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

April 2019

Table of contents

Chapter 10

Empirical evaluation

This chapter presents the evaluation the Parsimonious Vole parser. 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 grammar that is employed in this evaluation was already introduced in Chapter 4, the corpus annotations will be introduced in Section 10.1, while the evaluation method and the results will be covered in Sections 10.2 and **??**.

The evaluation data were generated by comparing the labelled segments available from the corpus annotations to the labelled segments from the parser output. The parser accuracy is measured in terms of *precision*, *recall* and F_1 scores. The parser *precision* measures how many segments have been produced by the parser that are also found in manual analysis; and the parser *recall* measures how many correct segments have been produced by the parser relative to the total number of produced segments. F_1 score is a harmonic mean of the precision and recall.

Each corpus was annotated in a manner that slightly differs from the analysis structure generated by the parser, due to differences in the annotation methodology. These differences will be described in detail in Section 10.1.3. Before then, to give you a sense of what the differences are consider the example segments in Listings 10.1 and 10.2. In this example, the segments differ in the way the spaces and the end of sentence punctuation are framed into the clause segments and so, in this evaluation, they are considered as partially (or closely) matching clauses. The exact matches are used to establish a base line accuracy.

Listing 10.1 Example segment from the corpus

```
587forced me into treatment611
```

Listing 10.2 Example segment from the parser output

```
583 and forced me into treatment . 612
```

The evaluation methodology, which will be described in detail in Section 10.2, considers perfect alignment between segment boundaries and their labels. Also it considers partial alignment of segments with the same label provided that the difference between them is not too large and there is not other better matching candidate at a shorter distance. This means that segment spans such as the ones in Listings 10.1 and 10.1 are given some credit in the alignment process. The main reason for taking segmentation discrepancies into consideration is to provide a wider sample for the systemic selections evaluation available for the MOOD and TRANSITIVITY system networks. Another reason for keep the partial matches is that discarding them completely is not entirely correct either. The next section describes the corpora used in current evaluation, followed by the evaluation mythology and finally the evaluation results are presented.

10.1 Evaluation corpus

This section briefly introduces the two corpora used in the evaluation of the Parsimonious Vole parser. They are the OCD corpus annotated with Mood features, and the OE corpus annotated with Transitivity features. Table 10.1 provides a summary of the corpora descriptions.

Corpus name	Meta-function	Characters	Clauses	Annotator(s)
OCD	Mood	16.200	529	Ela Oren & Eugeniu Costetchi
OE	Transitivity	51.800	1503	Anke Schultz & Tatsiana Markovic

Table 10.1 Evaluation corpus summary

Each corpus was annotated in a manner that slightly differs from the analysis structure generated by the parser. After having presented each corpus in more details in the next two sections, I present the main differences between then and the parser output in the last part of this section.

10.1.1 OE corpus

The OE corpus is a smaller part of BTC corpus annotated that was annotated in Cardiff Transitivity style during the PhD work of Anke Schulz. The Bremen Translation

Corpus (BTC) was created at the University of Bremen by Kerstin Fischer, Anatol Stefanowitsch and Anke Schulz. It consists of comparable and parallel texts. The comparable part consists of a series of newsgroup texts of about 10,000 words of English text and another 10,000 words of German, text taken from the same register. The parallel part, called EDNA, is much larger comprising about 100,000 words of parallel English-German text. Anke uses in her thesis 10,000 words of parallel text and about the same of comparable text (Schulz 2015: 31). In this evaluation only the English part is considered which is called OE corpus. It comprises 31 files spanning over 1503 clauses and 20864 words.

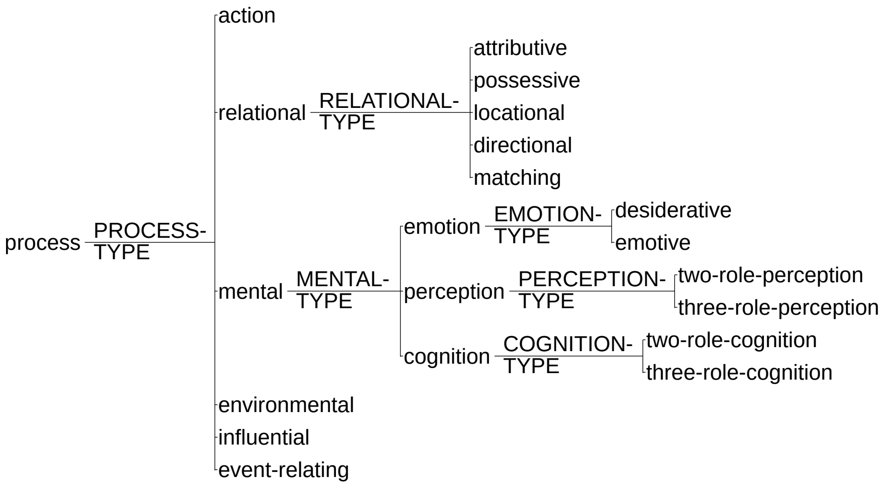


Fig. 10.1 The fragment of the TRANSITIVITY system network that has been used in the corpus

The corpus annotations, developed by Anke Schulz and Tatsiana Markovic (Schulz 2015: 36), cover Cardiff TRANSITIVITY, THEME and MODIFICATION system networks. The grammatical details and the annotation methodology are covered in detail in Schulz (2015: 48-161). In addition, Anke provided a set of annotations performed in the same manner for the “Little Red Riding Hood” fairy tale comprising 157 clauses which are also included in this evaluation as part of OE corpus.

For the purpose of the current evaluation only the TRANSITIVITY system was considered. The extent to which the system network is covered in the corpus annotations is limited to the top part of the original TRANSITIVITY system network. The used system network fragment is depicted in Figure 10.1, while the whole system network was provided in the Chapter 4. Employing the entire system network in the annotation process increases in difficulty as the delicacy increases due to the time needed to perform the task (McEnery et al. 2006: 33). The challenge of providing delicate (or fine-grained) corpus annotations using large if not the entire extent of a system network still has to be addressed in the SFL community.

10.1.2 OCD corpus

The OCD corpus was created by Ela Oren and myself during a two week scientific mission at the psychology faculty of the Tel-Aviv university. The aim of the mission was to design an empirical evaluation for the Parsimonious Vole parser (at that time still in development) using texts consisting of self reports on the challenge of overcoming Obsessive Compulsive Disorder (OCD).

My role was to offer technical support for the annotation tool (UAM Corpus Tool), to prepare the annotation guidelines (provided in Appendix H), present relevant literature and support with the annotation process; Ela's was to select the suitable texts and perform the annotation. The texts represent blog articles of people diagnosed with (OCD), who self-report on the challenge of overcoming OCD. The annotations contain syntactic constituency elements and clause MOOD features. The corpus contains four texts comprising all together 529 clauses and 8605 words.

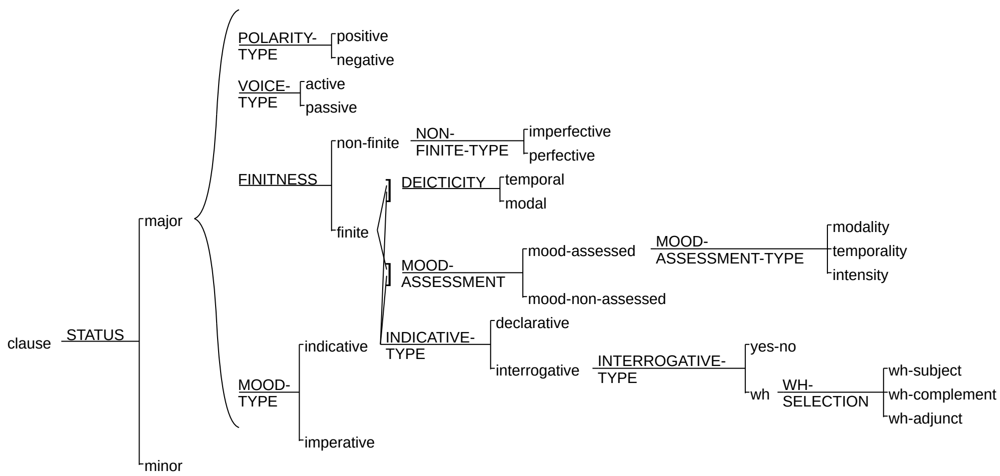


Fig. 10.2 The part of the MOOD system network that has been used in OCD corpus annotation

The corpus annotations account for the main unit classes and the main clause elements. The constituency annotation is based on the Cardiff grammar (Fawcett 2008) with some consulting of the traditional grammar (Quirk et al. 1985) for clarification. The MOOD systemic selections are based on the Sydney grammar (Halliday & Matthiessen 2013b) using the network fragment presented in Figure 10.2.

The systemic selections are restricted to the network fragment depicted in Figure 10.2. It is a sub-part of the MOOD system network supported by the Parsimonious Vole parser (described in Chapter 4 and depicted in Figure 4.1). Employing the entire system network in the annotations was difficult because, as the delicacy increases, the time spent for the annotation process increases drastically making the annotation process very tedious and slow (McEnery et al. 2006).

10.1.3 Differences between corpus annotation and parser output

This section describes the main known differences between how the parser structures output and the methodology used to annotate each of the corpora. These differences are mainly due to text normalisation and different treatment of conjunctions and punctuation. In OCD corpus there are also some segmentation errors described below.

The Parsimonious Vole parser first normalises the input text before further processing it. In this process the tab characters and extra spaces between words and are reduced and special characters such as quotes, parenthesis, dashes and other orthographic characters are re-represented in a uniform way. In the OE corpus most of the text is uniformly formatted but there are few deviations.

Listing 10.3 Sample of non-normalised raw text from the corpus

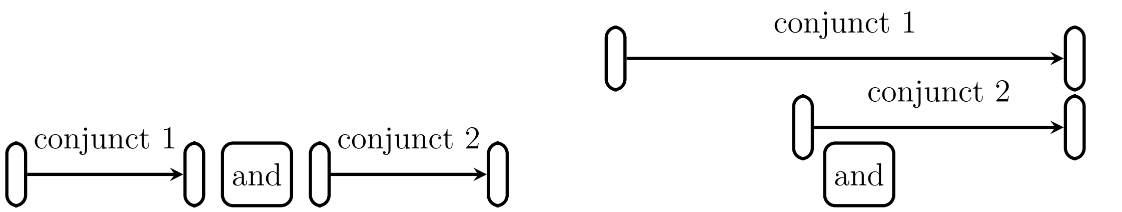
```
0 Red riding hood excerpt 24
25 "What have you in that basket,   Little Red Riding Hood?" 82
83
84 "Eggs and butter and cake, Mr. Wolf." 111
```

Listing 10.3 presents an example raw text from the annotation dataset containing an initial title line and two sentences separated by an empty line. The greyed index numbers at the beginning and end of each line indicate character offsets. In OE corpus files, the first line plays the role of a header containing the title or the file name and not considered for annotation or parsing. The extra spaces are eliminated during the normalisation process causing an index shift.

Before the annotation of the OCD corpus started the normalisation was not at all addressed and so the text contains some irregularities. Mostly it is organised as one sentence per line, but there are instances of extra blank lines or missing new line characters leading to few sentences per line as a block. The text may also contain tabs and extra blank spaces or blank lines as in Listing 10.3 at index [82,84].

It is noteworthy to mention that there are segmentation errors in a few cases from the OCD corpus. Some segments are either shifted and include the adjacent spaces (e.g. “ getting this push” instead of “getting this push”) or, the converse, leave out one or two characters of a marginal word (e.g. “the balanc” instead of “the balance”). Such

In the OCD and OE corpus annotations, the punctuation marks such as commas, semicolons, three dots and full stops are not included in the constituent segments while the parser includes them at the end of each adjacent segment. An example of such case was provided in Listings 10.1 and 10.2 in the beginning of this chapter.



(a) Conjuncts annotated as parallel segments

(b) Conjuncts annotated as subsumed segments

Fig. 10.3 Treatment of conjunctions in the corpus compared to the parser

The treatment of conjunctions that was discussed in Section 3.4.6 differs as well. In the corpus, the conjunctions (such as “and”, “but”, “so”, etc.) are excluded from the conjunct segments; they are considered markers in the clause/group complexes rather than part of the constituent. The parser, on the other hand, includes the conjunctions into the succeeding adjacent segment. For example, in the corpus we find the segment “forced me into treatment” while the parser the parser produces a slightly larger segment “and forced me into treatment.” that includes the conjunction at the beginning and the full-stop at the end.

Moreover the spans of the conjunct segments differ as well due to difference in treatment. Instead of being analysed in parallel, having sibling status as depicted in Figure 10.3a, the parse generated conjunct segments are subsumed in a cascade from the former to the latter as depicted in Figure 10.3b. This is one of the main reasons for long distances between the matched segments when “conjunct 1” in Figure 10.3a is matched to its counter part “conjunct 1” from Figure 10.3b.

In this section were mentioned the most important aspects by which the corpora annotations and parser output diverge. The evaluation methodology has been developed to take segmentation discrepancies into consideration in order to provide a wider evaluation ground to the systemic associations. Next section explains how this is done.

10.2 Evaluation methodology

This section explains the design of the current evaluation and how it was conducted. I start by explaining how the corpus annotations and the parser output are brought to a common representation as batches of mono-labelled segments. Then I discuss the method to compare the batches of segments in order to find (exact and partial) matches between segments. The matches mean that the parser has generated the same output as in the corpus annotations that is considered correct. The number of matched and that of non-matched segments is counted for each feature as part of the evaluation data. More details on the method, the matching algorithm and and different types of distance measurement that help dealing with partial matches in a controlled manner are presented in Section 10.2.3.

10.2.1 Corpus annotations as a set of mono-labelled segments

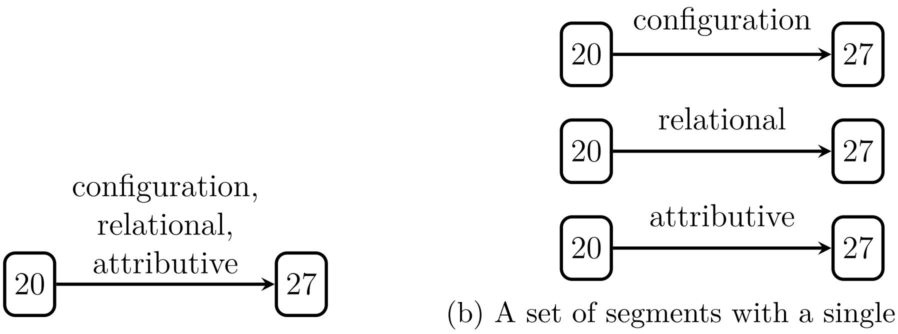
To compare the segment labels and boundaries we need to understand how they are represented in the annotations and parser output and how they can be brought to a common form comparison. This section explains the UAM Corpus tool representation of the corpus annotations and how are they treated for the purpose of this evaluation.

Both OCD and OE corpora annotations were created with the UAM Corpus Tool (O'Donnell 2008a,b) version 2.4. They are recorded as segments spanning in the text file from a start to an end index position and the set of features (selected from a systemic network) attributed to that segment. There are no constituency or dependency relations between segments. The example XML representation of an annotation segment is provided in Listing 10.4. 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, and the *features* attribute represents all the systemic features attached to this segment (as labels).

Listing 10.4 Segment example in UAM corpus tool

```
&lt;segment id="4" start="20" end="27"
features="configuration;relational;attributive"
state="active"/&gt;
```

In the current evaluation, the segments are constrained to carry only one label each. This means that the representation employed by the UAM corpus tool, with multiple labels per segments, as depicted in Figure 10.4a, is not suitable as such and needs an adaptation. When it is read from the disk, the segments with multiple features are



(a) A segment with a set of features

(b) A set of segments with a single feature each

Fig. 10.4 Example of breaking down a segment with multiple features into multiple segments with a single feature

broken down into multiple segments (over the same index span) for each feature in the original segment, as depicted in Figure 10.4b. Doing so permits the evaluation to focus on one or a set of features by freely and conveniently selecting only the segments that contain exactly those features.

The representation of the parser output, to be suitable in the current evaluation, needs a similar adaptation. In the next section I present how it is done.

10.2.2 Parser output set of mono-labelled segments

In order to compare the parser generated output to the corpus segments they need to be turned into the same form. In this section I describe the task of turning rich constituency graphs (CG) into labelled segments similar to those from the corpus presented in the previous section.

