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

January 2019

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

Declaration

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

Eugeniu Costetchi
January 2019

Acknowledgements

And I would like to acknowledge ...

Abstract

This is where you write your abstract ...

Table of contents

List of figures

List of tables

List of definitions

Chapter 1

The Empirical Evaluation

The present parser is evaluated by comparing the text analysis it outputs with manually analysed text. The evaluation seeks to measure the parser *precision* i.e. how many segments have been produced by the parser that are also found in manual analysis giving us; and the parser *recall* i.e. how many correct segments have been produced by the parser relative to the total number of produced segments.

In order to count correct and incorrect number of segments they need to be matched first. This task is not a trivial because of the annotation mistakes, some differences in grammar and methodology of treating certain phenomena described below. In this work slight divergences are tolerated provided that the segment label is the same and there is other segment that constitutes a better match. This is known, in computer science as the Stable Marriage problem defined in Section 1.4. The next two sections define segments and provide details on how and why they diverge. Section 1.3 describe how the segment labels differ and how they are mapped to extend the scope of the evaluation.

To establish the matches between manual and automatic segments I have implemented a variation of the Gale-Shapley algorithm (Gale & Shapley 1962) which is described in Section 1.4 below. Then, the section that follows, are finally described the empirical findings of the current evaluation.

In the current evaluation I use two sets of annotated texts. Table 1.1 summarises the corpora used for evaluation in this work.. The first dataset (OCD) was created by Ela Oren and I and is focused on syntactic constituency structure and clause Mood features. The texts represent blog articles of people diagnosed with Obsessive Compulsive Disorder (OCD) who self-report on the challenge of overcoming OCD. The second dataset (OE1) is the PhD work of Anke Shultz covering Cardiff Transitivity analysis. It comprises 31 files spanning over 1503 clauses and 20864 words. In addition

She provided another similar smaller corpus of 157 clauses that si also included into the evaluation.

Corpus	Elements	System Network	# characters (thousands)	# clauses	Annotator(s)
OCD	Syntax	Mood	16.2	147	Ela Oren & Eugeniu Costetchi
OE1	Semantics	Transitivity	51.8	1503	Anke Schultz
BTC	Semantics	Transitivity	5.6	157	Anke Schultz

Table 1.1 Evaluation corpus summary

The OE1 and BTC datasets have not been developed for the purpose of evaluating the current parser however they are compatible and can be adapted to evaluate (a) the boundaries of a constituent segment, (b) the constituent semantic function and (c) some Transitivity features. To enable each of these evaluations the annotation data and parser output need to be represented in such a way that they are comparable to each other. Specifically the feature names had to be harmonised with the ones from PTDB. The adjustments were the same as described in Section **??**.

1.1 Segment definition

To compare the segment boundaries we need to understand how they are represented in each output and how they can be brought to a common for comparison.

Listing 1.1 Segment example in UAM corpus tool

```
<segment numbersid="4" numbersstart="20" numbersend="27"
numbersfeatures="configuration;relational;attributive"
numbersstate="active"/>
```

Both datasets were created with UAM Corpus Tool (O'Donnell 2008a,b) version 2.4. The annotations, in this software, are recorded as segments spanning from a start to an end positions in the text file together with the set of features (selected from a systemic network) attributed to that segment. There are no constituency or dependency relations between segments. In Listing 1.1 is the XML representation for an example annotation segment where the *id* attribute indicates the unique identification number within the annotation dataset, the *start* and *end* attributes define the segment between two character offsets relative to the beginning of the text file.

Listing 1.2 Raw text example in annotation data

```
0 0Red riding hood excerpt24
1 25"What have you in that basket,  Little Red Riding Hood?"82
```

```
2 83
3 84"Eggs and butter and cake, Mr. Wolf."111
```

Listing 1.2 presents an example raw text from the annotation dataset containing an initial line and two sentences separated by an empty line. The greyed numbers in the beginning and end of each line indicate character offset. In some files, the first line plays the role of a header containing a title the file name. In this example it is a title. Either way, this, first line, is neither considered for annotation nor for parsing. Some files contain one sentence per line, others separate sentences by an extra blank line and in other files, the text is one continuous block. The text may contain sporadically tabs and blank spaces such as in the line 1 between the comma and the Little Red Riding Hood.

Parsimonious Vole, parses sentences one by one and not whole text document at once. Before parsing, the document is chunked into sentences guided by the punctuation delimiters. In addition, the Stanford dependency parser normalises the input text by trimming spaces and tabs, adjusting character encoding and replaces some special characters such as quotes, brackets, punctuation, etc.

Listing 1.3 presents the preprocessed text before it is parsed. The title line is cropped from the file. Each line contains a separate sentence. The text is tokenised such that words and punctuation marks are separated by a single space.

Listing 1.3 Parser preprocessed text example

```
0 0" What have you in that basket , Little Red Riding Hood ? "60
1 61" Eggs and butter and cake , Mr. Wolf . "102
```

After the parser preprocessing process, the input and output texts are no longer identical. The sentence and word offsets are no longer reliably findable with respect to the original text and the segments need to be matched. There are two fundamental kinds of segment misalignments: the segment offset shifting and the segment resizing (either shrinking or enlarging). Sometimes it can also be a combination of the two. Allen interval algebra (Allen 1983) systematizes the basic thirteen relations to position segments relative to each other. Nonetheless in the current work we are not interested to investigate how the segment offset shifts but rather what would be an efficient way to match them independent of their position.

To make manual and automatic annotations comparable, the constituency graphs need to be reduced to a set of segments corresponding to each constituent. The first task of the evaluation is to find matches or near-matches between two segment bundles that is addressed in the next section.

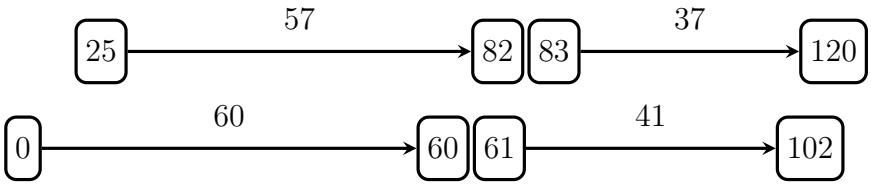


Fig. 1.1 Graphic representation of the sentence segment misalignment between Listing 1.2 and Listing 1.3

1.2 Reducing the CG to a set of segments

If we ignore the set of associated labels (systemic features), which we will consider in the next section, then two segments are we identical when (a) their start and end indexes and (b) the text in between is identical.

To address this issue in the current evaluation, the text processed by the parser is re-indexed back into the original raw text at the level of words (tokens), constituents and sentences. The high level perspective for generating segments for the constituency graphs indexed on the original text file is provided in Algorithm 1.

