

Automatic Processing of Local Grammar Patterns

Oliver Mason

Department of English
The University of Birmingham
O.Mason@bham.ac.uk

Abstract

This paper describes the application of finite-state methods to the recognition of grammar patterns. A preliminary evaluation reports a high degree of accuracy which indicates that grammar patterns are a robust way of describing syntactic structures well suited to unrestricted text.

1 Introduction

The work reported in this paper is part of a (PhD-)project to extract linguistic information from text corpora. The aim of that project is to automate the process of analysis as far as possible; previous (manual) studies have been limited in scope and typically investigated only a few words at a time. With an automated analysis it will be possible to get a view of the ‘big picture’, and patterns of usage will emerge from this which can give us useful insights into the nature of linguistic behaviour. This is not possible with the current manual approach.

This paper starts with a brief outline of the rationale on which this work is based, describing some of the typical problems that a computational analysis of language has to face. Then two formalisms are described which deal with syntax in a lexicalised way, Local Grammar and Pattern Grammar, which provides a solution to many of the problems faced by a traditional analysis. These two approaches are then combined, and an evaluation of two case studies of the resulting ‘local grammar pattern’ analysis is presented.

1.1 Rationale

Recent studies in lexis (eg. Sinclair (1991), Stubbs (2001)) present an analysis of either individual words, or relatively small sets of words. These case studies are sufficient to demonstrate the validity and use of a corpus-based approach to language analysis, but are not scaleable—they require a lot of careful effort by a specialist. Therefore it is not feasible to undertake a large-scale analysis of a language manually, and the application of NLP methods to research problems in corpus linguistics is still rather limited. However, a lot of basic problems of large-scale text analysis are being worked on in other contexts, such as information retrieval and extraction, and text understanding. Modules developed for specific

tasks (eg. named entity recognition) can also be useful when investigating the structure of language, rather than the content of a newspaper article.

The current project intends to fathom out how far procedures in corpus linguistics can be automated. A number of procedures have been selected on the basis of an intuitive assessment of their respective feasibility; one of the studies is concerned with the analysis of the syntactic environment of a word. In the grammar pattern approach words are assigned typical patterns in which they are used within a text. Syntax is then not concerned with applying phrase structure rules, but rather with sophisticated template matching (see examples below). In the past this has been done manually by a team of grammarians (Sinclair, 2001), but for any serious application this has to be automated.

1.2 Language Model

The basic underlying model of language is that a language is a system of choices/options/elements which make only sense in comparison with each other, ie. when a selection is made part of its meaning is in all the other available options which have not been chosen. A single choice looked at in isolation (without the range of possible alternatives) does not make (much) sense. However, it is difficult to comprehensively study all options manually, as analysing language data is time-consuming and requires a high level of skill. It also introduces the possibility of a bias, as it sometimes relies on human judgment and interpretation, which makes it difficult to keep the analysis objective and replicable.

The computer on the other hand has no bias—all words are equal(ly meaningless). It is also be vastly faster, and takes the drudge out of the repetitiveness that goes with a full-scale investigation. The main question is then, if relevant tasks can adequately be performed by the computer.

1.3 Computational Analysis

Some tasks can only be done by computer, namely those involving complex statistical procedures. An obvious candidate for this is the study of collocations; others might involve the postprocessing of outcomes through cluster analysis or multivariate statistics. This does usually require the setting of several parameters, but once

this has been done the computer can proceed without any further human intervention. Here the computer makes possible an analysis which could not have been done before, so that it makes a qualitative difference rather than simply a quantitative.

Other tasks are more difficult, as language is inherently vague and (when forced into precision) ambiguous. Its analysis is also often based on ill-fitting and inappropriate categories, for example when it comes to word classes: these have originally been specified in antiquity and are based on ancient Greek and Latin (as are a lot of other grammatical categories). As a result, many words in English cannot be unambiguously assigned a single word class, even by human experts. Modern part-of-speech taggers, typically operating with statistical language models, achieve an accuracy rate of around 95+ percent; however, the calculation of this rate is sometimes doubtful, given the lack of a reliable 100% benchmark, and methods of evaluation differ between authors.

In a hierarchical model of processing (which is how most NLP systems operate) errors made at an early stage can propagate upwards, making it difficult to reach accurate final results. A lot of ambiguities cannot be decided without recourse to additional higher-level information and effectively a decision will have to be postponed until that information becomes available. This in turn means that often obscure interpretations are reached through rare but not impossible word class ambiguities. Ideally, probabilities would make processing easier, but in the absence of statistical information options are limited.