To make the parser output segments comparable to the ones in the corpus they need to refer, in terms of their offsets and indexes, to the same raw text. This is not the case for two reasons. First, the parser processes one sentence at a time. This means that the original input text is chunked based on punctuation and the indexes are reset for each sentence. Correspondingly the parser output is generated with respect to the new indexes. Second, the text after being chunked is normalised. This means that the word indexes are readjusted within the sentence text and is directly reflected in the parser output. Before the evaluation can take place the parser output segments need to be re-indexed with respect to the original raw text.

To fulfil this task, the text processed by the parser is re-indexed back into the original raw text at the level of words (tokens), constituents and sentences. Algorithm 16 provides pseudo-code of the re-indexing process.

Algorithm 16: Sentence level re-indexing of CG according to the raw text

```
input  : CG bundle, text
begin
    offset ← 0
    for cg in CG bundle:
        generate segments for cg indexed on text given the offset
        offset ← the end of cg
end
```

In Section 8.3 was explained that the parser processes one sentence at the time. If more than one sentence is provided as input text, then the generated output is not only one but a set of constituency graphs. This is reflected in the input for Algorithm 16 which is the array of CGs produced by the parser and the original text. The result of this algorithm is a set of mono-labelled segments indexed according to the raw text, just as the corpus annotations.

This task is performed by sequentially iterating the array of output constituency graphs and re-indexing each of them with respect to the offset calculated after the re-indexing its predecessor. The process of re-indexing a CG structure is presented in Algorithm 17.

The way each CG is re-indexed is described by Algorithm 17. The returned result is a set of segments from the constituency graph considering a given offset.

In Algorithm 17, the indexing is performed first at the word (token) level. At the same time the mono-labelled segments are generated for each token. Then the CG constituent nodes that are group or clause rank are re-indexed based on constituents below them i.e. words that have just been re-indexed in the step before. The indexes of the constituent segments are set to be the beginning of the first word and the end of the last word. The labels assigned to the segments are the constituent unit class, function(s) and all the systemic features. As the segments can carry a single label only then for every feature, function and unit class a new segment is created. This is in line with the practice described above concerning usage of mono-labelled segments.

Now that I have shown how the parser output is aligned to corpus original text represented as a set of mono-labelled segments, it is possible now to compare the parser output and the corpus annotations in order to asses parser accuracy. The next section explains how this is done.

Algorithm 17: Constituent level re-indexing at the level of constituents according to the raw text

```
input  : cg, text, sentence offset
begin
    words ← get cg the list of words
    for word in list of sentence word segments:
        find the word in the text after a given sentence offset
        if word found:
            start ← get first word start index
            end ← get the last word end index
            create a new segment (start, end, word)
        else:
            generate a warning (manual adjustment needed)
    for node in cg in BFS postorder:
        find the word span of the constituent
        start ← get first word start index
        end ← get the last word end index
        labels ← get node class, function and features
        create new segment (start, end, labels)
    return set of segments
end
```

10.2.3 Segment alignment method and evaluation data

Both the corpus annotations and the parser output are represented as a set of mono-labelled segments on the raw corpus text and can be compared to evaluate the parser accuracy. This section explains how this comparison is done. I first present a strict method of evaluation and then introduce a more permissive/tolerant method of evaluation based on segment similarity and distance measurements.

First, a straight forwards evaluation method is to check for a perfect match between every segment in the parser output and a corresponding segment in the corpus annotations. A perfect match means that each parser segment has a counter part in the a corpus annotations whose start index, end index and label are the same. This signifies that the parser generated segment is correct.

Using this method of evaluation the resulting data contain the counts of (a) how many segments with the same label are matching, (b) how many corpus segments are not matching and (c) how many parser segments are left unmatched. Then the data are aggregated for each label in the corpus annotation and parser output combined.

In the Section 10.1.3 I presented some differences between the parser output and the corpus annotations. Most of these differences are comparable especially that they manifest as slight variations in the segment spans, i.e. shifted start and/or end segment index, while the segment labels are exactly the same. The above presented method is not satisfactory for the current evaluation because by design it leaves out about (20%) of valuable partial matches. I present next an alternative evaluation approach but I address how the differences between segments can be judged in an objective controlled manner.

A simple metric for the difference between the segments taking into account their start and end indexes is that of *geometric distance*. For two segments $S(start_S, end_S)$ and $T(start_T, end_T)$ the geometric distance is defined in Equation 10.1. We can replace the difference between start and end indexes with Δ_{start} and Δ_{end} notation and obtain the reduced form provided in Equation 10.2. This distance metric penalises errors in proportion to the length difference and the shift between segments.

$$d = \sqrt{(start_S - start_T)^2 + (end_S - end_T)^2} \tag{10.1}$$

$$d = \sqrt{\Delta_{start}^2 + \Delta_{end}^2} \tag{10.2}$$

Accounting for differences in the segment spans is a well known task in the mainstream computational linguistics called *text segmentation evaluation*. A variety of segmentation evaluation metrics have been proposed among which the most known are P_k (Beeferman et al. 1999: 198–200), *WindowDiff* (Pevzner & Hearst 2002: 10), *Segmentation Similarity* (Fournier & Inkpen 2012: 154-156) and *Boundary Edit Distance* (Fournier 2013). Each of these metrics have been shown to have strengths and flaws. For exampel both P_k and $WindowssDiff$ under-penalise errors (Lamprier et al. 2007) and have a bias towards favouring segmentation with few or tightly-clustered boundaries (Niekrasz & Moore 2010) while segmentation similarity tends to overly optimistic values due to its normalisation (Fournier 2013). Based on the evaluation data, these distances will be further discussed in Section 10.3.1.

Now that the metric for comparing segments have been introduced I move on to present the second evaluation method, which, in addition to accounting for the exact matches, accounts for close matches between segments using the distance measurements presented above.

The second evaluation method is to align two sets segments taking into consideration exact and partially matching pairs. This task is similar to the well know problem in computer science called *stable marriage problem* (Gusfield & Irving 1989). I adopt here the computer science framing of the problem to explain the second evaluation method.

The standard enunciation of the stable marriage 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 manually created one. The distance measures considered here have been introduced above. This matrix represents the complete and fully ordered set of preferences stipulated in the original problem formulation. In addition to having identical offsets, the segments need to carry the same labels in order to be considered a match. This condition is not expressed in the original problem but is integrated into the matching algorithm.

Algorithm 18 represents a modified version of Gale-Shapley algorithm. In the process each parser segments is attempted to match with closest corpus segment. A match is established if there is not other parser segment that is even closer to the candidate corpus segment. The left corpus and parser segments are marked as non-matching.

Using this method of evaluation we can count for every distinct label (a) how many segments match perfectly, i.e. the distance is zero, (b) how many segments

Algorithm 18: The algorithm for matching parser and corpus segments

```
input   : parser segments, corpus segments
begin
    mark all parser segments and corpus segments free
    compute distances from each corpus segments to every parser segments
    while exist free segments in parser segments:
        parser segment ← first free segment from parser segments
        if exist in corpus segments un-compared segment to parser segment:
            corpus segment ← the nearest among corpus segments to parser segment
              with identical label
            if corpus segment is free:
                match parser segment and corpus segment
                mark parser segment and corpus segment as non-free
            else:
                parser segment′ ← the current match of corpus segment
                if parser segment is closer to corpus segment than parser segment′:
                    match parser segment and corpus segment
                    mark parser segment and corpus segment as non-free
                    mark parser segment′ as free
        else:
            mark parser segment as non-free and non-matching
    fintq
end
```

partially match, i.e. the distance is greater than zero, (c) how many corpus segments are unmatched and (d) how many parser segments are unmatched. This way we get a four frequency counting numbers per label for all labels used in the corpus annotation and parser output combined. These frequency numbers are used to compute parser accuracy metrics. Further more the partial matches can be analysed to estimate the degree of the deviation and derive insights what can be done about it. Having presented the evaluation method, I further proceed with presenting the evaluation results in two parts: first part focusing on the syntagmatic aspects and second part dealing with paradigmatic aspects of the parser output.

10.3 Evaluation of syntactic structure generation

In this section I present the evaluation results for the parser accuracy to generate the syntagmatic aspects of an SFL analysis (described in Chapter 8). The discussion of evaluation results covers the accuracy of parser segmentation, the distribution of distances among the matched segments, the accuracy to detect the main unit classes and clause main Mood and Transitivity elements.

10.3.1 Segmentation evaluation

This section presents the evaluation data on text segmentation. As we will see below, over 60% of the parser segments coincide with the corpus segments, but not all of them. The differences, which were described in Section 10.1.3, are mainly due to differences in annotation approach, text normalisation and some segmentation errors in the OCD annotations (e.g. missing or including some extra characters).

The segmentation counts are provided in Table 10.2. The columns represent the number of matched segments, the segments from the corpus that have not been matched and the parser output segments left unmatched. There are over 6500 segments that perfectly align and over 11000 segments that align partially or completely.

	Matched	Corpus non-matched	Parser non-matched
Exact matches only	6665	1319	4332
Exact and close matches	11073	1319	4332

Table 10.2 Count statistics of matched and non-matched segments

	Precision	Recall	F1
Exact matches only	0.61	0.83	0.71
Exact and close matches	0.72	0.89	0.80

Table 10.3 Segmentation accuracy

The statistics provided in Table 10.2 translate into precision, recall and F_1 scores as provided in Table 10.3. The parser segmentation accuracy is 71% for exactly matching segments and 80% for partially matching ones. The distances between segments are measured in several ways. Next follows an analysis of these measurements.

The segmentation differences are measured, as introduced in Section 10.2, using a few distance metrics: (a) geometric (Euclidean) distance, (b) edit (Levenshtein), (c) generalised hamming distance (GHD) (Bookstein et al. 2002), P_k (Beeferman et al. 1999: 198–200), *WindowDiff* (Pevzner & Hearst 2002: 10). The data are calculated on a number of over 12500 segment pairs out of which 62% are exact matches and 38% are close matches.

Before providing the interpretations to the data I first show it is sufficient to discuss two of these distances as the other ones are strongly correlated to them and thus can be omitted from the discussion. Table 10.4 represents Pearson correlation matrix for the distance types.

	Levinstein	Geometric	Generalised Hamming	Pk	WindowDiff
Levinstein	1.00	0.99	0.53	0.45	0.54
Geometric	0.99	1.00	0.52	0.45	0.54
Generalised Hamming	0.53	0.52	1.00	0.84	0.92
Pk	0.45	0.45	0.84	1.00	0.92
WindowDiff	0.54	0.54	0.92	0.92	1.00

Table 10.4 Pearson correlation coefficients for pairs of distance measure types

The Pearson correlation coefficient measures the direction and strength of a linear correlation between two variables. In this case, the variables are distance types. The standard interpretation for this coefficient is as follows. A coefficient value between 0.1 and 0.3 indicated a weak linear relationship between the variables. If it is between 0.3 and 0.5 the relation is moderate, between 0.5 and 0.7 it is strong; and a score over 0.7 indicates a very strong correlation of variables.

Using the above interpretation we can partition the correlation matrix into two groups of strongly correlated variables. At the same time both groups are only moderately (50%) correlated to each other. The Levinstein and Geometric distances are nearly identical with a 99% correlation though from this point on I will exclude the Levinstein distance and employ the geometric distance only as representative for both.

The generalised Hamming distance, P_k and WindowDiff bear a strong correlation of 92% to WindowDiff. I exclude, from now on, P_k and generalised hamming distance from further discussions and use WindowDiff as the representative of the three distances. This choice is also based on the fact that WindowDiff was proposed to overcome weaknesses of P_k (Pevzner & Hearst 2002: 10).

Above, I have shown that the geometric distance and WindowDiff distance are the significant measures of distance in this segmentation evaluation. Next I provide descriptive data (in Table 10.5) and offer interpretations for each of them.

Table 10.5 presents the descriptive statistics (columns) for every distance type (rows). The first three columns provide mean, max and average values. The *std* column means *standard deviation* (σ) while the *relative std* represents *relative standard*

	Min	Max	Mean	Std	Relative std	Skew	Kurtosis
Levinstein	0.00	219.00	5.37	15.99	2.98	5.57	42.63
Geometric	0.00	219.00	4.99	14.67	2.94	5.56	43.14
Generalised Hamming	0.00	8.00	1.84	2.87	1.56	1.30	0.09
Pk	0.00	1.00	0.18	0.29	1.62	1.40	0.57
WindowDiff	0.00	0.86	0.15	0.23	1.59	1.32	0.28

Table 10.5 Descriptive statistics for each set of distance measurements between corpus and parser segments

deviation (or coefficient of variation). The Skew and Kurtosis are the last statistical indicators for describing the distance distributions.

The relative standard deviation is the ratio between the standard deviation and mean value (μ), i.e. (σ/μ) and measure how concentrated the data are around the mean: the more concentrated, the smaller the standard deviation. It is considered that a relative standard deviation between 0 and 0.5 indicates tightly clustered data around the mean; if it is situated between 0.5 and 1 then it means the data are more spread out; if, however the value is over one then it means the data are very scattered.

Besides the mean and (relative) standard deviation, *Skew* and *Kurtosis* are valuable statistical indicators for describing a distribution. Skewness measures the asymmetry of the bell of the normal distribution where skewness greater than 1 indicates that the data are highly skewed to the right, i.e that there are rare data points situated in a long tail. Kurtosis measures the outliers present in the distribution. Kurtosis smaller than the threshold of 3 indicates that the data has light tails or lack of outliers, whereas a value greater than 3 indicates heavy tail and requires additional investigation as it may indicate among others wrong data.

In the current evaluation, the geometric distance between corpus and parser segments spans from a minimum 0 to maximum 219 characters. The mean distance is 4.99, which is close to the minimum point, with a standard deviation of 14.67, which, in relative terms, indicates an extreme deviation of 290%. The skew over 1, in this case 5.56, indicates a strong asymmetry to the right, and the kurtosis of 43.14 (almost 15 times the threshold of 3) indicates that most of the data, about 80%, gravitate towards the left, between 0 and slightly over the mean, while the rest of the data point continue into a very long tail to the right. This is depicted in Figure 10.5.

As appears in Figure 10.5, the data do not follow a normal but a power law distribution (Newman 2005). This means that 62% of the segments are not shifted at

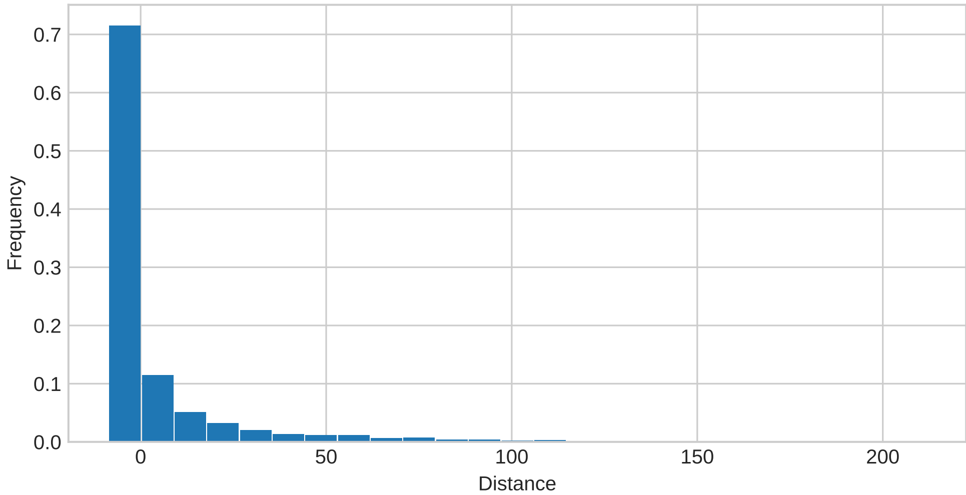


Fig. 10.5 Matched segments geometric distance distribution histogram (binning=25)

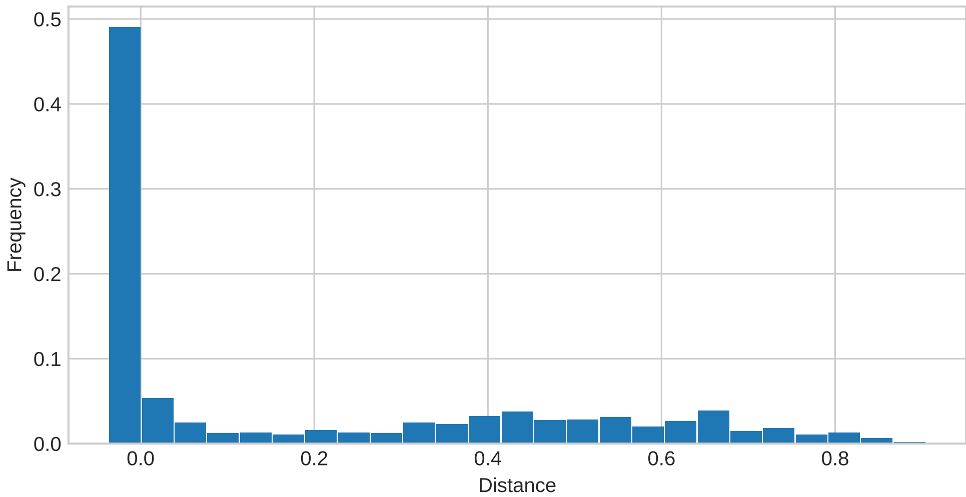


Fig. 10.6 Matched segments WindowDiff distance distribution histogram (binning=25)

all and 83% of the segments are slightly shifted up to 5 characters. There rest (17%) of the segments are shifted by more than 5 characters. These ratios approximate Patetto's 80-20 distribution law but may as well fit Zipf's law (Newman 2005). In future work, properties of these data should further be analysed, including the distribution fitting, that is selection of the theoretical distribution that fits best this dataset.