Algorithm 1: Algorithm to generate segments for the CG bundle indexed on the raw text

```
   input  : CG bundle, text
1  begin
2      offset ← 0
3      for cg in CG bundle:
4          generate segments for cg indexed on text given the offset
5          offset ← the end of cg
6  end
```

The result of the Algorithm 1 is a set of segments that are indexed according to the raw text by iterating the resulting constituency graphs one by one and indexing each with respect to the offset given by the previous one.

The way each CG is indexed is described by the Algorithm 2 which returns a set of segments from the constituency graph knowing given an offset. First a token level indexing is performed and corresponding segments generated. Then for each CG constituent, segments are generated based on its word span. The indexes are set from beginning of the first word to the end of the last word. The labels of the segment are the CG class, functions and systemic features discussed in the next section.

$$d = \sqrt{(start_S - start_T)^2 + (end_S - end_T)^2} \tag{1.1}$$

Algorithm 2: Algorithm to generate segments for CG constituents indexed on the raw text

```
   input  : cg, text, sentence offset
 1 begin
 2     words ← get cg the list of words
 3     for word in list of sentence word segments:
 4         find the word in the text after a given sentence offset
 5         if word found:
 6             start ← get first word start index
 7             end ← get the last word end index
 8             create a new segment (start, end, word)
 9         else:
10             generate a warning (manual adjustment needed)
11     for node in cg in BFS postorder:
12         find the word span of the constituent
13         start ← get first word start index
14         end ← get the last word end index
15         labels ← get node class, function and features
16         create new segment (start, end, labels)
17     return set of segments
18 end
```

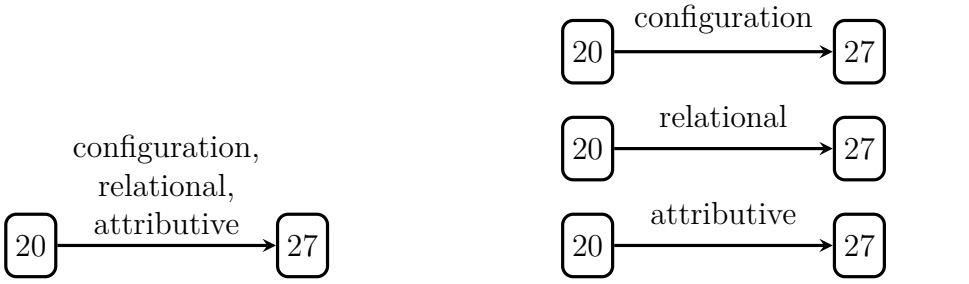
$$d = \sqrt{\Delta^2_{start} + \Delta^2_{end}} \tag{1.2}$$

Later in this chapter I address how the manually created segments and parser generated segments are aligned. Here it is noteworthy to mention that the manually created segments contain errors. Some segments are either shifted and include the adjacent spaces or, the converse, leave out one or two characters of a marginal word. For this reason the segment alignment process needs to tolerate a certain distance between the stand/end indexes of aligned segments. I adopted the geometric distance to measure the difference between the segment start and end indexes. For two segments $S(start_S, end_S)$ and $T(start_T, end_T)$ the geometric distance is defined in Equation 1.1. We can replace the difference between start and end indexes with Δ_{start} and Δ_{end} notation and obtain a reduced form provided in Equation 1.2.

1.3 Considering the segment labels

Provided that the task of indexing the segment indexes and content such that they "roughly" correspond to each other we need to start comparing the segment labels. However the labels of manually created segments differ from the segments provided by the parser. Besides shift in the offset mentioned above, another difference is in number of ascribed segments. In many instances, the manual annotation is less delicate and thus contain less features than the one provided by the parser. Still, some of the manual annotations contain features that are not provided by the parser (i.e. not yet implemented), such as the Thematic structure of the clause or the Modal Assessment features.

To address this issue, in the current evaluation, the segments are reduced to carry only one label. The consequence is that the segments with multiple features (Figure 1.2a) are broken down into multiple segments with the same span (Figure 1.2b) for each feature. When each segment contains exactly one feature the evaluation can be focused on one or a set of features of interest by selecting only the segments that contain exactly those.



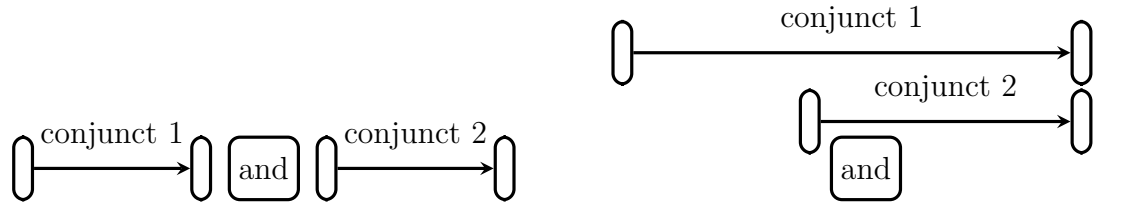
(a) A segment with a set of features (b) A set of segments with single features

Fig. 1.2 Example of breaking down a segment with multiple features into set of segments with a single feature

There are more differences in the grammar and annotation style between the manual and automatic processes. This means that both structure and feature names provided by the system networks differs (already mentioned above that the Theme system network is not implemented in the current version of the parser but provided in some manual annotations).

In the corpus, punctuation marks such as commas, semicolons, three dots and full stops are not included into the constituent segments while the parser includes them at the end of each adjacent segment. None of the conjunctions (such as "and", "but", "so", etc.) are included into the conjunct segments, they are considered markers in

the clause/group complexes rather than part of the constituent. The treatment of conjunction is discussed in Section **??**. The parser, on the other hand, includes the conjunctions into the following adjacent segment. Moreover the conjuncts spans differ as well. Instead of being annotated in parallel, as depicted in Figure 1.3a, the segments are subsumed in a cascade from the former to the latter as depicted in Figure 1.3b.



(a) Conjuncts annotated as parallel segments (b) Conjuncts annotated as subsumed segments

Fig. 1.3 Treatment of conjunctions in the corpus compared to the parser

The differences in the grammar are due to the slight variations in the feature names and in the system network structure. For example the element names are spelled differently "predeictic" and "pre-deictic", "root" and "clause" or "explative-marker" and "expletive". Similar spelling variations are present in the feature names for example "two role" and "two-role-action" and other variations alike. To remedy this issue the current evaluation establishes mappings between the manual and automatically generated features.

Now that the main divergences have already been outlined, I proceed to explain how the stable matches between the segments are established.

1.4 The matching algorithm

The task of aligning two sets of labelled segments is almost the same as the well know problem in computer science called *stable marriage problem* (Gusfield & Irving 1989). The standard enunciation of the problem is provided below and is solved in an efficient algorithm named Gale-Shapley (Gale & Shapley 1962) after its authors.

> Given n men and n women, where each person has ranked all members of the opposite sex in order of preference, marry the men and women together such that there are no two people of opposite sex who would both rather have each other than their current partners. When there are no such pairs of people, the set of marriages is deemed stable Iwama & Miyazaki (2008).

