class, function or any other grammatical functions. We know as well from works on parsing with formal generative grammars that such composition can always be expressed as a tree (or parse tree).

Each constituent can be decorated with its grammatical features . For example the word “he”, we know, is third person pronoun, masculine gender and singular number; or the word “gave” is a verb and the predicate of the sentence, and so on. The structure in Figure 1.2 depicts a syntactic constituency tree in which every node is richly decorated with syntactic and semantic features. The blue part of each node represents grammatical class and function fundamental for establishing a valid constituency structure; the red part represents the semantic functions (called Transitivity in SFL); and the green part other grammatical features. In practice, the feature set is much richer than what nodes in Figure 1.2 carry, the current limitation being simply the space constraints. Generating automatically graphs like the one in Figure 1.2 is the purpose of the current work.

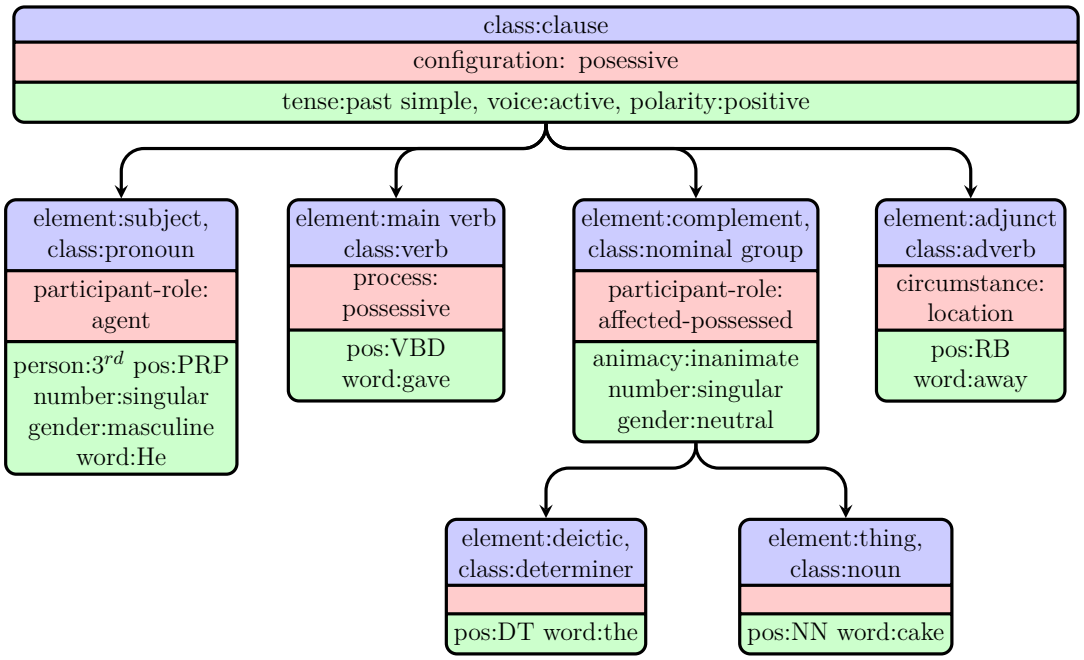

Fig. 1.2 Representation of Example 2 as feature rich constituency graph

1.4 The linguistic framework

Any description or analysis involving language implies some theory of how language works. In this thesis I chose the Systemic Functional Linguistic (SFL) framework because of its versatility in producing descriptions along *multiple semiotic dimensions* (Halliday 2003) (i.e. paradigmatic, syntagmatic, meta-functional, stratification and instantiation dimensions) and at different *delicacy levels* of the *lexico-grammatical cline* (Halliday 2002; Hasan 2014).

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 Butler (2003a,b) treats extensively.

Elaborating the foundations laid by his British teacher J. R. Firth, Hjelmslev (1953) from Copehagen School of linguistics and a group of European linguists from Prague School, Halliday develops the beginnings of SFL in his seminal paper Halliday (1961). Inspired by *oragnon model* formulated by Bühler (1934), Halliday refers to the language

functions as metafunctions or lines of meaning offering 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 *ideational metafunction*. Finally the two weave together into a coherent discourse flow whose mechanisms are characterised through the *textual metafunction*.

There are two models of SFG: the *Sydney Grammar* developed by Halliday & Matthiessen (2013), the founding fathers of *Systemic Functional Linguistics* (SFL), and *Cardiff Grammar* developed by Fawcett (2008), an extension and in a way simplification of Sydney Grammar. Each of the two grammars has advantages and shortcomings which I present in analyse and select based on theoretical soundness and suitability to the goals of the current project.

Cardiff and Sydney grammars had 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) attempted to reuse the grammars for the purpose of syntactic parsing within the borders of NL generation coverage. I come back to these works in more detail in Section 1.7.