Next distance that I discuss is WindowDiff distance between corpus and parser segments. One crucial difference to geometric distance is the normalisation to [0,1] interval. In the current evaluation, the WindowDiff distance distribution, depicted in Figure 10.6, spans from a minimum 0 to maximum score of 0.86. The mean distance is 0.15 with a standard deviation of 0.23. The mean value is close to the minimum point, the relative standard deviation indicates extreme deviations of 159%, the skew over 1 indicates a strong asymmetry to the right, and the kurtosis of 0.28 indicate the distribution does not have many outliers in the tail. These parameter values are similar to those pof the geometric distance but to a lesser degree.

One positive aspect of WindowDiff distance distribution is that the tail is not so long due to its normalised structure. This results in aggregation of the outliers we have observed in geometric distance distribution into a compact spectrum. This way, diminishing the kurtosis below 3, which no longer indicates an abnormally long tail of outliers.

The histogram in Figure 10.6 also resembles a power law distribution and more analysis work needs to be done in the future, including the distribution fitting and relation to the causes of partial matches in the first place.

This section presented the segmentation evaluation. The data show that the parser generates exact segments as provided in the corpus with an accuracy of 0.71 F_1 score, and segments that partially correspond to those in the corpus with an accuracy of 0.8 F_1 score. The distances of the partially matched segments, in about 83% of the cases do not exceed 5 characters, but can span, most probably by mistake, over 200 characters in less than 17% of cases. Next I move on to present the evaluation of the segment label assignments, which in our case are unit classes and functions.

10.3.2 Unit class evaluation

In this and the next section I present the parser syntactic accuracy. It aims to measure, in this section, how well the main unit classes, and in the next sections, the clause main elements have been generated by the parser compared to the corpus. The unit class evaluation is performed on the OCD corpus. This evaluation comprises the following unit classes: clause and nominal, prepositional, adverbial and adjectival groups. No clause complexes, group complexes or word types are included. The evaluation data are depicted in Figure 10.7. The names of the unit classes are provided on the x axis at the bottom of the graph while on the y axis the absolute number of occurrences is provided.

The meaning of exact and close match has been explained in Section 10.3.1 and from now on the label “Matched” will mean the segments that are either exactly or closely matched all together, while the label with a remark “(exact only)” means that it applies to only the portion of the exactly matched segments.

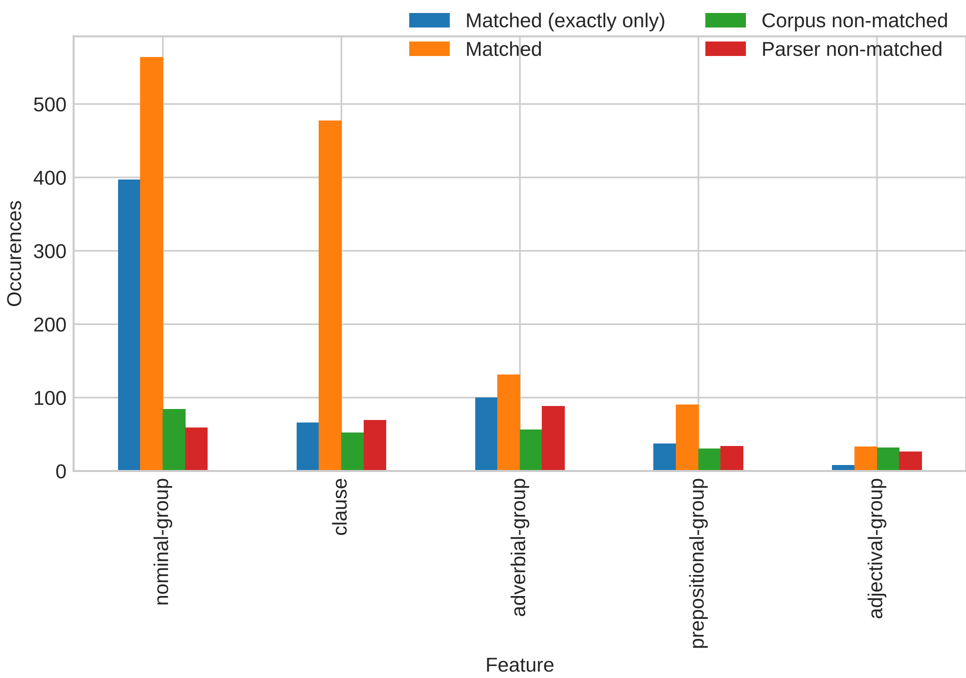


Fig. 10.7 Bar chart of matched and non-matched segments for the main unit classes

To make the data easier to read and interpret I present the evaluation data in a table form using relative values to the number of matched segments. The absolute values of the evaluation statistics are contained in the graphical form and also available in the appendices in the table form. The relative evaluation data in this and the following sections will be presented in tables with the same structure. Using Table 10.6 as example, the column meaning is as follows. The first column contains the name of the unit class, element or feature. The column "Matched" contains the absolute number of matched segments with a specific label. The three other columns represent the number of segments relative to the "Matched" ones. So the column "(%) Matched (exactly only)" means that out of all the matched segments that many represent exact matches and the rest up to 100% are partially matched segments. The column "(%) Corpus non-matched" represent the number of segments relative to the total number of segments of particular type in the corpus, which remain unmatched. The column signifies the fraction of segments that remain unmatched, while the rest up to 100% have, each, a corresponded in the parser output. The column "(%) Parser non-matched" represents the number of segments (relative to the total number of segments of particular type in the parser output) in the parser output that do not have a correspondent in the corpus.

	Matched	(%) Matched (exactly only)	(%) Corpus non-matched	(%) Parser non-matched
nominal-group	564.00	70.39	12.96	9.47
clause	477.00	13.84	9.83	12.64
adverbial-group	131.00	76.34	29.95	40.18
prepositional-group	90.00	41.11	25.00	27.42
adjectival-group	33.00	24.24	49.23	44.07

Table 10.6 The evaluation statistics relative to the number of matched segments for the main unit classes

The evaluation data from Table 10.6 indicate that most (over 70%) of the nominal and adverbial groups are identified with exact same borders as in the corpus while clause borders exhibit the most disagreement reflected by their low score of exact matches, only 13.84%. The proportion of unmatched unit class segments in both the corpus and parser output varies between 9% for clauses and nominal groups, and over 40% for adjectival and adverbial groups. These proportions, however, are better interpreted when they are embedded into precision and recall score, which are provided in Table 10.7.

	Exact match only			Exact and close match		
	Precision	Recall	F1	Precision	Recall	F1
nominal-group	0.87	0.83	0.85	0.91	0.87	0.89
adverbial-group	0.53	0.64	0.58	0.87	0.90	0.89
prepositional-group	0.52	0.55	0.54	0.73	0.75	0.74
clause	0.49	0.56	0.52	0.60	0.70	0.65
adjectival-group	0.24	0.20	0.22	0.56	0.51	0.53

Table 10.7 Parser accuracy statistics for for the main unit classes

The scores provided in the first three columns are calculated with respect to exact matches only and the last three columns with respect to all the matches. This can be seen reflected in the lower precision, recall and F_1 scores when compared to their correspondents in the last three columns. The exact match scores, nonetheless, constitute an appropriate baseline for the parser. Note that in all cases of a match, close or exact, the segments bear the same label, and so, as explained in Section 10.1.3, the source of the divergence is mainly in the segment index span.

The last three columns in Table 10.7 show that clause and normal group units are identified with almost 0.9 F_1 measure, which is an encouraging result, while the adjectival and adverbial groups score 0.53 and 0.65 indicates that there is some space for improvement. Further investigation is needed to discover the reason for the lower scores as there seem to be no obvious cause other than corpus and/or parser errors. Also, as visible in Figure 10.7, there is a contrast in the number of segments between the first two unit types and the last three with a ratio of one to four or more. The lower number of exemplars (close to and below 100) in this evaluation contributes, to a certain extent, to the lower accuracy statistics.

10.3.3 Clause Mood elements evaluation

In this section I describe the evaluation statistics reflecting the parser capacity to generate the main elements of a clause. The data available in the corpus unfortunately does not permit to evaluate elements of the lower rank units such as nominal, prepositional, adjectival and other groups. Next I present statistics on the clause Mood elements, which were described in Chapter 4. These elements are present in the annotations of the OCD corpus as explained in the beginning of this chapter in Section 10.1.

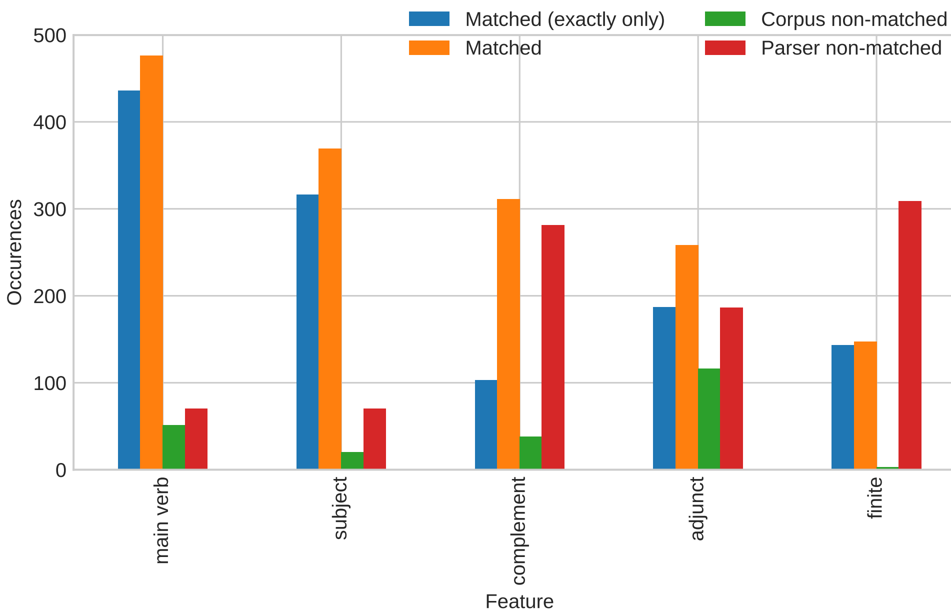


Fig. 10.8 Bar chart of matched and non-matched segments for the clause main Mood elements

The OCD corpus annotations provide with the main syntactic (Mood) elements in the clause. Some of them, such as Auxiliary verbs, Main verb extension, Negation particle, and others have been omitted in the corpus and are thus missing in the present evaluation. Figure 10.8 reflects the absolute values from the empirical data.

The parser accuracy measurements for the main syntactic functions are contained in Table 10.8. The F_1 score for subjects and main verbs is nearly 0.9 while the complements and adjuncts are over 0.6. Finite element scores nearly 0.5 which is a surprisingly low for this element. The reason for it lays in the incomplete corpus annotations. In the annotation process the conflated Finite and Main verb elements were not distinguished. So when there was a main verb that was also a finite, only the main verb function was marked, which is incomplete by SFL standards. This incompleteness is clearly reflected in the contrast between low precision of 0.32 and high recall of 0.98.

	Exact match only			Exact and close match		
	Precision	Recall	F1	Precision	Recall	F1
main verb	0.86	0.90	0.88	0.87	0.90	0.89
subject	0.82	0.94	0.88	0.84	0.95	0.89
complement	0.27	0.73	0.39	0.53	0.89	0.66
adjunct	0.50	0.62	0.55	0.58	0.69	0.63
finite	0.32	0.98	0.48	0.32	0.98	0.49

Table 10.8 Parser accuracy statistics for the clause main Mood elements

The number of complements unmatched in the parser output is nearly the same as the number of matched complements. This is reflected in the 0.53 precision score and nearly 0.9 recall rate which overall lead to an F_1 score lower than that of subject and main verb elements. This can be explained by a flaw, mentioned in Section 10.1, in the annotation methodology as follows. The clausal complements often were annotated as a new clauses omitting to draw the same segment and marking it to be a complement in the clause above. This required the corpus revision and correction. Adjuncts however have a higher number of unmatched segments on both sides and this may be due to bugs in the parser and other mistake or omissions in the corpus.

10.3.4 Clause Transitivity elements evaluation

OE corpus, provides with elements of Transitivity parsing described in Chapter **??**. The elements employed in this evaluation are Configuration, Participant role and Main verb while Circumstances are excluded from the study. Figure 10.9 presents the evaluation data.

The configuration segments, in SFG, correspond to clause segments, the participant role segments have as correspondents either the subject or complement segments, while the Main verb segments are shared. The aggregation of Subjects and Complements can be observed in Figure 10.9 where the number of participant roles is approximately double the number of configurations. A configuration can have between one and three participants, and current data show an average of two participants per clause.

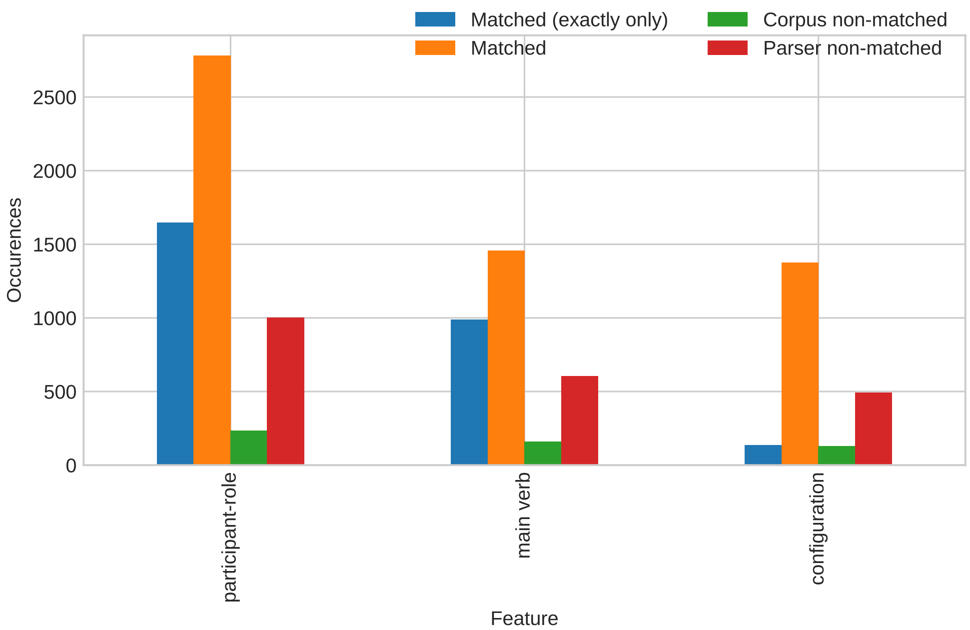


Fig. 10.9 Bar chart of matched and non-matched segments for the clause main Transitivity elements

Note that in Figure 10.9 the scale stretches over 2500 which reflects a much larger number of segments in OE corpus than that available in the OCD corpus. The size, and certainly a higher quality of annotations, is reflected in the fairly uniform evaluation results. The F_1 (0.82), precision (0.74) and recall (0.92) scores vary very little across elements when compared to scores of the syntactic elements shown in Table 10.8, which vary substantially from one element to another.

	Exact match only			Exact and close match		
	Precision	Recall	F1	Precision	Recall	F1
participant-role	0.62	0.88	0.73	0.74	0.92	0.82
configuration	0.22	0.52	0.30	0.74	0.92	0.82
main verb	0.62	0.86	0.72	0.71	0.90	0.79

Table 10.9 Parser accuracy statistics for the clause main Mood elements

In case of exact matches the evaluation scores are lower for the participant roles and main verbs, situating at 0.73 F_1 score, while the configuration accuracy plummeted to 30%. As configurations correspond in SFG to clauses boundary establishment methodology plays a significant role in achieving exact matches. Thus the discrepancy in the F_1 score between exact and combined matches is explainable by a discrepancy in the clause boundaries establishment and was already addressed in Section 10.1.3.