In the context of this evaluation the group of men is associated with the segments generated automatically by the parser and the group of women with the segments available from the manual analysis.

The standard stable marriage problem is formulated such that there is a group of men and a group of women and each individual from each group expresses their preferences for every individual from the opposite group as an ordered list. The assumption is that the preferences of every individual are known and expressed as a complete ordered list of individuals from the opposite group ranging from the most to the least preferred one. Thus the preference list must be *complete* and *fully ordered*.

To fulfil these requirements I construct a distance matrix from each automatically created segment to every and manually created one. The distance measure considered here is the Euclidean one provided in Equation 1.2 above. The matrix represents the complete and fully ordered set of preferences stipulated in the original problem formulation. In addition to distance the segments need to carry the same labels in order to be considered a match. This condition is not found in the original problem but is considerer in Algorithm 3 below.

Algorithm 3: The algorithm for matching automatic and manual segments

```
   input  : aut segments, man segments
 1 begin
 2     mark all aut segments and man segments free
 3     compute distances from each man segments to every aut segments
 4     while ∃ free aut segments:
 5         aut ← first free from aut segments
 6         if ∃ man segments not yet tested to match aut:
 7             man ← the nearest among man segments to aut with identical label
 8             if man is free:
 9                 match aut and man
10                 mark aut and man as non-free
11             else:
12                 aut′ ← the current match of man
13                 if aut is closer to man than aut′:
14                     match aut and man
15                     mark aut and man as non-free
16                     mark aut′ as free
17         else:
18             mark aut as non-free and non-matching
19     fintq
20 end
```

The input to the algorithm is the set of automatically generated segment list aut segments and the manually created segment list man segments. It begins with initialising all the segments as *free* meaning that none of them are matched yet (originally married). The segments can be marked either be as *non-free matching* or *non-free non-matching*. A Part of the initialisation is also creation of the distance matrix mentioned above representing a complete and ordered set of preferences for each segment.

The algorithm iterates over the aut segments for as long as there exist free ones. If there exists a man (manually created segment) which have not been attempted to match aut then attempt matching aut to the nearest man that has the same label. If man is free then it is a match. Otherwise compare aut to aut′ (currently assigned match of man) and then consider a match with the nearest one (i.e. shortest distance). When there are no man segments to test mark aut as non-free and non-matching meaning that it shall no longer be considered by the algorithm.

In the end the algorithm provides a list of successful matches out of which we can also deduce which aut segments man segments have not been matched. The next section provides the empirical measurements resulting from applying the current method.

1.5 The measurements

In this section I present a series of measurements addressing two quality criteria of the parser. Firstly, the degree of divergence in segment spans between manual and parse segments in all corpora (OCD, BTC and OE1). And secondly, the accuracy of the parser with respect to manual annotations measured through precision, recall and F_1 measures. The syntactic constituency and some Mood features are derived from OCD corpus only, followed by the semantic constituency and Transitivity features evaluation based on OE1 and BTC corpora.

1.5.1 Segment divergence: general findings

The segments don't perfectly align, as described in previous sections, due to minor differences in annotation approach, text normalisation and trimming before parsing (that was not performed before the manual annotation), errors in the annotations (missing or including characters). The misalignment is measured using the geometric distance on segments matched with Algorithm 3. The presented statistics are calculated on a number of over 12500 segment pairs whose mean distance is 4.84 with a standard deviation of 14.51. The distance is distributed between a minimum 0 and maximum

219. A mean close to the minimum point and a large standard deviation indicates that most of the data points are situated between 0 and slightly over the mean continued by a very long tail of rare and distant data points. This intuition can be seen in the histogram depicted in Figure 1.4 below. This dataset resembles the Pareto distribution where 74% are almost not shifted (by up to 1 character) or 83% of the segments are slightly shifted up to 5 characters.

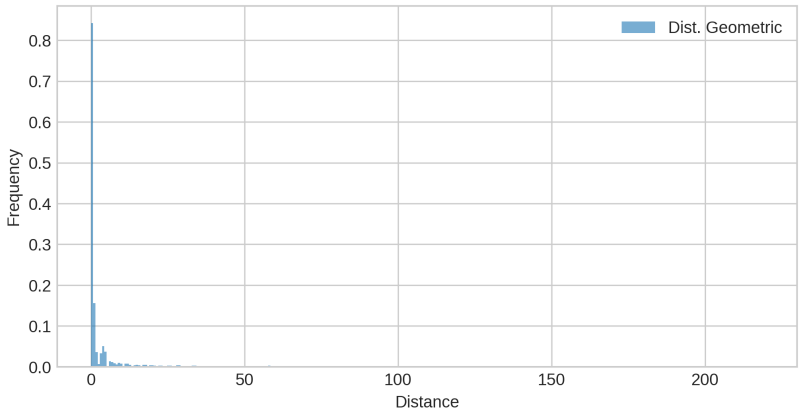


Fig. 1.4 Full histogram of the distance distribution of the matched segments (binning=300)

Because of a very long tail, I reduce the further analysis to the distance span between 0 and 40. Figure 1.6 depicts the histogram of distances between matched segments (in 300 bins). It shows that most segments (89%) are cumulated in first two bins with distance up to 2. The consequent three bins also contain a significant portion of segments with distance up to 5. The rest of the bins, represent a very long tail, spanning on distances up to maximum 219 and containing a small amount of segments.

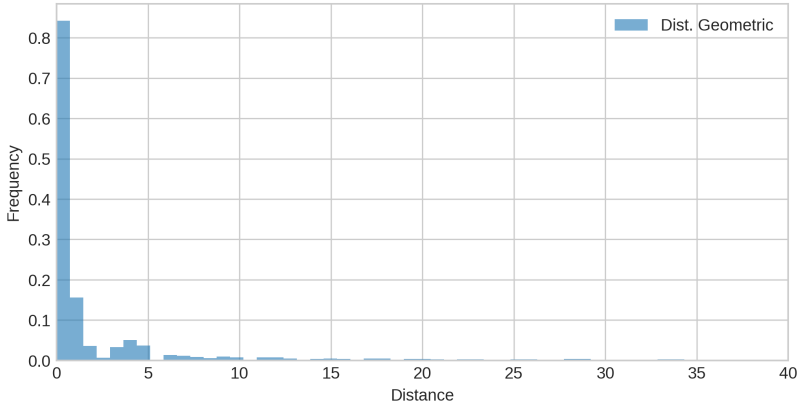


Fig. 1.5 Reduced histogram of the distance distribution of the matched segments (binning=300). View reduced to the a distance of 40 characters

This distribution can be viewed in it's cumulative form depicted in Figure 1.6. Here we see that over 51% of segments are perfectly aligned. 80% of the segments are slightly shifted up to a distance of maximum 5 characters. This can be explained by differences in (a) punctuation, (b) conjunction and (c)verbal group treatment described above. The next 5% of the segments are shifted between 5 and 10 characters. The last 10% cover the heavy shift of distances 20 to 219. This may be due differences in verbal group treatment, subsumed conjuncts (clause and group conjunctions) and erroneous prepositional phrase attachment (treated as Qualifier instead of Complement/Adjunct or vice versa).