1.5 The SFG complexity problem

Bateman (2008) thoroughly explains the reasons for such tremendous complexity after the attempts of Kasper (1988), Kay (1985), O'Donoghue (1991), O'Donnell (1993) and Day (2007), just a few to mention, none of which managed to parse broad coverage English with full SFG and without aid of some sort. Each had to accept limitations either in grammar or language size and eventually using simpler syntactic trees as a starting point of the parsing process. So what is it about?

Automatic analysis of text can be seen as a problem of searching though the space of possible solutions for an appropriate or even optimal solution. Here we speak of the Systemic Functional Grammar as a linguistic resource that shapes the search space and the way it access to that space is available. The systemic lexicogrammar is organised paradigmatically and was proven to be a good structure for natural language generation task but it turns out to be unusable for the reverse problem, that of natural language analysis. The principal issues is that of handling the *search space* leading back to Halliday's question "How big is a grammar?" (Halliday 1966).

The size of the search space defined by a grammar depends on the number of system networks and on the kind of connectivity and cross-classification it provides. For example given 50 system networks, the size of the search space lies somewhere between 51 and 2^{50}. This nevertheless is not such a big deal in the case of generation, as Halliday (1996) says that the "number of choice points [...] is actually rather small" as only few of the actual possibilities produced by a system network need to be explored when generating a clause. "Possible feature selections become relevant only when they are revealed to be relevant by prior paradigmatic choices and it is only those alternatives that need to be considered"(Bateman 2008).

Analysis is not symmetric with generation and the paradigmatic context of choice that is available during generation is no longer accessible in parsing. It is not known any longer which features of a systemic network are relevant and which are not. 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 virtually to explore entire search space in order to discover which features apply to the text (Bateman 2008).

In the analysis task first difficulty that needs to be addressed is 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 input and the grammatical description of *instantial syntagmatic organizations* involving *configurations of grammatical functions*. In a second stage these grammatical functions can serve as paradigmatic context for further traversing the system network and extend to the full set of systemic features. Moreover they will play a crucial role in restricting and organising the search space for relevant and applicable network parts during analysis task.

Addressing the gap of easily accessible syntagmatic account within SFG framework, can be done by first, providing information about what grammatical function operate at each rank, second which grammatical functions can be filled by which classes of units and third by providing relative and absolute account of ordering within each unit structure. This sort of information can guide building of a constituency backbone structure. As a second stage, as mentioned before, the unit classes and 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.

Alternatively the problem of structure construction can be outsources as parsing with other grammars especially that there has been a lot of progress recently. Then the problem changes into creating a transformation mechanism to obtain the SF

constituency structure rather than build it from scratch. Starting the SFG parsing process from a simple syntactic tree reduces the computational complexity by providing a set of reliable selections within the system network.

The second stage of constituent enrichment by network traversal can be further aided by checking an arbitrary set of patterns for preselecting even more features recoverable via lexico-syntactic patterns. The pattern recognition plays an essential role in current parsing method for fleshing out the constituent backbone with systemic selections.

1.6 On theoretical compatibility and reuse

In the past decades there have been made significant progresses 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 influences directly what and how is being 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.

Pareses for one theoretical framework may face common or different problems across theories, but as well as the solutions. The successes and achievements in any school of though should be regarded as valuable cross theoretical results to the degree the links and correspondences can be established. Therefore reusing components that have been proved to work and yield "good enough results" is a strong pragmatic motivation for deriving herein described parsing method.

Present work lays first some cross theoretical correspondences and then some inter-grammatical links. It demonstrates how selected grammatical frameworks namely *Systemic Functional Grammar*, *Dependency Grammar* and *Governance & Binding Theory* relate to each other and to which degree they are compatible to undergo a conversion process and to show that simple patterns carrying grammatical information can be used to enrich syntactically and semantically the parse structures. And here is a brief motivation for selecting these frameworks.

In the last years *Dependency Grammar* (Tensiere 2015) became quite popular in natural language processing world favoured over phrase structure grammars. The grammatical lightness and the 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.

I employ Stanford Dependencies (Marneffe & Manning 2008b,a; Marneffe et al. 2014) as a starting point which provides information about functional dependencies between words and grants direct access to the predicate-argument relations which are not readily available form the phrase structure parses and can be used off the shelf for real world applications. Regardless of being a simple grammatical framework which accounts for the syntactic relations between words, Stanford dependency grammar is structurally and functionally compatible to SFG.The account it provides for the word dependencies can be viewed also in functional terms and I expand this idea in Chapter **??**. It is a much more suitable foundation for building the SFG syntactic structure than phrase-structure trees, as well as for making more delicate grammatical distinctions (a process highlighted in Section 1.9 and explained in detail in Chapter **??**).

The current parsing process requires accounting for *null elements* which are not covered by the dependency grammar. As a solution I turned to a part of Chomsky's Transformational Grammar(Chomsky 1957), Government and Binding Theory (GBT) (Chomsky 1981; Haegeman 1991), to identify and create the Null Elements to support the semantic parsing. I introduce GBT and provide inter-grammatical links between towards dependency grammar in Chapter **??**.