We come to the end of the syntactic structure evaluation where I have shown that the parser generated segments exactly like those in the corpus with an accuracy of 70% on average; and partially matching segments with an average accuracy of 80%. The parser detects unit classes on average with 52% accuracy for exact matches and 74% for close matches. The Mood and Transitivity elements are detected on average with 71% and 81% accuracy.

A constituency parser that generates a syntactic analysis using comparable unit classes and functions (using phrase structure grammars) such as for example Chen & Manning (2014), Stern et al. (2017) or Kitaev & Klein (2018) reach an accuracy of 95% for English. This state of the art in parsing with other grammars reflects that there is a large space to improve the accuracy of Parsimonious Vole constituency. But it should not be separated from the context of this work which is to parse with constituency structures enriched with features from the system networks. Next I present the evaluation of the the systemic feature selections from the MOOD and from the TRANSITIVITY system networks.

10.4 Evaluation of systemic feature assignment

In this section I present the evaluation results for the parser accuracy to generate the paradigmatic aspects of an SFL analysis. It constitutes an evaluation of the approach described in Chapter **??**. The discussion of evaluation results covers selection of MOOD and TRANSITIVITY features.

In this section the differentiation between exact and close matches is no considered. The reason for this is the nature of the feature selection and assignment task, which is concerned with the paradigmatic aspect of grammar. The systemic features are assigned to already formed constituent units and is not directly affected by the segmentation errors relevant at the constituent creation.

10.4.1 Evaluation of MOOD systemic feature assignment

In this section I present the evaluation results for the systemic selections from the MOOD system network that were assigned to clause units in the constituency structure. The OCD corpus provides with MOOD annotations considered in this evaluation. The fragment of the MOOD system network that is covered by the corpus was presented in Section 10.1. It is noteworthy to mention that the parser provides with more feature selections from the MOOD system network, as described in Section 4.2.1, and that the current evaluation is limited to only what is available in the corpus.

In Table I.6 are provided the evaluation results for each of the MOOD features grouped by system names, which are marked with capital letters. On average the parser assigns systemic features with a precision of 59%. I do not discuss here the evaluation of each feature in part but analyse the results as a whole taking a few systems as discussion examples. A detailed discussion addressing each system in part and aiming at deeper evaluation understanding should be tackled in future work.

The order in which the features appear in Table I.6 roughly correspond to an increase in systemic delicacy. As delicacy increases there are increasingly fewer occurrences where a system is employed. This is associated also with a decrease in the accuracy although there are multiple factors influencing it among which parser errors, corpus quality and small population size.

The precision and recall values vary quite a lot from a minimum of 11% up to a maximum of 93% and their harmonic mean, the F_1 score, between 30% and 87% averaging to almost 60%. The details can be read in Table 10.11. The graphical representation of these values distribution can be seen in Figures 10.10 and 10.11. A noticeable feature is the presence of two peaks in the precision and recall distributions: one around 50% and the other one around 90%. They translate into a similar F_1 distribution with peaks at 60% and 85%, a phenomena which I address next.

	Match	Corpus non-matched	Parser non-matched	Precision	Recall	F1
POLARITY-TYPE						
positive	485	125	55	0.90	0.80	0.84
negative	57	10	70	0.45	0.85	0.59
VOICE-TYPE						
active	553	102	68	0.89	0.84	0.87
passive	11	11	28	0.28	0.50	0.36
FINITNESS						
non-finite	99	19	38	0.72	0.84	0.78
finite	526	33	554	0.49	0.94	0.64
NON-FINITE-TYPE						
perfective	71	12	16	0.82	0.86	0.84
imperfective	26	9	24	0.52	0.74	0.61
DEICTICITY						
temporal	446	74	55	0.89	0.86	0.87
modal	12	33	6	0.67	0.27	0.38
MOOD-ASSESSMENT-TYPE						
temporality	35	17	27	0.56	0.67	0.61
modality	15	32	8	0.65	0.32	0.43
intensity	12	14	43	0.22	0.46	0.30
MOOD-TYPE						
indicative	455	216	37	0.92	0.68	0.78
imperative	4	1	31	0.11	0.80	0.20
INDICATIVE-TYPE						
declarative	355	260	27	0.93	0.58	0.71
interrogative	47	7	63	0.43	0.87	0.57
INTERROGATIVE-TYPE						
wh	40	6	57	0.41	0.87	0.56
yes-no	5	3	8	0.38	0.62	0.48
WH-SELECTION						
wh-subject	9	3	7	0.56	0.75	0.64
wh-adjunct	11	15	3	0.79	0.42	0.55
wh-complement	8	0	62	0.11	1.00	0.21

Table 10.10 The evaluation statistics available for the MOOD system network

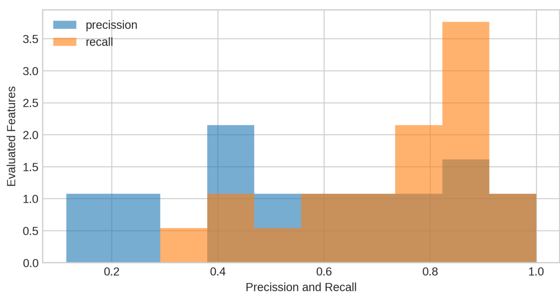


Fig. 10.10 The distribution of precision and recall for selected features from the MOOD system network

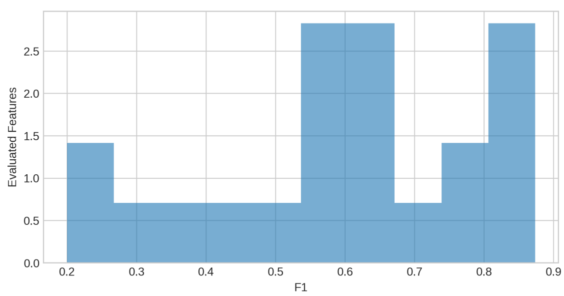


Fig. 10.11 The distribution of F1 score for selected features from the MOOD system network

Within most systems, the F_1 scores exhibit a contrast from one feature to the other. What this possibly mean is discussed in the next two cases. For example in the

	Precision	Recall	F1
mean	0.57	0.73	0.59
standard deviation	0.27	0.18	0.21
min value	0.11	0.32	0.20
25% quantile	0.41	0.62	0.48
50% quantile	0.56	0.80	0.61
75% quantile	0.82	0.86	0.78
max value	0.93	1.00	0.87

Table 10.11 Descriptive statistics of the precision, recall and F1 scores for evaluated MOOD features

POLARITY-TYPE system the positive polarity feature scores 84% accuracy while the negative one almost 60%. As per Definition 3.2.10, the system features are mutually exclusive. The polarity of an English clause is positive by default unless a negation marker is found and this represents only 10% of the clauses in the corpus. This leads us to the hypothesis that it should be sufficient for the parser to detect one feature with a reasonably high accuracy then the converse feature should be detectable with a similar accuracy if a selection in that system is expected. Yet the current data invalidate this hypothesis as the grammar does not represent exclusivity.

In the case of the POLARITY-TYPE system, the phenomena may be explained as an incomplete parser implementation. The current version of implementation determines polarity by checking for the presence of the negation verbal marker only without considering cases of nominal and adverbial negative markers. Nonetheless a more delicate polarity testing will have to take into consideration polarity indicators from the subject, complement and adjuncts of various types that have been taken into consideration during the annotation process. An incomplete, less delicate, implementation for systemic choices is certainly a source of errors but it requires further investigation to what extent it impacts these results.

Another way to look at the high discrepancy between the feature accuracy scores can be as follows. Considering POLARITY-TYPE as example, it might be the case that most instances of positive polarity are easy to detect. But, there is a portion of cases, regardless of the polarity, that are difficult for the parser to distinguish and that the negative polarity selections fall mostly within these ambiguous cases.

The phenomena of unbalanced accuracy scores among siblings of the same system can be seen in multiple other cases. Let's look at the VOICE-TYPE system. the detection mechanism for VOICE-TYPE is implemented similarly to POLARITY-TYPE.

The parser checks whether there is a passive order of elements in the clause, otherwise the active voice is selected. The detection of the active voice scores a significantly higher accuracy of 87% than the passive one of only 36%. There is no delicacy variation problem and still the discrepancy between the F_1 scores of the two features is there. But this could be explained by small sample of passive voice instances: only 22 are annotated in the corpus.

This section presented the evaluation results for the MOOD systemic selections. On average parser assigns to clause constituents MOOD features with an accuracy of almost 60%. A more thorough analysis for each system in part would indicate how to improve the current parser. At the same time the quality of the OCD annotations has not been assesses and therefore a more reliable corpus with MOOD annotations is highly desirable for a similar evaluation. Next section provides the evaluation results for the TRANSITIVITY system network.

10.4.2 Evaluation of TRANSITIVITY systemic feature assignment

In this section I present the evaluation of the TRANSITIVITY system network. As was explained in Section 10.1 above, the OE corpus contains annotations covering only a fragment of the TRANSITIVITY system, which is depicted in Figure 10.1. The full Cardiff TRANSITIVITY system network was presented in Figure **??** in Chapter 4. The parser provides more feature selections than those available in the corpus, which are described in Section 4.2.2, but the current evaluation is limited to only what is available in the corpus.

Transitivity analysis is semantic in nature and poses challenges in meaning selection beyond constituent class or function. The approach of assigning this sort of features was explained in Chapter **??**. Here I would like to remind an important aspect of this approach that impacts the evaluation results, explaining in some cases abnormally high recall rates.

In the grammar, described in Chapter 4, a clause can be assigned a single process configuration and the participant constituents can take only one role each. In the semantic role labelling task the situation is similar: a clause takes one semantic frame and each constituent only one semantic label. This means that a parser shall generate as output one semantic configuration that fits best the text.

The approach implemented into the Parsimonious Vole parser is such that it does not always provide a single semantic configuration. Instead it generates one or several possible configurations for each clause instead of providing exactly a one. The reason for this is the mechanism by which semantic analysis is generated. The constituency structure is tested against a set of semantic graph patterns and the matching patterns enrich the constituency structure with semantic features immediately. In some cases more than one pattern matches the constituency structure leading to enrichment by multiple graph patterns, which is not entirely correct.

Multiple features assignment from the same system, however, are represented as disjunctive sets in the constituency structure which ought to be interpreted as alternating possibilities rather than actual assignments. This interpretations was explained in Chapter 7 when the disjunctive sets were introduced.

Intuitively, this should reduce accuracy on all the elements but the effects are mostly manifested at the level of participant roles as will be described below. First let's discuss the evaluation results for the process types, which are provided in Table I.8, and after we will turn to the evaluation of participant roles.

	Match	Corpus non-matched	Parser non-matched	Precision	Recall	F1
PROCESS-TYPE						
mental	277	231	87	0.76	0.55	0.64
relational	338	297	174	0.66	0.53	0.59
influential	38	51	62	0.38	0.43	0.40
action	170	231	352	0.33	0.42	0.37
event-relating	1	28	0	1.00	0.03	0.07
RELATIONAL-TYPE						
attributive	169	239	107	0.61	0.41	0.49
directional	30	13	127	0.19	0.70	0.30
locational	39	20	207	0.16	0.66	0.26
matching	2	0	69	0.03	1.00	0.05
MENTAL-TYPE						
three-role-cognition	45	51	34	0.57	0.47	0.51
two-role-cognition	95	102	86	0.52	0.48	0.50
two-role-perception	13	12	102	0.11	0.52	0.19
three-role-perception	0	2	6			
desiderative	0	0	81			
emotive	0	0	87			

Table 10.12 The evaluation statistics available for the PROCESS-TYPE system and few of its subsystems from the TRANSITIVITY system network

In Table I.8 *mental* and *relational* processes are the ones with highest F_1 scores: 0.64 and 0.59. They are followed by *influential* and *action* process types while results for the *event-relating* are not conclusive because of the very small number of occurrences in the dataset. Note that considerable volume of annotations for process type sub-types are provided for mental and relational processes explained in Schulz (2015: 153-155).

Among the *mental* processes, *two-role-cognition* and *three-role-cognition* are parsed with highest accuracy of 51% and 50% correspondingly; whereas among *relational* ones the *attributive* process type scores the highest, 49% while rest of them score much lower. This can be seen also in the higher number of non-matched segments for each process type for every feature.

	Precision	Recall	F1
mean	0.35	0.48	0.36
standard deviation	0.32	0.26	0.19
min value	0.00	0.00	0.05
25% quantile	0.07	0.42	0.24
50% quantile	0.33	0.48	0.39
75% quantile	0.59	0.55	0.51
max value	1.00 (0.76)	1.00 (0.70)	0.64

Table 10.13 Descriptive statistics of the precision, recall and F1 scores for evaluated TRANSITIVITY features

Looking at the entire set of evaluation results for process types, the precision and recall values vary quite a lot from a minimum of 3% up to a maximum of 100% and the F_1 score, between 7% and 64% averaging to 41%. The summary of the descriptive statistics can be read in Table 10.13. The maximum of 100% precision is a bit unfortunate because there is one instance of the event-relating process found by the parser which also failed to find the other 28 thus the recall of 3% only. So I decided to ignore this value and use the next maximum which is 76% corresponding to the mental process types. A similar case of 100% is for recall of the matching process type which was provided only two times in the corpus but the parser generated 67 different instances of it. Therefore I consider the next maximum recall value, 70% for the directional process type.

Next I provide an analysis of the evaluation data indicating a potential relation between the increase in the delicacy and effect it has on the parser accuracy. The accuracy of mental process detection is 64% whereas the average accuracy for the mental sub-types (cognition, perception, desiderative and emotive) is 40%. The same holds for relational process whose accuracy is 59% whereas the average of its sub-types (attributive, directional, locational, matching) is only 26%. I start by comparing the number of mental and relational segments to the sum of mental sub-type segments and sum of relational sub-type segments correspondingly.

Features	Manual	Parse	/
mental	508	364	**0.72**
mental sub-types (sum of)	320	549	**1.72**
/	**0.63**	**1.51**	

Table 10.14 The ratios between *mental* segments and the sum of mental sub-type segments

Features	Manual	Parse	/
relational	635	512	**0.8**
relational sub-types (sum of)	512	750	**1.47**
/	**0.8**	**1.47**	

Table 10.15 The ratios between *relational* segments and the sum of mental sub-type segments

Table 10.14 and 10.15 contain four frequency counts and four ratios. The total number of segments available in the corpus and generated by the parser are provided by column and the features are provided in rows. On the bottom and right sides of the table two pairs of ratios between frequency numbers are provided.

Next I discuss only the case of the mental process type (see Table 10.14) because the values of the ratios are very similar and the same holds for relational processes. Perhaps this holds for other process types but we lack data for testing this hypothesis further.

The first pair of ratios, provided in the lowest row, compare the number of segments with *mental* feature to the sum of segments with any sub-type of *mental* feature (i.e. *cognition*, *perception*, *emotive*, etc.). This ratio measures how well are the feature dependencies preserved across delicacy levels. The second pair of ratios, provided in the last column, compares the number of segments provided by the parser and to that available in corpus for both the mental feature and the sum of its sub-types.

Table 10.14 shows that in the corpus the number of segments with *mental* feature is almost one fourth higher than what the parser provides (72 %). This result means that probably not all the instances of a mental process have been detected by the parser (i.e. 28% undetected). The same comparison ran on the sub-types of *mental* process shows diametrically opposite results, i.e. three fourths more parser generated results than in the corpus (172%) which is an indication of multiple false positives. A possible explanation is the correlation between increase in delicacy and uncertainty i.e. the more delicate features are less precise in the parser results. As mentioned in the beginning of the section, uncertainty, in this case, is manifested as excessive generation of process types represented in disjoint sets of possible options. Currently no ranking mechanism in put in place that would suggest the best option from the candidate ones. Hence the parser provides multiple feature selections from the same system (in this case MENTAL-TYPE) for a constituent whereas there should be a single one. In the future this needs to be addressed by introducing a discrimination mechanism. It could, for instance, collect all the possible matches first and then only the most suitable to be

	Match	Corpus non-matched	Parser non-matched	Precision	Recall	F1
emoter	91	70	57	0.61	0.57	0.59
phenomenon	359	223	294	0.55	0.62	0.58
carrier	267	263	244	0.52	0.50	0.51
cognizant	82	84	104	0.44	0.49	0.47
agent	267	210	428	0.38	0.56	0.46
possessed	71	24	155	0.31	0.75	0.44
attribute	162	241	170	0.49	0.40	0.44
affected	93	70	663	0.12	0.57	0.20

Table 10.16 The evaluation statistics available for the PARTICIPANT-ROLE-TYPE system from the TRANSITIVITY system network

assigned to the constituent unit, possibly by using frequencies available from a corpus annotations like OE or other sources.