In general, tasks involving assumed grammatical categories are always problematic and any outcomes have to be carefully analysed to assert their validity; the problem here is that it cannot be easily determined whether it is the program that is at fault, or whether the grammatical categories are simply inadequate. One should be prepared to question any 'received wisdom', even if it has not been challenged for literally millennia.

Grammatical analysis is a hard enough task with invented sentences, but it becomes even more difficult when working with authentic data, which often does not adhere to the traditional rules of grammar. In addition, a lot of real language is 'messy', involves names of people or entities, dates, and a whole host of other phenomena which linguists do not normally pay any attention to. Grammars of, for example, dates might have been created for existing NLP systems, but are often not re-usable due to differences in tokenisation or word class labels.

In real life, language, however, is rarely ambiguous due to additional information supplied by common sense or shared contextual knowledge. It is only isolated sentences (which do not have access to such information) that often have hundreds of different readings, depending on the type of grammar used. But most of linguistics has so far concentrated on the analysis of isolated sentences. Phrase structure grammar only operates on

word classes, and arguably any system of word classes in general use is not sufficient to model the distributional behaviour of word forms. Studies in corpus linguistics have shown that even different inflected forms of a word often have complementary distributions. Sinclair (1991) demonstrates that *eye* and its plural form *eyes* are used in different contexts and cannot be interchanged, yet they both share the word class 'noun'.

As a consequence, more emphasis has to be placed on the behaviour of individual lexical items. While this allows a more fine-grained description, it also increases the number of elements by several orders of magnitude: instead of, say, 40 word classes we might be dealing with 40,000 different word forms. Automation is essential if a reasonably comprehensive analysis is to be attempted. This is one of the aims of this work.

2 Local Grammar Patterns

Most grammatical formalisms are based on either *constituency* or *dependency*, the two basic principles of structural description. An alternative approach is to view language not as a basically flexible system which is driven by constraints to avoid overgeneration, but instead as a set of prefabricated building blocks that can be combined to form a sentence. While there is undeniably a capacity for creativity in language, most language use is based on routine (Stubbs, 1993).

Starting the description of language from a lexical point of view goes together well with a pattern or template based approach. As stated above, word classes are not suited to capture the distributional regularities of a language, as they are far too coarse. In the Firthian tradition meaning and use are related, which means that a large part of the meaning of a word is in the context it is used in. This on the other hand requires words with a similar distribution to have similar meanings, so that a description of such regularities purely based on word classes becomes infeasible.

2.1 Local Grammar

Local Grammar (eg. Gross (1993)) is a way of describing the syntactic behaviour of groups of individual elements, which are related but whose similarities cannot easily be expressed using phrase structure rules. Recursive Transition Networks (RTN, see Winograd (1983)) are used as a formalism, with both word classes and sets of lexical items as possible labels of transitions. This allows the grammarian to be very specific if a certain construction requires a particular lexical choice.

The basic idea is that local grammars for recurring elements (eg. date expressions) are constructed, which can then be re-used in the description of larger linguistic constructions; basically a bottom-up approach towards a comprehensive description of a language.

Local grammars are also well-suited for (semi-)fixed phrases, where some limited variation in form is possible, but with minimal change in meaning. Gross (1993) gives

the examples (only 5 out of 11 are reproduced here):

1. *Bob lost his cool.*
2. *Bob lost his temper.*
3. *Bob lost his self-control.*
4. *Bob blew a fuse.*
5. *Bob blew his cool.*

All these sentences are (according to Gross) synonymous, and their shared features can easily be captured in a local grammar.

Grammar creation is facilitated by a graphical interactive development environment, INTEX (Silberztein, 1993).

2.2 Pattern Grammar

Pattern Grammar (eg. Hunston and Francis (2000)) abandons a rule-based hierarchical structure in favour of patterns or templates that describe typical environments a word occurs in. Patterns are derived from analysing corpus data, so they are based on actual usage. Like local grammar, patterns contain a mixture of actual lexical items (mostly prepositions) and abstract elements (such as noun group, *that*-clause, etc).

Unlike local grammar, pattern grammar attempts a broader description of syntactic behaviour, which is less lexicalised. Typically the only lexically restricted elements in a pattern will be the main word the pattern belongs to, and perhaps a preposition (which is more specific than the general class of all prepositions).

