



Using role and reference grammar to support computer-assisted assessment of free-text answers

Elizabeth Guest
Sally Brown

Leeds Metropolitan University

Contents

Summary	3
1 Background	3
1.1 Parsing for Role and Reference Grammar	5
2 Methods	6
2.1 Outline of the parsing algorithm	6
2.2 Tagging	7
2.3 Stripping the operators	10
2.4 Basic chart parser	14
2.5 Parsing template	15
2.6 Parsing with fixed, free and constrained word order	17
2.7 Modifications to the parsing algorithm	17
3 Results	19
4 Conclusions and future work	24
References:	26
Appendix A: Results for parsing student texts	28
Appendix B: RRG templates for parsing student texts	38

Summary

Computer-assisted assessment (CAA) is increasingly being used in universities worldwide, but to date its use has mainly involved multiple-choice or other forms of constrained-choice questions and answers. Early work on assessing free text largely involves looking for word strings in text without attempting to ensure that the required words form part of connected and linguistically accurate text. The research covered in this report has attempted an analytical approach of free text involving breaking it down into its component grammatical elements. When further developed, this should allow the use of role and reference grammar, which will enable the checking of text for both content words and meaning. The results to date are promising, but considerable further work is required before this approach can be meaningfully applied to the assessment of student work.

1 Background

Numerous authors on effective assessment (Brown, Bull *et al.* 1997; Boud 1995; Brown and Knight 1994) argue that assessment is potentially of even greater importance with regard to its impact on the student learning experience than just about any other aspect of their pedagogic experiences:

“Assessment defines what students regard as important, how they spend their time and how they come to see themselves as students and then as graduates. Students take their cues from what is assessed rather than from what lecturers assert is important.” (Brown, Bull and Pendlebury 1997 p7)

In the current climate of increasing student numbers and decreased funding per student in many HEIs internationally, it is necessary to find economies of scale in teaching and supporting undergraduate students. Economies of scale are possible to a certain extent for lectures and tutorials, but this is less possible for assessment unless radical re-engineering of assessment processes is undertaken. As staff-student ratios decrease, the assessment burden on staff will increase unless alternative approaches are used.

One solution to this dilemma is to seek ways to mark student work automatically. This is being done at present using variations on multiple-choice questions, with a variety of innovative question types that test learning beyond simple recall. If designed correctly, these kinds of tests can provide students with immediate feedback on how well they are doing and can provide valuable formative pointers for further learning. Extensive evidence demonstrates that increased formative assessment can impact positively on student learning and retention (Sadler 1989; Sadler 1998; Rust 2002; Sambell and Hubbard 2004; Yorke 2001). However, it can be difficult to design if we want to make it truly an integral part of learning and if we want to avoid encouraging inappropriate student behaviour, such as random guessing of answers.

Considerable work has been undertaken in recent years to investigate and implement approaches to CAA that foster deep learning (Beevers, Foster *et al.* 1989; Brown, Race *et al.* 1999; Brown, Bull *et al.* 1997 *op. cit.*), but significant advances still remain to be made. Some have argued that it is currently possible to assess essays by automatic means, but we remain unconvinced since examples to date have been somewhat crude. However, it would be very helpful if it were possible automatically to mark short free-text answers using CAA approaches,

thus reducing the drudgery for markers. This would allow more scope in the setting of questions and would give students more opportunity to show what they understand and can do. Much research has been aimed at this question, but this generally either involves pattern matching (Sukkarieh, Pulman *et al.* 2003; Sukkarieh, Pulman *et al.* 2004) or latent semantic analysis (Wiemer-Hastings 2001; Landauer, Laham *et al.* 1997), or a combination of these (Pérez and Alfonsa 2005). These methods work to a certain extent, but because they are not based on the meaning of the text, they are quite easy to fool. For instance, latent semantic analysis can be fooled by writing down the right kinds of words in any order. The problem with current approaches to pattern matching, on the other hand, is that if the student writes down a correct answer in a different way, it will be marked wrong.

Our innovative approach is based on the grammatical tradition of ‘parsing’, which is breaking down language into its functional components like verbs, nouns and adverbs. Role and Reference Grammar (RRG) (Van Valin and LaPolla 1997; Van Valin 2005) is a relatively new linguistic theory that majors on predicates and their arguments. It separates the most vital parts of the sentence from the modifiers (adverbs, adjectives, auxiliaries and articles). This means that the core meaning can be extracted first, and then the modifiers fitted in at a later stage. As long as the arguments and the verbs are in the correct order for English (subject, verb, object) then the sentence can be understood. It doesn’t matter if (for example) Chinese students forget the articles, the sentence can still be parsed and the meaning extracted. The core meaning of the sentence is extracted via the use of templates. This makes it easier to extract the important parts of the meaning of the sentence: we just need to identify the predicate and the arguments, which are clearly labelled branches within the templates.

In this work we describe a method for using the RRG paradigm for parsing student texts, which do not have to be grammatically correct. This work can be used as a pre-processing step to those methods that use latent semantic analysis or pattern matching. There is evidence to suggest that latent semantic analysis gives better results when the subject, verb and object of the sentence is used rather than an unstructured “bag of words” (Wiemer-Hastings 2001). Our method will provide a mechanism for extracting some structure. If structure can be extracted, then this structure can also be passed to a pattern matcher, which will decrease the number of possibilities that have to be included. This method will also enable accurate marking of ungrammatical sentences.

Role and Reference Grammar (RRG) (Van Valin and LaPolla 1997; Van Valin 2005) is a promising theory for extracting the meaning from sentences from a computational viewpoint for several reasons. Because it posits multiple projections, the various aspects of a sentence can be dealt with separately. For example, words that modify other words are removed from the constituent projection and placed in an operator projection, which shows how these words modify the meanings of the words in the constituent projection. As a result, only the main constituents of a sentence have to be parsed, simplifying the parsing process. RRG has a strong link with semantics, and the grammatical constructs are designed both to be cross-linguistically valid and to make the meaning relatively easy to extract. The grammatical constructs are based on templates rather than rules. This means that more information can be encoded into the grammatical construct, which in turn makes the meaning easier to extract.

However, there are aspects of RRG that make it harder to implement. It is much harder to parse with templates than with rules; RRG templates are particularly hard because lines are allowed to cross and the parse trees are not simply made up of parents and children, but nodes can

have modifiers (such as PERIPHERY) attached to them. In addition, although RRG says nothing explicit about word order constraints, they are implicit in the templates in that the theory contains examples from many languages that include fixed and free word order, and varieties in between.

Existing parsing techniques are not suited to solve all these problems without modification. Most parsing algorithms, such as those based on Head-Driven Phrase Structure Grammar (HPSG) and probabilistic context-free grammar, are based on rules. Although templates can easily be reduced to rules, a lot of information is lost. The other main problem is the word order flexibility. The two most popular parsing grammars are HPSG (Wahlster 2000; Hou and Cercone 2001; Kešelj 2001) and dependency grammar (Covington 2003). HPSG is good for fixed word order, but poor for free word order; dependency grammar is good for free word order, can do fixed word order, but it is difficult to enforce other constraints. Which of these methods is chosen depends on the language: dependency grammar is popular for free word order languages such as Czech (Holan 2002) and Korean (Chung and Rim 2003; Chung and Rim 2004), while HPSG is the method of choice for many other languages. However, some languages, such as Japanese, have had work done using both methods (Ito, Hori *et al.* 2001; Kanayama, Torisawa *et al.* 2002; Wahlster 2000). Other methods exist that match the spirit behind RRG more closely. For example, Grootjen (2001) classifies words into three classes, predication, modification and qualification, and relates these words in a 'bottom-up' fashion. This method works quite well on English headline titles, but has not been tested on more complex sentences. Galicia Haro *et al.* (2001) combine dependency and constituency resources. Their method is based on a standard context-free grammar, government patterns and a semantic network that provides the relationship between words. However, it still suffers from the problems of rules and varying degrees of word order flexibility.

In this paper, a new parsing method is described. This method is based on a chart parser, which is an old but effective parsing algorithm that can be modified to handle the unification used in HPSG (Jurafsky and Martin 2000). The chart parser has been modified to handle both templates and varying degrees of word order flexibility and is used to parse the main constituents once the modifying words have been taken away. Modifying words are removed via a relatively simple process that finds related words using information about the grammar of the language stored in a file.

1.1 Parsing for Role and Reference Grammar:

Role and Reference Grammar (RRG) (Van Valin and LaPolla 1997; Van Valin 2005) was developed as a result of asking the question: "What would a linguistic theory look like if it was based on Lakhota and Tagalog rather than English?" The result is a theory that is suited to describe a huge range of languages, including English. Of all the linguistic theories, it is most closely related to functional grammar, but there are important differences.

Role and Reference Grammar posits algorithms to go from syntax to semantics, and semantics to syntax. The main contribution is the use of parsing templates and the notion of the CORE. A CORE consists of a predicate (generally a verb) and (normally) a number of arguments. It must have a predicate. Everything else is built around one or more COREs. Simple sentences contain a single CORE; complex sentences contain several COREs.

The fact that RRG focuses on COREs, means that the semantics is relatively easy to extract from a parse tree. You just have to look for the PRED and ARG branches of the CORE to obtain the predicate (PRED) and the arguments (ARG). Who did what to whom will depend either on the ordering of the ARG branches (in the case of English), or on their cases, or both.

RRG makes extensive use of templates. These templates consist of whole trees and are thus harder to use in a parsing algorithm than rules. The templates can easily be reduced to rules, but only at a loss of much important information. The first example in Figure 1 consists of one large template that gives the overall structure and some simple templates (which are equivalent to rules) so that elements such as NP and PP can be expanded. An NP is a noun phrase and in this theory consists of a noun, pronoun or question word. Templates are required to parse complex noun phrases, such as those with embedded clauses. A PP is a prepositional phrase and consists of a preposition followed by a NP. Clearly, if we reduce the large template in the example in Figure 1 to the rule $CLAUSE \rightarrow NP1 V2 NP ADV/PP$, a lot of the information inherent in the structure of the template is lost. A further feature of RRG is that the branches of the templates do not have to have a fixed order and lines are allowed to cross. The latter is important for languages such as German and Dutch, where the adverb that makes up the periphery normally occurs within the core. This feature will be important in our application for marking work by students for whom English is not their first language.

The above features pose challenges for parsing according to the RRG paradigm. We have overcome these challenges by making some additions to the standard chart parsing algorithm. The main innovations are:

- a modification to enable parsing with templates

- a modification to allow variable word order.

In addition, parsing also includes elements of dependency grammar to find operators and to determine which word they belong to. At present the most popular methods of parsing are HPSG (Wahlster 2000; Hou and Cercone 2001; Kešelj 2001) and dependency grammar (Holan 2002; Covington 2003; Chung and Rim 2004). HPSG is good for fixed word order languages, and dependency grammar is good for free word order languages. The approach to parsing described below is novel in that it allows parsing with templates and because of the range in flexibility of word order allowed.

2 Methods

2.1 Outline of the parsing algorithm:

The parsing algorithm relies on correctly tagged text. We use Toolbox (available from SIL: www.sil.org/computing/toolbox) to tag sentences. Toolbox is a semi-interactive tagging program. It was chosen because the user can define their own tags and because it is easy to ensure all tags are correct. This is a good program to use for experimentation. Once the tags have been finalised an appropriate automatic tagger can be used, or written using standard techniques. Details of the tags used are given below, and the dictionary used to tag sentences will be found in Appendix C.

a complex data structure in such a way that relevant parts are deleted as parsing progresses, leaving the correct option according to the data.

3. Draw the resulting parse tree.

Details of the extensions to the chart parser are given below.

2.2 Tagging:

The main things to bear in mind when designing a tagging schema for Role and Reference Grammar are:

1. It should be easy to separate the operators from the constituents.
2. It should be easy to distinguish between different classes of operators.
3. Words denoting discourse features is words that link sentences together, need to be handled in a sensible way as these do not feature in the standard RRG description.
4. Words that join clauses together need to be labelled in such a way that the labelling helps to identify the structure.

The constituents that have operators are nouns and verbs. The tags for verbs are:

V1	A verb that takes one argument
V12	A verb that can take either one or two arguments
V2	A verb that takes two arguments
V23	A verb that can take either two or three arguments
V3	A verb that takes three arguments
BE	The verb "to be"
DO	The verb "to do"
HAVE	The verb "to have"

The verbs "be", "do" and "have" are tagged separately because they can function either as main verbs in a sentence or as auxiliaries to other verbs.