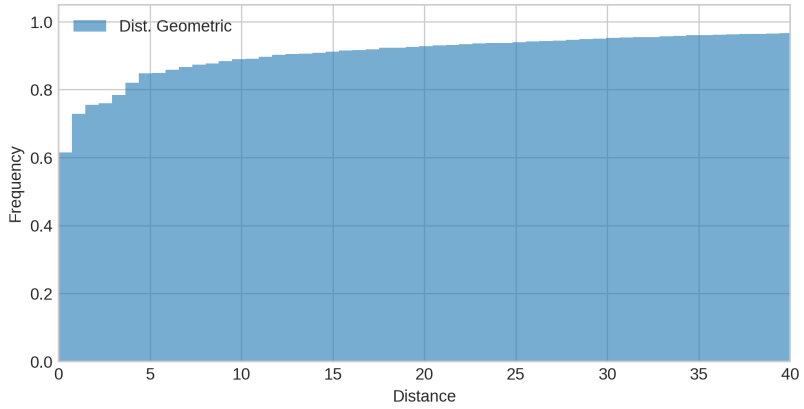


Fig. 1.6 Cumulative histogram of the distance distribution of the matched segments (binning=300). View reduced to the a distance of 40 characters

1.5.2 Segment divergence breakdown by element type

If we take a closer look at the Figure 1.5 we will notice some spikes and a tendency towards some local values. What would the histogram look like if we group values in a different manner considering these spikes and the long tail of the distribution. In Figure 1.5 we can observe several dips at around distances 3, 6, 11, 13, 17, 19, 22, 24 and so on. I decided to consider these dips for deciding bin borders in the new bin design. And, since the tail is very long with very few values, as the bins advance to the left, each is designed over longer span this way compressing the tail. In Table 1.2 is presented such binning design. Also each interval is assigned a category that means the degree of deviation.

Besides custom binning described above, looking at each segment label in part brings more clarity to the evaluation. The most relevant segment labels are the unit elements. In the current evaluation, the annotators provide clause level syntactic

degree	insignificant	tiny	little	moderate	significant	high
distance intervals	0-3	3-5	5-10	10-20	20-50	50-250

Table 1.2 The progressive binning scale considering the dataset properties

(subject, complement, adjunct, predicator, finite) and semantic (configuration, main verb, participant) elements. These elements are used a dimension in the breakdown of the segment divergence that follows.

Using the bins defined in Table 1.2 the distribution of segment deviations grouped by the main syntactic elements is provided in Table 1.3 that is also depicted in Figure 1.7.

element	**% of segments per degree of deviation**					
	insignificant (0-3)	**tiny (3-5)**	**little (5-10)**	**moderate (10-20)**	**significant (20-50)**	**high (50-250)**
predicator	29.38	0.95	0.30	0.10	0.00	0.05
subject	22.70	0.10	0.25	0.30	0.05	0.00
adjunct	13.86	0.45	0.60	0.15	0.85	0.20
complement	12.10	0.75	1.46	2.26	1.96	1.86
finite	9.19	0.00	0.00	0.00	0.05	0.05

Table 1.3 Percentage of segments deviated to a given degree for major syntactic elements

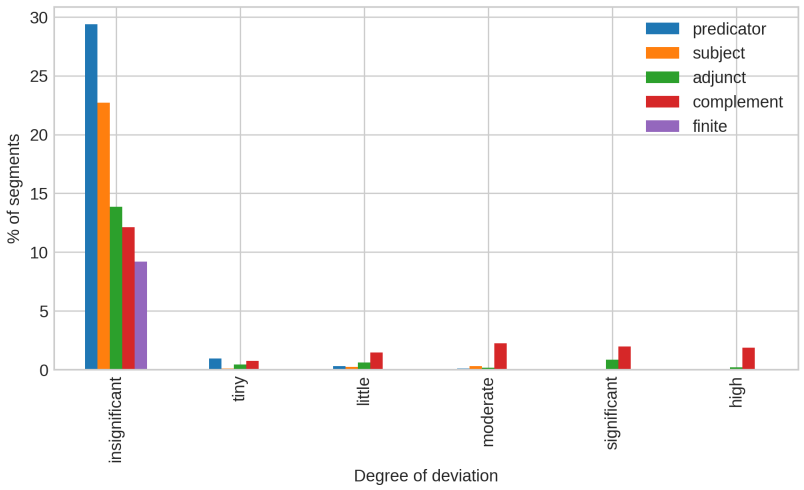


Fig. 1.7 Bar chart of the segments deviated to a given degree for major syntactic elements

In Figure 1.7 we can see that most deviations are insignificant. The number of segments in the rest of the bins, from tiny to high, is below 1% in every bin which perhaps reflects errors in annotation and/or matching algorithm requiring further

investigation. In case of complements, however, the proportion of segments is slightly higher (0.7–2.2%). This may be explained by the problem of prepositional phrase attachment and further analysis is needed to test this hypothesis.

When we switch to a grouping by the main Transitivity elements maintaining the same bins defined in Table 1.2, then the distribution of segment deviations looks as outlined in Table 1.4. The same data are depicted in Figure 1.8.

feature	**% of segments per degree of deviation**					
	insignificant (0-3)	**tiny (3-5)**	**little (5-10)**	**moderate (10-20)**	**significant (20-50)**	**high (50-250)**
participant-role	36.78	2.21	3.48	2.80	3.04	1.67
main	20.75	3.48	1.23	0.39	0.00	0.00
configuration	7.75	3.73	2.80	3.04	4.56	2.31

Table 1.4 Percentage of segments deviated to a given degree for major semantic elements

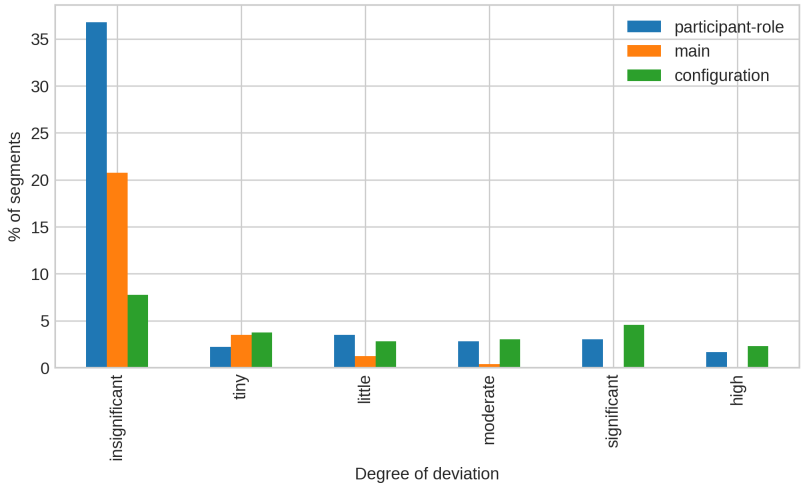


Fig. 1.8 Bar chart of the feature segments deviated to a given degree for major semantic elements

Similarly to Figure 1.7, in Figure 1.8 we can see that most of the deviations are insignificant (0-3 characters). In the rest of the bins representing higher degrees of deviation the amount of segments is up to 4.7% which reflects higher number of shifted segments. One exception is the main verb whose occurrences decreases towards higher degrees of deviation, i.e. it is shifted by an insignificant, tiny or little degree that is up to about 10 characters. This is explainable by the fact that the main verb is most of the time (if not always) a single word, which is always shorter than the configuration or participant elements which usually span clauses of groups comprising multiple words.