One of the important issues with pattern grammar is the correspondence between form and meaning: it can be seen that words which share aspects of their meaning also occur in similar sets of patterns. Typically a pattern will be used to realise several different meanings, and each of these meanings will be associated with a different set of related words which go together with this pattern. In Sinclair (1996) the example of **V n by n** is given, which has five meaning groups:

1. 'begin/end': verbs used are *answer, begin, start*, as in *He answered the question by denying that...*
2. 'grab': *catch, grab, hold*, as in *He grabbed Rivers by the shoulders and...*
3. 'call': *call, know*, as in *In three years I had never called him by name.*
4. 'raise/lower': *cut, devalue, lift, raise*, as in *The Irish government was forced to devalue its pound by 10 percent within...*
5. 'others': *replace, run, surround*, for example *I think you'd better run it by me again.*

Similar groups of meanings can be established for other patterns as well; for a comprehensive list see Sinclair (1996).

3 System Outline

The pattern recogniser consists of several components. A pre-processor, which would turn the input text into a list of sentences is not used at present, but will be required for the processing of free text input. Presently, the input data is assumed to be concordance lines, with each input record on a separate line. Sentence boundaries are not taken into account, as the aim is the recognition of a pattern and not yet the analysis of a complete sentence. This will be added at a later stage.

A simple tokeniser splits the input into tokens, separating punctuation marks (which are treated as separate elements). A stochastic parts-of-speech tagger (Qtag) is then used to assign ambiguity classes to the input tokens. An ambiguity class contains all possible word class labels which can be assigned to the input token, ordered by their respective likelihood. At present, all word classes are treated equally, as the basic approach is lexical, and would hopefully disambiguate any problem cases automatically. Leaving out potential word classes which are less likely might result in a loss in recall.

The final stage in processing is an RTN parser, which takes as input a finite-state network describing a set of patterns for a word. As additional resource it has available a set of 'constituent networks', RTNs which describe the individual elements which can be part of a pattern. These are eg. noun groups, verb groups, and other clause elements. The RTN parser ranks each path through the RTN according to length; typically a pattern can match in several ways. A noun phrase with a determiner would also match without the determiner, so the same constituent would match twice with different starting positions. Longest-matching is a very successful strategy in avoiding mistakes at this stage.

The networks describing the word patterns have been generated from a list of patterns from Sinclair (2001). This yields more than 19,600 patterns for about 4,800 verbs (patterns of nouns and other word classes have initially not been included). All patterns (see examples below) are stated as linear sequences of elements; these have been converted into simple finite-state automata, which were then merged and minimised to form a single FSA for each word. They have been designed to be compatible in format with the INTEX system (Silberztein, 1993), as it is envisaged that they might form part of a local grammar development project at a later stage. In any case, it is always advisable to follow an existing standard if possible. Some automatically generated automata have already been successfully imported into INTEX.

4 Evaluation

For evaluating the effectiveness of both the pattern approach and the pattern recogniser, a few sample words were chosen (randomly) for testing. In this section the results of the tests are described.

In order to measure the effectiveness of changes to the system automatic test have been set up. A number of

sample lines was annotated with the correct pattern that the system should identify, so that the performance could be evaluated fully automated. This is particularly important as the system is still under development; this way any changes can immediately be evaluated as to whether they actually increase the overall performance of the system.

4.1 *blend*

The first word chosen was *blend*, which has the following patterns: **V n with n**, **V with n**, **V n**, **V pl-n**, **pl-n V**, **V-ed**, **N of n**, and **V-ed**. From a selection of corpora (3 million words in size) 56 lines for *blend* and its inflected forms were retrieved and tested on the automaton generated from the pattern list. An analysis of the errors showed a possible variation in the patterns, the use of *into* instead of *with*, which was then added to the FSA. As the system is still under development, the RTNs used for recognising phrase components were manually adjusted to deal with cases where an element had not been recognised properly. This test thus represents 'ideal' conditions, where no errors from the syntactic analysis interfere with the pattern recognition. Such adjustments will continue throughout, as the RTNs are applied to more and more data. The amount of changes necessary will decrease as coverage of frequent constructions increases.

There are four possible outcomes of a pattern match:

1. The correct pattern is the highest ranking pattern returned by the recogniser
2. The correct pattern has been found, but is not at the highest rank
3. The correct pattern has not been identified at all
4. A pattern has been recognised, but there is actually no pattern in the data

The output of the pattern recogniser for *blend* showed that the correct pattern had been identified in 54 cases (always as the highest-ranking result), and in two lines a pattern had been recognised where the word had actually been used without a pattern. Restated in terms of *precision* and *recall* we have a recall of 100% and a precision of 96.4%, yielding an *F*-measure of 98.16 (van Rijsbergen, 1979).

