Chapter 11

Conclusions

This thesis aims at a reliable modular method for parsing English text with Systemic Functional Grammars. To achieve this goal I designed a pipeline, which, starting from a dependency parse of a sentence, generates a SFL-like constituency structure serving as a syntactic backbone, and then enriches that structure with various systemic features.

In this process, the first milestone is the creation of the constituency structure. Chapter 3 describes the essential theoretical foundations of two ~~SFL schools~~: the Sydney and Cardiff schools. It also provides a critical analysis in order to reconcile the diverging points on rank scale, unit classes, the constituency structure, treatment of coordination, grammatical unit structure and clause boundaries; and states what is the adopted position on each point.

In order to create the constituency structure from the dependency structure there needs to be a mechanism in place to provide a mapping between the two both at the theoretical and grammatical levels. The theoretical account of the dependency grammar and how it is related to SFL ~~is~~ described in Chapter 5. The practical aspects of the process such as the algorithms and the enactment of inter-grammatical mapping rules ~~are~~ described in Chapter 8.

Before describing the parsing pipeline, to make clear what the basic ingredients of this implementation are and how the algorithms are coded, Chapter 7 introduces the basic data structures and operations. These structures are defined from a computer science point of view emulating the needed SFL concepts. The main structures are attribute-value dictionaries, ordered lists with logical operators and a few graph types. In addition, the basic operations relevant for the parsing pipeline such as conditional traversal and querying of nodes and edges, graph matching, pattern-based node selection, insertion and update are also described.

Once the constituency structure is created, the second milestone is to enrich it with systemic features. Because systemic features can be associated with or derived from the (dependency and constituency) graph fragments, in this work, graph pattern matching is a cornerstone operation used for adding features to constituent units and inserting new or missing constituents. These operations are described in detail in the second part of Chapter 7, and Chapter 9 outlines how these operations are used for the enrichment of the constituency backbone with systemic features.

The quality of graph patterns impacts directly the outcome of the parser. The more precise graph patterns are the smaller the false omission and miss rate in the parser output is, and thus the number of errors in general decreases while the accuracy of feature enrichment increases. This was shown in the evaluation result discussion in Chapter 10 in general, and in Section 10.4 in particular.

It is often the case for the TRANSITIVITY network~~,~~ that the graph patterns, in their canonical form, list the mandatory participants of a semantic configuration. In practice, however, instances of such configurations may leave unrealised up to two mandatory participants. And so, if applied in their canonical form the patterns will not identify ~~with~~ such instances. In this thesis, mock constituents (null elements) are created in the places where the presumed constituents should be, allowing in this way matches with canonical graph patterns.

To identify and create the covert participants I turned to Government and Binding theory which accounts for this. In doing so, this thesis brings two more contributions: (a) the theoretical mapping from GBT into dependency structures covered in Chapter 6 and (b) a concrete implementation of how to perform creation of the null elements described in Chapter 6.3.

In Chapter 10 I describe the empirical evaluation. The aim of the assessment, in general, is to determine how accurately the text analysis is generated; and, in particular, how well the parser performs at unit boundary detection (i.e text segmentation), unit class assignment, element assignment and feature selections.

The data show that the parser assigns classes to the constituent units with an accuracy of 74% and clause main Mood elements are detected with an accuracy of 71.2%, while the Transitivity elements are detected with an accuracy of 81%.

When it comes to evaluating the accuracy of systemic assignments, the measured accuracy varies drastically across delicacy levels and between the sibling features within the same system. This has been addressed for the MOOD and TRANSITIVITY systems in Section 10.4. The features from the MOOD system network are assigned, on average, ~~with~~ an accuracy of 59%. The accuracy of TRANSITIVITY system network was measured separately for the PROCESS-TYPE system and the PARTICIPANT-ROLE-TYPE system. The accuracy of the former, on average, is 36% and of the latter, on average, is 46%.

Next I present how the main research questions were addressed and what ~~are~~ the main contributions of this thesis.