Operator tags for verbs are:

PAST	Past-tense marker (-ed)
CONT	Continuous-aspect marker (-ing)
AUX	Auxiliary ("would", "should", "may" etc)
NEG	The word "not"
HAVE,BE&DO	The verbs "to be", "to do", and "to have"

There are just four tags for nouns:

N	Noun
PN	Proper noun
PRO	Pronoun
POSSPRO	Possessive pronoun

The tags for nouns are (but note that PRO and POSSPRO do not have operators):

DET	Determiners ("a", "an", "the")
-----	--------------------------------

QNT	Quantifier (“all”, “some” etc)
NUM	Number
DEM	Demonstrative (“this”, “these” etc)
ADJ	Adjective
NEG	The word “not”

The other tags are:

ADV	Adverbs, including words like “tomorrow” and “yesterday”
P	Preposition
P-BY	The preposition “by”. This is included separately because of its importance in the passive
P-V	Prepositions that occur with verbs, e.g. “down”, “out”
DISC	Discourse words, e.g. “also”
CONJ	Conjunction
RSN	Word denoting reason (e.g. “because”, “so that”)
ETHR	The word “either”
IF	The word “if”
SUBCONJ	The word “that”
EX	Word denoting an example (e.g. “such as”)
WH1	WH words that necessitate a pre-core slot (“what”, “which”)
WH2	WH words that do not necessitate a pre-core slot (“why”, “where”, “who”)

Note that in RRG, the key constituent of a sentence is the CORE, which consists of a predicate and its arguments. In English, there is a strict word order to the constituents of the CORE: subject, verb, object. In questions, the object comes first if this is the subject of the question. For example, if we have the sentence “John ate pizza”, we can ask the questions: “What did John eat?” and “Who ate pizza?” In the latter question, there is no change within the CORE. “Who” is the subject and we just need to fill in that slot. In the first question, however, “what” appears at the start of the sentence. As in English, this is the object and would normally appear at the end of the sentence; RRG theory says that this is outside the CORE. Since it is still really part of the CORE it goes in a ‘pre-core slot’ (PrCS). The pre-core slot simply tells us that something that is normally found within the CORE has been moved outside. This is useful information because we can then investigate why it has been moved outside and discover, for example, that we have a question.

The other RRG constituent that is important for English is the ‘left detached position’ (LDP). We can say “John ate pizza yesterday” and “Yesterday, John ate pizza”. In RRG theory, the first sentence gives us the canonical form and the fact that the position of “yesterday” has changed in the second sentence is signalled by putting it in a left detached position. This is useful for working out the emphasis of a sentence. Note that “yesterday” is not considered to be part of the CORE, but is peripheral information. It therefore goes in a PERIPHERY.

The concepts of pre-core slot and left detached position (LDP) are illustrated in Figure 2.

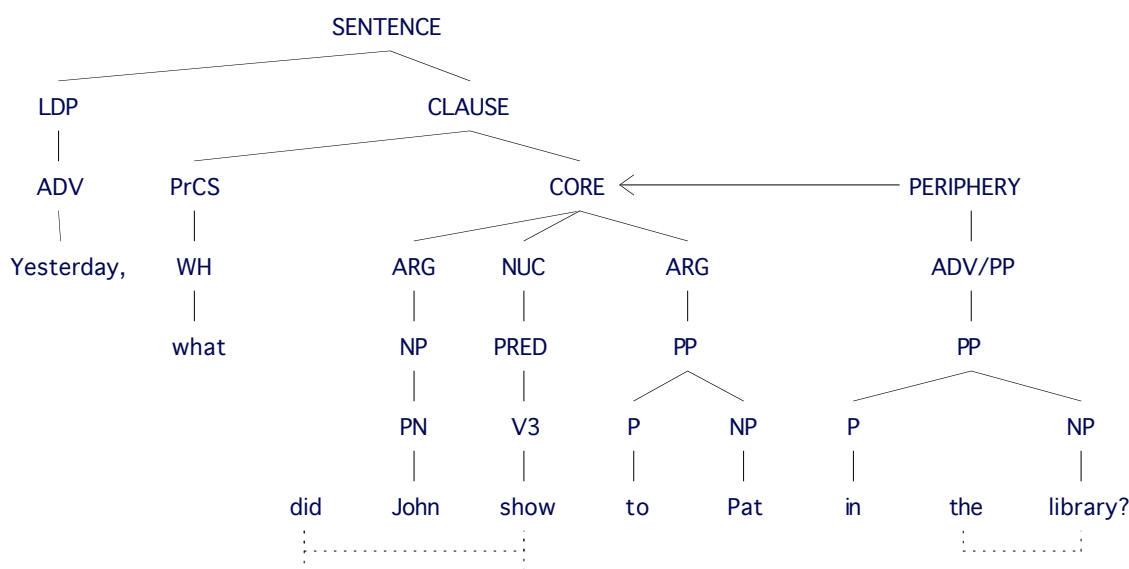
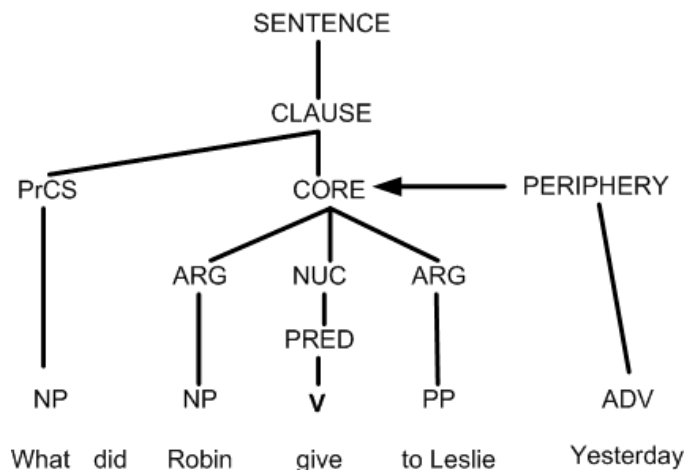


Figure 2: RRG trees showing the use of the left detached position (LDP), pre-core slot (PrCS) and PERIPHERY

2.3 Stripping the operators:

Before a sentence is parsed, its words are divided into predicates and operators, and the operators are attached to their predicates. Predicates are words that are modified by operators. So, for example, adjectives are operators for nouns. In many languages, operators do not always occur immediately next to their predicates. For this reason, the scope of the search for the operators for a particular word may include the whole of the clause. However, this is not always the case. For example, in English the modifying words for nouns always occur immediately before the noun.

In order to provide some information about where operators may occur for any given language, the predicates and their associated operators are specified in a grammar file. Examples of such a file for English and Dyrirbal are given in Figure 3. We include information about languages other than English in this section because RRG was developed with languages other than English in mind, and because we would like our parser to handle other languages.

```

#
# OPGrammar file for English
#

@IGNORE 3S PL

@PREDICATE N PN
@OPERATORS DET QNT ADJ NUM
@SKIP CONJ
@INSERTHEAD CONJ DET NUM :
ADJ
@TYPE BEFORE CONTIGUOUS

@PREDICATE V V1 V2 V3 V12 V23
@OPERATORS AUX PAST CONT NEG
@TYPE BEFORE NON_CONTIGUOUS
@TYPE AFTER CONTIGUOUS

@NEXT_PASS

@PREDICATE AUX
@OPERATORS PAST NOT

```

```

#
# OPGrammar file for Dyirbal
#

@IGNORE G.1 G.2

@PREDICATE N
@OPERATORS DEIC
@DEFAULT CASE C.ABS
@TYPE BEFORE
NON_CONTIGUOUS
@TYPE AFTER NON_CONTIGUOUS
@TYPE AGREEMENT CASE

@PREDICATE V V1 V2 V3 V12 V23
@OPERATORS TNS

```

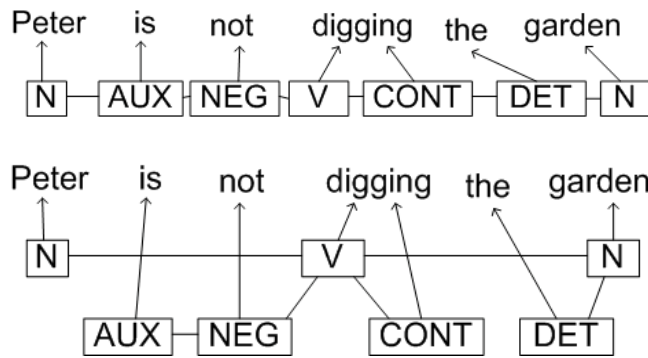
Figure 3: Grammar files for English and Dyirbal

This file also specifies the strategy for locating operators. There are four search types:

- BEFORE CONTIGUOUS
- AFTER CONTIGUOUS
- BEFORE NON_CONTIGUOUS
- AFTER NON_CONTIGUOUS.

Up to two of these types will be needed depending on whether or not the operators occur before or after their predicate and whether they are next to (CONTIGUOUS) or separated (NON_CONTIGUOUS) from their predicate. For example, operators for English nouns occur immediately before the noun so the search strategy is BEFORE CONTIGUOUS; for French, where adjectives can occur before or after the noun, two search strategies are required: BEFORE CONTIGUOUS and AFTER CONTIGUOUS. For languages with free word order, such as Dyirbal, where operators can occur anywhere in the sentence, the search strategy is BEFORE NON_CONTIGUOUS and AFTER NON_CONTIGUOUS.

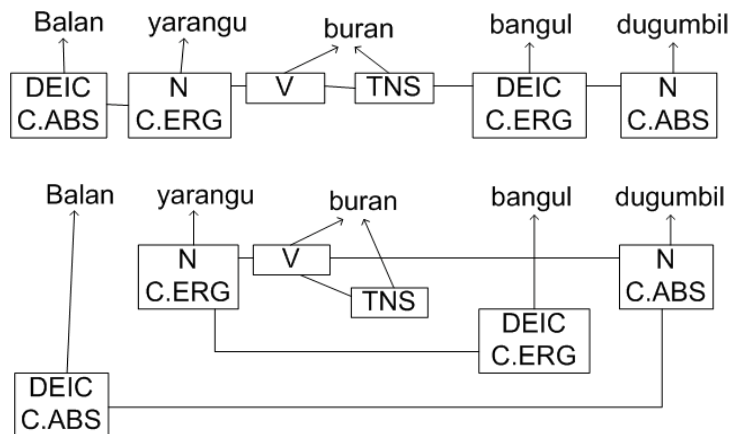
The algorithm for attaching operators to predicates is as follows. The sentence is searched from left to right. When a predicate tag is found, the search strategy for this tag is employed to locate any operators. If the search strategy is BEFORE CONTIGUOUS, then the search looks back from the predicate and attaches all words with appropriate operator tags to the predicate. The search stops when a word with a tag that is not in the list of operator tags is found. If the search is NON_CONTIGUOUS then the search continues until the start of the clause is reached. Note that as operators are attached to their predicates, they are removed from the sentence. An example of removing operators from an English sentence is shown below:



Notice that when the operator removal process is complete, only the tags N V N are left in the sentence. This is clearly much easier to parse than the original list of tags.

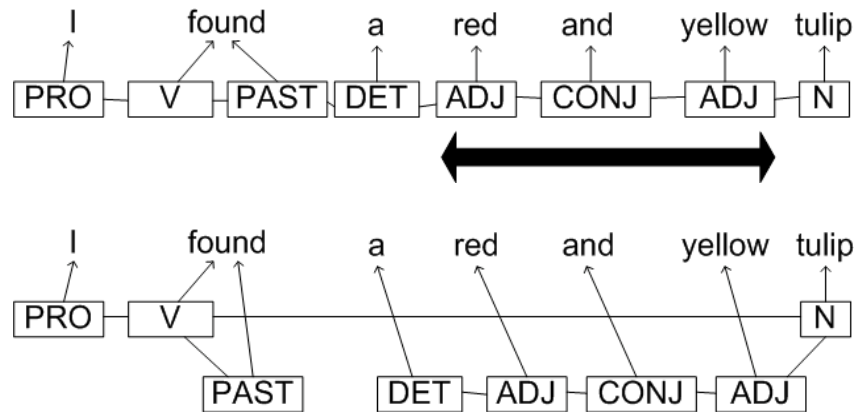
An example for Dyrirbal is shown below. The sentence means "The man (yarangu) saw the woman (dugumbil)".