Current work is the first one to parsing with a dependency backbone, all previous ones using context-free grammars. There are other good candidates to serve as backbone (CCG, TAG, FCG, HPSG etc.) but a broad investigation of parsers and the compatibility of their linguistic theoretical frameworks to SFL is outside the scope of this thesis.

1.7 Previous works on parsing with Systemic Functional Grammars

There have been various attempts to parsing with SFGs. This section covers the most significant attempts to parse with a Systemic Functional Grammar. The firs 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, Honnibal (2004); Honnibal & 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 representation covering sentence mood and Thematic constituency (a kind of analysis in SFL which is not considered in current work). Transitivity has not been covered there because of its inherently semantic nature but it is in the current work.

1.7.1 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 as 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).

1.7.2 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,

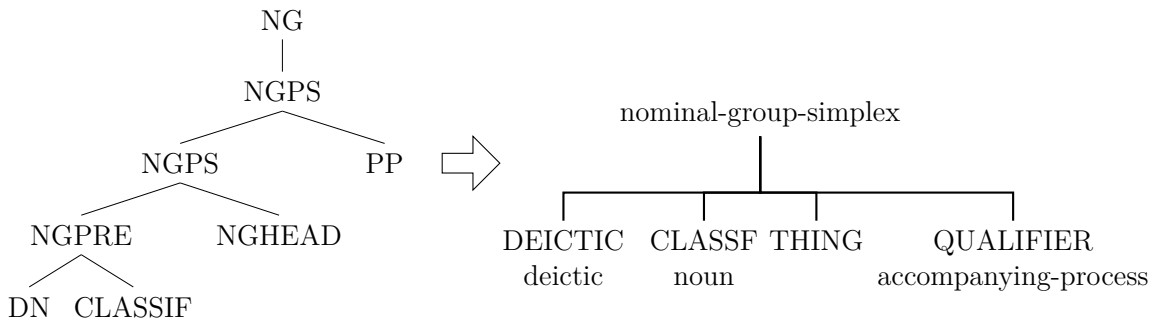

Fig. 1.3 Transformation from phrase structure into systemic constituency structure. Rule example from O'Donnell & Bateman (2005).

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 Chomsky (1957). He created a set of rules for mapping the phrase structure (PS) into a parallel systemic tree like the one depicted in Figure 1.3. 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.

1.7.3 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 UAM 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.

1.7.4 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 also created a lexical resource indicating for each word which elements can expanded it.

The parsing procedure is a simple lookup of words in the lexical resource selecting all possible elements it can expound and then selecting possible strips starting with the elements expounded by the word. Advancing from left to right for each sentence word more strips compatible with the previously selected ones are selected within the collocation network constrains. The parser finds all possible combinations of strips composing parse trees representing possible output parses.

The corpus from which the vertical strips were extracted is 100,000 sentences large and was generated with Fawcett's natural language generation system and was tested on the same corpus leaving unclear how would the parser behave on a real corpus. In 98% of cases the parser returns a set of trees (between 0 and 56) that included the correct one with an average of 6.6 trees per parse.

Actually, using a larger corpus could potentially lead to a combinatorial explosion in the step that looks for vertical strips. It would decrease the accuracy of the parse because of the higher number of possible trees per parse.

1.7.5 Honnibal

Honnibal (2004; 2007) describes how Penn Treebank can be converted into a SFG Treebank. Before assigning to parse tree nodes synthetic features such as mood, tense, voice and negation he first transforms the parse trees into a form that facilitates the feature extraction.

The scope of SFG corpus was limited to a few Mood and Textual systems leaving aside Transitivity because of its inherently lexico-semantic nature. He briefly describes how he structurally deals with verb groups, complexes and ellipses as functional structures are much flatter than those exhibited in the original Treebank. Then he describes how are identified metafunctional features of unit class, mood function, clause status, mood type, polarity, tense, voice and textual functions.

The drawback of his approach is that the Python script performing the transformation does not derive any grammar but rather implements directly these transformations as functions falling into the same class of problems like Winograd's SHRDLU. By doing so the program is 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.

1.8 Current approach to parsing

The main problem in using SFGs for parsing is that they are very large and complex. Some parsing approaches use a syntactic backbone which is then flashed out with SFG description. Other ones use a reduced set or a single layer of SFG representation the third ones use an annotated corpus as the source of a probabilistic grammar. Regardless of the approach each limits the SFG in a 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.*

Current approach is aligned with works of Honnibal, Kasper and O'Donnell with respect to using a backbone structure and enriching it with syntactic and semantic features. It relies on parse structures produces in other grammars and then translated to systemic functional constituency structure. The contributions on theoretical compatibility and inter-grammatical transformations are briefed in the Section 1.6 coming up next.

Current method employs rules for graph traversal in order to build a parallel backbone constituency tree and rules for graph matching to enrich it with systemic features. This aims at keeping the language coverage broad at the expense of higher systemic delicacy. I cover basic Mood systems which is way less than what is available in Nigel grammar and leave it for the future work to extend the grammatical delicacy.