11.1 Research questions and main findings

In Chapter 1 six research questions were asked. This section shows how those research questions are addressed in the current thesis and what ~~are~~ the theoretical and practical outcomes.

One of the main theoretical contributions of this thesis is the investigation to what degree ~~can~~ cross-theoretical bridges be established between SFL and other theories of grammar, formulated as Research question 1. The approach to answering this broad question was to further ask more specific questions. In particular I have focused on studying correspondences to Dependency Grammar and Government and Binding Theory, formulated in Research questions 2 and 3.

First, Research question 2 on the degree to which the syntactic structure of GD and SFG are compatible to undergo a transformation from one into another was fully addressed in this thesis showing that they are entirely compatible and that the goal is feasible. The support for this claim is provided in Section 5.6, which addresses in detail the cross-theoretical links between the Dependency and the Systemic Functional theory of grammar. This cross-theoretical bridge constitutes a fundamental principle for further deriving transformation rules from a dependency representation into a systemic functional one. Such rules are enacted, in the parsing pipeline, to create the systemic constituency structure as laid out in Section 8.3.

Second, Research question 3 about the usability of GBT for detecting places of null elements in the context of SFL constituency structure was explored in depth with positive results. This is addressed in Section 6.3, where rules, principles and generalisations from GB theory are translated into DG and SFG frameworks. These translations serve directly the goal of identifying places where (and by which relations) the null elements should be injected. The translated rules are realised in the form of graph patterns explained in detail in Section 9.3.

Addressing Research questions 2 and 3 and establishing cross-theoretical bridges to DG and GBT constitute answers for Research question 1. The conducted theoretical investigation accompanied by the practical implementation and the evaluation results show that persuading reuse, in SFL contexts, of positive results from other areas of computational linguistics constitutes not only a desirable but also a feasible goal. This, however, does not guarantee in practice maximal accuracy and the extent to which the goal is achieved depends to a large extent on the implementation.

As another contribution, this thesis offers an investigation on how suitable ~~are~~ graph patterns in detecting systemic features and enriching the constituency structure. In the current approach to parsing, graph patterns play a similar role as the realisation rules play in the process of natural language generation. They serve as language for systematising grammatical realisations, and constitute a convenient form of representing grammatical features employing both structural and lexico-structural patterns. Graph patterns and the matching methods developed in this thesis can potentially be applied for expressing many other grammatical features than the ones presented here.

Research question 5 on the extent ~~at~~ which graph patterns can be used to represent systemic features based solely on structural aspects was addressed in Section 9.1 (focused MOOD system network). It is shown that some of the ~~top~~ system can be dealt with only by structural patterns, however, as ~~the~~ delicacy increases, the ~~employment~~ of lexis into the graph patterns is inevitable. Moreover, for TRANSITIVITY features, escaping ~~the~~ lexis is not possible at all, and constitutes the main reason for employing a lexical-semantic resource such as PTDB. This only confirms the already known strong link between grammar and lexis, which SFL considers a unitary lexico-grammar (defined in Section 3.1).

In the end, Research question 6 on whether the PTDB is suitable as a lexical-semantic resource for Transitivity parsing is addressed in Chapters 9 and 10. I explain how ~~automatically~~ graph patterns can be generated from PTDB before describing how it was turned into a machine readable resource. Nonetheless, the evaluation of the currently implemented method to assign Transitivity features does not provide encouraging results reaching only 42% accuracy (10% less that for Mood features). These performance indicators are explained in Section 10.4.2, and further discussed in Section 10.5. This level of accuracy, among others, is due to currently implemented enrichment mechanism, which applies all the matching graph patterns to the constituency structure instead of applying the one with highest probability. This means that higher accuracy can be achieved provided that the implemented approach is improved by reducing the number of patterns per feature. The degree to which the accuracy will improve if enrichment mechanism is enhanced remains a question for future work.