Balan			dugumbil	buran		Bangul			yarangu
ba	-la	-n	dugumbil	bura	-n	ba	-ngu	-l	yara -ngu
DEIC	-C.ABS	-G.2	N	V2	-TNS	DEIC	-C.ERG	-G.1	N -C.ERG
Deic	-ABS	-G.2	woman	see	-TNS	deic	-ERG	-G.1	man -ERG

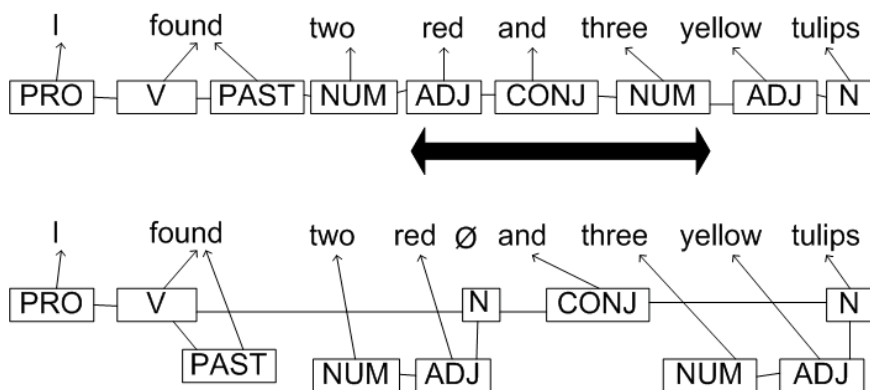


Notice how the nouns are matched up according to their cases. This is done with the additional search type AGREEMENT CASE. In Dyrirbal, absolutive case is the default case and is often not marked. This means that the absolutive case has to be inferred for "dugumbil". This is achieved with the additional line @DEFAULT CASE C.ABS.

Noun phrases can be quite complicated. A couple of extra search routines have been added to handle things like a "red and yellow tulip" or "two red and three yellow tulips". In the first instance the conjunction "and" is used to join two adjectives. Normally "and" would mark the end of the noun phrase because it is often used to join two clauses. This source of confusion is solved by the addition of @SKIP CONJ. This line means that if you find CONJ between two identical tags from the list of operator tags then, skip over CONJ, and include it in the list of operators. An example of the use of this feature is shown below:

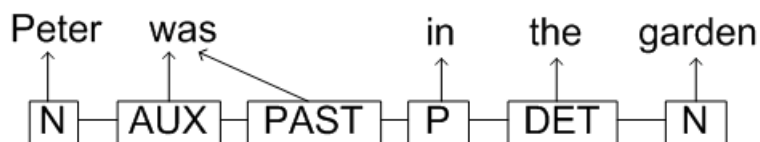


In the second case, “two red and three yellow tulips”, there is a noun missing. This will not parse correctly unless the missing noun is added. The addition of the search routine @INSERTHEAD solves this problem: @INSERTHEAD CONJ DET NUM : ADJ. This means that if you find a CONJ preceded by a DET or a NUM and followed by an ADJ, then insert a predicate tag before the ADJ. The predicate tag is given by the one currently under investigation. Note that this follows the search direction, which is from right to left in this example:

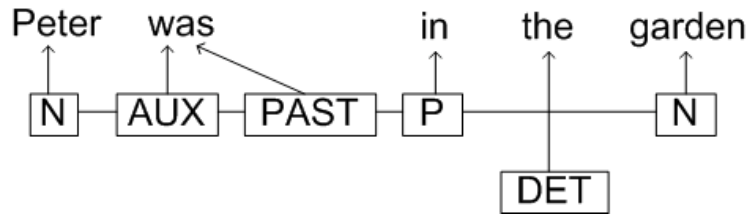


For languages (such as English and many other European languages) that use auxiliaries both as modifiers of a main verb and as a stand alone verb when there is no main verb, a single pass will not pick up the auxiliary as a predicate because AUX is listed in a list of operators, not predicates. To solve this problem, provision is made for more than one pass through the sentence. On the first pass, main verbs are sought and their operators are attached to them. The second pass can mop up any AUX that are left over.

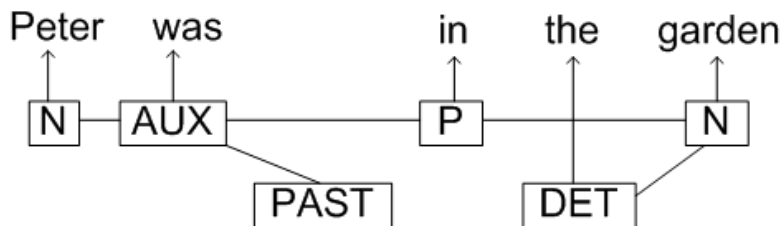
Example:



The first pass simply removed the DET and attaches it to the N.



On the second pass, AUX is identified as a predicate and PAST is attached to it.



2.4 Basic chart parser:

A chart parser, or the Earley algorithm, is an old, but efficient method for parsing. It is a method of ensuring that all appropriate rules get applied to the data, so that all possible parses are found. This is done in a systematic way. The chart parser was chosen as the basis for RRG parsing because of its simplicity and efficiency. However, it has been modified to handle templates (rather than rules) and variable word order flexibility. Note that it does not attempt to model how people parse sentences.

The chart parser is described in most books on natural language processing (e.g. Allen 1995; Jurafsky and Martin 2000), but we include it here so that the modifications can be better understood. The modified chart parser took as its basis a code developed by David Perelman-Hall and Jamshid Afshar. The code can be obtained from the internet via a search for the file "chartp10.zip". The basic chart parsing algorithm outlined here describes the implementation found in this code.

A chart parser consists of two main objects: a chart and an agenda. It is designed to parse rules that are contained in a list. The agenda and the chart contain "edges". Edges contain a start index and a finish index (which refer to the position in the sentence), a parse tree and a list of items that need to be satisfied for the rule to be complete. For example, given the sentence "John ate pizza", three initial edges are added to the agenda:

```
[0] john [1]
[1] ate [2]
[2] pizza [3]
```

where the numbers indicate the positions (start and finish) in the sentence. These edges are "inactive" because the list of items that needs to be satisfied is empty. Edges that have items that need to be satisfied are "active".

The agenda holds a list of edges to be considered. These edges are fed to the chart one at a time for processing. The result of processing an edge is that this edge gets added to the chart,

and often other edges are generated and placed on the agenda. New edges are generated either by combining the edge with another edge in the chart or by locating rules that apply to the edge that has just been added. When an edge is combined with another edge a (partial) parse tree is generated, often by combining two partial parse trees and the start and finish indices are updated because the rule will now cover more of the sentence.

The chart consists of a list of active edges and a list of inactive edges. This distinction is important because active edges can only combine with inactive edges and vice versa. The list of rules is consulted only when an inactive edge is added to the chart. If an applicable rule is found, a new edge is created from the rule and added to the agenda. After an edge has been processed, the chart takes the last edge that was added to the agenda. If this last edge was a rule, then the rule edge immediately gets combined with the last edge that was processed.

When the agenda is empty and there are no more edges to process, the parse trees can be obtained from the chart by looking for inactive edges with the appropriate goal (generally S or SENTENCE) at the top of the tree.

The important methods in this process are those for combining edges and those for finding rules that apply. It is these methods that have been modified in order to enable parsing with templates and parsing with varying degrees of word order flexibility.

The actual grammar file used for English is shown in Figure 4. There are a couple of extra commands in this file: COLLECT and CHANGE. The command COLLECT simply puts all the DISC tags into a list and then adds them at the top level of the SENTENCE when parsing is complete. This seemed to be the most sensible of way of dealing with DISC words that link sentences together. The CHANGE command is included so that passive mood can be picked up. The lines:

```
@CHANGE VP23 V23
@BEFOREOPS BE
@AFTEROPS PAST
```

mean that the tag V23 should be changed to VP23 if the tag BE is found before the tag V23 and the tag V23 has the PAST tag attached to it. All this is saying is that if you have a verb that is in past tense and has the auxiliary “to be” attached to it, then you have passive mood, and this signalled by changing the verb tag to VP instead of V.

2.5 Parsing templates:

The reason for parsing with templates rather than rules is that templates contain a lot more information. In addition, RRG contains peripheries and links that do not fit into trees in the normal way but via arrows, as illustrated in Figure 3, which shows an automatically generated parse tree. Also, by using a template, it is easier to ensure that in sentences with a pre-core slot (PrCS), for example, an argument really is missing from the CORE. However, parsing with templates is much harder than with rules.

```

#
# OPGrammar file for English
#
@IGNORE 3S PL

@COLLECT DISC

@PREDICATE V V1 V2 V3 V12 V23 BE DO HAVE
@OPERATORS PAST CONT
@TYPE AFTER CONTIGUOUS

@NEXT_PASS

@PREDICATE N PN
@OPERATORS DET QNT ADJ NUM NEG DEM
@INSERTHEAD CONJ DET NUM : ADJ
@TYPE BEFORE CONTIGUOUS

@PREDICATE V V1 V2 V3 V12 V23
@OPERATORS NEG AUX ADV HAVE BE DO
@TYPE BEFORE CONTIGUOUS
@OPERATORS NEG
@TYPE AFTER CONTIGUOUS

@CHANGE VP2 V12 V2
@BEFOREOPS BE

@AFTEROPS PAST

@CHANGE VP23 V23
@BEFOREOPS BE
@AFTEROPS PAST

@CHANGE VP3 V3
@BEFOREOPS BE
@AFTEROPS PAST

@NEXT_PASS

@PREDICATE HAVE BE DO NEG
@LAST
@OPERATORS NEG AUX HAVE BE DO
@TYPE BEFORE CONTIGUOUS
@OPERATORS NEG
@TYPE AFTER CONTIGUOUS

@PREDICATE ADV
@OPERATORS QNT NEG
@TYPE BEFORE CONTIGUOUS

@PREDICATE ADJ
@LAST
@OPERATORS QNT ADJ
@TYPE BEFORE CONTIGUOUS

```

Figure 4: The complete grammar file for English

Templates are parsed by collapsing all the templates to rules and then rebuilding the correct parse tree once parsing is complete. This is done by including the template tree in the rule, as well as the left- and right-hand sides. When rules are combined during parsing, we make sure that the right-hand side elements of the instantiated rule, as represented in the partial parse tree, point to the leaves of the appropriate rule template tree. This is especially important when the order of the leaves of the template may have been changed. The reference number for the rule that has been applied is also recorded so that it can be found quickly.

Modifying nodes, such as PERIPHERY, cause problems with rebuilding the tree. This is because such nodes can occur anywhere within the template, including at the root and leaf levels. Also, if we are dealing with a sub-rule whose root node in the parse tree has a modifying node, it is not possible to tell whether this is a hangover from the previous template, or part of the new template. To solve this problem, modifying nodes have flags to say whether they have been considered or not. There is a potential additional problem with repeated nested rules: if processing is done in the wrong order, the pointers to the rule template tree get scrambled. To overcome this problem, each leaf of a template is dealt with before considering sub-rules.

The algorithm for building the tree is:

1. Get the appropriate rule and rule template tree.
2. If the rule tree is of depth 1 and has no embedded modifying nodes (that is modifying nodes that point to a node other than the root), then simply continue by looking at each of the children in turn, starting at step 1.

3. If the rule tree is of depth greater than 1 or there are embedded modifying nodes, then make the rule template tree point to the appropriate places in the parse tree. This is done using the links made from the parse tree to the rule template tree during parsing. Note that the parse tree will consist of simple rule structures of depth 1 and modifying nodes will show up as children.
4. Clear all the children in the parse tree. This will have the effect of removing any embedded modifying nodes.
5. Copy all the children of the template tree and copy into the appropriate place in the parse tree.
6. If the template has modifying nodes, copy that part of the template tree and insert into the appropriate place in the parse tree.
7. Replace the leaves of the copied template trees with the original leaves. This is possible because the template leaves are pointing to the original leaves (step 3).
8. Consider each leaf in turn, modifying the parse tree as above (start at step 1 for each leaf).

2.6 Parsing with fixed, free and constrained word order:

There were two main problems to solve in order to modify the chart parser to handle varying degrees of word order flexibility:

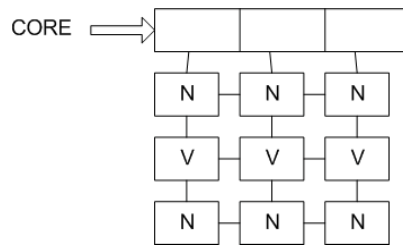
- working out a notation for denoting how the word order can be modified
- working out a method of parsing using this notation.

The first was achieved by the following notation on the ordering of the leaves of the template, treating the template as a rule:

1. Fixed word order: leave as it is [N V N].
2. Free word order: insert commas between each element [N, V, N]. (Note that case information is included as an operator so that the undergoer and actor can be identified once parsing is complete.)
3. An element has to appear in a fixed position: use angular brackets: [N, <V>, ADV] this means that N and ADV can occur before or after v, but that V *must* occur in second position. Note that this is second position counting constituents, not words.
4. Other kinds of variation can be obtained via bracketing. So, for example, [(N, V) CONJ (N, V)] means that the N's and V's can change order, but that the CONJ must come between each group. If we had [(N,V),CONJ,(N,V)] Then the N's and V's must occur next to each other, but the group do not have to be separated by the CONJ, which can occur at the start, in the middle or at the end, but which cannot break up an [N,V] group.