The large variations in participant seems to correlate with complement variation and might be explained by the attachment errors. In cases of high configuration deviation however further investigation are needed because these may possibly be errors in the segment matching algorithm or other unknown anomalies.

1.5.3 Syntactic evaluation: Constituency elements

The evaluation data in this and the following sections will be presented in tables with the same structure. Using Table 1.5 as example I explain next what the columns mean. The first column will contain the name of the unit type, element or feature. The next three columns *Match*, *Manual mn* and *Parse mn* represent the number of segments that are considered identical between the corpus and parser, the number of unmatched the corpus (manually created) segments and the number of unmatched parser (automatically generated) segments. The next three columns *Precision*, *Recall* and F_1 represent standard accuracy metrics indicating the fraction of relevant instances among the retrieved instances, fraction of relevant instances that have been retrieved over the total amount of relevant instances and the harmonic mean of the previous two. In addition, the column *%Total matched* represents the percentage the current item (row) in the table while the *% Manual nm* and *%Parse nm* represent the number of remaining unmatched segments of the current item (row) that represents a translation of the Manual and Parse mn columns into relative terms.

Unit type	Matched	Manual nm	Parse nm	Precision	Recall	F1	%Total matched	%Manual nm	%Parse nm
clause	612.00	64.00	78.00	0.89	0.91	0.90	37.00	9.47	11.30
nominal-group	717.00	108.00	67.00	0.91	0.87	0.89	43.35	13.09	8.55
prepositional-group	119.00	39.00	39.00	0.75	0.75	0.75	7.19	24.68	24.68
adverbial-group	161.00	79.00	103.00	0.61	0.67	0.64	9.73	32.92	39.02
adjectival-group	45.00	36.00	38.00	0.54	0.56	0.55	2.72	44.44	45.78

Table 1.5 The evaluation statistics for the main constituency unit types

The syntactic accuracy aims to measure how well the main unit types and the clause main elements have been detected by the parsed compared to the corpus. The evaluation is performed on the OCD corpus. This evaluation is restricted to the clause and four group types: nominal, prepositional, adverbial and adjectival. No clause complexes, group complexes or word types are included. The evaluation results are presented in Table 1.5 where we can see that clauses and nominal groups have an F_1 score of about 90%. Prepositional, adverbial and adjectival group scores decrease to 55% and requires investigation of the errors in the parsing and matching algorithms. There is also a contrast in the number of segments, visible in the bar chart from Figure

1.9, between the first two element types and the last three with a ratio of one to four or more.

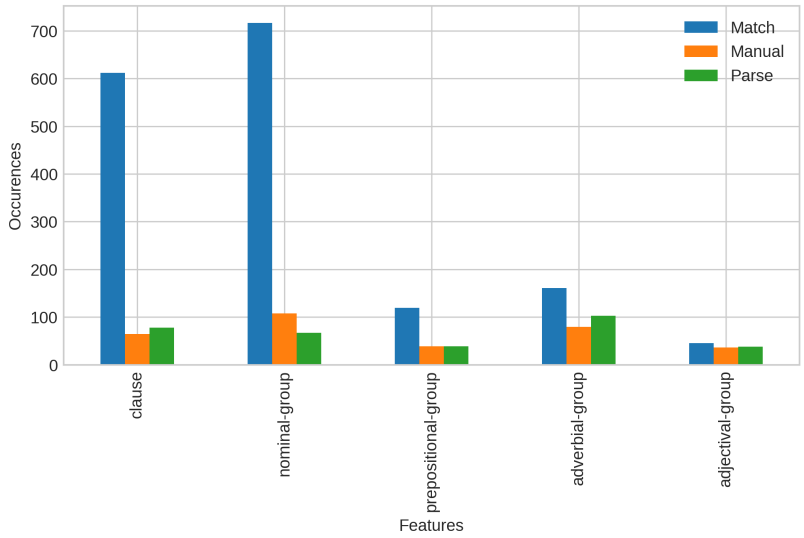


Fig. 1.9 Bar chart of matched and non-matched (manual and parse) segments of the main constituency unit types

Table 1.6 presents the evaluation result for the main clause elements. Some of them such as, auxiliary verbs, main verb extension, negation particle, and others have been omitted in the corpus and thus missing in the present evaluation. For the predicator (i.e main verb) and subject elements the F_1 measure raises to 90%. For the complements and finite the F_1 score is 67% and 63%. Surprisingly the complements have a small number of corpus unmatched segments and a high number of parser unmatched segments. This is explained by a flaw in the annotation methodology because the clausal complements were often annotated directly as new clause and omitting to draw the same segment with complement element. This required the corpus revision and correction. Adjuncts however have a higher number of unmatched segments on both sides and this may be dues to bugs in the parser.

Clause element	**Matched**	**Manual nm**	**Parse nm**	**Precision**	**Recall**	**F1**	**%Total matched**	**%Manual nm**	**%Parse nm**
predicator	613	60	79	0.89	0.91	0.90	30.79	8.92	11.42
subject	466	22	86	0.84	0.95	0.90	23.41	4.51	15.58
complement	406	43	350	0.54	0.90	0.67	20.39	9.58	46.30
adjunct	321	159	224	0.59	0.67	0.63	16.12	33.12	41.10
finite	185	3	392	0.32	0.98	0.48	9.29	1.60	67.94

Table 1.6 The evaluation statistics for the clause main elements

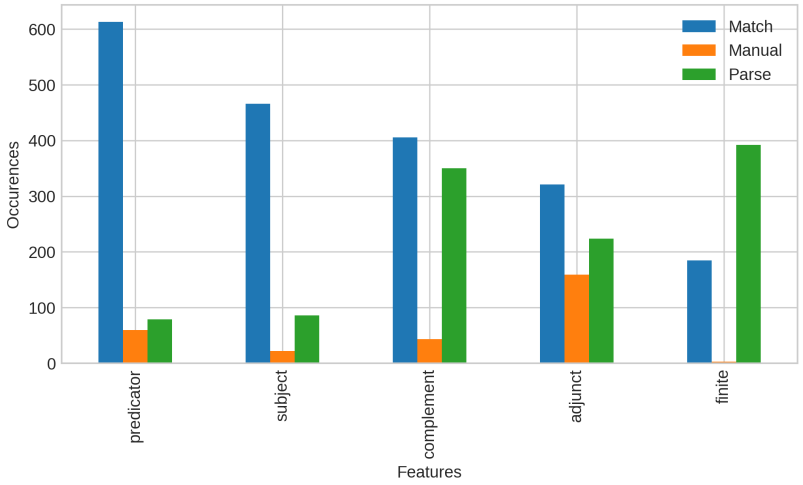


Fig. 1.10 Bar chart of matched and non-matched (manual and parse) segments of the clause main elements

1.5.4 Syntactic evaluation: Mood feature selections

In this section I present the evaluation of Mood system network selections. The corpus contains selections from a pat of the system network that is depicted in Figure 1.11. The full system network is provided in Figure **??**. Employing the entire system network in the annotations was difficult because as the delicacy increases the time spent for the annotation process increases.