The two lines where a pattern has been falsely recognised are:

1. ...an opera with some kind of model for dramatic themes in which were blended history - in the sense of a distant past that could be upheld as...
2. ...colors on the market are more lively and interesting to me than blended tones - which is one of the reasons why my palette is made of more...

In the first line, the pattern **V n** is identified; there is not sufficient context to correctly analyse the structure of this rather convoluted sentence even for a human analyst. In the second line the adjectival use is mis-interpreted and

the pattern **V pl-n** is proposed. Here we see a case where arguably our traditional system of grammatical description does not quite apply to English. Cases where past or continuous participles of verbs are used as noun modifiers are usually treated as adjectives, but one could imagine them as embedded non-finite clauses with a verb an object.

In order to keep matters simple, the adjective-noun interpretation seems generally to be preferable, in which case the recognised pattern would be wrong. This is a consequence of the lexical approach which gives a lexical match preference over part-of-speech match. A full parse of the sentence might resolve that problem, or alternatively the part-of-speech could be taken into account. However, the latter alternative might lead to errors in other cases and would have to be carefully evaluated.

In summary, this first attempt is not a realistic test for the overall performance, but rather shows the optimal result in an ideal setting. It does provide reassurance that the method itself is capable of operating with high levels of accuracy, but the true performance needs to be evaluated on different data.

4.2 *link*

As a second example, the word *link* was chosen at random. It has the patterns **N between/with/to n**, **V n to/with n**, **V-ed**, **V pl-n**, **V n prep/adv**, **V n**, **V P with n**, **pl-n V P**, and **be V-ed P to n**.

This time 116 lines were selected at random from the same 3 million word corpus selection as before, split proportionately across the various inflected forms. No further adjustments were made to the system; the result therefore is a more realistic reflection of its current capabilities. Of the 116 lines, 73 were correctly analysed (62.9%), 9 patterns were found but were not highest-ranking (7.7%), 6 patterns were found where no patterns had been used (5.1%), and 28 patterns were not correctly identified (24.1%). With recall at 100% (again, at least one pattern was recognised for each line) strict precision is at 62.9% (if only highest ranking matches are counted) ($F = 77.2$) and the less strict precision (where any correct identification is counted) is at 70.7% ($F = 82.8$).

Most of these errors were due to non-canonical patterns, especially passive clauses. Supplementing the automaton with the missing variants, **V to/with n** and **pl-n be V-ed** we significantly improve the precision, which is now at 80.1% (strict) and 91.3% (loose), with the respective *F* values of 88.9 and 95.4. Given that the transformed patterns can be added to the patterns of any word automatically once a general list of corresponding passive patterns has been drawn up for each active one, the result looks very encouraging. At present, non-canonical patterns are not listed in the dictionary and are therefore not used by default. A previous study (Mason and Hunston, to appear) reports that 6 out of 100 patterns for the lemma *decide* were in non-canonical form.

In most cases where the correct pattern was not the

highest ranking one, a word preceding the pattern has erroneously been incorporated due to the strategy of longest-matching. This happens easily with continuous forms, as in

1. *surprise, that a Labour government would not be averse to **cultivating** close links with an arch-conservative industrialist such as himself.*
2. *well be poached by European big business with many schools already **forming** links with companies such as the West German car makers, BMW, as they are finding*

In all the above cases there is a (local) structural ambiguity as exemplified by the following (correctly identified) example: ... *day that could lead the way to lifting the international boycott on **sporting links** with South Africa.* Looking at the wider context the ambiguity can usually be resolved. This, however, does not apply to the first case, where *averse* has the pattern **v-link ADJ to n**, where *cultivating close links* would nicely fill the **n** slot, but in the second line the clause would be missing a verb, and the noun phrase *many schools* could not easily be followed by another noun phrase *already forming links*. Here the verbal reading of *forming* would be necessary to fulfil grammatical constraints.

This example shows that often local context is not sufficient for a correct analysis, and pattern grammar has to be embedded in a more 'global' approach to structural description. This could simply mean a comprehensive analysis of all patterns occurring in a text, which should handle most of the existing ambiguities.

The remaining four cases in which the pattern was not identified at all are split evenly between parsing errors and missing patterns with two instances each. The parsing errors are due to the developing nature of the system and should not occur anymore once the system is more stable; the missing patterns will always remain a problem. However, once a systematic study of patterns is undertaken with the aid of the computer, the existing pattern lists can be extended to include any omissions.