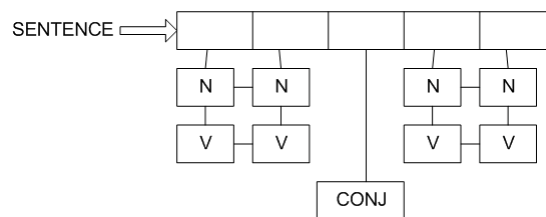
2.7 Modifications to the parsing algorithm:

Parsing was achieved via a structure that encoded all the possible orderings of a rule. So, for example, the rule CORE→N, V, N would become:



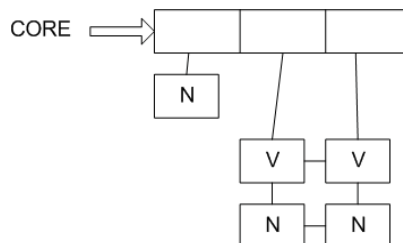
This means that N or V can occur in any position, and N has to occur twice. The lines between the boxes enable the “rule” to be updated as elements are found.

Using this schema, $\text{SENTENCE} \rightarrow (\text{N}, \text{V}) \text{ CONJ} (\text{N}, \text{V})$ would become:

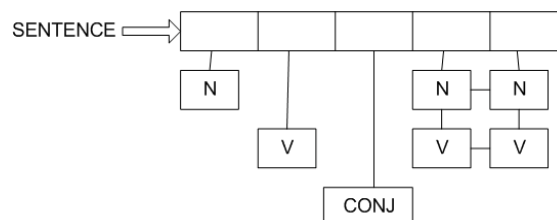


In this case, the CONJ in the middle is by itself because it has to occur in this position as the grouping word order is fixed. The groupings of N’s and V’s show where the free word ordering can occur.

To apply a rule, the first column of the left-hand side of the rule is searched for the token. Any tokens that do not match are deleted along with the path that leads from them. In the first example, after an N is found, we would be left with:



In the second example, after an N is found we would be left with:



Note that in order for the rule to be satisfied, we *must* find a V and then a CONJ: there are no options for position 2 once the element for position 1 has been established.

In this way, we can keep track of which elements of a rule have been found and which are still to be found. Changes in ordering with respect to the template are catered for by making sure that all instantiated rules point back to the appropriate leaves of the rule template, as described above.

The different possibilities for each rule are obtained via a breadth first search method that treats tokens in brackets as blocks. Then the problem becomes one of working out the number of ways that blocks of different sizes will fit into the number of slots in the rule.

3 Results

The above algorithm was tested on read student texts that formed part of a statistics assignment. The task was to identify biases in a certain scenario and to suggest a less biased way of collecting data. There are over 100 of these texts, but as tagging is fairly time consuming, to date only 15 of these texts have been considered. Some texts were discarded because they had some formatting in them, such as bulleted lists. Of those texts we could use, we used the first 45 sentences (six complete texts) to design a set of templates to parse these sentences. The complete list of templates used will be found in Appendix B, along with a few comments where these differ from standard RRG theory. These changes were necessary to ensure correct parses and to reduce the number of parses found. Note that this is the first attempt to use RRG theory to parse texts. We managed to parse all but one sentence, and this sentence has an ungrammaticality in it, which is hard to work around. We then applied these templates to the next six texts (47 sentences) to see how well our current set-up works. Of these sentences, 15 were parsed correctly. This is about 32%, or 36% if you discount the ungrammatical sentences that did not parse. Many of the sentences that did not parse did not do so because their structure had not been modelled. This is not a surprise given that the templates had only been determined using six texts.

In addition, the lengths of sentences vary from seven words to 63 words, and the distribution can be seen in the graph in Figure 5, which shows the frequency of the number of words in the sentences. This shows that most of the sentences are long and thus have a high degree of complexity. If we add this to the fact that many kinds of ungrammaticalities are handled, we have to conclude that the results are promising and it is worth continuing with this work.

Although, the preliminary results of applying these algorithms to student texts are very promising, some issues have been highlighted. The method parses relatively simple sentences correctly, and the main arguments and verbs are found. In addition, some very long and complicated sentences are parsed correctly, and many kinds of grammatical errors do not cause any problems. A complete set of results can be found in Appendix A.

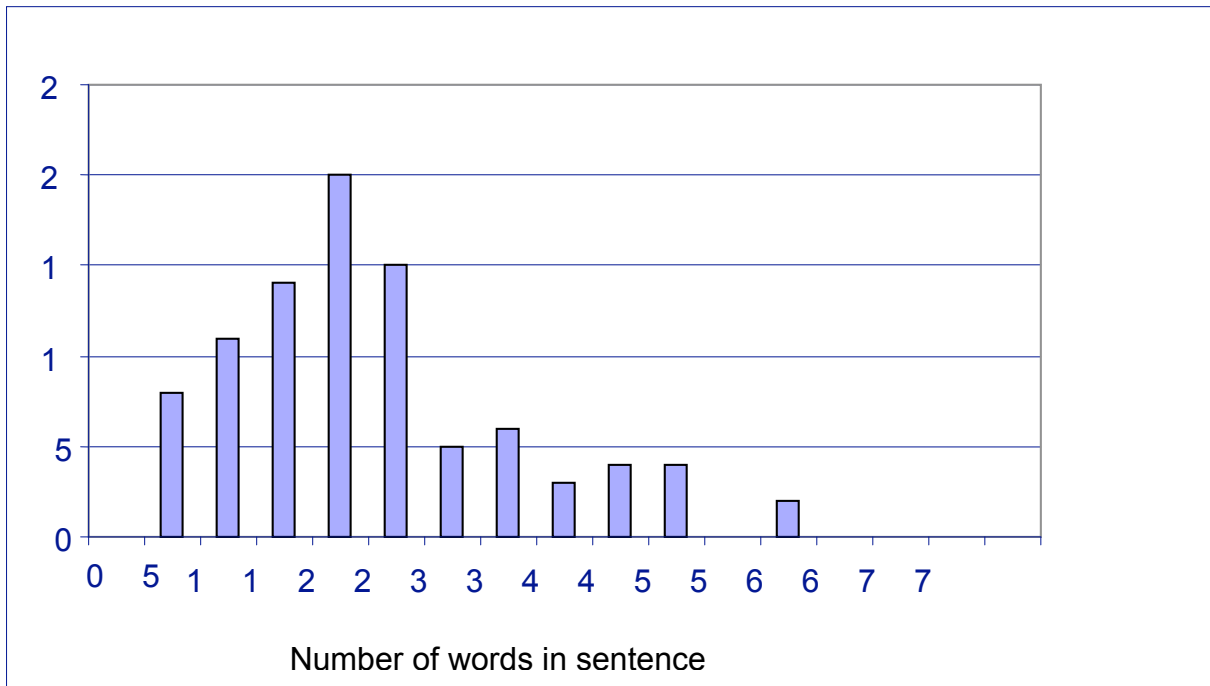


Figure 5: Graph showing the frequency of sentence lengths in the analysed texts

An example of a correctly parsed sentence is: “I would target main areas populated by students and would attend the same place at different times and during the day.” The parse tree for this example is given in Figure 6. Note that the complex object “main areas populated by students” has been parsed correctly and that the tree attaches the qualifying phrase to “area” so that it is clear what is being qualified. An important source of ambiguity in English sentences is caused by prepositional phrases, and this is a main cause of multiple parses of a sentence. In this example, the phrases “at different times” and “during the day” are placed together in the periphery of the CORE, although arguably they should have a different structure. This is a design decision to limit the number of parses. This kind of information needs semantic information to sort out what attaches to what. This cannot be obtained purely from the syntax.

An example of an ungrammatical sentence that is correctly parsed is: “Results from the observations would be less bias if the sample again was not limit the students in the labs between 9:30 and 10:30 on a Thursday morning.” (The parse tree for which is given in Figure 7.) This sentence parses correctly because the affix that should be on “limit” is an operator and the correctness of the operators is not checked during the parsing process. The word “bias” is labelled as a noun and gets attached as the second argument to “would be”, although it should be “biased”, which would get it labelled as an adjective. Despite these errors, the meaning of the sentence is clear and the parse will enable the meaning to be deduced.

The following sentence produces two parses (one correct and one incorrect): “Therefore, asking only the students present on a Thursday morning will exclude all the students that either have

no lessons or are not present.” The incorrect parse breaks up “Thursday morning” to give two clauses: (1) “Asking only students present on a Thursday” and (2) “Morning will exclude all the students that either have no lessons or are not present.”

In the first clause, the subject is “asking only students”, the main verb is “present” and the object is “on a Thursday morning”. This does not make sense, but it is syntactically correct as far as the main constituents are concerned. Similarly, the second clause is also syntactically correct, although it does not make sense. There are two ways of eliminating this parse. The first is to do a semantic analysis; the second is to not allow two clauses juxtaposed next to each other without punctuation such as a comma. However, students tend to not be very good at getting their punctuation correct. The current implementation of the parsing algorithm ignores all punctuation other than full stops for this reason. In fact, there is a trade-off between allowing the system to parse ungrammatical sentences and the number of parse trees produced. More flexibility in grammatical errors increases the number of parse trees.

An issue that makes parsing problematic is that of adverbs. These tend to be allowed to occur within several places within the core and some, such as “yesterday”, modify groups of words rather than a single word. The best solution, given their relative freedom of placing and the fact that sorting out where best to put them is more a meaning than a syntactic issue, would be to remove them and work out where they belong once the main verb and arguments have been identified.

Most of the above issues have to be left to an analysis of meaning to sort out the correct parse. There is no clear division between syntax and semantics. However there is another issue that has been highlighted to do with grammar and punctuation. How tolerant of errors should the system be? We have shown that errors in the operators do not cause problems for the parser, and errors in the placing of adverbs are relatively easy to deal with, but errors in the main constituents are not handled. For example, the phrase “the main people you need to ask will not be in the labs so early unless that have got work to hand in” occurs in one of the texts. The current algorithm will not handle these kinds of mistakes. Should the system be able to handle these kinds of mistakes, or should students be encouraged to improve their writing skills?