If we look again at Table 10.14 and compare the number of all mental sub-type occurrences (see Figure **??**) to the number of mental type occurrences, then we see that the ratio is quite low (63%). As the delicacy of the features increases fewer of these features are provided in the corpus. This is a direct manifestation of difficulty in annotating with ever more delicate features. This ratio, therefore, measures the degree of incompleteness at this level of delicacy. Comparing the same ratio for the parser generated segments we notice an opposite result (151%). This is in fact another measurement of noise generated due to uncertainty when advancing to a more delicate features as explained above.

So far we have discussed evaluation for the process types and now let's turn attention towards the feature assignments from the PARTICIPANT-ROLE-TYPE system. Table 10.16 presents the data considered for this evaluation along with the precision, recall and F_1 score for each participant role sorted according to F_1 score in descending order.

In this evaluation only the participant roles that appear at least 100 times in the corpus are considered. This restriction is inherited from Schulz (2015: 160-162) study on OE corpus. The entire set of evaluation results for participant roles is provided in Table I.9 in Appendix I.

The evaluation results for considered set of participant roles is summarised in Table 10.17. The precision vary from 12% to 61% with an average of 43%, and the recall values between 40% and 75% with an average of 56%. The data is characterised by lower precision and higher recall, which is directly reflected in the F_1 score averaging to 46%.

The distribution of precision, recall and F_1 scores can be seen in Figures 10.12 and 10.13. The noticeable feature is the peak of precision near the 0.5 mark and that of recall around 0.6. They translate into F_1 distribution as three groups: a small first group near the minimum pole, formed by the affected feature; a second tall tower at 45% accuracy formed by cognizant, agent, possessed and attribute features; and a third group around 55% accuracy on the right side of the graph formed by the emoter, phenomenon and carrier.

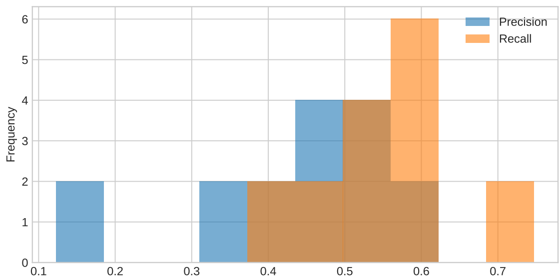


Fig. 10.12 The distribution of precision and recall for selected features from the PARTICIPANT-ROLE system

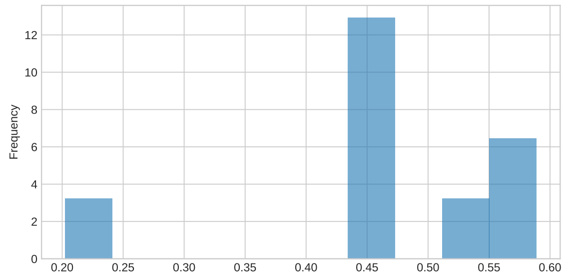


Fig. 10.13 The distribution of F1 score for selected features from the PARTICIPANT-ROLE system

In Table I.9 a suite of eight features have F_1 scores descending from almost 60% to 44%. The affected features follow with half of the previous (20%). As the accuracy decreases we can notice that few features display high recall especially in the case of affected feature, which spiked to 600% the number of matched segments. This is a direct manifestation of the assignment of multiple roles to one participant by multiple

	Precision	Recall	F1
mean	0.43	0.56	0.46
standard deviation	0.16	0.10	0.12
min value	0.12	0.40	0.20
25% quantile	0.37	0.50	0.44
50% quantile	0.46	0.56	0.46
75% quantile	0.53	0.58	0.53
max value	0.61	0.75	0.59

Table 10.17 Descriptive statistics of the precision, recall and F1 scores for evaluated PARTICIPANT-ROLE features

graph patterns (for the reasons explained in the beginning of this section). This abnormally high number of participants generated by the parser must be addressed in future work starting with an investigation of the Transitivity graph patterns generated from the PTDB, in particular for the affected feature. The next section will conclude this evaluation chapter.

10.5 Discussion

In this chapter we have discussed how the empirical evaluation of Parsimonious Vole parser has been conducted. The stage is set through a general presentation of the corpora and what the task at hand is, i.e. identifying and comparing segments available in the corpus annotations to those generated automatically by the parser. The accuracy is determined by the parser ability to generate identical or partially overlapping segments to those in the corpus.

Section 10.1 presents the OCD and OE corpora, which are employed in the current evaluation exercise. OCD corpus is used for measuring the accuracy of the constituency structure generated by the parser and MOOD feature assignments to clause units. The OE corpus is used to evaluate TRANSITIVITY feature assignments to configuration and participant constituents. The parser output does not follow entirely the annotation methodology used in annotation of the corpus therefore there are a few differences to account for, which are explained in Section 10.1.3.

Section 10.2 explains how the current evaluation is performed. It starts by defining what is evaluated i.e. labelled segments, then explains how corpus annotations and parser output are represented as sets of segments, and finally presents how these batches are compared to one another deriving from that parser accuracy measurements and how the measurement data is structured. The alignment algorithm, presented in the same section, takes into consideration not only the exact but also the partial matches.

The next two sections, 10.3 and 10.4, present the evaluation data and discuss the findings. The evaluation of the segmentation task revealed that 71% of the segment have identical spans and that 83% of the segments are identical or shifted slightly (up to 5 characters). There are several ways to measure distance and among the tested ones the most significant was the geometric distance and the WindowDiff distance, while the other distances were strongly correlated to one of these two and were omitted from the discussion. The parser assigns classes to the constituent units with an accuracy of 74%; furthermore, clause main Mood elements are assigned with an accuracy of 71.2% while Transitivity elements with an accuracy of 81%.

When it comes to evaluating the accuracy of systemic assignments, the measured accuracy varies across delicacy levels and between sibling features within the same system, which was addressed for MOOD and especially for TRANSITIVITY in Section 10.4. The accuracy measurements are provided for a fraction of the MOOD system network and a fraction of TRANSITIVITY system network. This is based on their availability in the corpus annotations, which was described in Section 10.1. The features from the MOOD system network are assigned, on average, with an accuracy of 59%. The accuracy of TRANSITIVITY system network was measured for the PROCESS-TYPE system and the PARTICIPANT-ROLE-TYPE system separately. The accuracy of the former, on average, is 36% and the latter, on average, is 46%.

The present evaluation results are significant in at least two major respects. First, the parser overall accuracy is not too satisfactory when compared to the accuracy achieved by constituency parsers in the mainstream computational linguistics. Nonetheless, as the parser generates feature rich output which makes Parsimonious Vole parser different, the current results could be already considered useful in some practical situations. Second, this evaluation shows which areas are in need of improvement and provides some hints on what could be the reason for the poor performance. Also, this evaluation is the first one and constitutes the baseline for further incremental developments.

Even if it is completely separate action, this evaluation can be useful for further corpus improvements as well. When I mention corpus improvement I bear in mind the OCD corpus in particular, which needs to be annotated by at least one more annotator and tested for reliability. In addition, the corpora size is fairly small and many systemic features are under-represented or missing completely as is the case, for example, of event-relating, environmental, action sub-types and other processes. It would be desperately necessary but quite unlikely to happen (McEnery et al. 2006: 33) extending the corpus annotation and include more delicate MOOD and TRANSITIVITY features in order to enable study how they vary and how accurately the parser detects them.

Next chapter will put the entire thesis into perspective and conclude the work done so far providing new ideas and setting a tone for what needs to be done in future work to improve the current results.

Chapter 11

Conclusions

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

In this process a primary milestone is the creation of constituency structure. Chapter 3 describes the essential theoretical foundations of two SFL schools, namely the Sydney and Cardiff schools, and provides a critical analysis of the two to reconcile 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 thesis.

In order to create the constituency structure from the dependency structure there needs to be a mechanism in place providing a theoretical and practical mapping between the two. The theoretical account of the dependency grammar and how it is related to SFL is described in Chapter 5. The practical aspects and concrete algorithms are described in Chapter 8 together with the mapping rules used in the process.

To make clear what the basic ingredients are and how the algorithms are coded, Chapter 7 introduces all the data structures and operations on them. These structures are defined from a computer science 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 this, 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 with 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 Chapter 7. Then in Chapters 8 and 9 I show how these operations are used for enrichment of the syntactic backbone with systemic features.

The more precisely graph patterns are defined the less instances will be matched to them and thus the number of errors decreases and the accuracy increases. 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 leave unstated up to two participants. If applied in their canonical form the patterns will not identify with such instances. One solution would be to reduce the specificity of the patterns, which leads to an increase in erroneous applications, or to populate where possible the covert participants to yield successful matches. It is the second approach that was implemented in this thesis. 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 6 and then a concrete implementation described in Chapter 9.

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 by Anke Schultz covering the Transitivity system. The performance shown in the evaluation results varies between 0.6 and 0.9 F_1 score 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) but constitute a good baseline for future improvements.

As discussed in Section 10.5 further investigation needs to be conducted to determine the error types, shortcomings in the corpus and the parser. Since for both syntactic and semantic annotations there was 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 low performance in order identify potential problems.

11.1 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 ones 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 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 texts (usually short) as much knowledge as possible is crucial for 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 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 thus the deeper linguistic understanding possible the better.

A parser of the type aimed 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 and others. 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 and many others.

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 the potential task at hand such as satisfaction level, requirement or complaint discovery, etc.

11.2 Future work

Pattern graphs and the matching methods developed in this thesis can be applied for expressing many more grammatical features than the ones presented in this thesis. They can serve as language for systematising grammatical realisations especially that the realisation statements play a vital role in SG grammars. The graph matching method itself can 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 annotate large amounts of text to come up with conclusive statements or formulate hypothesises. 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 arise in the process of theoretical development and parser implementation.

11.2.1 Towards semantic verbal groups

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. The alternative is adopting the concept of verbal group, simple or complex, as proposed by the Sydney school in Halliday & Matthiessen (2013b: p.396–418, 567–592), which provides a richer semantically motivated description. However, analysis with the verbal group complex is potentially complex 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*

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

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

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.

Generalisation 11.2.1 (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 Generalisation 11.2.1. 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 the Cardiff grammar and the system of verbal group complex described in the Sydney grammar (Halliday & Matthiessen 2013b: 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 11.1 and 11.2 – but also of the modal assessment that includes modality, as demonstrated by the Examples 127 and 128.

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

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

Examples 127 and 128 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*.

11.2.2 Nominal, Quality, Quantity and other groups of the Cardiff grammar

Cardiff unit classes are semantically motivated as compared to the more syntactic ones in the Sydney grammar. This has been presented in Section 3.3 and discussed in Section 4.1.

For instance, Nominal class structure proposed in the Cardiff grammar (discussed in Section 4.1.3), 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 the 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 resource to help determine 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. 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 straightforward and further adaptations need to be done for the context of SFL.

The same holds for Adverbial and Adjectival groups (discussed in Section 4.1.4) 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) can yield positive results in parsing with Cardiff unit classes.

Just as 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.

11.2.3 Taxis analysis and potential for discourse relation detection

Currently the 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 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.

Then, in the 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 so remains highly desirable future work. Further improvements and developments can be performed based on incremental testing and corrections of the taxis pattern database.

The methods employed in this thesis 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 et al. 1992; Mann & Thompson 1988).

To increase the accuracy of taxis analysis, I believe the following additional elements should 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.

11.2.4 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 the Searle (1969) which, in mainstream linguistics, are placed under the umbrella of pragmatics.

Halliday proposes a simple system of speech functions (Halliday & Matthiessen 2013b: 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 the Hallidayan MOOD system as a foundation, especially because it has already been implemented and described in the current work.

11.2.5 Process Types and Participant Roles

The PTDB (Neale 2002) is the first lexical-semantic resource for the 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 exhausting its potential as a lexical-grammatical resource for semantic parsing.

In 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 occurs, each with associated semantic representations of participant roles and some first order predicates. However, the systematisation 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 the Transitivity system, could be used as a training corpus for machine learning algorithms. Another potential benefit would be generating semantic constraints (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 the Nigel grammar (Matthiessen 1985). To achieve this, one would have to first map the Cardiff and the Sydney Transitivity systems and second extract lexical entries from the Nigel grammar along with adjacent systemic selections.

11.2.6 Reasoning with systemic networks

Systemic networks are a powerful instrument to represent the 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 realisation rules useful for generation purpose but they could potentially be used in parsing as well.

In this thesis 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 was the first to attempt formalization 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. 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.

11.2.7 Creation of richly annotated corpus

In order to evaluate a parser, a gold standard annotation corpus is essential. The bigger the corpus, covering various 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 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 the Cardiff grammar already exists but, from personal communication with Prof. Robin Fawcett, it has not yet been released to the public because it still is considered 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 the 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 the Penn Treebank. This task had been already explored by Honnibal in 2004; 2007.

11.2.8 The use of Markov Logic for pattern discovery

Markov Logic (Domingos et al. 2010; Richardson & Domingos 2006) is a probabilistic logic which applies ideas of Markov networks to first order logic enabling uncertain inference. 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.

In the current approach I am using graph pattern matching techniques to generate a rich set of features for the constituent units. However creating those patterns is a considerable effort. Since graph patterns can be expressed via first order functions and individuals, and assuming that there would already exist a richly annotated corpus, Markov Logic instruments (for example Alchemy[1], Tuffy[2] and others) could 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 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.

11.3 A final word

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

References

Abney, S. 1987. *The English noun phrase in its sentential aspects*. MIT Press.

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

Bach, Emmon. 1966. *An introduction to transformational grammars*. Holt, Rinehart and Winston. Inc.

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>.

Bar-Hillel, Yehoshua. 1953. A quasi-arithmetical notation for syntactic description. *Language* 29. 47–58. Reprinted in Y. Bar-Hillel. (1964). *Language and Information: Selected Essays on their Theory and Application*, Addison-Wesley 1964, 61–74.

Barendregt, Hendrik Pieter, Marko CJD van Eekelen, John RW Glauert, J Richard Kennaway, Marinus J Plasmeijer & M Ronan Sleep. 1987. Term graph rewriting. In *International conference on parallel architectures and languages europe*, 141–158. Springer.

Bateman, John A. 1996a. KPML-compatible linguistic resources: Release 3. Tech. rep. Institut für Integrierte Publikations- und Informationssysteme (IPSI), GMD Darmstadt, Germany.

Bateman, John A. 1996b. *KPML Development Environment: multilingual linguistic resource development and sentence generation*. German National Center for Information Technology (GMD), Institute for integrated publication and information systems (IPSI) Sankt Augustin. <http://www.fb10.uni-bremen.de/anglistik/langpro/kpml/Doc/kpml1/kpml-1-1-documentation.pdf>. (Release 1.1).

Bateman, John A. 1997. Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering* 3(1). 15–55. doi:10.1017/S1351324997001514. <http://www.journals.cambridge.org/abstract{_}S1351324997001514>.

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

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

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

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

Beeferman, Doug, Adam Berger & John Lafferty. 1999. Statistical models for text segmentation. *Machine learning* 34(1-3). 177–210.

Bloomfield, Leonard. 1933. *Language*. New York and London: Henry Holt and Co. and Allen and Unwin Ltd.

Böhmová, Alena, Jan Hajič, Eva Hajičová & Barbora Hladká. 2003. The prague dependency treebank. In Anne Abeillé (ed.), *Treebanks: Building and using parsed corpora*, 103–127. Dordrecht: Springer Netherlands. doi:10.1007/978-94-010-0201-1_7. <https://doi.org/10.1007/978-94-010-0201-1_7>.

Bohnet, Bernd. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd international conference on computational linguistics* COLING '10, 89–97. Stroudsburg, PA, USA: Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=1873781.1873792>.

Bondy, John Adrian, Uppaluri Siva Ramachandra Murty et al. 1976. *Graph theory with applications*, vol. 290. Citeseer.

Bookstein, Abraham, Vladimir A. Kulyukin & Timo Raita. 2002. Generalized hamming distance. *Information Retrieval* 5(4). 353–375. doi:10.1023/A:1020499411651. <https://doi.org/10.1023/A:1020499411651>.