Finally, the evaluation results presented in Chapter10 are significant in at least two major respects. First, the parser accuracy of generating SFL constituency structures is comparable to or slightly lower than the accuracy achieved by previous attempts, e.g. 76% by Souter (1996) and 81% by O'Donoghue (1991a). ~~Moreover,~~ ~~as~~ the parser generates feature rich output which sets apart Parsimonious Vole from other parsers. The features in the generated output could be already considered useful in some practical situations. Second, this study shows which areas are in need of improvement and provides hints on what can be improved. Also, this evaluation can be considered initial baseline for incremental development~~s~~ in future work.

11.2 Limitations and future work

~~It should be borne in mind that~~ this work has a number of limitations. This section introduces the most important ones along with improvements that are desirable or ~~at least~~ worth considering.

Parsimonious Vole parser grammar

The grammar proposed in Chapter 4 is a combination of elements taken from both the Cardiff and the Sydney grammars. Even if the chosen grammar parts have been carefully motivated, explained and argued for~~,~~ as a whole, how well they fit together still requires ~~a~~ scrutiny by grammarians and a validation with larger corp~~us~~.

The graph patterns ~~have been~~ manually created, which is error prone and requires careful validation. This work can be supported and facilitated by a workbench editor, debugger and validator for graph patterns and systemic features combined. The workbench could build on top and extend UAM Corpus tool functionalities. As no editor for grammatical graph patterns exists yet ~~and~~ developing one in the future is desirable.

Graph patterns from Nigel grammar

One important experience this thesis provides is the use of graph patterns for detecting systemic features, based on structural and lexical cues in the provided constituency structures. For the parser implementation, all of the MOOD patterns were created manually while the TRANSITIVITY patterns were created from a simple lexical-semantic database, the Process Type Database (Neale 2002).

At the same time, Nigel grammar (Matthiessen 1995), the largest ~~known~~ SFL grammar, was not employed in this work even if it is very relevant. In part this is due to the reasons explained in Chapter 2, i.e. the previous attempts to parse with full

SFL grammars directly had limited success. Also, not being familiar with Lisp, the programming language in which Nigel is represented, made it difficult to reuse it. In future work, however, investigating how graph patterns can be generated from the system network realisation rules available in Nigel will be very valuable and highly desirable work. Not only can it save time and reduce potential errors of the manual authoring of graph patterns, but it can provide a very rich set of graph patterns covering system networks outside the scope of this work.

Adoption of verbal group

The current grammar does not include the verbal group unit but treats the elements of what would be a verbal unit as elements of the clause. This decision is motivated in Section 4.1.1 and is in line with the proposal put forward by the Cardiff grammar. This resolves the problem of discontinuity in the syntactic units which was an issue for the current implementation.

(127) Are you feeling cold?

A simple example of a discontinuity is provided in Example 127. The verbal group here is formed of the Auxiliary "are" and the Main verb "feeling". In principle, in the syntactic analysis, the units of analysis should be continuous. This is known to not always be the case as illustrated by Example 127 where the subject "you" splits the verbal group in two.

Adopting a gap resilient constituency structure would permit inclusion into the generated analysis not only of verbal groups but also enable Thematic analysis, which often employs discontinuous units, and the adoption of other unit classes.

Transition to semantically motivated unit classes

Cardiff unit classes are semantically motivated ~~if~~ compared to the more syntactic ones in the Sydney grammar. This is stated in Fawcett (2000: 193–194) and was presented in Section 3.3 and further discussed in Section 4.1.

For example, nominal structure proposed in the Cardiff grammar (discussed in Section 4.1.3), uses elements that are more semantic in nature than the syntactic ones offered in the Sydney grammar. For example compare various types of determiners: representational, quantifying, typic, partitive etc. in the Cardiff grammar and the deictic determiner in the Sydney grammar.

In order to shift towards semantically motivated nominal unit structure two problems need to be addressed: (a) how to detect semantic heads and (b) how to craft (if none exists) a lexical-semantic resource to support detection of various determiners in the nominal group. Building lexical-semantic resources asked at point (b) represents a potential solution for point (a) as well. Employing some of the existing resources such as Nigel grammar, even if it is built in Sydney style, could and most likely ~~is~~ suitable starting point for addressing point (b). In addition, other non-SFL lexical resources such as WordNet (Miller 1995) or FrameNet (Baker et al. 1998) could be considered in this context. Yet resorting to these lexical resources would not be a straightforward solution and would require more adaptations so that they are useful in the SFL domain.