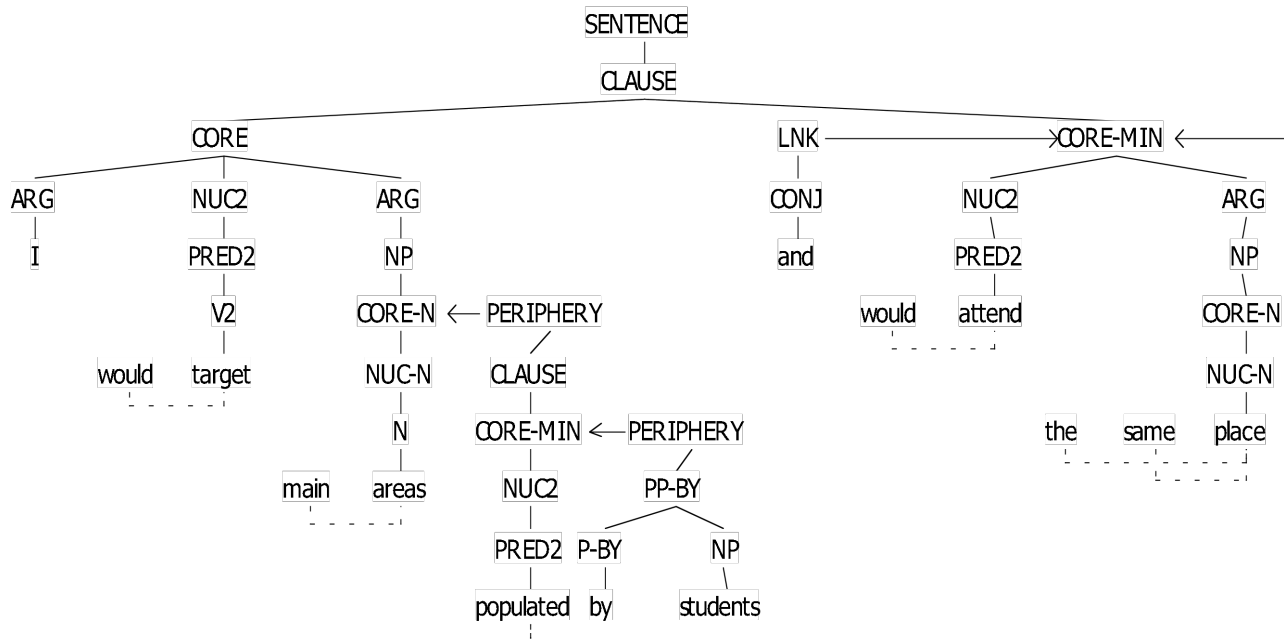


Figure 6: An example of a correctly parsed sentence

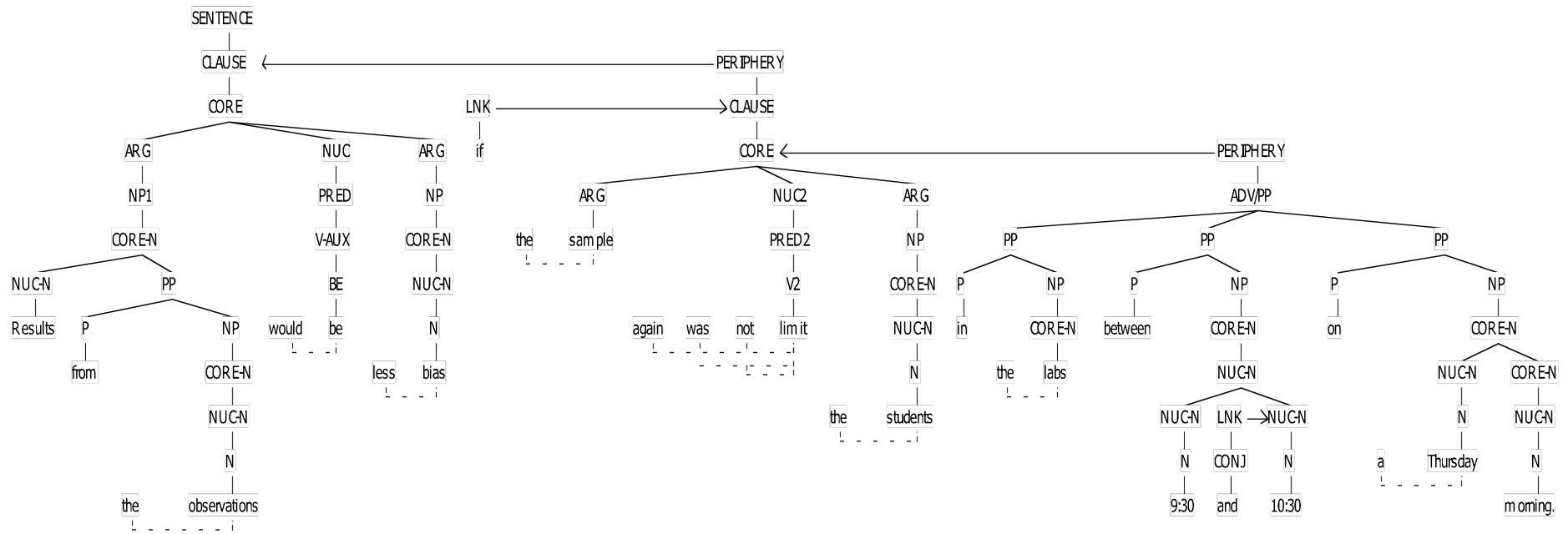


Figure 7: An example of a correctly parsed ungrammatical sentence

4 Conclusions and future work

Although, the initial results are promising, a lot more work needs to be done before this work can be included in an automatic marking system. Determining a good set of templates according to RRG theory is hard, and it is extremely difficult to keep the number of parses down to a small number. The work has now got to a point where experts on RRG need to look at the sentences and the templates to correct them and to see if any modifications need to be made to the theory. At present, there is no way of finding out which templates were applied in order to construct a parse, but this addition will be essential in order to do more analysis about the templates used.

We also need to look at the tagging process, and again professional linguists need to be involved with this. Compared to tagged corpuses, few tags have been used in this work, and it may be that it will be easier to achieve better results via the use of more verb and/or noun classes. At present, the operators that get attached to head words are not included in parsing, but again results may be improved by changing this. One way of changing this is to change the tags in the grammar file as currently happens to identify passive mood. For example, we could have a look to see if "CONT" is included and then change the tag, as this plays a part in many constructions. However, we have to be careful about how much we do this because the more we tie down the verb forms, the fewer ungrammaticalities are likely to be handled.

One particularly difficult aspect of parsing has been the noun phrases and the forms of arguments of verbs. When an argument is a straightforward noun phrase, parsing is straightforward, but we have numerous examples where the arguments are very complex noun phrases or another complex structure. An example of a complex noun phrase is: "possible sources of bias introduced into finding out students clubbing habits by asking those in the labs between 9:30 and 10:30 on a Thursday morning". This noun phrase is parsed correctly, but so far other kinds of complex arguments are not, such as "by randomly selecting an equal amount of students out of each year on varying courses by using their student ID email and sending them a questionnaire or just asking at popular locations within the campus, keeping a tally in order to meet the specification of the sample". All the latter is a single argument to "This could be done". Clearly the sentence would be easier both to parse and for other people to understand if it was broken up into several sentences. If the computer is to be able to mark student text automatically, however, this level of complexity really should be handled.

One line of attack to handle complexity would be to compare these examples with texts where (unlike for these examples) students are given marks for presentation and grammar. It may be that these texts are much easier to parse, and if this is the case, then we may be able to put some constraints on the complexity of sentences that students are allowed to input to an automatic marking system. One constraint that it would be useful to add would be to constrain the formatting that students are allowed to use. Formatting in the form of bulleted lists, especially of paragraphs, is a topic we have not even considered. It should be possible to handle other types of formatting such as simple lists or the use of colons, which are not handled at present, but even this adds a significant level of complexity to the parsing problem.

At present, verbs are simply categorised depending on how many arguments they take. Parsing may be easier if we could take into account the kind of arguments they take. For example, the verb "eat" is generally found with relatively simple arguments, whereas "suggest" or "recommend" generally take much more complex arguments. In the future, we plan to use this

work as the first stage in a system that uses a new semantic framework, ULM (Universal Lexical Metalanguage) [17], to compare the meaning of student texts with a (single) model answer. A core part of ULM is to link syntactic structure and semantics together, and part of this is to specify the types of arguments in more detail along with the syntactic structure that goes with them. For example, the verb “see” takes simple arguments when used with standard simple sentence structure (such as “I see lots of trees”), but more complex arguments when used in a more figurative sense, as in “I see that Jim has gone home”.

The ultimate aim of using ULM would be to enable us to convert text to a meaning representation. The aim is to build up a meaning representation from several sentences and then compare the meaning of the student text with the model answer – even when the words used are not the same. Clearly much work needs to be done before this aim is achieved.

References:

- Allen, J. (1995) *Natural Language Understanding*. Redwood City, CA: The Benjamin/Cummings Publishing Company Inc.
- Anderson, J. A. (2003) The Challenges of Collaborative Knowledge. In: C. M. Glynn and S. R. Acker, *Learning Objects: Contexts and Connections*. Ohio, USA: The Ohio State University.
- Beevers, C. E., Foster, M. G. *et al.* (1989) Integrating Formative Evaluation into a Learner Centred Revision Course. *British Journal of Educational Technology*. 20 (2), 115-19.
- Boud, D. (1995) Assessment and Learning: Contradictory or Complementary? In: P. Knight, *Assessment of Learning in Higher Education*. London: Kogan Page, 35-48.
- Brown, G., Bull, J. *et al.* (1997) *Assessing Student Learning in Higher Education*. London: Routledge.
- Brown, S. and Knight, P. (1994) *Assessing Learners in Higher Education*. London: Kogan Page.
- Brown, S., Race, P. *et al.* (1999) *Computer Assisted Learning in Higher Education*. London: Kogan Page.
- Chung, H. and Rim, H.-C. (2003) A New Probabilistic Dependency Parsing Model for Head-Final, Free Word Order Languages. *IEICE TRANS. INF. & SYST.* E86-D (11), 2490-3.
- Chung, H. and Rim, H.-C. (2004) Unlexicalized Dependency Parser for Variable Word Order Languages based on Local Contextual Pattern. In: *Lecture Notes in Computer Science: Computational Linguistics and Intelligent Text Processing (5th International Conference CICLING): Seoul 15-21 Feb. 2945*, 112-23.
- Covington, M. A. (2003) *A Free Word Order Dependency Parser in Prolog*. Available from: <http://www.ai.uga.edu/mc/dparser/dparser.pdf> [last accessed Sep 2008]
- Galicia-Haro, S. N., Gelbukh, A. *et al.* (2001) Combining Dependency and Constituent-Based Resources for Structure Disambiguation. In: *IEEE International Conference on Systems, Man, and Cybernetics: Tucson, AZ Oct 7-11. 5*, 2873-9.
- Grootjen, F. (2001) Relational Indexing Using a Grammarless Parser. In: *IEEE International Conference on Systems, Man, and Cybernetics: Tucson, AZ Oct 7-11. 5*, 2886-9.
- Holan, T. (2002) Dependency Analyser Configurable by Measures. In: *Text, Speech and Dialogue 5th International Conference TSD: Brno, Czech Republic, Sep 9-12.*, 81-8.
- Hou, L. and Cercone, N. (2001) Extracting Meaningful Semantic Information with EMATISE: an HPSG-Based Internet Search Engine Parser. In: *IEEE International Conference on Systems, Man, and Cybernetics: Tucson, AZ Oct 7-11. 5*, 2858-66.
- Ito, A., Hori, C. *et al.* (2001) Language Modeling by Stochastic Dependency Grammar for Japanese Speech Recognition. *Systems and Computers in Japan*. 32 (12), 10-15.
- Jurafsky, D. and Martin, J. H. (2000) *Speech and Language Processing*. Upper Saddle River, New Jersey: Prentice Hall.
- Kanayama, H., Torisawa, K. *et al.* (2002) A Hybrid Japanese Parser with Hand-Crafted Grammar and Statistics. In: *Proceedings of 18th COLING: Taipei, Taiwan 24 Aug - 1 Sep.* 411-17.
- Kešelj, V. (2001) Modular HPSG. In: *IEEE International Conference on Systems, Man, and Cybernetics: Tucson, AZ Oct 7-11. 5*, 2867-72.
- Landauer, T. K., Laham, D. *et al.* (1997) How well can Passage Meaning be Derived without using Word Order? A Comparison of Latent Semantic Analysis and Humans. In: *Proceedings of 19th Annual Conference of the Cognitive Science Society: Stanford University 7-10 Aug.* 412-17.
- Pérez, D. and Alfonsa, E. (2005) Adapting the Automatic Assessment of Free-Text Answers to the Students. *Paper delivered at 9th Computer Assisted Assessment Conference, Loughborough, UK: 5-6 July*

- Rust, C. (2002) The Impact of Assessment on Student Learning. *Active Learning in Higher Education*. 3 (2), 145-58.
- Sadler, D. R. (1989) Formative Assessment and the Design of Instructional Systems. *Instructional Science*. 18 (2), 119-44.
- Sadler, D. R. (1998) Formative Assessment: Revisiting the Territory. *Assessment in Education: Principles, Policy and Practice*. 5 (1), 77-84.
- Sambell, K. and Hubbard, A. (2004) The Role of Formative 'Low Stakes' Assessment in Supporting Non-Traditional Students' Retention and Progression in Higher Education: Student Perspectives. *Widening Participation and Lifelong Learning*. 6 (2), 25-36.
- Sukkarieh, J. Z., Pulman, S. G. *et al.* (2003) Auto-marking: using computational linguistics to score short, free text responses. *Paper presented at International Association of Educational Assessment, Manchester, UK: 5-10 Oct.*
- Sukkarieh, J. Z., Pulman, S. G. *et al.* (2004) Auto-Marking 2: An Update on the UCLES-Oxford University research into using Computational Linguistics to Score Short, Free Text Responses. *Paper presented at International Association of Educational Assessment, Philadelphia: 13-18 June.*
- Van Valin, R. D. J. (2005) *Exploring the Syntax-Semantics Interface*. Cambridge: Cambridge University Press.
- Van Valin, R. D. J. and LaPolla, R. (1997) *Syntax: Structure, Meaning and Function*. Cambridge, Cambridge University Press.
- Wahlster, W. (2000) *Verbmobil: Foundations of Speech-to-Speech Translation*. Berlin: Springer.
- Wiemer-Hastings, P. (2001) Rules for Syntax, Vectors for Semantics. *In: Proceedings of 22nd Annual Conference of the Cognitive Science Society: Edinburgh 1-4 Aug.*
- Yorke, M. (2001) Formative Assessment and its Relevance to Retention. *Higher Education Research and Development*. 20 (2), 115-26.

Appendix A: Results for parsing student texts

In the following tables the first column is the sentence, the middle column gives the number of paragraphs contains comments on what is going on.