Bresnan, Joan. 1982. Control and complementation. *Linguistic Inquiry* 13(3). 343–434.

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Blackwell. <http://books.google.lu/books/about/Lexical{_}Functional{_}Syntax.html?id=vMxgevXoq{_}gC{&}redir{_}esc=y>.

Bresnan, Joan, Ash Asudeh, Ida Toivonen & Stephen Wechsler. 2015. *Lexical-functional syntax*, vol. 16. John Wiley & Sons.

Buchholz, Sabine & Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning* CoNLL-X '06, 149–164. Stroudsburg, PA, USA: Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=1596276.1596305>.

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

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

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

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

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

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

Carroll, John, Ted Briscoe & Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of the international conference on language resources and evaluation*, 447–454.

Carroll, John, Guido Minnen & Ted Briscoe. 1999. Corpus Annotation for Parser Evaluation. In *Proceedings of the eacl workshop on linguistically interpreted corpora (linc)* June, 7. Association for Computational Linguistics. <http://arxiv.org/abs/cs/9907013>.

Cer, Daniel D.M., Marie-Catherine M.C. De Marneffe, Daniel Jurafsky & C.D. Manning. 2010. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *Lrec 2010*, vol. 0, European Language Resources Association. <http://171.64.67.140/pubs/lrecstanforddeps{_}final{_}final.pdfhttp://www.lrec-conf.org/proceedings/lrec2010/pdf/730{_}Paper.pdf>.

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

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory* 2. 113–124.

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

Chomsky, Noam. 1957b. *Syntactic structures*. The Hague: Mouton.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, Massachusetts: M.I.T. Press.

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

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

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

Chomsky, Noam. 1993a. A Minimalist Program for Linguistic Theory. In K. Hale & S. J. Keyser (eds.), *The View from Building 20*, Cambridge, Mass: MIT Press.

Chomsky, Noam. 1993b. *Lectures on government and binding: The pisa lectures* 9. Walter de Gruyter.

Clegg, Andrew B & Adrian J Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC bioinformatics* 8(1). 24.

Costetchi, Eugeniu. 2013. Semantic role labelling as SFL transitivity analysis. In *Esslli student session 2013 preproceedings*, 29–37.

Courcelle, Bruno. 1990. Graph rewriting: An algebraic and logic approach. In *Formal models and semantics*, 193–242. Elsevier.

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

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

Fawcett, R. 1988a. Language Generation as Choice in Social Interaction. In Zock, M. & G. Sabah (eds.), *Advances in natural language generation*, vol. 2, 27–49. Pinter Publishers. (Paper presented at the First European Workshop of Natural Language Generation, Royaumont, 1987).

Fawcett, Robin. 1973. Generating a sentence in systemic functional grammar. In M.A.K. Halliday & James Martin (eds.), *Readings in systemic linguistics*, Bastford.

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

Fawcett, Robin P. 1987. The semantics of clause and verb for relational processes in English. In Robin P. Fawcett & David J. Young (eds.), *New developments in systemic linguistics: theory and description*, vol. 1, London: Pinter.

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

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

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

Fawcett, Robin P. 1996. A systemic functional approach to complementation in English. In Christopher Butler, Margaret Berry, Robin Fawcett & Guowen Huang (eds.), *Meaning and form: systemic functional interpretations*, Norwood, NJ: Ablex.

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

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

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

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

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

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

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

Fournier, Chris. 2013. Evaluating text segmentation using boundary edit distance. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)*, vol. 1, 1702–1712.

Fournier, Chris & Diana Inkpen. 2012. Segmentation similarity and agreement. In *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 152–161. Association for Computational Linguistics.

Gaifman, Haim. 1965. Dependency systems and phrase-structure systems. *Information and Control* 8(3). 304 – 337. doi:https://doi.org/10.1016/S0019-9958(65)90232-9. <http://www.sciencedirect.com/science/article/pii/S0019995865902329>.

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>.

Garde, Paul. 1977. Ordre lineaire et dependance syntaxique. contribution a une typologie. In *Bulletin de la societe de linguistique de paris*, vol. 1 72, 1–26. Paris.

Gerdes, Kim & Sylvain Kahane. 2013. Defining dependencies (and constituents). *Frontiers in Artificial Intelligence and Applications* 258. 1–25.

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

Grzegorz, Rozenberg. 1999. *Handbook of graph grammars and computing by graph transformations, vol 2: Applications, languages and tools.* world Scientific.

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

Haegeman, J. 1991a. *Introduction to Government and Binding Theory.* Oxford (UK) and Cambridge (USA): Blackwell.

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

Hajic, Jan, Eva Hajicová, Petr Pajas, Jarmila Panevová, Petr Sgall & Barbora Vidová-Hladká. 2001. Prague dependency treebank 1.0 (final production label). cdrom cat: Ldc2001t10. Tech. rep. ISBN 1-58563-212-0.

Halliday, M.A.K. 1968a. Language and experience. *Educational Review* 20(2). 95–106.

Halliday, MAK. 1997. Linguistics as metaphor. *Reconnecting Language: Morphology and Syntax in Functional Perspectives* 154. 3–27.

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

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

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

Halliday, Michael A. K. 1967. Notes on transitivity and theme in English — Parts 1 and 2. *Journal of Linguistics* 3. 37–81 and 199–244.

Halliday, Michael A. K. 1968b. Notes on transitivity and theme in English — Part 3. *Journal of Linguistics* 4. 179–215.

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

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

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

Halliday, Michael A. K. 2003b. Systemic theory. 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. & Christian M. I. M. Matthiessen. 2013a. *Halliday's Introduction to Functional Grammar*. London and New York: Routledge 4th edn.

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

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

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

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

Harris, Z.S. 1951. *Methods in structural linguistics* Methods in Structural Linguistics. University of Chicago Press. <https://books.google.lu/books?id=a6nYjgEACAAJ>.

Harrison, Philip, Steven Abney, Ezra Black, Dan Flickinger, Claudia Gdaniec, Ralph Grishman, Donald Hindle, Robert Ingria, Mitch Marcus, Beatrice Santorini & Tomek Strzalkowski. 1991. Evaluating syntax performance of parser/grammars. In *Proceedings of the natural language processing systems evaluation workshop, berekely, ca, june 1991* Rome Laboratory Technical Report, RL-TR-91-362, .

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.

Hays, David G. 1960. Grouping and dependency theories. *Proceedings of the National Symposium on Machine Translation* 2538. 258–266.

Hays, David G. 1964. Dependency theory: A formalism and some observations. *Language* 40(4). 511–525. <http://www.jstor.org/stable/411934>.

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

Hockett, Charles F. 1958. *A course in modern linguistics*. New York: Macmillan.

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>.

Hudson, Richard. 2010. *An Introduction to Word Grammar*. Cambridge University Press.

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

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.

Jackendoff, Ray. 1977. *$\overline{X}$ Syntax: a study of phrase structure*. Cambridge, MA: The M.I.T. Press.

Johnson, Christopher & Charles J. Fillmore. 2000. The framenet tagset for frame-semantic and syntactic coding of predicate-argument structure. In *Proceedings of the 1st north american chapter of the association for computational linguistics conference* NAACL 2000, 56–62. Stroudsburg, PA, USA: Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=974305.974313>.

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

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

King, Tracy Holloway & Richard Crouch. 2003. The PARC 700 Dependency Bank. In *4th international workshop on linguistically interpreted corpora (linc03)*, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.15.3277>.

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.

Kitaev, Nikita & Dan Klein. 2018. Multilingual constituency parsing with self-attention and pre-training. *arXiv preprint arXiv:1812.11760* .

Kübler, S., R. McDonald & J. Nivre. 2009. *Dependency parsing* Online access: IEEE (Institute of Electrical and Electronics Engineers) IEEE Morgan & Claypool Synthesis eBooks Library. Morgan & Claypool. <https://books.google.lu/books?id=k3iiup7HB9UC>.

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>.

Lamprier, Sylvain, Tassadit Amghar, Bernard Levrat & Frederic Saubion. 2007. On evaluation methodologies for text segmentation algorithms. In *19th ieee international conference on tools with artificial intelligence (ictai 2007)*, vol. 2, 19–26. IEEE.

Lecerf, Yves. 1961. Une representation algebrique de la structure des phrases dans diverses langues naturelles. In *Compte rendu de l'academie des sciences de paris* 252, 232–234. Academie des Sciences.

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

Lin, Dekang & Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering* 7(4). 343–360.

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

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

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

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

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

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

Marneffe, Marie-Catherine, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre & Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the ninth international conference on language resources and evaluation (lrec-2014)(vol. 14)*, European Language Resources Association (ELRA). <http://www.lrec-conf.org/proceedings/lrec2014/pdf/1062{_}Paper.pdf>.

Marneffe, Marie-Catherine, Bill MacCartney & Christopher D Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Lrec 2006*, vol. 6 3, 449–454. Stanford University. <http://nlp.stanford.edu/manning/papers/LREC{_}2.pdf>.

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

Marneffe, Marie-Catherine & Christopher D. Manning. 2008b. The Stanford typed dependencies representation. *Coling 2008 Proceedings of the workshop on CrossFramework and CrossDomain Parser Evaluation CrossParser 08* 1(ii). 1–8. doi:10.3115/1608858.1608859. <http://portal.acm.org/citation.cfm?doid=1608858.1608859>.

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

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

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

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

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

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

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

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

McEnery, Tony, Richard Xiao & Yukio Tono. 2006. *Corpus-based language studies: An advanced resource book*. Taylor & Francis.

Mel'čuk, Igor. 1988. Semantic description of lexical units in an Explanatory Combinatorial Dictionary: basic principles and heuristic criteria. *International Journal of Lexicography* 1. 165–188.

Mel'čuk, Igor A. & Nikolaj V. Pertsov. 1986. *Surface syntax of english*. John Benjamins Publishing. doi:10.1075/llsee.13. <http://www.jbe-platform.com/content/books/9789027279378>.

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

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

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

Müller, Stefan. 2018. *Grammatical theory: From transformational grammar to constraint-based approaches*, vol. 1. Language Science Press.

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

Newman, Mark EJ. 2005. Power laws, pareto distributions and zipf's law. *Contemporary physics* 46(5). 323–351.

Niekrasz, John & Johanna D Moore. 2010. Unbiased discourse segmentation evaluation. In *2010 ieee spoken language technology workshop*, 43–48. IEEE.

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

Nivre, Joakim. 2015. Towards a universal grammar for natural language processing. In *Computational Linguistics and Intelligent Text Processing*, 3–16. Springer.

Nivre, Joakim, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel & Deniz Yuret. 2007a. The conll 2007 shared task on dependency parsing. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (emnlp-conll)*, 915–932.

Nivre, Joakim, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov & Erwin Marsi. 2007b. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13(2). 95–135. doi:10.1017/S1351324906004505. <https://doi.org/10.1017/S1351324906004505>.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty & Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph

Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the tenth international conference on language resources and evaluation (lrec 2016)*, Paris, France: European Language Resources Association (ELRA).

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

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

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

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

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

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

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

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

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

Pevzner, Lev & Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics* 28(1). 19–36.

Plasmeijer, Rinus, Marko Van Eekelen & MJ Plasmeijer. 1993. *Functional programming and parallel graph rewriting*, vol. 857. Addison-wesley Reading.

Pollard, Carl & Ivan Sag. 1987. *Information-Based Syntax and Semantics*. CSLI.

Pollard, Carl J. & Ivan A. Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

Pollock, Jean-Yves. 1989. Verb movement, universal grammar, and the structure of ip. *Linguistic inquiry* 20(3). 365–424.

Postal, P. M. 1974. *On Raising*. MIT Press.

Pradhan, S, E Loper, D Dligach & M Palmer. 2007. Semeval-2007 task-17: English lexical sample, srl and all words. *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)* 87–92. <papers2://publication/uuid/74521E4E-811F-4EB7-A1BE-973D69EBC6C2>.

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.

Ross, John Robert. 1967. *Constraints on variables in syntax.*: Massachusetts Institute of Technology dissertation.

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.

Schank, Roger. 1969. Conceptual dependency as a framework for linguistic analysis. *Linguistics* 7(49). 28–50.

Schuler, Karin Kipper. 2005. *Verbnet: A broad-coverage, comprehensive verb lexicon*: University of Pennsylvania dissertation.

Schulz, Anke. 2015. *Me, myself and i: A corpus-based, contrastive study of english and german computer-mediated communication from a systemic functional perspective*: Technische Universität, Darmstadt dissertation.

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>.

Silveira, Natalia, Timothy Dozat, Marie-Catherine De Marneffe, Samuel Bowman, Miriam Connor, John Bauer & Chris Manning. 2014. A Gold Standard Dependency Corpus for English. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the ninth international conference on language resources and evaluation (lrec'14)*, European Language Resources Association (ELRA). <http://nlp.stanford.edu/pubs/USD{_}LREC14{_}paper{_}camera{_}ready.pdf>.

Sleator, Daniel Dominic & David Temperley. 1995. Parsing english with a link grammar. *CoRR* abs/cmp-lg/9508004. 93. <http://arxiv.org/abs/cmp-lg/9508004>.

Snow, Rion, Daniel Jurafsky & Andrew Y Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems*, 1297–1304.

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

Sowa, John F. 1976. Conceptual graphs for a data base interface. *IBM Journal of Research and Development* 20(4). 336–357. doi:10.1147/rd.204.0336. <http://dx.doi.org/10.1147/rd.204.0336>.

Stern, Mitchell, Jacob Andreas & Dan Klein. 2017. A minimal span-based neural constituency parser. *arXiv preprint arXiv:1705.03919* .

Stockwell, Robert P., Barbara Hall Partee & Paul Schacter. 1973. *The major syntactic structures of english*. New York: Holt, Rinehart and Winston. <http://hdl.handle.net/2027/mdp.39015002216417>. Bibliography: p. 811-840.

Stowell, T.A. & E. Wehrli. 1992. *Syntax and the lexicon* Syntax and semantics. Academic Press. <https://books.google.lu/books?id=yiEcAQAAIAAJ>.

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.

Teich, Elke. 1991. *Functional and systemic linguistics: approaches and uses* chap. A treatment of raising and control in systemic grammar, 107–120. De Gruyter Mouton CY. doi:10.1515/9783110883527.107.

Tesniere, Lucien. 1959. *Elements de syntaxe structurale*. Paris: Klincksieck.

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

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

Tucker, Gordon H. 1998. *The Lexicogrammar of Adjectives: A Systemic Functional Approach to Lexis*. Bloomsbury.

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

Žolkovskij, Alexander K. & Igor A. Mel'čuk. 1967. O semantičeskom sinteze. *Problemy kibernetiki* 19(?). 177–238.

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

West, Douglas Brent et al. 2001. *Introduction to graph theory*, vol. 2. Prentice hall Upper Saddle River.

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

Zeman, Daniel, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis M. Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jaroslava Hlavácová, Václava Kettnerová, Zdenka Uresová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çagri Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali El-Kahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj & Josie Li. 2017. Conll 2017 shared task: multilingual parsing from raw text to universal dependencies. *Proceedings of the CoNLL Shared Task* 1–19.

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

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

Zwicky, Arnold M. 1985. Heads. *Journal of Linguistics* 21. 1–30.

Appendix A

SFL Syntactic Overview

A.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)

A.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)

A.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)

A.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)

A.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)

A.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)

A.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)

A.2 Sydney Syntax

A.2.1 Logical

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

A.2.2 Textual

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

A.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

A.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

A.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

Appendix B

Stanford Dependency schema

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

- arg - argument
  - agent - agent
  - comp - complement
    - acomp - adjectival complement
    - attr - attributive
    - ccomp - clausal complement with internal subject
    - xcomp - clausal complement with external subject
    - compl - complementizer
    - obj - object
      - dobj - direct object
      - iobj - indirect object
      - pobj - object of preposition
    - mark - marker (word introducing an advcl)
    - rel - relative (word introducing a rcmod)
  - subj - subject
    - nsubj - nominal subject
      - nsubjpass - passive nominal subject
    - csubj - clausal subject
      - csubjpass - passive clausal subject
- ...