Nonetheless I attempt to cover some the lexico-semantic features as well. Parsing Transitivity system, a task similar to Semantic Role Labelling, requires large lexicogrammatical resource describing verb meanings in terms of their process type and participant roles. The semantically-oriented decomposition of clauses offered by SFL is still sufficiently closely tied to observable grammatical distinctions as to offer a powerful bridge to automatic analysis. Such descriptions are analogous to frame representations (Fillmore 1985) as found in FrameNet (Baker et al. 1998) or VerbNet (Kipper et al. 2008) applied in Semantic Role Labelling Task (Carreras & Màrquez 2005).

O'Donnell approaches this task by providing possible process types directly to the verb by employing self constructed lexicon where each word has syntactic and semantic features. Only recently a resource comparable to FrameNet and VerbNet has been produced in the SFL framework called Process Type Database (PTDB) (Neale 2002). PTDB which provides for each verb a process configuration (similar to a semantic frames) in terms of process type and participant roles. Current work represents the first attempt of using the PTDB to produce semantic (or Transitivity) analysis which combined with pattern patching method has an advantage over O'Donnell's parser. It enables to simultaneously assign, if matched, the systemic features to all clause constituents or not at all.

Another major advantage, as compared to Honnibal's approach is that the grammar and the program are carefully disconnected so that the code is maintainable and scalable with the respect to size of the grammar. This makes it possible to choose rather pragmatically which graph patterns to consider for parsing depending on the scope of task at hand.

1.9 The parsing process overview

Te parser follows a pipeline architecture depicted in Figure 1.4 where staring from an input text gradually a rich systemic functional constituency structure is built. This sections provides an overview to the building process.

In the figure there are three types of boxes. The rounded rectangles represent the parsing steps. They linearly flow from one to the next one via green trapezoids boxes, on the left side, which represent intermediary data. On the right side are positioned double edged orange trapezoids representing some fixed resources used as additional input for some steps. For example *constituency graph creation* step takes a normalised dependency graph for input and produces a constituency graph as output.

The entire process starts with some input English text and ends with production of Rich Constituency Graph. The Input text is first parsed with a dependency parser. For the current work Stanford dependency parser was chosen for its dependency relations, parse accuracy and the continuous efforts put into its development (motivated in Section 1.6).

The dependency graphs often contain errors some of which are predictable, easy to identify and correct. Also some linguistic phenomena are treated in a slightly different manner than proposed in the current thesis. Therefore the dependency graph