Text 1

Possible sources of bias introduced into finding out students clubbing habits by asking those in the labs between 9:30 and 10:30 on a Thursday morning include the possible absence in the labs of anyone who has been out clubbing and had a late night the night before as they may be too tired or hung-over to attend university so early the next day.	8	
A better way of obtaining this information would be to ask students over a variety of times on every day of the week so students who have been out the night before clubbing would not be excluded from answering by the sampling technique employed being unsuitable.	12	Periphe
Alternatively you could track down or email every tenth (or different depending on the sample size wanted) person on a list of student ID numbers obtained from the university.	2	Everyth Multiple periphe

Text 2

There is a lack of a good sampling frame because most students who go out clubbing will tend to choose nights where they don't have lectures or tutorials in the morning.	3	Peripheries and difference in placement of "lectures or tutorials".
Therefore, asking only the students present on a Thursday morning will exclude all the students that either have no lessons or are not present.	1	
Also, not all students use the labs on a Thursday and some don't use them at all.	1	
Therefore this isn't a representative sample, it is biased towards students who have lessons in the lab on Thursday morning and towards those students who don't tend to go out during the mid week.	44	Periphery is something of a mess.
To obtain a representative sample the survey should be carried out at regular intervals during the day and at different locations.	1	
I would recommend stratified sampling in clusters.	1	
I would target main areas populated by students and would attend the same place at different times during the day.	2	
I would also target tutorial groups of different faculties to see if there is a significant difference in the clubbing habits of students in different schools.	2	
I believe this would provide a more representative sample.	1	

Text 3

Here the population being observed are students and their clubbing habits.	2	The "population being observed" is either NP1->CORE-PASS or a CORE-N with a periphery, which is a CORE-MIN with a VP2.
This type of sample described here could be termed a convenience sample "a sample that consists of population items that are easily accessed".	1	
A lab full of students is not an accurate sample of the population of all students that go clubbing.	1	
In order to accurately choose a sample for collecting data, it would be more prudent to not limit the sample group to that of the students in the labs.	1	
Not all students use the labs.	1	
Results from the observations would be less bias if the sample again was not limit the students in the labs between 9:30 and 10:30 on a Thursday morning.	1	UNGRAMMATICAL
The sample should try to get feedback from students that would be in University at a different times in a day (students have lectures and classes at different times in the day) and on a number of days in the week.	7	Peripheries
The time at which the sample is being employed "between 9:30 and 10:30" may be considered early for those students out clubbing the night before.	5	
This would lead to feedback from a sample which is not an accurate reflection of the population being observed.	6	
The type of sample sought here needs to be more of a subset (a smaller group), of the population being observed.	1	
A more random sample of a population could be used to achieve the goal of finding a representation of the entire population without any preference.	2	
In order to get more accurate results, a stratified random sample should be employed here.	1	
This is where the population is divided into similar, non-overlapping groups.	1	
Then a simple sample is chosen from each group.	1	
This would be more of a prudent approach as students are in University in most cases not everyday, students go to University at different times of the day and they are on different types of courses.	1	
The fact that students go to different University should also be an included factor.	1	UNGRAMMATICAL

Text 4

The scenario about the clubbing habits of students is a very bias way of collecting information.	1	
There are many better ways of obtaining this information and making it more accurate and that will make it more consistent.	5	
I would start off by changing the type of people which are being asked this is because you are only targeting the students which are in are using the labs.	5	
This means you are only asking people who are using computers and not asking other types of students such as art students, PE students etc.	4	
So this means that the results will not be accurate because students from all areas have not been covered.	2	
The second problem is that the time which the information is collected is not suitable firstly that people that do go out which are the main people you need to ask will not be in the labs so early unless that have got work to hand in.	6	
The best solution would be to change the time and ask people through out the whole day starting at 9:30 and finishing at 5:30.	8	

Text 5

This is not handled because of the format of its sentences: headings followed by ‘:’.

Text 6

Taking results on a Thursday morning between 9:30 and 10:30 in the morning is bias as students who go out clubbing are not likely to be in the labs.	2	
A better idea would be to take the results at a later time or even do the survey via e-mail where students can submit an answer at any time.	4	
This means that no students will be left out because they could not make it to the early lab session due to clubbing.	1	

Text 7

The first part of this scenario that may cause area of bias is that the time of the questioning is going on at a very early time, considering the fact that they are asking the question of clubbing habits.	1	
It would be a much better idea to start the survey a bit later so to make sure you have not missed any of the clubbers of the previous night.	1	
Another bad point of the survey is that it only goes on for one hour; this is just not enough time to get enough questions asked to the students.	2	Location of the “the students”.
A much better time scale would be to conduct the survey between maybe 10:30 and 16:30, so you would get the students at all peak time.	1	
The survey is also only being conducted on one day, which is Thursday.	2	
This would also show bias as some days may have different kinds of nights and different kind of offers which would in turn creates a different kind of “Clubber”.	6	
Therefore it would be necessary to conduct the survey on different days, such as Monday too Friday, so you don’t miss vital information.	0	UNGRAMMATICAL: “too” instead of “to”.

A set of templates was made based on the above texts. Out of 45 sentences, only one failed to parse.

The same set of templates was applied to the following texts. Of these 47 sentences, 16 parsed, giving a success rate of 34%. This rises to 38% if you discount the ungrammatical sentences that failed. However, as we will see below, some of this is because some structures have not been included in the templates. When these are also incorporated, the success rate should increase.

Text 8

This is not handled because of bulleted lists.

Text 9

This is not handled because of formatted lists.

Text 10

Students who are actively taking part in clubbing entertainment are less likely to be present at the 9:30 – 10:30 labs at university, and even if they were they may be less likely to share their beliefs as accurately as the other students not participating in clubbing due to their current state of health!	0	The problem here is “9:30 – 10:30”: we ignore punctuation, and so this does not make sense.
This means that the sample would not be representative of the student population.	1	
It should also be taken into consideration that the typical sample size of a morning lab would be too small to represent the whole student body.	2	
An example of serious bias could be if it was a lab of final year students, this samples work ethics on average would be expected to be far more disciplined than a student earlier on in the course with more available time to participate in such activities.	0	UNGRAMMATICAL and we do not yet deal with comparative constructions.
In order to obtain a more accurate reading of students clubbing habits it would be better to observe students while clubbing, maybe contact local clubs and ask which nights attract the most and least students.	0	
This would have an accurate backbone as student nights often require ID upon entry.	2	
If a sample was to be targeted at the university a broader sample should be chosen throughout all courses and course years.	0	The problem here is that construction like “was to be” are not handled.
This could be done by randomly selecting an equal amount of students out of each year on varying courses by using their student ID email and sending them a questionnaire or just asking at popular locations within the campus keeping a tally in order to meet the specification of the sample.		We do not handle prepositions next to each other, such as “out of”, although we could amalgamate these into a single preposition.

Text 11

The best way to collect the information about students clubbing habits is the questionnaire, in which you could ask variety of questions such as logical, open and close questions depending on the depth of the research.	0	We cannot yet handle “the best way to”, although we could turn it into an expression like “in order to”.
However the information can also be collected by interviewing individuals about their clubbing habits face to face, which is more time consuming than the questionnaire but more efficient method.	0	Comparative construction not yet handled.

Text 12

Students who had been out clubbing the night before would not be in the labs between 9:30 and 10:30!	1	
Initially I would suggest going and interviewing people in club queues at night, but then you would only get students who go out clubbing.	2	The construction "I would suggest ..." is not handled properly.
So I would suggest asking people within the university campus between the hours of 2 and 4 pm.	0	"I would suggest ..." is not yet handled.
This would give an unbiased collection of data.	2	

Text 13

The sample chosen will produce inaccurate results.	1	
For one, the time chosen to conduct the survey is limiting the sample to those students who attend in the morning.	0	"For one" is not handled.
Generally students who go out clubbing may not get up in time for 9:30am!	1	
Also the sample may not be representative of the student population.	1	
I would recommend using stratified sampling, breaking up the students into age groups 18 – 20, 21 -23, 23+ etc. and then using random sampling techniques to select an equal number of students from each age group using their student id.	0	"I would recommend ..." is not handled.

Text 14

In the survey I found that the there were a few bias areas.	0	UNGRAMMATICAL: “that the there were”.
These are that it has poor sampling because it is only a small sample due to it being in the labs in short time period.	6	
Also due to the fact that it was between 9.30 and 10.30 in the morning students that had been out the night before may not have attended the early morning lessons.	0	“Due to the fact that ...” is not handled.
With it only being in the labs you will only target one class, this would give inaccurate data the students in the group that go out a lot or never go out at all this would not represent all students.	22	
My suggestion to retrieve better data would be to use random sampling, i.e stand out side the James Graham building between the times of 1 and 2 this way the students who have been out the night before will be more likely to be in.	0	The structure of the sentence is not handled.
Depending on the time limit you have to retrieve this information you could take samples from different days and different areas i.e outside the bar.	0	The structures “depending on ...” and “i.e.” are not handled.

Text 15

The scenario demonstrates the use of a poor sampling frame.	2	
The population in this case would be simply all students (it does not specify whether they attend LMU or even if they go to university!).	2	
So from this we can say that the sampling frame (the group of people that we will consider sampling) is all students that attend LMU Beckett Park Campus (being that we are in Leeds I assume the labs that the scenario refers to are in Leeds Metropolitan University's Beckett Park campus!).	0	"From this, we can say that ..." is not handled.
This is a poor sampling frame for this investigation because the night life in Leeds might be better/worse than the night life in York, Nottingham, Birmingham, Newcastle etc.	0	Comparative and a list.
This would mean that students go out more/less in Leeds than the other cities and because of this it would not give us a representative sample.	0	Comparative and "more/less".
Another reason that the results of this investigation will not be representative is because it also demonstrates the use of a poor sampling unit (the people that are going to get sampled).	4	
The sampling unit in this case is all students that are in the labs between the time 9.30 and 10.30.	0	Doesn't like "the sampling unit in this case".
This is a poor choice of sampling unit because the students that do go out to clubs regularly would probably go to the clubs the night before, making them liable to not be present when the sample is being taken the following morning.	0	The use of "when" is causing problems.
On the same kind of lines, Wednesday night might be the best night to go clubbing so all the students that do go clubbing will go on a Wednesday night also making them liable to miss the sampling leaving us with a set of students that don't particularly like clubbing.	0	
Along side the previous two sources of bias, is the fact that the sample is far too small, for one were only in one city out of all students and further still were only at one campus, and further more, were narrowing the investigation down to students studying within a set of labs at Beckett Park campus in the space of 60 minutes.	0	Problems with "along side".
This is quite obviously too small for the investigation that we wish to carry out in the given scenario.	0	Two adverbs next to each other.
A bias could also be set on the scenario by the questions that we decide to ask the students about clubbing habits.	0	Fails to pick up the passive construction because "set" is marked with past or pass.
The person carrying out the investigation might favour one set of clubs over	0	The structure of the sentence is not

another and therefore making them liable to ask question directed at those set of clubs as opposed to the clubbing scene as a whole.		handled.
Another case of putting a bias on the scenario is that of actually going to labs.	0	We cannot yet handle "is that".
Students will (should) be in the labs to do work at that time and if were just going to go and ask them questions on clubbing, they might not be interested in clubbing and want to carry on with their work instead.	0	UNGRAMMATICAL: "if were" (no subject).
This would leave us with the group of students that do like clubbing.	0	
The problem with this is that the sampling unit has become even smaller and that the people that were left with all (in general) like clubbing, which would swing the results in favour of clubbing.	0	
A better way of obtaining this information would be to randomly select some towns and cities in England and then randomly select universities within these.	2	
Then I would wait outside of the lecture theatre and survey students via questionnaire that way between the times 0900 and 1700.	0	UNGRAMMATICAL: "outside of" and "via questionnaire that way".
This might have a problem of students having additional timetable classes following the lecture meaning that they would not be willing to participate in such a survey but a lot of lectures finish before the scheduled hour so it wouldn't be an as bigger problem as we would have first thought.	0	UNGRAMMATICAL: "an as bigger problem" and construction is too complex.
This still might have the problem of eliminating the students that go out all the time and are at university just for the night life instead of the degree but this probably isn't an as bigger problem either because you can't really class them as students if their not attending classes.	0	Too complex.
So the survey would be changed to finding out the clubbing habits of students that attend classes.	2	

Appendix B: RRG templates for parsing student texts

SENTENCE

CLAUSE

CORE

<p>Core1</p> <pre> graph TD CORE --> ARG CORE --> NUC1 ARG --> NP1 </pre>	<p>Core1periph</p> <pre> graph TD CORE --> ARG CORE --> NUC1 PERIPHERY --> CORE PERIPHERY --> ADV_PP[ADV/PP] ARG --> NP1 </pre>
<p>Core2</p> <pre> graph TD CORE --> ARG CORE --> NUC2 CORE --> ARG ARG --> NP1 ARG --> NP </pre>	<p>Core2periph</p> <pre> graph TD CORE --> ARG CORE --> NUC2 CORE --> ARG PERIPHERY --> CORE PERIPHERY --> ADV_PP[ADV/PP] ARG --> NP1 ARG --> NP </pre>
<p>Core2-2</p> <pre> graph TD CORE --> ARG CORE --> NUC2 CORE --> ARG ARG --> NP1 ARG --> PP </pre>	<p>Core2-2periph</p> <pre> graph TD CORE --> ARG CORE --> NUC2 CORE --> ARG PERIPHERY --> CORE PERIPHERY --> ADV_PP[ADV/PP] ARG --> NP1 ARG --> PP </pre>
<p>Core2-complex</p> <pre> graph TD CORE --> ARG CORE --> NUC2 CORE --> ARG ARG --> NP1 NUC2 --> NUC2 NUC2 --> LNK LNK --> NUC2 NUC2 --> NP LNK --> TO </pre>	<p>Core3</p> <pre> graph TD CORE --> ARG CORE --> NUC3 CORE --> ARG CORE --> ARG ARG --> NP1 ARG --> NP ARG --> PP </pre>
<p>Coreaux</p> <pre> graph TD CORE --> ARG CORE --> NUC CORE --> ARG ARG --> NP1 NUC --> PRED PRED --> V_AUX[V-AUX] ARG --> NP </pre>	<p>Coreaux-periph</p> <pre> graph TD CORE --> ARG CORE --> NUC CORE --> ARG PERIPHERY --> CORE PERIPHERY --> ADV_PP[ADV/PP] ARG --> NP1 NUC --> PRED PRED --> V_AUX[V-AUX] ARG --> NP </pre>
<p>Coreaux2</p> <pre> graph TD CORE --> ARG CORE --> NUC ARG --> NP1 </pre>	<p>Coreaux2-periph</p> <pre> graph TD CORE --> ARG CORE --> NUC PERIPHERY --> CORE PERIPHERY --> ADV_PP[ADV/PP] ARG --> NP1 </pre>

CORE-MIN

A CORE-MIN is a core where the first argument is missing. It is not included in official RRG theory, but the notion is useful to ensure that there really is an argument missing. In English, this is the second argument. This definition of CORE-MIN does not include the case where the first argument is missing.