Fig. B.1 The Stanford dependency scheme - part one

```
dep - dependent
├─ ...
├─ mod - modifier
│  ├─ abbrev - abbreviation modifier
│  ├─ amod - adjectival modifier
│  ├─ appos - appositional modifier
│  ├─ advcl - adverbial clause modifier
│  ├─ purpcl - purpose clause modifier
│  ├─ det - determiner
│  ├─ predet - predeterminer
│  ├─ preconj - preconjunct
│  ├─ infmod - infinitival modifier
│  ├─ partmod - participial modifier
│  ├─ advmod - adverbial modifier
│  │  └─ neg - negation modifier
│  ├─ rcmod - relative clause modifier
│  ├─ quantmod - quantifier modifier
│  ├─ tmod - temporal modifier
│  ├─ measure - measure-phrase modifier
│  ├─ nn - noun compound modifier
│  ├─ num - numeric modifier
│  ├─ number - element of compound number
│  ├─ prep - prepositional modifier
│  ├─ poss - possession modifier
│  ├─ possessive - possessive modifier ('s)
│  └─ prt - phrasal verb particle
└─ ...
```

Fig. B.2 The Stanford dependency scheme - part two

```
dep - dependent
├─ ...
├─ aux - auxiliary
│  ├─ auxpass - passive auxiliary
│  └─ cop - copula
├─ cc - coordination
├─ conj - conjunct
├─ expl - expletive (expletive "there")
├─ parataxis - parataxis
├─ punct - punctuation
├─ ref - referent
└─ sdep - semantic dependent
   └─ xsubj - controlling subject
```

Fig. B.3 The Stanford dependency scheme - part three

Appendix C

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 C.1 Penn Treebank tag set

Appendix D

Mapping dependency to constituency graph

key	***operation***	***parameter***
acomp	new constituent	COMPLEMENT
advcl	new constituent	ADJUNCT
advmod	extend current	None
amod	new constituent	EPITHET_CLASSIFIER_OR_ORDINAL
agent	new constituent	COMPLEMENT_AGENT
appos	new constituent	APPOSITION
aux	extend current	None
auxpass	extend current	None
complm	new constituent	MARKER
conj	extend current	None
csubj	new constituent	SUBJECT
csubjpass	new constituent	SUBJECT
det	new constituent	DEICTIC
dobj	new constituent	COMPLEMENT
expl	new constituent	EXPLETIVE_MARKER
infmod	new constituent	QUALIFIER
iobj	new constituent	COMPLEMENT_DATIVE
mark	new constituent	MARKER
mwe	extend current	None
neg	extend current	None
nn	extend current	None
npadvmod	new constituent	ADJUNCT
nsubj	new constituent	SUBJECT
nsubjpass	new constituent	SUBJECT
num	new constituent	CARDINAL_NUMERATIVE
number	extend current	None
parataxis	new constituent	CLAUSE
partmod	new constituent	QUALIFIER
vmod	new constituent	QUALIFIER
pobj	extend current	None
poss	new constituent	POSESSOR
possessive	new constituent	POSESSOR
preconj	extend current	None
predet	new constituent	PREDEICTIC
prepc	new constituent	COMPLEMENT_ADJUNCT
prt	new constituent	MARKER
punct	extend current	None
purpcl	new constituent	CLAUSE
quantmod	extend current	None
rcmod	new constituent	QUALIFIER
ref	extend current	None
rel	new constituent	CLAUSE
tmod	new constituent	ADJUNCT
xcomp	new constituent	COMPLEMENT
xsubj	new constituent	SUBJECT
discourse	new constituent	DISCOURSE
goeswith	extend current	None

Table D.1 The rule table mapping generic dependency context to generative operations

Key	***Operation***	***Value***
JJ-dep-IN	new constituent	MARKER
VB-dep-IN	new constituent	MARKER
VB-dep-VB	new constituent	CLAUSE
NN-dep-NN	extend current	None
NN-dep-VB	new constituent	CLAUSE
VB-dep-WP	new constituent	COMPLEMENT_ADJUNCT
VB-dep-NN	new constituent	ADJUNCT
RB-dep-IN	extend current	None
WR-dep-JJ	extend current	None
VB-dep-JJ	new constituent	ADJUNCT
VB-conj-VB	new constituent	CLAUSE
VB-cc-CC	new constituent	MARKER
NN-cc-CC	extend current	None
VB-prep-NN	new constituent	COMPLEMENT_ADJUNCT
VB-prep-JJ	new constituent	COMPLEMENT_ADJUNCT
VB-prep-PR	new constituent	COMPLEMENT_ADJUNCT
VB-prep-WP	new constituent	COMPLEMENT_ADJUNCT
VB-prep-CD	new constituent	COMPLEMENT_ADJUNCT
NN-prep-NN	new constituent	QUALIFIER
NN-prep-PR	new constituent	QUALIFIER
RB-npadvmod-NN	extend current	None
NN-npadvmod-NN	extend current	None
VB-npadvmod-NN	new constituent	ADJUNCT
JJ-npadvmod-RB	extend current	None
VB-advmod-RB	new constituent	ADJUNCT
VB-advmod-JJ	new constituent	ADJUNCT
VB-advmod-WR	new constituent	COMPLEMENT
NN-advmod-RB	new constituent	PREDEICTIC
VB-ccomp-NN	new constituent	COMPLEMENT
VB-ccomp-VB	new constituent	COMPLEMENT
IN-pcomp-IN	new constituent	COMPLEMENT_ADJUNCT
IN-pcomp-NN	new constituent	COMPLEMENT_ADJUNCT
IN-pcomp-CD	new constituent	COMPLEMENT_ADJUNCT
IN-pcomp-JJ	new constituent	COMPLEMENT_ADJUNCT
NN-amod-CD	new constituent	CARDINAL_NUMERATIVE
NN-infmod-VB	new constituent	QUALIFIER
CD-prep-NN	new constituent	QUALIFIER
NN-vmod-VB	new constituent	QUALIFIER
NN-prep-JJ	new constituent	QUALIFIER
DT-prep-NN	new constituent	QUALIFIER
JJ-prep-NN	extend current	None

Table D.2 The rule table mapping specific dependency context to generative operations

Appendix E

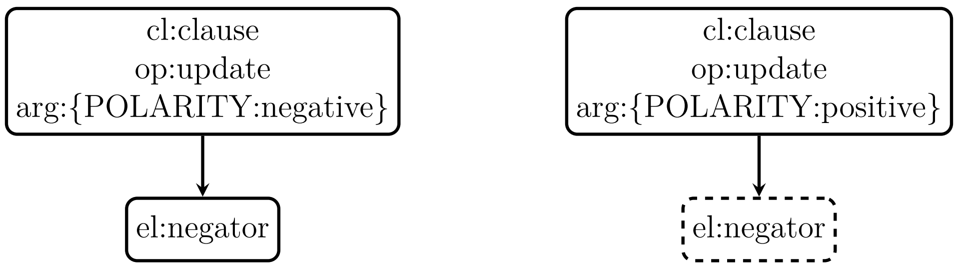
Normalization of PTDB and Cardiff TRANSITIVITY system

The process type column in PTDB contains two words separated by comma. The first one I call *major* as it represents a high level selection in the TRANSITIVITY system and the second one I call *minor* hints at selecting a particular participant configuration. The re-indexing consists of replacing the two features with a more delicate selection in the TRANSITIVITY system.

major	*minor*	*new index*	*min # roles*	*max # roles*
action	one role	one-role-action	1	1
	two role	two-role-action	2	2
	three role	three-role-action	3	3
relational	attributive	attributive	2	3
	locational	locational	2	3
	directional	directional	2	5
	possessive	possessive	2	3
	matching	matching	2	3
emotion	desiderative	desiderative	2	2
	plux xxx	emotive	2	3
perception	xxx	two-role-perception	2	2
	3 p Ag	three-role-perception	3	3
cognition	xxx	two-role-cognition	2	2
	3 p Ag/ matchee	three-role-cognition	3	3
environmental		environmental	x	x
influential	starting	starting	1	2
	continuing	continuing	1	2
	ceasing	ceasing	1	2
	succeeding	succeeding	1	2
	failing	failing	1	2
	causative	causative	1	2
	permissive	permissive	1	2
	enabling	enabling	1	2
	preventing	preventing	1	2
	delaying	delaying	1	2
	tentative	tentative	1	2

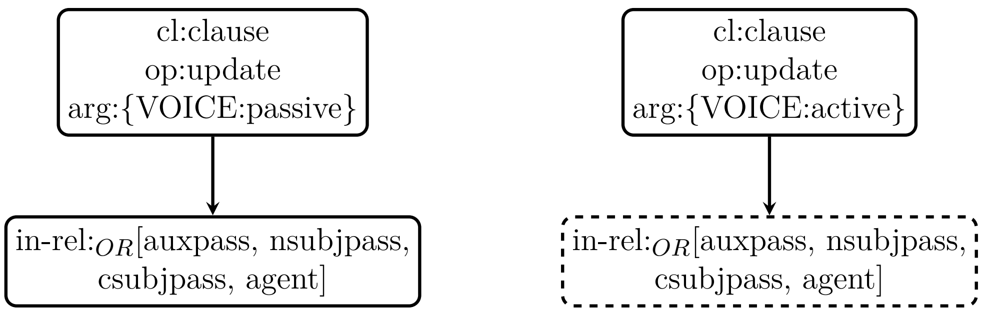
Appendix F

A selection of graph patterns



(a) Negative polarity (b) Negative polarity

Fig. F.1 Polarity detection graph patterns



(a) Passive voice (b) Active voice

Fig. F.2 Voice detection graph patterns

Tense simple

Tense progressive

Tense perfect

Tense perfect

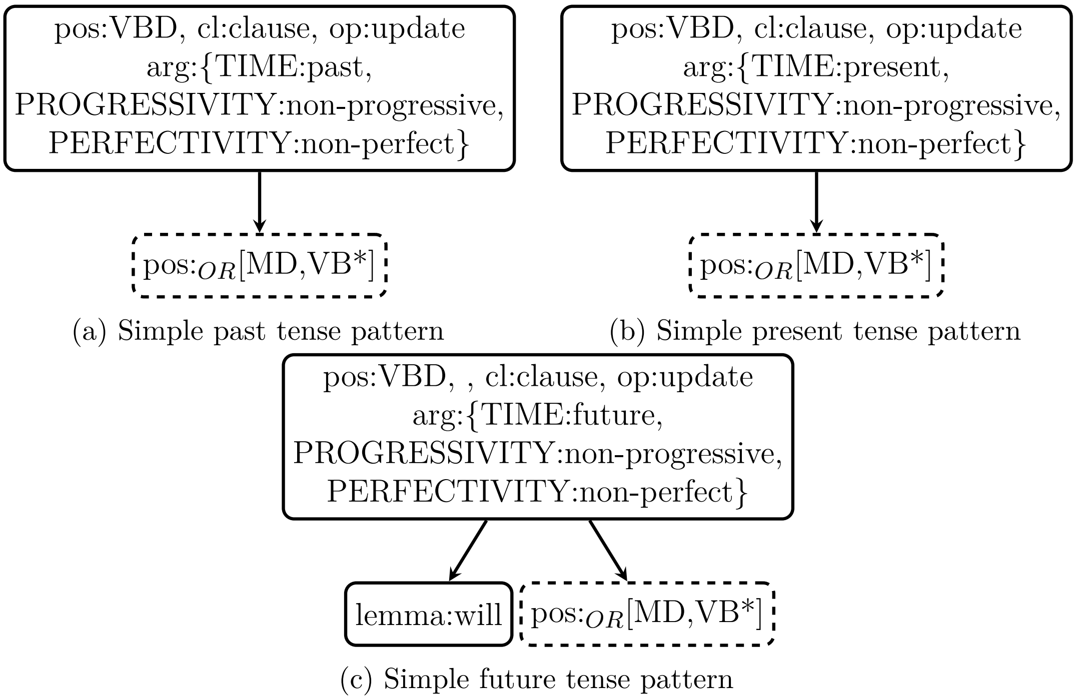


Fig. F.3 Simple past, present and future tense patterns

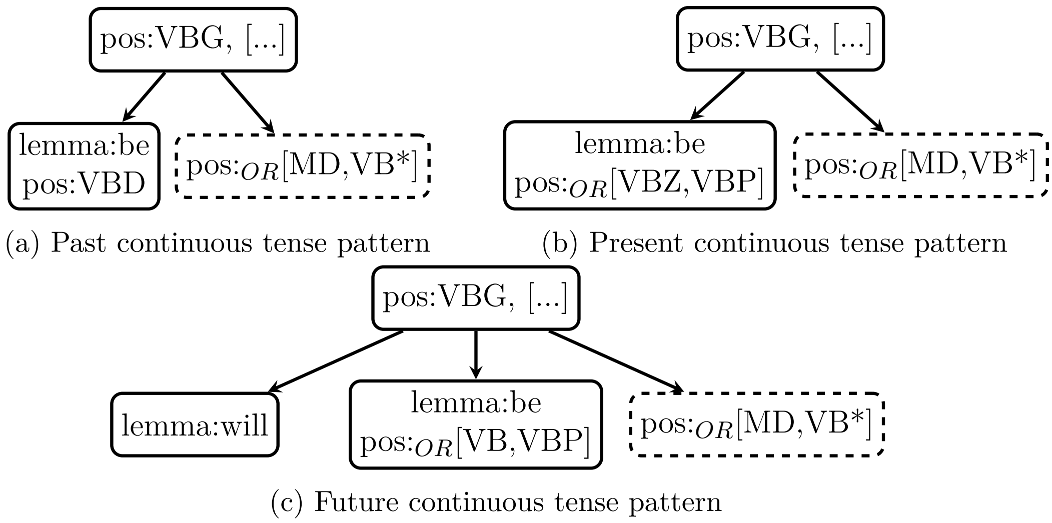
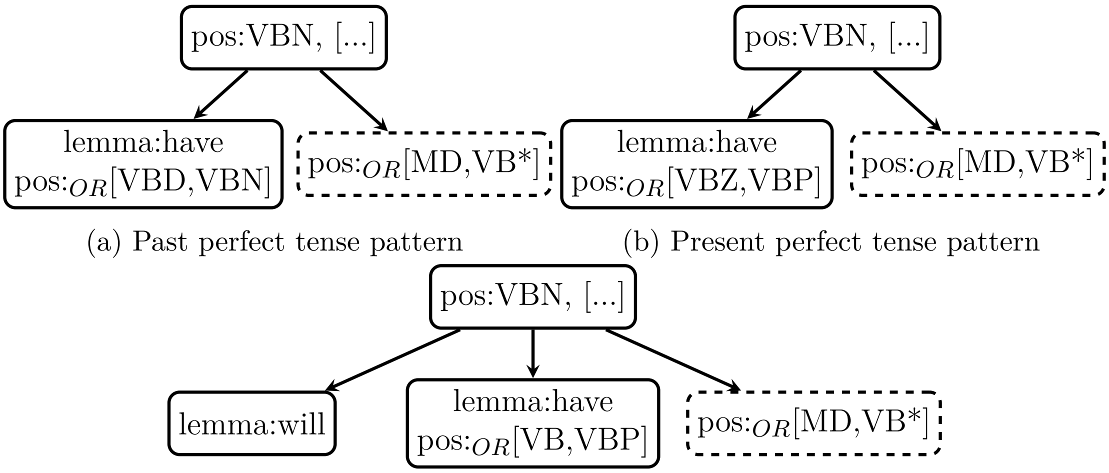


Fig. F.4 Past, present and future continuous tense patterns

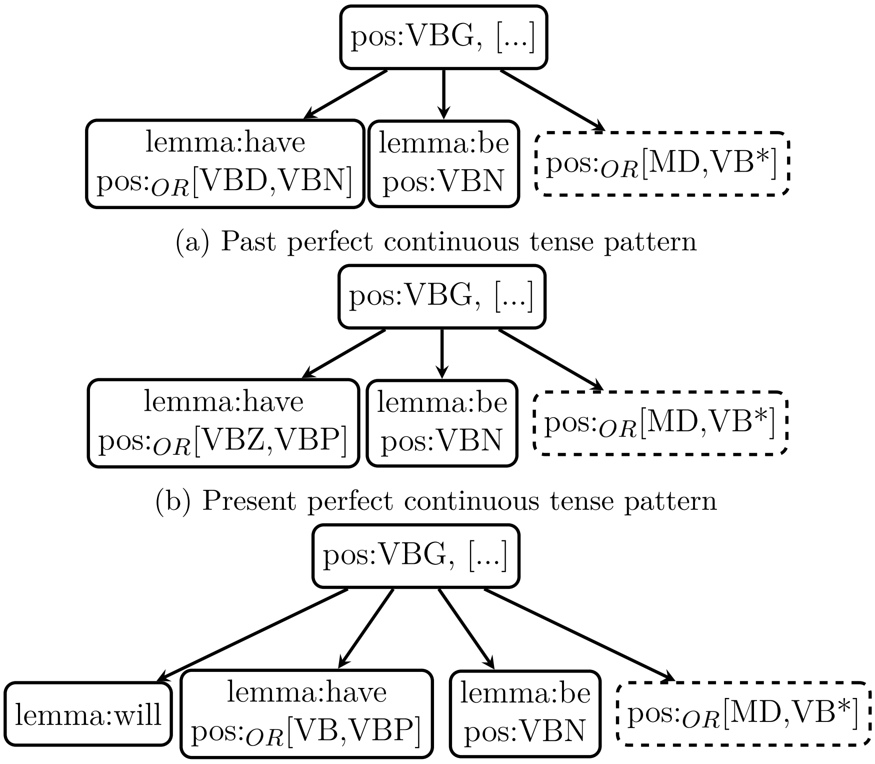


(a) Past perfect tense pattern

(b) Present perfect tense pattern

(c) Future perfect tense pattern

Fig. F.5 Past, present and future perfect tense patterns



(a) Past perfect continuous tense pattern

(b) Present perfect continuous tense pattern

(c) Future perfect continuous pattern

Fig. F.6 Past, present and future perfect continuous tense patterns

Appendix G

Auxiliary algorithms

The Algorithm 19 and 20 below are part of the Algorithm 12 for creating the Wh null elements.