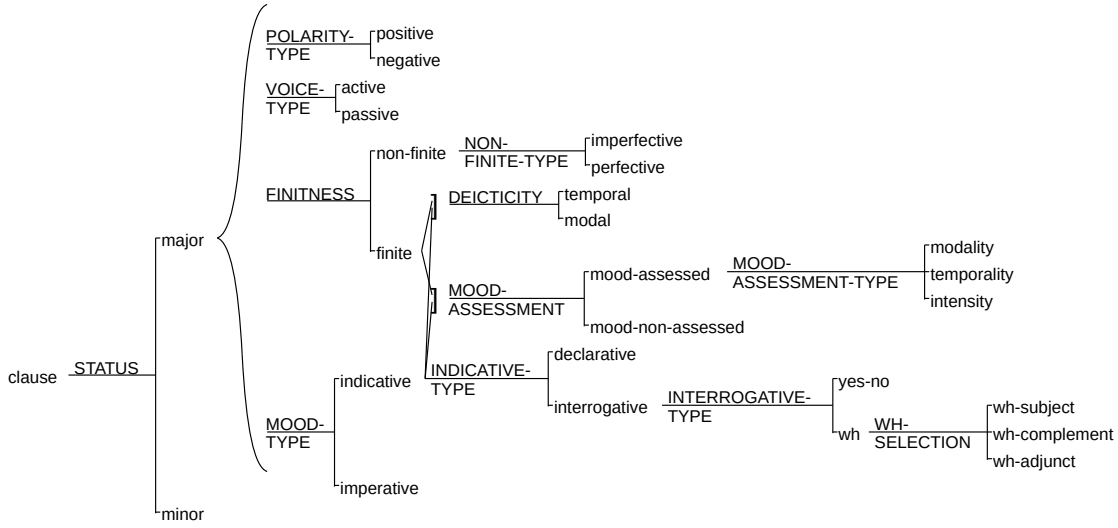


Fig. 1.11 The part of the Mood system network that has been used in OCD corpus annotation

Feature	Matched	Manual nm	Parse nm	Precision	Recall	F1	%Total matched	%Manual nm	%Parse nm
POLARITY-TYPE									
positive	485.00	125.00	55.00	0.90	0.80	0.84	89.48	20.49	10.19
negative	57.00	10.00	70.00	0.45	0.85	0.59	10.52	14.93	55.12
VOICE-TYPE									
active	553.00	102.00	68.00	0.89	0.84	0.87	98.05	15.57	10.95
passive	11.00	11.00	28.00	0.28	0.50	0.36	1.95	50.00	71.79
FINITNESS									
non-finite	99.00	19.00	38.00	0.72	0.84	0.78	15.84	16.10	27.74
finite	526.00	33.00	554.00	0.49	0.94	0.64	84.16	5.90	51.30
NON-FINITE-TYPE									
perfective	71.00	12.00	16.00	0.82	0.86	0.84	73.20	14.46	18.39
imperfective	26.00	9.00	24.00	0.52	0.74	0.61	26.80	25.71	48.00
DEICTICITY									
temporal	446.00	74.00	55.00	0.89	0.86	0.87	97.38	14.23	10.98
modal	12.00	33.00	6.00	0.67	0.27	0.38	2.62	73.33	33.33
MOOD-ASSESSMENT-TYPE									
temporality	35.00	17.00	27.00	0.56	0.67	0.61	56.45	32.69	43.55
modality	15.00	32.00	8.00	0.65	0.32	0.43	24.19	68.09	34.78
intensity	12.00	14.00	43.00	0.22	0.46	0.30	19.35	53.85	78.18
MOOD-TYPE									
indicative	455.00	216.00	37.00	0.92	0.68	0.78	99.13	32.19	7.52
imperative	4.00	1.00	31.00	0.11	0.80	0.20	0.87	20.00	88.57
INDICATIVE-TYPE									
declarative	355.00	260.00	27.00	0.93	0.58	0.71	88.31	42.28	7.07
interrogative	47.00	7.00	63.00	0.43	0.87	0.57	11.69	12.96	57.27
INTERROGATIVE-TYPE									
wh	40.00	6.00	57.00	0.41	0.87	0.56	88.89	13.04	58.76
yes-no	5.00	3.00	8.00	0.38	0.62	0.48	11.11	37.50	61.54
WH-SELECTION									
wh-subject	9.00	3.00	7.00	0.56	0.75	0.64	32.14	25.00	43.75
wh-adjunct	11.00	15.00	3.00	0.79	0.42	0.55	39.29	57.69	21.43
wh-complement	8.00	0.00	62.00	0.11	1.00	0.21	28.57	0.00	88.57

Table 1.7 The evaluation statistics for POLARITY-TYPE systemic choices

1.5.5 Semantic evaluation: Constituency elements

This section describes the empirical evaluation of the semantic annotations. This evaluation is based on the OE1 and BTC corpora created by Anke Schultz for her PhD project. She annotated the text with Cardiff Transitivity system.

1.5.6 Semantic evaluation: Transitivity feature selections

For the semantic evaluation has been used a corpus created by Anke Schultz for her PhD project annotated with Transitivity features i.e Configuration, Process and Participant which from constituency point of view correspond to the Clause, Predicator and Participants(predicate arguments) to Subjects and Complements cumulated.

Name	*M/N*	*M/%*	*A/N*	*A/%*	*Prec*	*Rec*	F_1
Configuration	1466	-	1833	-	0.736	0.915	0.814
Process	1423	-	1814	-	0.707	0.902	0.791
Participant	2652	-	3398	-	0.732	0.925	0.816

Table 1.8 Transitivity System evaluation statistics

Transitivity analysis is semantic in nature and poses challenges in meaning selection beyond constituent class or function. Current parser does not select one meaning but rather proposes a set of possible configurations for each clause. If top level Transitivity features correspond fairly to the number of syntactic constituents, the more delicate features are parsed with an average of 2.7 proposed features for each manually annotated one.