The same holds for Adverbial and Adjectival groups (Section 4.1.4), which in Cardiff grammar are split into Quality and Quantity groups. Existent lexical resources such ~~as such~~ as WordNet (Miller 1995) or FrameNet(Baker et al. 1998) combined with the delicate classification proposed by Tucker (1997) ~~can~~ yield positive results in parsing with Cardiff unit classes.

Just as in the case of verb groups discussed in the previous sections, moving towards semantically motivated unit clas[illegible] would greatly benefit applications requiring deeper natural language understanding.

More delicate TRANSITIVITY graph patterns

The PTDB (Neale 2002) is the first and only lexical-semantic resource for the Cardiff Transitivity metafunction. In ~~the~~ original form, this resource was ~~barely~~ machine readable, with its usability limited to dictionary-like search by linguists in the process of manual text analysis. It was rich in human understandable comments and remarks across all fields and not fully formal enough to be employed in computational tasks. In the scope of current work the PTDB has been cleaned and brought into a machine readable form.

In mainstream computational linguistics, there are several lexical-semantic resources used for Semantic Role Labelling (a task similar to Transitivity parsing), such as FrameNet (Baker et al. 1998) and VerbNet (Kipper et al. 2008). Mapping or combining PTDB with these resources into a new one would yield benefits for both: potentially inspiring the internal organisation for VerbNet and extending the coverage of PTDB.

Combining PTDB with VerbNet for example, would be my first choice in the task of improving Transitivity analysis for the following reasons. PTDB is well semantically systematised according to the Cardiff Transitivity system, however, it lacks any links to syntactic manifestations. VerbNet, on the other hand, contains an excellent mapping to the syntactic patterns in which each verb occurs, each with associated semantic representations of participant roles and some first order logic representation. Also, the lexical coverage of VerbNet is twice ~~wider than~~ that of PTDB.

Resorting to resources like FrameNet or WordNet could bring other benefits. For example, FrameNet has a set of annotated examples for every frame which, after transformation into the Transitivity system, could be used as a training corpus for machine learning algorithms.

Towards speech act analysis

As Robin Fawcett explains (Fawcett 2011), Halliday's approach to Mood analysis differs from that of Transitivity in the way that the former is not "pushed forward towards semantics" as the latter is. This claim, however, is controversial and not endorsed by the Sydney grammarians. The meaning proposed by Fawcett in the Cardiff MOOD system network is similar to and incorporates concepts from ~~the~~ Speech Act Theory (Austin 1975) or its later advancements (Searle 1969). Such theories, in mainstream linguistics, are placed under the umbrella of pragmatics. Operating with concepts such as speech acts (which Sydney grammarians reject) would take the interpersonal text analysis to a new level of meaning with potential benefits in applications where interactivity is a feature of primary concern.

Halliday proposes a simple system of speech functions (Halliday & Matthiessen 2013b: 136) which Fawcett develops into a quite delicate system network (Fawcett 2011). It is worth exploring ways to implement Fawcett's latest developments especially that the two are not conflicting but complementing each other. In future work can be explored how to use the Hallidayan MOOD system as a foundation to transit towards the Cardiff MOOD system. Such exploration can be facilitated by the fact that Sydney MOOD system network has already been implemented and described in the current work.

Adoption of group complexing

The group complexing structures are well described in the Sydney grammar (Halliday & Matthiessen 2013b: 567–592). Such structures are not considered in the current work ~~at large~~ except for the particular case of conjunction treatment, which is described in Section 3.4.6. Adopting a general framework of unit complexing is highly beneficial as it contributes to a more meaningful analysis. The immediate applications of group complexing, in the context of this thesis, can be seen in the case of verbal group complexes presented next.