Algorithm 19: Creating the Adjunct (circumstantial) Wh-traces

```
input  : wh-group, dg, cg
begin
    check the tense and modality for all the clauses
    for clause in cg: from the clause of wh-group to lowest
        /* create the adjunct trace in the first clause that has non
           present simple tense                                      */
        if clause tense is not present simple:
            create Adjunct Wh-trace for Wh-group
            return
end
```

Algorithm 20: Creating the Theta (participant) Wh-traces

```
input  : wh-group, dg, cg
begin
    get possible configurations for each clause from the PTDB
    /* check if the higher clause is a projection and and has an
       extra argument                                                 */
    for config in higher clause configurations:
        if (config is two role cognition and config takes expletive subject) or
          (config is three role cognition and clause is passive voice):
            higher is eligible ← True
            break
    /* check if the lower clauses might miss an argument              */
    for clause in lower clauses:
        for config in clause configurations:
            if number of clause theta constituents < number of config arguments:
                lower is eligible ← True
                break
    if higher is eligible and lower is eligible:
        if higher clause has "that" complementizer:
            create Object Wh-trace in the lowest clause
        else:
            if Wh-group has case:
                if Wh-group case is nominative(subjective):
                    create Subject Wh-trace in lowest clause
                else:
                    create Object Wh-trace in lowest clause
            else:
                create Wh-trace with Subject function and attempt to assign theta
                  roles
                if theta roles not successfully assigned in lower clause:
                    change the Wh-trace to Object function and assign theta roles
end
```

Appendix H

Annotation guidelines for OCD corpus

H.1 Constituency

The constituency annotation is based on the Cardiff grammar (Fawcett 2008) with some consulting of the traditional grammar (Quirk et al. 1985) for clarification; while MOOD systemic selections are based on the Sydney grammar (Halliday & Matthiessen 2013b). Below follows a set of short descriptions aimed at helping annotators identify main unit types and clause elements.

Clause – the main processing unit onto which meanings of different kinds are mapped and integrated into.

- the punctuation (.?!) at the end of the sentence(the matrix clause) shall be left outside the segment
- in clause complexes, the punctuation(,;“”-), conjunctions(and, or , but . . .) and other nexus markers(if, or,) shall be left outside the segment.

Finite – a part of the verbal group expressing the tense or modality. It either precedes the Predicator or is conflated with it in present and past simple tenses.

- If the finite is conflated with the predicate do not mark it
- The finite is the first auxiliary verb in the verb group before the subject in declarative clauses and the auxiliary that precedes the subject in interrogative clauses.

Subject – the nominal group or a nominal clause that precedes the Predicator in a clause and it is something by reference to which the proposition can be affirmed or denied. The subject is the nearest nominal group or clause that precedes the predicator.

Predicator – the part of verbal group minus the finite constituent when they are not conflated. It specifies additional temporal and aspectual relations, voice and the process type (e.g. action, relation, mental process etc.) that is predicated about the Subject. To enforce the syntactic and functional analysis proposed in the Cardiff analysis methodology (Fawcett 2008), the complex clauses need to be separated into individual clauses so that each comply with the "one main verb per clause" principle (see below). The predicator is the entire verb group(main verb + auxiliaries) minus the first auxiliary(which is the finite element)

Complement – the part of the clause that follows the Predicator and has the potential of becoming a Subject, i.e. it can become an axis of the argument. Usually it is a nominal group and rarely a prepositional phrase.

- The nominal group, prepositional group or clause that follows the Predicator
- exception are the copulative clauses (when the main verb is "to be"), then the adjectives following the verb receive complement function because they receive participant role of attribute (which is a quality)
- prepositions that can introduce complements are enumerated in table 1
- most clauses have 0 - 2 complements, exception are directional processes that can have 0-4 complements

Adjunct – do not have the potential of becoming a Subject; therefore arguments cannot be constructed around adjunct elements. They are realized by adverbial and prepositional groups.

- Prepositional phrases after the clause complements
- adverbs preceding and following the predicator
- Adjuncts can occur in front of the subject, then the clause becomes thematically marked.
- We do not annotate circumstantial adjuncts (bearing experiential information) but ONLY comment adjuncts (serving an interpersonal modification function in modality or appraisal). For more details see Schulz (2015) guidelines.

Markers – prepositions, conjunctions, expletives and verb particles. We do not annotate them. [Looking back at this rule I don't know what was behind this decision back in 2013].

Group – a set of words executing a particular function in a clause. The head of the group dictates the group type and the other words in the group may have other parts of speech and contribute to specifying enriching and further specifying the meaning of the group.

- note: we do not make distinction between a simple group and a group complex
- prepositional group is a nominal group preceded by a preposition

H.2 Clause partition

Follow the "One main verb per clause" rule. Use the semantic analisys to guide the sentence clause division into clauses.

When the clauses are connected by a conjunction and have their own subject/objects then the conjunction is the clause border marker.

(129) The lion chased the tourist but she escaped alive.

(130) The lion[Ag-Ca] chased[Pr] the tourist[Af-Pos]

(131) she[Ag] escaped[Pr] alive[Ra].

When the predicators are conjoined and share subject and/or objects then each predicator will form a new clause and borrow the subject/objects from the other clause.

(132) The lion chased and caught the tourist.

(133) the lion[Ag-Ca] chased[Pr] the tourist[Af-Pos]

(134) the lion[Ag-Ca] caught[Pr] the tourist[Af-Pos]

In the case of mental(e.g. know, think, feel, want, like), influential(e.g. start, stop, try, continue, fail) and event relating (e.g. cause) processes the predicates are often complex. Verbs in these classes are known as control and raising verbs (Haegeman 1991a) where a super-ordinate controls subordinate non-finite verb and binds its participants (Subject/Complement).

In order to comply with "one main verb per clause" principle, each Main Verb of the complex clause becomes a governor of a distinct clause. The subordinate verb with all of its dependent nodes is assigned to a place-holder. The super-ordinate verb receives the place-holder as Complement with the role of Phenomena. If the subject is missing in the subordinate clause then it is copied from the super-ordinate one.

(135) The lion wanted/began to chase the tourist.

(136) the lion[Cog] wanted/began[Pr] X[Phen]

(137) X= the lion[Ag-Ca] to chase[Pr] the tourist[Af-Pos]

The meaning of complex clause decomposition can be expressed with an equivalent rephrasing by inserting “something that is” between the Main Verbs, as in example below.

(138) The lion wanted/began something that is to chase the tourist.

H.3 The tricky case of prepositional phrases

There are cases in mood analysis when deciding the unit type is impossible by relying solely on syntactic analysis (including typed dependency analysis). Prominent cases are the prepositional phrases. These can fill both a Complement and an Adjunct role. For mood analysis this implies that the same syntactic unit can fill a Complement and an Adjunct, while for transitivity analysis, it implies that the same syntactic unit can fill a Participant or a Circumstance.

(139) John goes home through London.

(140) John is building a house for Bob.

(141) Her teardrop shines like a diamond.

(142) John is building a house for ten years now.

(143) John goes to London by fast train.

In examples 139 and 140 the prepositional phrases “through London”1 and “for Bob” are Complements and Participants (Path and Beneficiary roles) while in the latter examples “like a diamond”, “for ten years now” and “by fast train” are Adjuncts and Circumstances (of comparison, temporal duration and manner-means).

H.4 Making selection from the MOOD system network

Here is a brief description how to make selections in the MOOD system network. Agency is of primary concern in this work. The rest of the features are annotated as much as the time permits.

Agency is a clause level feature and is attributed depending on the experiential analysis of the clause (be it via Transitive or Ergative model). It expresses whether there is an active participant that brings about the unfolding of the process. Is the process brought about from within or from outside? With regards to this an important distinction to be made is between doings and happenings. The probing for doings is usually by asking the questions "What did X do (to Y)?" If there is a suitable X then we say that clause has an effective voice. Now effective voice can be (a) operative (that corresponds to active voice in mood analysis) and thus realised with an agent (doer) as subject or (b) receptive (that corresponds to passive voice in mood analysis) and thus goal/medium or beneficiary takes the subject place while the agent is given a secondary role. It can be either overt i.e. agentive (a prepositional phrase complement marked in English via "by" preposition) or covert i.e. non-agentive (unspecified).

Many intransitive verbs (with only one participant) are better analysed from Ergative perspective. In this case no external participant is implied to actuate the process and they are realised with a Medium subject. The best question to probe such clauses is by asking "What happened?". If this probing is more suitable than the one for doing the the clause receives middle selection in the agency system. The middle configurations do not have a active or passive voices thus voice can be used as a probing method (Halliday & Matthiessen 2013b: 336-354).

Modality should be annotated as described in Schulz (2015).

Remaining MOOD features should be annotated as described in (Halliday & Matthiessen 2013b).

Appendix I

Empirical evaluation data

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

Table I.1 The progressive binning scale considering the dataset properties

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)**
Main verb	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 I.2 Percentage of segments deviated to a given degree for major syntactic elements

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 I.3 Percentage of segments deviated to a given degree for major semantic elements

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 I.4 The evaluation statistics for the main constituency unit types

Element	**Matched**	**Manual nm**	**Parse nm**	**Precision**	**Recall**	**F1**	**%Total matched**	**%Manual nm**	**%Parse nm**
Main verb	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 I.5 The evaluation statistics for the clause main elements

Feature	**Matched**	**Manual nm**	**Parse nm**	**Precision**	**Recall**	**F1**	**%Total matched**	**%Manual nm**	**%Parse nm**
POLARITY-TYPE									
positive	485	125	55	0.90	0.80	0.84	89.48	20.49	10.19
negative	57	10	70	0.45	0.85	0.59	10.52	14.93	55.12
VOICE-TYPE									
active	553	102	68	0.89	0.84	0.87	98.05	15.57	10.95
passive	11	11	28	0.28	0.50	0.36	1.95	50.00	71.79
FINITNESS									
non-finite	99	19	38	0.72	0.84	0.78	15.84	16.10	27.74
finite	526	33	554	0.49	0.94	0.64	84.16	5.90	51.30
NON-FINITE-TYPE									
perfective	71	12	16	0.82	0.86	0.84	73.20	14.46	18.39
imperfective	26	9	24	0.52	0.74	0.61	26.80	25.71	48.00
DEICTICITY									
temporal	446	74	55	0.89	0.86	0.87	97.38	14.23	10.98
modal	12	33	6	0.67	0.27	0.38	2.62	73.33	33.33
MOOD-ASSESSMENT-TYPE									
temporality	35	17	27	0.56	0.67	0.61	56.45	32.69	43.55
modality	15	32	8	0.65	0.32	0.43	24.19	68.09	34.78
intensity	12	14	43	0.22	0.46	0.30	19.35	53.85	78.18
MOOD-TYPE									
indicative	455	216	37	0.92	0.68	0.78	99.13	32.19	7.52
imperative	4	1	31	0.11	0.80	0.20	0.87	20.00	88.57
INDICATIVE-TYPE									
declarative	355	260	27	0.93	0.58	0.71	88.31	42.28	7.07
interrogative	47	7	63	0.43	0.87	0.57	11.69	12.96	57.27
INTERROGATIVE-TYPE									
wh	40	6	57	0.41	0.87	0.56	88.89	13.04	58.76
yes-no	5	3	8	0.38	0.62	0.48	11.11	37.50	61.54
WH-SELECTION									
wh-subject	9	3	7	0.56	0.75	0.64	32.14	25.00	43.75
wh-adjunct	11	15	3	0.79	0.42	0.55	39.29	57.69	21.43
wh-complement	8	0	62	0.11	1.00	0.21	28.57	0.00	88.57

Table I.6 The evaluation statistics available for the Mood systems

Element	Match	Manual nm	Parse nm	Precision	Recall	F1	%Total matched	%Manual nm	%Parse nm
Participant role	2780	233	1002	0.74	0.92	0.82	49.54	7.73	26.49
Configuration	1376	127	493	0.74	0.92	0.82	24.52	8.45	26.38
Main Verb	1456	160	605	0.71	0.90	0.79	25.94	9.90	29.35

Table I.7 The evaluation statistics for the main semantic elements

Feature	Match	Manual nm	Parse nm	Precision	Recall	F1	%Total matched	%Manual nm	%Parse nm
PROCESS-TYPE									
mental	277	231	87	0.76	0.55	0.64	33.62	45.47	23.90
relational	338	297	174	0.66	0.53	0.59	41.02	46.77	33.98
influential	38	51	62	0.38	0.43	0.40	4.61	57.30	62.00
action	170	231	352	0.33	0.42	0.37	20.63	57.61	67.43
event-relating	1	28	0	1.00	0.03	0.07	0.12	96.55	0.00
RELATIONAL-TYPE									
attributive	169	239	107	0.61	0.41	0.49	70.42	58.58	38.77
directional	30	13	127	0.19	0.70	0.30	12.50	30.23	80.89
locational	39	20	207	0.16	0.66	0.26	16.25	33.90	84.15
matching	2	0	69	0.03	1.00	0.05	0.83	0.00	97.18
MENTAL-TYPE									
three-role-cognition	45	51	34	0.57	0.47	0.51	29.41	53.12	43.04
two-role-cognition	95	102	86	0.52	0.48	0.50	62.09	51.78	47.51
two-role-perception	13	12	102	0.11	0.52	0.19	8.50	48.00	88.70
three-role-perception	0	2	6	0.00	0.00		0.00	100.00	100.00
desiderative	0	0	81	0.00			0.00		100.00
emotive	0	0	87.00	0.00			0.00		100.00

Table I.8 The evaluation statistics available for the PROCESS-TYPE system and few of its subsystems from the Transitivity network

	Match	Corpus non-matched	Parser non-matched	Precision	Recall	F1
emoter	91	70	57	0.61	0.57	0.59
phenomenon	359	223	294	0.55	0.62	0.58
carrier	267	263	244	0.52	0.50	0.51
possessive	75	48	99	0.43	0.61	0.51
cognizant	82	84	104	0.44	0.49	0.47
agent	267	210	428	0.38	0.56	0.46
possessed	71	24	155	0.31	0.75	0.44
attribute	162	241	170	0.49	0.40	0.44
destination	18	10	121	0.13	0.64	0.22
affected	93	70	663	0.12	0.57	0.20
location	17	40	226	0.07	0.30	0.11
source	1	6	13	0.07	0.14	0.10
created-phenomenon	29	12	647	0.04	0.71	0.08
path	1	4	19	0.05	0.20	0.08
range	11	19	240	0.04	0.37	0.08
affected-carrier	33	28	899	0.04	0.54	0.07
affected-possessed	20	6	575	0.03	0.77	0.06
affected-cognizant	22	34	624	0.03	0.39	0.06
manner	1	7	25	0.04	0.12	0.06
perceiver	10	8	344	0.03	0.56	0.05
agent-carrier	17	12	745	0.02	0.59	0.04
created	3	15	228	0.01	0.17	0.02
matchee	1	2	91	0.01	0.33	0.02
agent-perceiver	6	1	657	0.01	0.86	0.02
agent-cognizant	2	3	633	0.00	0.40	0.01
affected-perceiver	1	0	544	0.00	1.00	0.00
affected-destination	1	0	593	0.00	1.00	0.00
affected-emoter	1	0	631	0.00	1.00	0.00
affected-path	0	0	494			
affected-source	0	0	490			

Table I.9 The evaluation statistics available for the PARTICIPANT-ROLE-TYPE system from the TRANSITIVITY system network