<p>Core1-min CORE-MIN</p> <p>NUC1</p>	<p>Core1-min-periph CORE-MIN ← PERIPHERY</p> <p>NUC1 ADV/PP</p>
<p>Core2-min CORE-MIN</p> <p>NUC2 ARG</p> <p> </p> <p> NP</p>	<p>Core2-min-periph CORE-MIN ← PERIPHERY</p> <p>NUC2 ARG ADV/PP</p> <p> </p> <p> NP</p>
<p>Core2-2-min CORE-MIN</p> <p>NUC2 AAJ</p> <p> </p> <p> PP</p>	<p>Core2-3-min CORE-MIN</p> <p> </p> <p> NUC2ARG</p> <p> / \</p> <p> PRED2 NP ADJ</p>
<p>Core-min CORE-MIN</p> <p>NUC</p>	<p>Core-min-periph CORE-MIN ← PERIPHERY</p> <p>NUC ADV/PP</p>
<p>Core-min2 CORE-MIN ← PERIPHERY</p> <p> </p> <p> NUC ADV/PP</p> <p> / \</p> <p> V-AUX P-V ADJ</p>	<p>Core-min-conj CORE-MIN</p> <p> </p> <p> CORE-MIN LNK → CORE-MIN</p> <p> </p> <p> CONJ</p>
<p>Core-min3 CORE-MIN</p> <p>NUC NP</p> <p> </p> <p> V-AUX</p>	<p>Core-min3-periph CORE-MIN ← PERIPHERY</p> <p>NUC NP ADV/PP</p> <p> </p> <p> V-AUX</p>
<p>Core-min4 CORE-MIN</p> <p>NUC</p> <p> </p> <p> V-AUX AAJ</p> <p> </p> <p> PP</p>	<p>Core-min4-periph CORE-MIN ← PERIPHERY</p> <p>NUC ADV/PP</p> <p> </p> <p> V-AUX AAJ</p> <p> </p> <p> PP</p>

CORE-PASS

A CORE-PASS is included in order to model passive mood. Again, this is not part of official RRG theory, but it is useful for ensuring that the second argument (the object) has been promoted to the first argument (subject position). It is also useful for ensuring that the verb has the correct form.

<p>Core2-pass CORE-PASS</p> <pre> graph TD CP[Core2-pass CORE-PASS] --- ARG CP --- NUC2PASS[NUC2-PASS] ARG --- NP1[NP1] </pre>	<p>Core2-pass-periph CORE-PASS ← PERIPHERY</p> <pre> graph TD CP[Core2-pass-periph CORE-PASS ← PERIPHERY] --- ARG CP --- NUC2PASS[NUC2-PASS] CP --- ADVPP[ADV/PP] ARG --- NP1[NP1] PERIPHERY --- ADVPP </pre>
<p>Core2-pass-min CORE-PASS-MIN</p> <pre> graph TD CP[Core2-pass-min CORE-PASS-MIN] --- NUC2PASS[NUC2-PASS] </pre>	
<p>Core3-pass CORE-PASS</p> <pre> graph TD CP[Core3-pass CORE-PASS] --- ARG CP --- NUC3PASS[NUC3-PASS] CP --- PP ARG --- NP1[NP1] </pre>	<p>Core3-pass-periph CORE-PASS ← PERIPHERY</p> <pre> graph TD CP[Core3-pass-periph CORE-PASS ← PERIPHERY] --- ARG CP --- NUC3PASS[NUC3-PASS] CP --- PP CP --- PPBY[PP-BY] ARG --- NP1[NP1] PERIPHERY --- PPBY </pre>

NUC

<p>Nuc-coord</p> <pre> NUC2 / \ NUC2 LNK → NUC2 CONJ </pre>	
<p>Nuc1</p> <pre> NUC1 PRED1 </pre>	<p>Nuc1-2</p> <pre> NUC1 / \ PRED1 P-V ADJ </pre>
<p>Nuc1-3</p> <pre> NUC1 / \ PRED1 ADJ </pre>	<p>Nuc1-3</p> <pre> NUC1 / \ PRED1 P-V </pre>
<p>Nuc2</p> <pre> NUC2 PRED2 </pre>	<p>Nuc2-2</p> <pre> NUC2 / \ PRED2 P-V </pre>
<p>Nuc2-3</p> <pre> NUC2 / \ PRED2 P-V ADJ </pre>	<p>Nuc2-4</p> <pre> NUC2 / \ V-AUX LNK → NUC2 TO </pre>
<p>Nuc3</p> <pre> NUC3 PRED3 </pre>	<p>Nuc4</p> <pre> NUC V-AUX </pre>
<p>Nuc</p> <pre> NUC / \ V-AUX PRED </pre>	<p>Nuc-2</p> <pre> NUC / \ V-AUX P-V </pre>
<p>Nuc-3</p> <pre> NUC / \ V-AUX P-V ADJ </pre>	<p>Nuc3-pass1</p> <pre> NUC3-PASS PPRED3 </pre>
<p>Nuc2-pass1</p> <pre> NUC2-PASS PPRED2 </pre>	<p>Nuc2-pass2</p> <pre> NUC2-PASS / \ PPRED2 P-V </pre>

PRED

<p>Pred1 PRED1 V1</p>	<p>Pred1-2 PRED1 V12</p>
<p>Pred2 PRED2 V2</p>	<p>Pred2-2 PRED2 V12</p>
<p>Pred2-3 PRED2 V23</p>	<p>Pred2-4 PRED2 V</p>
<p>Pred3 PRED3 V3</p>	
<p>Pred-1 PRED ADJ</p>	<p>Pred-2 PRED / \ PRED LNK → PRED CONJ</p>
<p>PPred2-1 PPRED2 VP2</p>	<p>PPred2-2 PPRED2 VP23</p>
<p>PPred3-1 PPRED3 VP3</p>	<p>PPred3-2 PPRED3 VP23</p>

V-AUX

<p>V-aux1 V-AUX DO</p>	<p>V-aux2 V-AUX BE</p>
<p>V-aux3 V-AUX HAVE</p>	

ADV/PP

<p>Adv/pp1 ADV/PP ADV</p>	<p>Adv/pp2 ADV/PP PP</p>
<p>Adv/pp3 ADV/PP / \ PP ADV</p>	<p>Adv/pp4 ADV/PP / \ PP PP</p>
<p>Adv/pp5 ADV/PP / \ PP PP PP</p>	<p>Adv/pp6 ADV/PP AP</p>
<p>Adv/pp7 ADV/PP / \ AP ADV</p>	<p>Adv/pp8 ADV/PP PP-B</p>
<p>Adv/pp9 ADV/PP / \ PP PP ADV</p>	<p>Adv/pp10 ADV/PP PP-BY</p>

PP

<p>PP</p> <pre> graph TD PP --> P PP --> NP </pre>	<p>PP2</p> <pre> graph TD PP --> P PP --> NP NP --> CORE-N CORE-N --> CORE-MIN </pre>
<p>PP3</p> <pre> graph TD PP --> PP1[PP] PP --> LNK LNK --> PP2[PP] LNK --> CONJ </pre>	<p>PP4</p> <pre> graph TD PP --> TO PP --> NP </pre>
<p>PP5</p> <pre> graph TD PP-B --> NP1[NP] PP-B --> P NP1 --> CORE-N CORE-N --> NUC-N NUC-N --> N </pre>	<p>PP6</p> <pre> graph TD PP --> TO PP --> ADJ </pre>
<p>PP-N</p> <pre> graph TD PP-N --> P-N PP-N --> NP </pre>	<p>PP-N2</p> <pre> graph TD PP-N --> P-N PP-N --> NP NP --> CORE-N CORE-N --> CORE-MIN </pre>
<p>AP</p> <pre> graph TD AP --> ADV AP --> NP </pre>	<p>PP-BY</p> <pre> graph TD PP-BY --> P-BY PP-BY --> NP </pre>
<p>PP-BY2</p> <pre> graph TD PP-BY --> P-BY PP-BY --> NP NP --> CORE-N CORE-N --> CORE-MIN </pre>	<p>PP-BY3</p> <pre> graph TD PP-BY --> P-BY PP-BY --> NP NP --> CORE-N CORE-N --> CORE </pre>

NP

<p>NP</p> <p>NP</p> <p> </p> <p>CORE-N</p>	<p>NP-CONJ</p> <p>NP</p> <p>├── NP</p> <p>└── LNK → NP</p> <p> </p> <p> CONJ</p>
<p>NP-NPIP</p> <p>NP</p> <p>├── NPIP</p> <p>└── CORE-N</p>	<p>NPIP</p> <p>NPIP</p> <p> </p> <p>POSSPRO</p>

NP1

<p>NP1</p> <p>NP1</p> <p> </p> <p>CORE-N</p>	<p>NP1-2</p> <p>NP1</p> <p> </p> <p>CORE-MIN</p>
<p>NP1-3</p> <p>NP1</p> <p> </p> <p>CORE-N</p> <p>├── NUC-N</p> <p>└── PP</p>	<p>NP1-4</p> <p>NP1</p> <p> </p> <p>CORE-PASS</p>

CORE-N


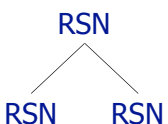
<p>Core-N CORE-N NUC-N</p>	<p>Core-N-PPN CORE-N / \ NUC-N PP-N</p>
<p>Core-N-nuc CORE-N / \ NUC-N CORE-N</p>	<p>Core-N2 CORE-N ← PERIPHERY \ NUC-N ADJ PP-N</p>
<p>Core-N-subconj CORE-N / \ \ NUC-N LNK → CLAUSE SUBCONJ CORE</p>	<p>Core-N-subconj2 CORE-N ← PERIPHERY NUC-N CLAUSE / \ PrCS CORE-MIN SUBCONJ</p>
<p>Core-N-subconj3 CORE-N ← PERIPHERY NUC-N CLAUSE / \ PrCS CORE-PASS-MIN SUBCONJ</p>	<p>Core-N-ETHR CORE-N / \ \ \ \ \ NUC-N LNK → CLAUSE SUBCONJ ETHR CORE-MIN LNK → CORE-MIN CONJ</p>
<p>Core-N-WH1 CORE-N ← PERIPHERY NUC-N CLAUSE / \ PrCS CORE-MIN WH1</p>	<p>Core-N-WH1-2 CORE-N ← PERIPHERY NUC-N CLAUSE / \ PrCS CORE-PASS-MIN WH1</p>
<p>Core-N-WH1-3 CORE-N ← PERIPHERY NUC-N CLAUSE / \ PrCS CORE-PASS WH1</p>	<p>Core-N-WH2 CORE-N ← PERIPHERY NUC-N CLAUSE / \ PrCS CORE WH2</p>