Name	*M/N*	*M/%*	*A/N*	*A/%*	*Prec*	*Rec*	F_1
CONFIGURATION-TYPE	1466	23.12	1308	7.57	0.542	0.483	0.508
action	359	24.82	453	33.99	0.315	0.409	0.333
relational	546	37.79	445	34.77	0.645	0.530	0.574
mental	452	29.86	318	24.27	0.744	0.530	0.605
influential	81	6.69	91	7.39	0.556	0.519	0.484
event-relating	28	3.48	1	1.85	1.0	0.5	0.667
environmental	0	0	0	0	0	0	0
other	0	0	0	0	0	0	0

Table 1.9 Configuration type evaluation statistics

The semantic evaluation yields surprisingly positive results for relational and metal processes. At the same time, actions would have been expected to score a high accuracy but as seen in the results it is not the case. Despite a high number of them both in the corpus (24%) and in the parses (33%) they seem to be misaligned and perhaps not matched in the process. Further investigation should be conducted to spot the source of such a low score. Next I present a short critical discussion of the overall evaluation.

1.6 Discussion

Further investigation shall be conducted to determine the error types, shortcomings in the corpus and the parser. But beyond the simple improvement to the corpus such as adding the missing Finite elements it would benefit tremendously from a second annotator in order to evaluate reliability of the annotation itself and how much of a gold standard it can be considered for the current work.

Also the corpus size is very small and many features are missing or severely underrepresented and bear no statistical significance. For example event relating, environmental and other processes are missing from the corpus. Also the number of other features that the parser provides are missing from the manual analysis and it would be interesting to add some of them to study how varies the accuracy distribution as the delicacy of features increases.

Since for both syntactic and semantic annotations there is only a single author annotation available, the results shall be considered indicative and by no means representative for the parser performance. Nevertheless they can already be considered as a good feedback for looking into certain areas of the grammar with considerably low performance in order identify the potential problems.

Chapter 2

Conclusions

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

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

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

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

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

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

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

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

As discussed in the Section 1.6 further investigation shall be conducted to determine the error types, shortcomings in the corpus and the parser. Since for both syntactic and semantic annotations there is only a single author annotation available, the results shall be considered indicative and by no means representative for the parser performance. Nevertheless they can already be considered as a good feedback for looking into certain areas of the grammar with considerably low performance in order identify the potential problems.

2.1 Practical applications

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

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

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

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

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

2.2 Impact on future research

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

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

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

2.2.1 Verbal group again: from syntactically towards semantically sound analysis

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

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

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

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

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

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

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

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

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

The benefits of such a merger leads to an increased comprehensiveness, not only of the transitivity analysis – demonstrated by the examples in Tables 2.1 and 2.2 – but also of the modal assessment that includes modality, as demonstrated by the Examples 1 and 2.

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

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

Examples 1 and 2 represent cases when the modal assessment of the lower clause is carried on by the higher one. In both examples, the higher clause can be replaced by the modal verb *maybe* or the adverb *perhaps*.

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

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

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

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

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

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

2.2.3 Taxis analysis and potential for discourse relation detection

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

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

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

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

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

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

2.2.4 Towards speech act analysis

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

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

2.2.5 Process Types and Participant Roles

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

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

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

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

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

2.2.6 Reasoning with systemic networks

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

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

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

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

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

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

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

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

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

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

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

2.2.8 The use of Markov Logics for pattern discovery

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

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

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

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

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

2.3 A final word

References

Allen, James F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11). 832–843. doi:10.1145/182.358434. <http://portal.acm.org/citation.cfm?doid=182.358434>.

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

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

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

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

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

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

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

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

Fawcett, R. 1988. Language Generation as Choice in Social Interaction. In Zock, M. & G. Sabah (eds.), *Advances in natural language generation*, vol. 2, 27–49. Pinter

Publishers. (Paper presented at the First European Workshop of Natural Language Generation, Royaumont, 1987).

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

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

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

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

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

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

Gale, D. & L. S. Shapley. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* 69(1). 9–15. <http://www.jstor.org/stable/2312726>.

Gusfield, Dan & Robert W. Irving. 1989. *The stable marriage problem: Structure and algorithms*. Cambridge, MA, USA: MIT Press.

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

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

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

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

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

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

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

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>.

Iwama, Kazuo & Shuichi Miyazaki. 2008. A survey of the stable marriage problem and its variants. In *International conference on informatics education and research for knowledge-circulating society*, 131–136. IEEE.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