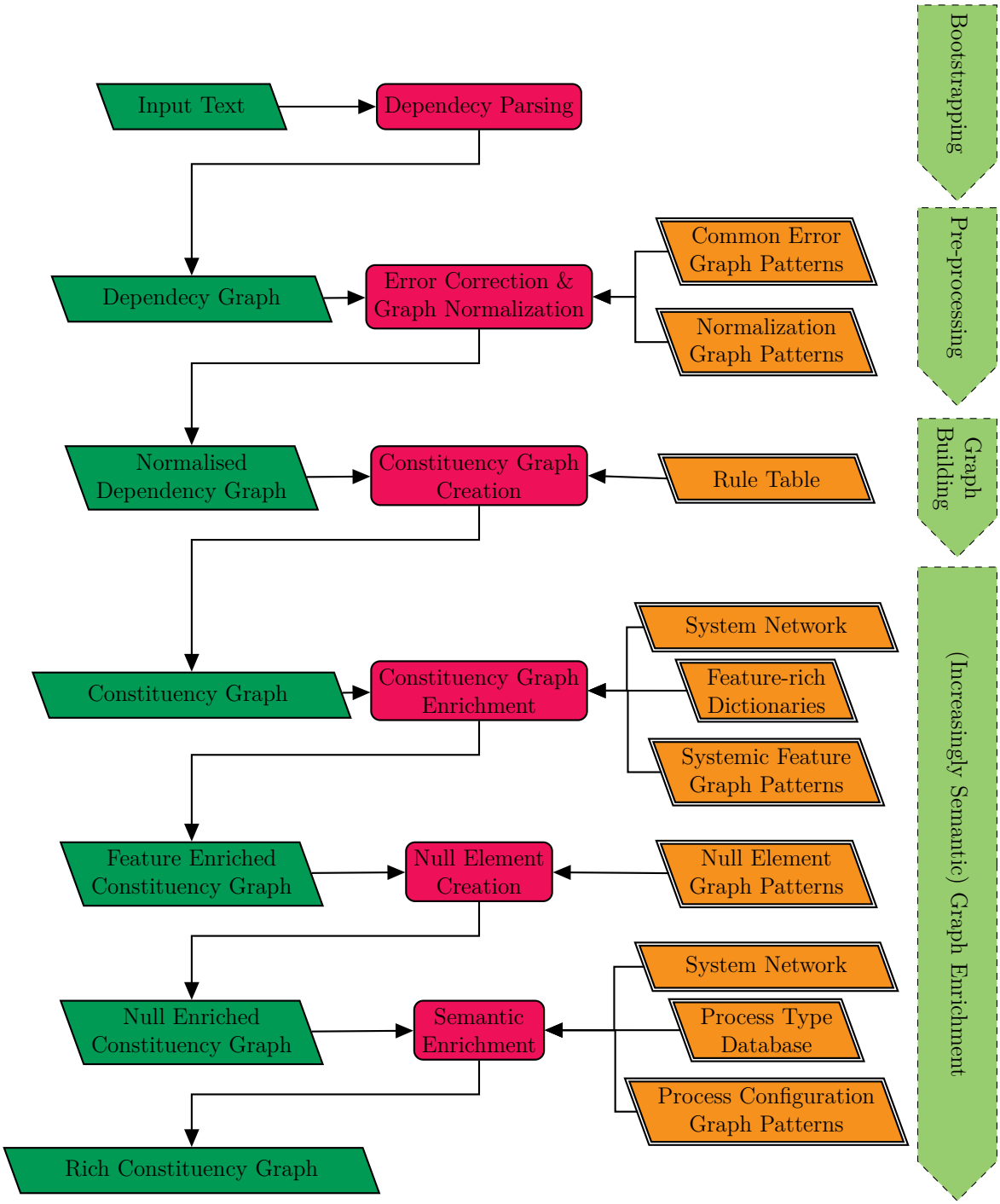

Fig. 1.4 The parsing process pipeline

produced by Stanford parser is *corrected and normalised* using pattern matching against a collections of known errors and one of normalization rules.

Once normalised the dependency graph is ready to guide the *building process* of the systemic functional constituency graph. It represents, in a way, transformation of the dependency graph, and serves a syntactic backbone on which the subsequent enrichment phases are performed.

Next follow two phases where the syntactic backbone is *enriched* with features some of which bear a *syntactic* whereas other a *semantic* nature. In between these enrichment phases there is a construction process which produces structural changes to the backbone adding some *empty constituents* that play a role in semantic enrichment. The enrichment phases use additional resources such as *system networks*, *feature rich lexicons*, *graph patterns* and *semantic databases*. The *null element creation* process also needs a collection of graph patterns for identifying where and what kind of null elements occur. The final result of the process is a *Rich Constituency Graph* of the original text comprising a plenitude of systemic feature selections associated with constituting units of structure.

1.10 Research questions and contributions

This thesis addresses the following questions:

- What is a computationally feasible method to parse with systemic functional grammars with a syntactic backbone?
- To what degree are Stanford Dependencies suitable as a syntactic backbone for Systemic Functional Grammar parsing?
- How can Process Type Database be used as a resource for SFG Transitivity parsing?
- How can Government and Binding Theory be used for detecting external predicate arguments in the context of SFG Transitivity parsing with PTDB?

Also it brings the following contributions:

- The analysis of theoretical and practical compatibility between the syntactic structures of Stanford Dependency and Systemic Functional Grammars along with an implemented method to transform from one structure to another.
- A fast engine for graph pattern matching which can also update and insert new nodes.

- A flexible and expressive method to represent systemic features as graph patterns together with two strategies for choice propagation in the systemic networks.
- A set of pattern graphs covering Mood, Transitivity and other smaller system networks.
- A method to transform PTDB into a set of Transitivity graph patterns.
- Derived principles and generalizations from the Government and Binding Theory (GBT) and represented them as graph patterns used to identify the covert elements of the clause that are explicitly mentioned outside the clause borders. These generalizations serve for semantic parsing where is very helpful to identify the external arguments of verbs.
- Development of a test corpus and evaluation of the parser.

1.11 Thesis organisation

The remaining of this thesis is organised as follows.

Chapter **??** explains in parallel Cardiff and Sydney theories of grammar followed by a discussion of structure units of each grammar. When juxtaposed, weaknesses and strengths of each school emerge in contrast to each other on aspects like *structure, dependency relations, unit classes, systemic networks, rank scale and unit complexing*. I use elements of both grammars therefore I explain my stance on each of the above issues and argument the choices. In similar manner I discuss the syntactic and semantic units of each grammar even if the systemic functional linguistics does not make such distinction it is useful in establishing links to mainstream methods for language processing. Basically, this chapter presents the mixed grammar and its theoretical underpinning through the comparative discussion between two schools in SFL.

Chapters **??** and **??** introduce *Dependency Grammar* and *Governance and Binding Theory (GBT)*. Both frameworks are used as departing points to build the SFG structure. The cross-theoretical correspondences together with specific inter-grammatical links are developed in the same chapters.

Chapter **??** formally defines the structures used in this thesis and the operations on them. Important to mention structures are *feature rich graphs*, *ordered conjunction sets*, *feature structures* and *system networks*; whereas important operations are the varieties of *graph matching* and *pattern graph matching*.