The *one main verb per clause* principle of the Cardiff school that I adopted in this thesis (briefly discussed in Section 4.1.1) provides a basis for simple and reliable syntactic structures. Also, it represents a simple clause boundary detection rule. The alternative is adopting the concept of verbal group, simple and complex, as proposed by the Sydney school in Halliday & Matthiessen (2013b: 396–418, 567–592), a much richer and complex approach. The verb complex provides a richer semantically motivated description (Halliday & Matthiessen 2013b: 567–592), however, analysing text with such constructs is difficult and subject to ambiguities.

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

Table 11.1 Sydney sample analysis of a clause with a *verbal group complex*

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

Table 11.2 Cardiff sample analysis of a clause *embedded* into another

One way to approach this is in two steps (similarly to semantic head detection discussed in Section 3.4.5): first, generating the syntactic analysis and then ~~uplifting~~ it to a more meaningful analysis. Even though an approach in two steps such as the one suggested here is subject to criticism, in part, it can already be implemented by considering Cardiff influential process types (implemented as part of Transitivity parsing).

Consider the sample analyses in Tables 11.1 and 11.2. The two-clause analysis proposed by the Cardiff school can be quite intuitively transformed into a single experiential structure with the top clause expressing a set of aspectual features of the process in the lower (embedded) clause just like the Sydney analysis in Table 11.1.

The class of *influential* processes proposed in the Cardiff transitivity system was introduced to handle expressions of process aspects through other lexical verbs. I consider it as a class of pseudo-processes with a set of well defined and useful syntactic functions but with incomplete semantic descriptions. The analysis with influential process could be used as an intermediary step towards a more meaningful analysis, such as the one suggested by Sydney grammar. Alternatively, the analysis process could be redesigned to generate complex verbal units directly taking into account the available lexical-syntactic resources.

Generalisation 11.2.1 (Merging influential clauses)**.** When the top clause has an influential process and the lower (embedded) one has any of the other processes, then the two clauses ~~shall~~ be merged into one and the two verbs into a verb complex enriched with aspectual features.

This rule of thumb is described in Generalisation 11.2.1. Of course, this raises a set of problems that are worth investigating. First, the connections and mappings between the influential process system network described in the Cardiff grammar and the system of verbal group complex described in the Sydney grammar (Halliday & Matthiessen 2013b: 589) should be investigated. Second~~ly~~, one should investigate how this merger impacts the syntactic structure.

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

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

The benefits of such a merger lead to an increased comprehensiveness, not only of the Transitivity analysis, illustrated by the examples in Tables 11.1 and 11.2, but potentially apply to the modal assessment illustrated by Examples 128 and 129 and similar phenomena.

Taxis analysis

Currently, the Parsimonious Vole parser implements a simple taxis analysis technique based on graph pattern matching, similar to the one described in Sections 7.4 and 7.5. Description of this work, however, is not included in this thesis because it has not yet been tested.

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

In the taxis analysis process, each pair of adjacent clauses in the sentence is tested for compliance with TAXIS pattern in the database. The matches (there ~~might~~ be multiple ones for a single system feature) represent potential manifestation of the corresponding relation with no way to distinguish at the moment which pattern is, in fact, more likely to be correct. A similar problem was described for the TRANSITIVITY system and a potential solution was also described in terms of a discrimination mechanism in Section 10.4.2. More work, however, needs to be conducted in this area.

Dealing with covert elements and ellipsis

In the current approach to Transitivity parsing, accounting for the covert (or the so-called null) elements was taken as an instrumental goal to increase accuracy of parsing. Whether such elements should be accounted for in the grammar or whether they exist at all is still under debate in the linguistic literature, and, of course, arguments exist for and against the null elements.

One future development would be to change the way graph patterns are generated from PTDB. The resulting graph patterns would need to be shaped such that the null elements are no longer a requirement for Transitivity parsing. This would, among other things, eliminate the need to create null element units in the constituency structure and would make the cross-theoretical links to GBT obsolete in this task.