SFL Syntactic Overview

.1 Cardiff Syntax

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

.1.1 Clause

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

.1.2 Nominal Group

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

.1.3 Prepositional Group

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

.1.4 Quality Group

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

.1.5 Quantity Group

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

.1.6 Genitive Cluster

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

.2 Sydney Syntax

.2.1 Logical

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

.2.2 Textual

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

.2.3 Interactional

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

.2.4 Experiential

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

.2.5 Taxis

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

Stanford Dependency schema

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

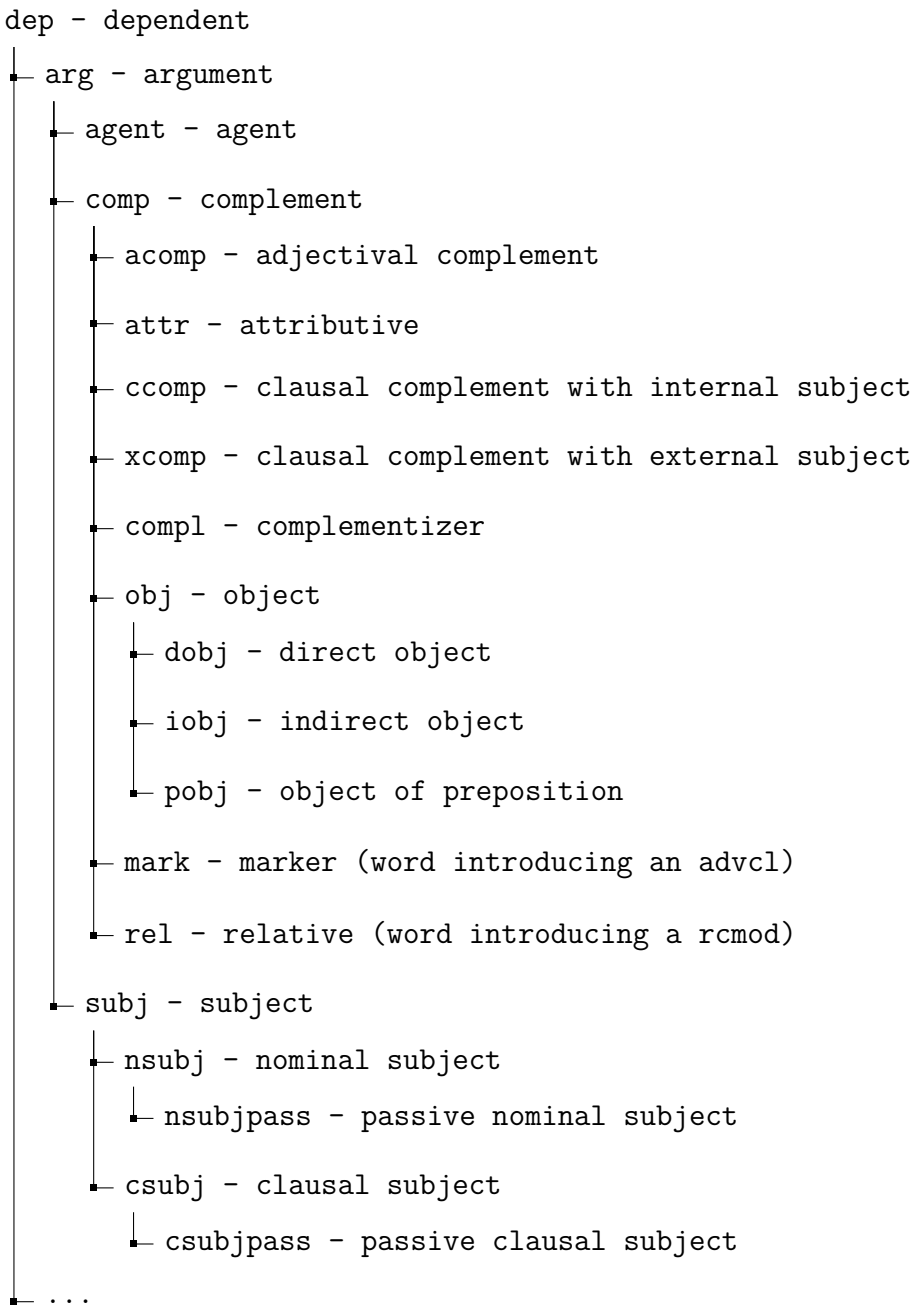


Fig. 1 The Stanford dependency scheme - part one

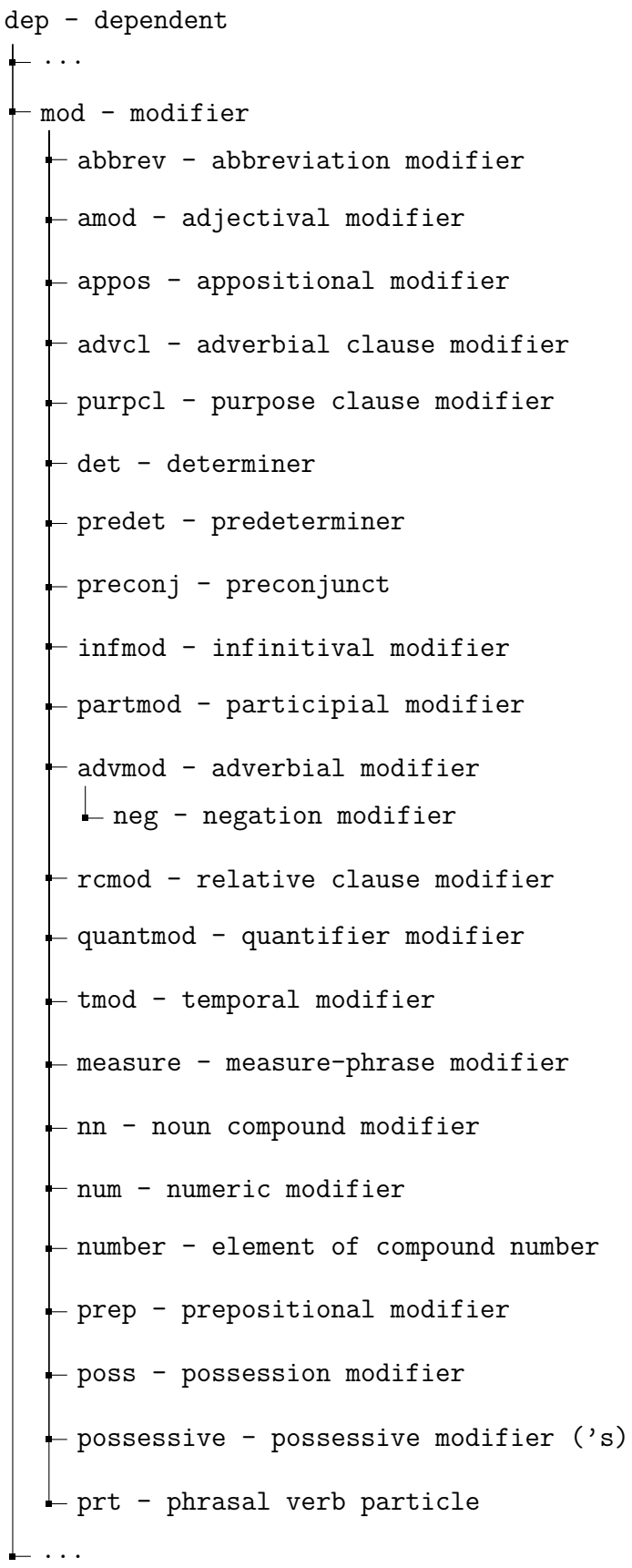


Fig. 2 The Stanford dependency scheme - part two

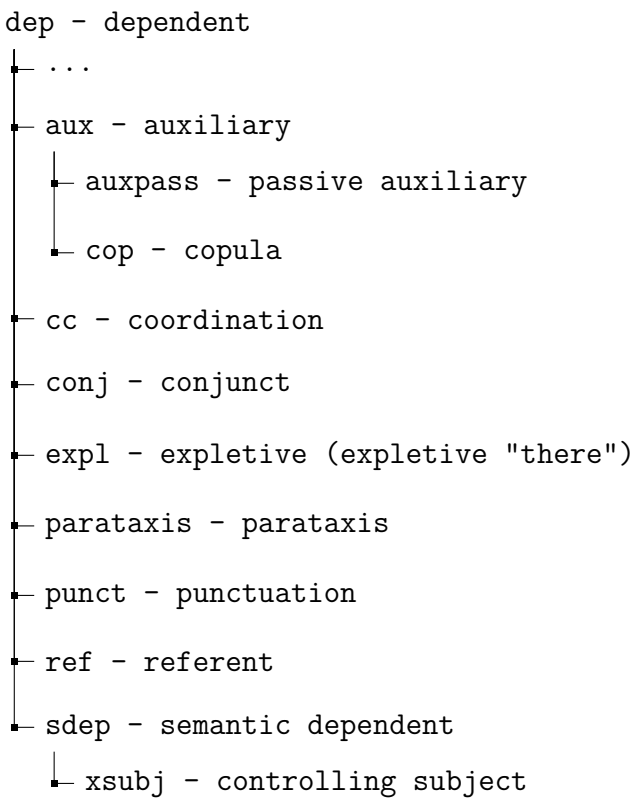


Fig. 3 The Stanford dependency scheme - part three

Penn treebank tag-set

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

Table 3 Penn Treebank tag set