Chapters **??** and **??** explain how the parsing process evolves starting from the dependency graph towards a constituency graph and then towards increasingly semantic constituency graph through its feature features. This suite of algorithms and the pipeline has a Python implementation called Parsimonious Vole[1]. A limited empirical evaluation of the parser is provided in Chapter **??**. It describes the evaluation methodology, the gold standard used and highlights strengths and weaknesses of the current implementation.

The last part of the thesis sets future directions explore and concludes on the current work (Chapter **??**).

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

Chapter 2

Introduction

2.1 Practical relevance

Developed over thousands and thousands of years, human language has become a nuanced form of communication that carries a wealth of meaning that 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 formal, strict and rigorous manner via graphical user interfaces, command line terminals and programming languages. Advancements in *Natural Language Processing* (NLP) which is a branch of *Artificial Intelligence* (AI) are a game changer in this domain. NLP starts to unlocks the information treasure 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 ours, where technology is ubiquitous and pervasive in almost all aspects of our life, NLP becomes of great value and importance regardless whether it materializes as a spell-checker, intuitive recommender system, spam filter, (not so) clever machine translator or a voice controlled device.

Every time you ask Siri or Alexa 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. Now they are learning our language.

2.1.1 NLP for businesses

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. 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 whether it is pizza or a smartphone. Popular sources for such inquiry are the blogs, forums, reviews, social media, reports, news, company websites, etc. All of them 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 sentence "The pizza was amazing but the waiter was awful!" and connect it to the following sentence "I adore when it is topped with my favourite artichoke" about pizza and not the waiter and discover a topping preference.

NLP is heavily used in customer service in order to figure out what 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 fo increased responsiveness by providing immediate answers to simple needs and also decrease the load of 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 socio-cultural context where it occurs.

2.2 The linguistic framework

Any description or analysis involving language implies some theory of how language works. In this thesis I chose 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* (Halliday 2003) (i.e. paradigmatic, syntagmatic, meta-functional, stratification and instantiation dimensions) and at different *delicacy levels* of the *lexico-grammatical cline* (Halliday 2002; Hasan 2014).

Elaborating the foundations laid by his British teacher J. R. Firth, Hjelmslev (1953) from Copehagen School of linguistics and a group of European linguists from Prague School, Halliday develops the beginnings of SFL in his seminal paper Halliday (1961).

This 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: p.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: p.55)

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 Butler (2003a,b) treats extensively.

Inspired by *oragnon model* formulated by Bühler (1934), Halliday refers to the language functions as metafunctions or lines of meaning offering 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 *ideational metafunction*. Finally the two weave together into a coherent discourse flow whose mechanisms are characterised through the *textual metafunction*.

There are two models of SFG: the *Sydney Grammar* developed by Halliday & Matthiessen (2013), the founding fathers of *Systemic Functional Linguistics* (SFL), and *Cardiff Grammar* developed by Fawcett (2008), an extension and in a way simplification of Sydney Grammar. Each of the two grammars has advantages and shortcomings which I present in analyse and select based on theoretical soundness and suitability to the goals of the current project.

Cardiff and Sydney grammars had 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) attempted to reuse the grammars for the purpose of syntactic parsing within the borders of NL generation coverage. I come back to these works in more detail in Section 1.7.

As we part away from the surface form of text and aim for rich semantics or aim at analyses higher than the clause level i.e. discourse, the functional is increasingly useful and revealing of meanings in text. Such analyses have been done manually by linguists, semioticians and educators in an informal manner as there have not been any tools to automate such processes. Besides Linguistics, there is a plethora of linguistic analysis using SFL framework in other fields of research. SFL has been used extensively as a descriptive framework in Critical Discourse Analysis and in Education studies. Automatising the language analysis with SFL framework will unlock the potential of these fields. Next I provide a glimpse of what opportunities they offer.

2.3 The Opportunity

Second, a large amount of description of this kind has traditionally been done, and done a lot, in SFL. Here again you need to find a good collection of example 'applications' in SFL (not computational) where the deeper analysis has been found useful and give references: education, text/discourse analysis (critical)(Tenorio 2011) [Encarnacion Hidalgo Tenorio 2011, Critical Discourse Analysis, An overview], whatever plus references. Survey of studies in systemic functional language description(Mwinlaaru & Xuan 2016) functional information is found useful for text analysis, but this has only been done informally well, look at the education work, the typology work, the CDA work: these are the main areas where SFL appears and has papers.

2.4 The Barrier

But, 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, etc. etc. (add as many points as occur to you).

2.5 Previous Attempts

There have been attempts to make this work (which you will come back to and describe in Chapter X in detail), however, but these have not worked. As you you say you will describe in detail in Chapter X, there is however a strong diagnostic as to just why these attempts have not been successful: i.e., the lack of structural detail that SFG descriptions typically provide. This is argued in general in Bateman (2008) and Teich (1999) [and any other references you can find].