I need to make, however, a reference here to *ellipsis*, a well studied linguistic phenomenon. An elliptical construction is the omission from a clause of one or more words that are nevertheless understood in the context of the remaining elements. There is a variety of ellipsis types, among which the null elements mentioned above. Whether to fill the gaps in the syntactic structure and which ones is a question that should not be abandoned too soon as providing rich and explicit structures can have positive outcomes in practical contexts.

Bridges to other grammars and linguistic theories

In this thesis exploration of cross-theoretical bridges is limited to two other traditions: that of Dependency grammars (specifically Stanford Dependency Grammar) and that of Phrase-Structure Grammars (specifically Government and Binding Theory). There is a wider set of useful cross-theoretical correspondences to establish that can materialise as positive reuse outcomes.

Due to compatible approaches to language analysis, among the most interesting correspondences would be Lexical Functional Grammars (Bresnan et al. 2015), Head-Driven Phrase Structure (Pollard & Sag 1994), Combinatory Categorial Grammar

(Steedman 1993, 2000) and Tree Adjoining Grammars (Kroch & Joshi 1985) (whose correspondence to SFG has already been address in Yang et al. (1991)) to name just a few. Having traced these new correspondences it will become possible to create the constituency backbone in the SFL style in a similar fashion as it is currently done from the Dependency Grammar.

The current implementation also requires an immediate upgrade to the latest vesrion of the Stanford parser. Between 2006 and 2015 the Stanford parser (Marneffe et al. 2006) was employing the Stanford dependency model for English (and a few other languages). Afterwards, in 2016, Nivre et al. (2016) proposed the language independent Universal Dependency scheme which was integrated into the Stanford Parser and replaced the Stanford dependency model. Around 2015–2016 the Parsimonious Vole parser was developed based on the Stanford dependency model. No transition to universal dependency was considered at that time because it was not mature or stable enough. For this reason the current thesis employs the legacy Stanford grammar and so a transition to universal dependency model must be considered in future work, in order to keep up the pace with the latest developments in the Stanford dependency parser.

Efficient graph rewriting method

In the current work the SFL style constituency backbone is created from dependency graphs. This is treated in computer science as *graph/tree rewriting*. There is extensive literature addressing this task such as Barendregt et al. (1987), Courcelle (1990), Plasmeijer et al. (1993) and Grzegorz (1999).

As at the time of developing the Parsimonious Vole parser I was not aware of this work I implemented my own method of graph rewriting. And so in the prototype implementation no pre-existing algorithm has been used. Future work needs to integrate the state of the art methods in graph rewriting and potentially improve or replace the current graph rewriting algorithm. Such a decision would need to be based on the efficiency and ease of providing the transformation rules.

Execution order of graph patterns

For a given constituency structure the current enrichment mechanism fires all the available graph patterns and any of the matching ones enrich the constituency structure. This can be costly when the number of patterns increases dramatically. Such a risk is imminent if, for example, the graph patterns are generated from the Nigel grammar as mentioned above. That richness poses the danger that too many graph patterns will make the parsing if not uncomputable, then at least too slow to be practical.

This risk can be countered, to an extent, by putting in place a selection mechanism that would seek to minimise the number of fired graph patterns for a given constituent unit. Such mechanism needs to implement a search mechanism in the space of features covered by the graph patterns taking into account the systemic dependency between features and, therefore, between patterns. Moreover, a fitness function measuring information gain per graph and execution cost must be considered. Such a mechanism may already speed up the current implementation to an extent.

Dealing with multiple patterns per systemic feature

In the current implementation for each process type configuration in PTDB multiple patterns graphs were generated. This is one of the leading causes ~~to~~ decreased TRANSITIVITY parsing accuracy ~~and~~ was described in Section 10.4.2.

To prevent features from the same system from being assigned to constituent units simultaneously (even if clearly marked as a disjunctive set of possibilities) a discrimination mechanism ~~shall~~ be implemented. Such a mechanism collects all the possible pattern matches first, and then assigns only the most suitable one to the constituent unit. This mechanism can be based on calculated probabilities or frequency in a corpus. More investigations are needed ~~in~~ on this issues.