<p>Core-N-WH1-TO</p> <pre> CORE-N ← PERIPHERY NUC-N CLAUSE / \ PrCS CORE-MIN LNK → CORE-MIN WH1 TO </pre>	<p>Core-WH1-TO2</p> <pre> CORE-N ← PERIPHERY NUC-N CLAUSE / \ PrCS CORE-MIN LNK → CORE-MIN WH1 NUC2 TO </pre>
<p>Core-N-periph</p> <pre> CORE-N ← PERIPHERY NUC-N N CLAUSE CORE-MIN NUC2 </pre>	<p>Core-N6-periph-BY</p> <pre> CORE-N ← PERIPHERY NUC-N N CLAUSE CORE-MIN ← PERIPHERY NUC2 PP-BY </pre>
<p>Core-N-periph2</p> <pre> CORE-N ← PERIPHERY NUC-N N CLAUSE CORE-MIN PPRED2 </pre>	<p>Core-N-complex</p> <pre> CORE-N ← PERIPHERY NUC-N CLAUSE ← PERIPHERY / \ PrCS CORE-MIN CLAUSE SUBCONJ </pre>
<p>Core-N-periph3</p> <pre> CORE-N ← PERIPHERY NUC-N N CLAUSE CORE-MIN / \ NUC2 ARG PP </pre>	<p>Core-N-periph3-BY</p> <pre> CORE-N ← PERIPHERY NUC-N N CLAUSE CORE-MIN ← PERIPHERY / \ NUC2 ARG PP PP-BY </pre>
<p>Core-N-periph4</p> <pre> CORE-N ← PERIPHERY NUC-N N CLAUSE / \ CORE-MIN CORE-MIN NUC2 NUC PRED </pre>	

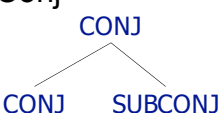
NUC-N

<p>Nuc-N NUC-N N</p>	<p>Nuc-N2 NUC-N PN</p>
<p>Nuc-N3 NUC-N PRO</p>	<p>Nuc-N4 NUC-N DEM</p>
<p>Nuc-N5 NUC-N QNT</p>	<p>Nuc-N6 NUC-N WH1</p>

RSN

<p>RSN  <pre> graph TD RSN --> TO RSN --> V12 RSN --> IF </pre> </p>	<p>RSN2 RSN IF</p>
<p>RSN3  <pre> graph TD RSN --> RSN1[RSN] RSN --> RSN2[RSN] </pre> </p>	

MISCELLANEOUS

<p>Conj  <pre> graph TD CONJ --> CONJ1[CONJ] CONJ --> SUBCONJ </pre> </p>	
---	--

Appendix C: Dictionary used for tagging

a	<i>DET</i>
a lot	<i>ADV</i>
about	<i>P</i>
absence	<i>N</i>
access	<i>V2</i>
account	<i>N</i>
accurate	<i>ADJ</i>
accurately	<i>ADV</i>
achieve	<i>V2</i>
actively	<i>ADV</i>
activity	<i>N</i>
actually	<i>ADV</i>
additional	<i>ADJ</i>
again	<i>ADV</i>
age	<i>N</i>
all	<i>QNT</i>
along	<i>P</i>
also	<i>DISC</i>
alternative	<i>ADJ</i>
alternatively	<i>DISC</i>
am	<i>ADV</i>
amount	<i>N</i>
and	<i>CONJ</i>
another	<i>ADJ</i>
answer	<i>V12</i>
answer	<i>N</i>
any	<i>QNT</i>
anyone	<i>PRO</i>
approach	<i>N</i>
area	<i>N</i>
art	<i>N</i>
as	<i>RSN</i>
as opposed	<i>RSN</i>
ask	<i>V23</i>
assume	<i>V2</i>
at	<i>P</i>
attend	<i>V2</i>
attract	<i>V2</i>
available	<i>ADJ</i>

average	<i>N</i>
backbone	<i>N</i>
Bad	<i>ADJ</i>
Bar	<i>N</i>
Be	<i>BE</i>
because	<i>RSN</i>
Beckett Park	<i>PN</i>
become	<i>V2</i>
before	<i>ADV</i>
Belief	<i>N</i>
believe	<i>V12</i>
Best	<i>ADJ</i>
better	<i>ADJ</i>
between	<i>P</i>
bias	<i>V1</i>
bias	<i>N</i>
biased	<i>ADJ</i>
big	<i>ADJ</i>
Birmingham	<i>PN</i>
bit	<i>N</i>
body	<i>N</i>
break	<i>V12</i>
broad	<i>ADJ</i>
building	<i>N</i>
but	<i>CONJ</i>
by	<i>P-BY</i>
cafeteria	<i>N</i>
campus	<i>N</i>
can	<i>AUX</i>
carry	<i>V2</i>
case	<i>N</i>
cause	<i>V2</i>
change	<i>V2</i>
choice	<i>N</i>
choose	<i>V2</i>
city	<i>N</i>
class	<i>N</i>
close	<i>V2</i>
club	<i>N</i>
clubber	<i>N</i>

clubbing	<i>ADJ</i>
clubbing	<i>V1</i>
cluster	<i>N</i>
collect	<i>V2</i>
collection	<i>N</i>
computer	<i>N</i>
conduct	<i>V2</i>
consider	<i>V2</i>
consideration	<i>N</i>
considering	<i>RSN</i>
consist	<i>V1</i>
consistent	<i>ADJ</i>
consume	<i>V2</i>
contact	<i>V2</i>
convenience	<i>ADJ</i>
could	<i>AUX</i>
course	<i>N</i>
cover	<i>V2</i>
create	<i>V2</i>
current	<i>ADJ</i>
data	<i>N</i>
day	<i>N</i>
decide	<i>V2</i>
degree	<i>N</i>
demonstrate	<i>V12</i>
depend	<i>V1</i>
depth	<i>N</i>
describe	<i>V2</i>
difference	<i>N</i>
different	<i>ADJ</i>
direct	<i>V23</i>
discipline	<i>V2</i>
divide	<i>V1</i>
do	<i>DO</i>
down	<i>P-V</i>
due to	<i>RSN</i>
during	<i>P</i>
each	<i>QNT</i>
early	<i>ADV</i>
early	<i>ADJ</i>
easily	<i>ADV</i>
-ed	<i>PAST</i>

efficient	<i>ADJ</i>
either	<i>ETHR</i>
eliminate	<i>V2</i>
email	<i>V2</i>
email	<i>N</i>
employ	<i>V2</i>
England	<i>PN</i>
enough	<i>QNT</i>
entertainment	<i>N</i>
entire	<i>ADJ</i>
entry	<i>N</i>
equal	<i>ADJ</i>
-er	<i>COMP</i>
etc	<i>DISC</i>
ethics	<i>N</i>
even	<i>ADV</i>
every	<i>QNT</i>
everyday	<i>ADV</i>
example	<i>N</i>
exclude	<i>V23</i>
expect	<i>V2</i>
face	<i>N</i>
fact	<i>N</i>
factor	<i>N</i>
faculty	<i>N</i>
far	<i>ADJ</i>
favour	<i>V2</i>
feedback	<i>N</i>
few	<i>QNT</i>
final	<i>ADJ</i>
find	<i>V2</i>
finish	<i>V12</i>
first	<i>QNT</i>
firstly	<i>CONJ</i>
following	<i>ADJ</i>
for	<i>P</i>
frame	<i>N</i>
friday	<i>N</i>
from	<i>P</i>
full	<i>ADJ</i>
further more	<i>ADV</i>
further still	<i>ADV</i>

gather	V2
general	ADJ
generally	ADV
get	V12
give	V23
go	V1
goal	N
good	ADJ
Graham	PN
group	N
habit	N
hand	V2
hangover	N
have	HAVE
health	N
here	ADV
hour	N
however	DISC
hung-over	ADJ
I	PRO
ID	N
idea	N
ie	EX
if	IF
in	P
in	P-V
in order to	TO
in turn	ADV
inaccurate	ADJ
include	V2
included	ADJ
individual	N
information	N
-ing	CONT
initially	ADV
instead	ADV
interested	ADJ
interval	N
interview	V2
into	P
introduce	V2
investigation	N

it	PRO
item	N
James	PN
just	ADJ
keep	V2
kind	N
lab	N
lack	N
late	ADJ
lead	V1
least	QNT
leave	V12
lecture	N
Leeds	PN
less	QNT
lesson	N
liable	V2
life	N
like	V2
likely	ADJ
limit	V2
line	N
list	N
LMU	N
local	ADJ
location	N
logical	ADJ
main	ADJ
make	V2
many	QNT
may	AUX
maybe	CONJ
mean	V1
meet	V2
method	N
metropolitan	ADJ
mid	ADJ
might	AUX
minute	N
miss	v2
monday	N
more	ADJ

morning	<i>N</i>
most	<i>QNT</i>
much	<i>QNT</i>
my	<i>POSSPRO</i>
-n	<i>PASS</i>
narrow	<i>V2</i>
necessary	<i>ADJ</i>
need	<i>V2</i>
never	<i>ADV</i>
Newcastle	<i>PN</i>
next	<i>ADJ</i>
night	<i>N</i>
no	<i>NEG</i>
non-overlapping	<i>ADJ</i>
not	<i>NEG</i>
Nottingham	<i>PN</i>
number	<i>N</i>
observation	<i>N</i>
observe	<i>V2</i>
obtain	<i>V2</i>
obviously	<i>ADV</i>
of	<i>P-N</i>
off	<i>P-V</i>
offer	<i>N</i>
often	<i>ADV</i>
on	<i>P</i>
on	<i>P-V</i>
one	<i>NUM</i>
only	<i>ADV</i>
open	<i>ADJ</i>
or	<i>CONJ</i>
order	<i>V</i>
other	<i>QNT</i>
out	<i>P-V</i>
outside	<i>P</i>
over	<i>P</i>
overlapping	<i>ADJ</i>
part	<i>N</i>
participate	<i>V2</i>
particularly	<i>ADV</i>
PE	<i>N</i>
peak	<i>N</i>

people	<i>N</i>
period	<i>N</i>
person	<i>N</i>
picture	<i>N</i>
place	<i>N</i>
pm	<i>ADV</i>
point	<i>N</i>
poor	<i>ADJ</i>
popular	<i>ADJ</i>
populate	<i>V2</i>
population	<i>N</i>
possible	<i>ADJ</i>
preference	<i>N</i>
present	<i>V1</i>
previous	<i>ADJ</i>
probably	<i>ADV</i>
problem	<i>N</i>
produce	<i>V2</i>
provide	<i>V2</i>
prudent	<i>ADJ</i>
put	<i>V23</i>
question	<i>N</i>
question	<i>V2</i>
questioning	<i>N</i>
questionnaire	<i>N</i>
queue	<i>N</i>
quite	<i>ADV</i>
-r	<i>COMP</i>
random	<i>ADJ</i>
randomly	<i>ADV</i>
read	<i>V12</i>
reading	<i>N</i>
really	<i>ADV</i>
reason	<i>N</i>
recommend	<i>V2</i>
refer	<i>V2</i>
reflection	<i>N</i>
regular	<i>ADJ</i>
regularly	<i>ADV</i>
represent	<i>V2</i>
representation	<i>N</i>
representative	<i>ADJ</i>

require	V2
research	N
response	N
result	N
retrieve	V2
-s	PL
-s	3S
same	ADJ
sample	N
sampling	N
say	V23
scale	N
scenario	N
scene	N
scheduled	ADJ
school	N
second	QNT
see	V12
seek	V12
select	V2
send	V2
serious	ADJ
session	N
set	V12
share	V2
short	ADJ
should	AUX
show	V2
side	N
significant	ADJ
similar	ADJ
simple	ADJ
simply	ADV
size	N
small	ADJ
so	QNT
so	RSN
so	DISC
solution	N
some	QNT
sought	
source	N

space	N
specification	N
specify	V2
stand	V12
start	V12
state	N
still	ADV
stratified	ADJ
stratify	V
student	N
study	V12
submit	V2
subset	N
such	SUBCONJ
such as	EX
suggest	V1
suggestion	N
suitable	ADJ
sure	ADV
survey	N
swing	V2
take	V23
tally	N
target	V2
technique	N
ten	NUM
tend	V2
term	V
-th	QNT
than	COMP
that	SUBCONJ
that	DEM
the	DET
theatre	N
their	POSSPRO
them	PRO
then	DISC
there	DEM
therefore	DISC
these	DEM
they	PRO
think	V12

this	DEM
those	DEM
through	P
through out	P
Thursday	N
time	N
timetable	N
tire	V
tired	ADJ
to	TO
to make sure	RSN
too	QNT
towards	P
town	N
track	V2
try	V2
turn	N
tutorial	N
two	NUM
type	N
typical	ADJ
unbiased	ADJ
unit	N
university	N
unless	RSN
unsuitable	ADJ
up	P-V
upon	P
us	PRO
use	V2
use	N
variety	N
varying	ADJ
very	ADJ
via	P
vital	ADJ
wait	V12
want	V2
way	N
we	PRO
wednesday	N
week	N

when	WH2
where	WH2
whether	CONJ
which	WH1
while	P
who	WH1
whole	ADJ
will	AUX
wish	V2
with	P
within	P
without	P
work	N
worse	ADJ
would	AUX
wrong	ADJ
year	N
York	PN
you	PRO