2.6 Some interesting examples

You then give EXAMPLES of some difficult cases, where you illustrate what an SFG analysis would like look and you point out the lack of structural detail, informally so that it can be understood directly without further technical detail. Preferably bringing out some cases where it is evident that there is no information, e.g., about raising and control (Teich) and anything else which would make interpretation difficult.

2.7 Proposed Solution

Your proposed solution to this problem, and the goal of the thesis, is therefore to add some more structural information to a complete augmented SFG account by drawing on frameworks which have demonstrated coverage of structural detail and which also have supported computational instantiation. This will be shown and evaluated in the thesis.

2.8 Thesis Goal

So, the thesis goal and outline will be to (and you list them explicitly like this too):
- characterise SFL in its two major variants
- characterise the previous attempts to parse with SFL and their problems
- set out two further linguistic frameworks which (a) have strong accounts of structural relationships, (b) have shown themselves supportive of computational instantiation, and (c) can be shown to exhibit suggestive theoretical/descriptive links with SFG: in particular, DG and GB. Chapter X does this for DG Chapter Y does this for GB.

2.9 Provisional Thesis Structure

1: introduction, reasons and goals
2: SFG
3: State of the Art in approaches to Parsing with SFG and complexity
4: DepGrammar
5: GBT
6: Single architecture
: : Empirical Evaluation
: Conclusions (what has been achieved and outlook)

2.10 References

[Butler2003] Structure and Function [Hjelmslev1953] - Prolegomena-to-a-Theory-of-Language-by-Luis-Hjmeslev [Elke Teich 1999] - Systemic Functional Grammar & Natural Language Generation - Ch5

% feedback Chapter 1 + 2
Dear Eugene,

thanks for file; attached are the detailed comments and corrections and suggestions for Chapters 1 + 2. I suggest some reorganisation of the introduction and how the materials in the current chapter 2 are described, you will n sense of this. But, in short, I think an organisation along the lines:

Chapter 1: introduction, reasons and goals
Chapter 2: SFG
Chapter 3: State of the Art in approaches to Parsing with SFG and complexity
Chapter 4: DepGrammar
Chapter 5: GBT (perhaps, haven't read these yet)

would get the thesis off to a better start. Also you need to think about whether all the detail of the SFG variants is important enough for your task. You will need to provide some more detail of the organisation of the actual grammars as well in any case, as otherwise you can't talk about Mood and Transitivity and the like. This is all clarified in the comments. Alternatively you say very little about these and introduce them when you get to the later chapters: that might make sense; I'll see when I get that far. If one went that road, it would mean not including comments about Mood and Transitivity in the current chapter 2 though, which might be awkward.

I'll proceed with the other chapters, but as you will see, you have a fair bit to get going with in any case.

I will not be able to work on the thesis after the end of March.

theses don't really work like that; so we'll see how far you get. You (and I) don't want a repeat of the Daniel situation.

Let me know if anything is unclear.

Best,
John.

%feedback Chapter 1 + 2

Dear Eugene,

Comments/corrections for chapters 3 + 4 attached.

Now I'm getting more of a view of the thesis, I'd say that
at present, systemicists will get confused because they'd
wonder why alien things like GB and dependency grammar
appear, and formal/computational linguists would get
confused because they wouldn't be clear why one would
want to take something like SFL. This can be managed
fairly straightforwardly I suspect by setting up the
argument in the Introduction in a clear way, so that
everyone knows just why these things are coming together.
I'd suggest the following kind of outline for the
introduction to make that work, let me know if you
have any problems or questions about this as it
would seem (to me) to be a good way of making all
the bits fits together in a reasonably convincing
fashion. This would also help avoid a reoccurring problem
in your text at the moment, where you frequently want
to talk about things that you have not yet introduced - this
just makes the text confused and impossible to follow (many
examples of this are picked out explicitly in the comments).

So...

Structure the Intro to the thesis more like this:

First, point to the increasing and increasingly recognized
need for deeper, richer semantic/pragmatic analyses across
a broad range of applications: corpora, human-machine
interaction, intelligent interfaces and assistance robotics,
whatever you can find with references supporting the
claim.

Second, a large amount of description of this kind has
traditionally been done, and done a lot, in SFL. Here again
you need to find a good collection of example 'applications'
in SFL (not computational) where the deeper analysis
has been found useful and give references: education,

text/discourse analysis (critical), whatever plus references.

But, 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, etc. etc. (add as many points as occur
to you).

There have been attempts to make this work (which you
will come back to and describe in Chapter X in detail), however,
but these have not worked. As you you say you will describe in detail
in Chapter X, there is however a strong diagnostic as to
just why these attempts have not been successful: i.e., the
lack of structural detail that SFG descriptions typically
provide. This is argued in general in Bateman (2008)
and Teich (1999) [and any other references you can find].

You then give EXAMPLES of some difficult cases, where
you illustrate what an SFG analysis would like look and
you point out the lack of structural detail, informally
so that it can be understood directly without further
technical detail. Preferably bringing out some
cases where it is evident that there is no information,
e.g., about raising and control (Teich) and anything
else which would make interpretation difficult.