5 Future Work

The work described in this paper can be further extended in a number of different directions. First, a complete description of all patterns in a sentence can be attempted. At present there are problems with selecting the correct automaton for the right word in an input sequence, which is why the analysis best works with a single pattern automaton only. This is the first restriction that needs to be remedied. Once this has been tackled it becomes possible to assess the overall coverage of grammar patterns: how dense are texts in terms of their usage of patterns? Or, in other words, how much of a text can we describe in terms of patterns? This should give a thorough overview which might at the same time show up patterns that have been left out of the initial description.

A further step is the automatic identification of patterns, rather than their recognition. Using machine learning techniques it ought to be possible to generalise from the syntactic context of a word and identify what patterns are used with it. That would be one way to overcome the problem of incomplete description, but Hunston and Francis (2000) give some examples where identifying the correct pattern is rather difficult, so it remains to be seen if this can be done automatically.

For a linguistic description it will be useful to investigate the typical instances of pattern elements. For example, in *blend n with n* the two **n** slots will be filled with words describing 'blendable' entities. The pattern *bite into n* could provide us with a list of items that are typically bitten into. There is quite a lot of scope for collecting sets of semantically related words which could be used for describing the meaning of words in terms of distributional similarity.

Finally, there are the basic steps of processing non-canonical forms of patterns, which requires dealing with unusual word order, passive transformations, and similar phenomena. This would increase the coverage substantially, and would only be necessary once for each pattern. In the list of verb patterns of 4,800 verbs mentioned above the number of unique patterns is about 1,100. Assuming the usual distribution we will only have to deal with a few hundred patterns to cover most actual occurrences.

6 Conclusion

Pattern grammar is a novel way of describing the syntactic structure of sentences. It does not assume a hierarchical structure, but instead is based on a sequential/linear model where patterns follow each other or even flow into each other (Hunston and Francis, 2000). Pattern grammar can be implemented using efficient finite-state technology, and can thus be integrated into the precise description of (lexicalised) local grammars, as described by Gross (1993) and Gross (1997).

The initial results of individual case studies are very encouraging, however, more automata need to be evaluated as the system develops to provide more accurate and robust measures of performance. It also needs to be investigated how overlapping patterns can contribute to the disambiguation of structural ambiguities: if one pattern is followed by a noun, the next pattern cannot interpret that noun as a verb; the question then is which pattern is right.

Another factor is the modular nature of the system and the influence of multiple factors on the overall result. The interaction between the parts-of-speech tagger, the constituent networks, and the pattern automata needs to be fine-tuned to work best; errors in any of these components can lead to problems with recognition. For testing purposes automated benchmarks have been set up to investigate the influence of adjustments to any component on the overall system performance.

In summary, local grammar patterns are a promis-

ing approach to the description of syntactic structures. They allow a description of language based on re-usable component networks and can be processed efficiently by computer.

References

- M. Gross. 1993. Local grammars and their representation by finite automata. In M. Hoey, editor, *Data, Description, Discourse*, pages 26–38. HarperCollins, London.
- M. Gross. 1997. The construction of local grammars. In E. Roche and Y. Schabes, editors, *Finite State Language Processing*, pages 329–354. Bradford, Cambridge, MA.
- S. Hunston and G. Francis. 2000. *Pattern Grammar: A corpus-driven approach to the lexical grammar of English*. Benjamins, Amsterdam.
- O. Mason and S. Hunston. to appear. The automatic recognition of verb patterns: a feasibility study. *International Journal of Corpus Linguistics*.
- M. Silberztein. 1993. *Dictionnaires électroniques et analyse automatique de textes: le système INTEX*. Masson, Paris.
- J. McH. Sinclair. 1991. *Corpus, Concordance, Collocation*. Oxford University Press, Oxford.
- J. McH. Sinclair, editor. 1996. *Collins COBUILD Grammar Patterns 1: Verbs*. HarperCollins, Glasgow.
- J. McH. Sinclair, editor. 2001. *Collins COBUILD English Dictionary for Advanced Learners*. HarperCollins, Glasgow.
- M. Stubbs. 1993. British traditions in text analysis—from firth to sinclair. In M. Baker, G. Francis, and E. Tognini-Bonelli, editors, *Text and Technology: In Honour of John Sinclair*, pages 1–33. Benjamins, Amsterdam.
- M. Stubbs. 2001. *Words and Phrases: corpus studies of lexical semantics*. Blackwell, Oxford.
- C. J. van Rijsbergen. 1979. *Information Retrieval*. Butterworths, London.
- T. Winograd. 1983. *Language as a cognitive process*. Addison-Wesley.