Analysis of errors from the current evaluation

The evaluation performed in the current work does not go into ~~too much~~ detail analysing the types of errors ~~this~~ parser commits. In order to improve the performance of the current implementation the known errors need to be investigated down to the level of transformation rules, graph pattern and systemic feature disjunctions. Therefore, it is essential to carry on further investigation of segmentation errors (e.g. distance distribution for each feature) and errors in the constituency structure (false positives in the parser generated analysis and true negatives in the corpus). Results of a deeper error analysis will ~~show~~ how to correct the transformation rules from the dependency into SF constituency structures. Similar benefits can be achieved by investigating the errors in the systemic feature assignments.

Investigation of probabilistic logics for SFG parsing

The problems of computational complexity in parsing with SFGs is explained in Chapter 1 and treated at ~~large~~ in Bateman (2008). At the heart of this problem lies the combinatorial explosion ~~sourced~~ by the complex network of disjunctive systems. One way to deal with large combinatorial spaces is by using search approximations. For ~~the~~ logical systems such an approximation is materialised in the form of probabilistic logics.

Martin Kay was the first to attempt formalisation of ~~systemics~~ that would become known as Functional Unification Grammar (FUG) (Kay 1985). This formalisation was adopted in other linguistic frameworks such as HPSG, Lexical Functional Grammars and Typed Feature Structures. For SFGs, however, using first order or even description logic reasoners has been shown to have ~~terrible~~ complexity problems (Bateman 2008). Employing probabilistic logics, therefore, may ~~constitute a key to overcome~~ that complexity issue.

Markov Logic (Domingos et al. 2010; Richardson & Domingos 2006) draws my attention in particular, which I consider a good candidate for parsing with SFGs. It is a probabilistic logic, which applies ideas of Markov networks to first order logic enabling inference under uncertainty. What is very interesting about this logic is that tools implementing it have learning capabilities not only of formulas weights but also of new logical clauses. Moreover, it has been shown to be computationally feasible on large knowledge bases. The extent of such clauses should, however, still be investigated.

Markov logics can be employed in addressing the graph pattern creation problem. Besides creating the graph patterns manually or from existing resources such a PTDB or advanced grammars such as Nigel, another possibility worth exploring is learning them from a corpus.

Since graph patterns can be expressed via first order functions and individuals, and assuming that a (richly) annotated corpus in SFL style is available, Markov Logic tools such as Alchemy[1], Tuffy[2] and others, can be employed in an experiment to inductively learn patterns structure (and features) from the corpus.

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

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

11.3 Practical applications

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

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

A parser of the type aimed at in this thesis would be useful to solve the latter set of tasks. The rich constituency parses could be an essential ingredient for further tasks such as anaphora resolution, clausal taxis analysis, rhetoric relation parsing, speech act detection, discourse model generation, knowledge extraction. All these tasks are needed for creating an intelligent interactive agent for various domains such as call centres, ticketing agencies, intelligent cars and houses, personal companions or assistants.

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

11.4 Final word

In this work I have advanced the work on automatic text analysis in SFL style. The current implementation did not succeed to employ a full SF grammar, and, just like previous attempts, had to accept limitations in the grammar size while maintaining broad language coverage. This task is particularly difficult because of the richness of such grammars. Nonetheless, ~~the~~ modern applications desperately need deep feature-rich text analysis functionalities.

My view is that building on top of successful results achieved with other grammars by mapping them to parts of SF grammar constitutes a viable solution to the creation of SFL style constituency structures. Furthermore, employing graph patterns to enrich the structure with systemic features is the key ingredient for performing a delicate feature-rich text analysis.

By further advancing the proposed methods and exploring new ways to cut through complexity, my hope is that one day automatically generating feature-rich text analysis will become the *de facto* approach employed in truly intelligent agents that can, to a large extent, do with language what people do.