Your proposed solution to this problem, and the goal of the thesis,
is therefore to add some more structural information to a
complete augmented SFG account by drawing on frameworks which
have demonstrated coverage of structural detail and which
also have supported computational instantiation. This will
be shown and evaluated in the thesis.

So, the thesis goal and outline will be to (and you list them
explicitly like this too):
- characterise SFL in its two major variants

- characterise the previous attempts to parse with SFL and their problems
- set out two further linguistic frameworks which (a) have
strong accounts of structural relationships, (b) have shown
themselves supportive of computational instantiation, and (c) can
be shown to exhibit suggestive theoretical/descriptive
links with SFG: in particular, DG and GB.

Chapter X does this for DG
Chapter Y does this for GB.

- Following this, Chapter Y+1 brings these altogether in a single
architecture (can be short: material from the current introduction
about the system architecture goes here, or can be longer, if
you take the material about merging GB and DG and then with SFG
here too: this might be best).

- rest of chapters go into details.

- Chapter $-1 Evaluation
- Chapter $ What has been achieved and outlook.

I think this kind of explicit form in the Introduction of
the thesis would tell a convincing story that
would make the most of what you currently have and simply wrap
this in a structure that readers can follow and accept. Then
you strengthen the existing bits of text to explicitly draw
attention to these goals as you go so that the reader
remembers where they are and what you are trying to do (and why).
I think this is a fair bit of work still, but relatively
straightforward as it is more about imposing structure and
getting things in the right order. Definitely a thesis in
there struggling to get out! :-)

Best,
John.

%feedback Chapter 5

Hi Eugen,

here is chapter 5 commented. In this one, there are many more
comments about content that will need fixing up, so not just
style of presentation. Many of the problems though come, I suspect,
because you have not yet introduced the algorithm and pipeline
and its datastructures sufficiently that the reader has any
idea what your formalisations here are attempting to do. I think
many of them can just disappear, since you certainly won't
be able to use them anywhere. To define a data structure, you
don't need a full first-order theory, that is overkill. You
do not get any points for formalisation; you'd only get points
for appropriate, necessary and well motivated formalisation,
and many of the definitions in this chapter do not meet
this requirement. You only need as much formalism as necessary
to get the job done. And the job is the task that you need
to have described as the pipeline of the system: probably best
immediately after the discussion of GB. There are many
interesting decisions made in this chapter, but they are
just lost in the mass of probably hardly relevant detail.
So introducing the pipeline and its data structures first,
would give you a better way of picking out just that which
is a crucial contribution of your thesis, i.e., the stuff
that makes parsing work. Providing definitions of
morphisms between graphs does *not* do that; and it is
hardly your job and has been done more or less completely
before in appropriate formal texts in any case.

In short, you need to provide the new architecture and pipeline
chapter and rewrite this one accordingly.

Let me know when that has happened, as that will be the next
major version that it would be sensible for me to comment on

I think. The actual details of the parsing algorithm that
occurs in subsequent chapters will I hope be more straightforward,
once the groundwork is out of the way.

Best,
John.

Am 08.03.18 um 22:00 schrieb Eugen Costezki:
> I wanted to say that this chapter 5 represented a special kind of struggle as I wa

yes, I noticed! :-) Fortunately, you do not need to do this...
so simplifications are ahead!

Best,
John

References

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.

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

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 & 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>.

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

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

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

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

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>.

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. 1961. Categories of the theory of grammar. *Word* 17(3). 241–292.

Halliday, Michael A. K. 1966. Some notes on 'deep' grammar. *Journal of Linguistics* 2(1). 57–67.

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

Halliday, Michael A.K. 2003. 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 (4th Edition)*. Routledge 4th edn.

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.

Kasper, Robert. 1988. An Experimental Parser for Systemic Grammars. In *Proceedings of the 12th 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.

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, 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, 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>.

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>.

Mwinlaaru, Isaac N. & Winfred Wenhui Xuan. 2016. A survey of studies in systemic functional language description and typology. *Functional Linguistics* 3(1). 8. doi:10.1186/s40554-016-0030-4. <https://doi.org/10.1186/s40554-016-0030-4>.

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. 1994. Sentence Analysis and Generation: a systemic perspective. Tech. rep. Department of Linguistics, University of Sydney.

O'Donnell, Michael. 2005. The UAM Systemic Parser. *Proceedings of the 1st Computational Systemic Functional Grammar Conference* <http://www.wagsoft.com/Papers/ODonnellUamParser.pdf>.

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

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

Stevan Harnad. 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>.

Tenorio, Encarnacion Hidalgo. 2011. Critical Discourse Analysis, an overview. *Nordic Journal of English Studies* 10(1). 183–210.

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

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

Weerasinghe, Ruvan. 1994. *Probabilistic Parsing in Systemic Functional Grammar*: University of Wales College of Cardiff dissertation.

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