Can Punctuation Help Parsing?

Bernard E M Jones

Centre for Cognitive Science, University of Edinburgh,
2 Buccleuch Place, Edinburgh EH8 9LW, Scotland
Email: B.E.M.Jones@ed.ac.uk

July 1994

Acquilex-II Working Paper 29, Cambridge University Computer Laboratory[1]

1 Introduction

There are no current text based natural language analysis or generation systems that make full use of punctuation, and while there are some that make limited use, like the Editor's Assistant [Dale 1990, Douglas & Dale 1992], they tend to be the exception rather than the rule. Instead, punctuation is usually removed from text before any analysis is carried out, or else the computational systems are designed to ignore any punctuation occurring in the text. In the generation framework, similarly, text tends to be produced without punctuation.

Intuitively, this seems very wrong. Punctuation is such an integral part of written language that it is difficult to imagine naturally producing any significant body of unpunctuated text, or being able to easily understand any such body of text.

However, this is what has been done in the computational linguistics field. The reasons why treatments of punctuation in text have never risen above the cursory level is that there is no clear, coherent treatment or theory of punctuation available. Little real, linguistic work has ever been carried out in the field, so any concepts of punctuation that are incorporated into computational systems have to rely more on personal intuitions than any theoretical account.

Punctuation has long been considered to be intimately related to intonation: that is that different punctuation marks simply give the reader cues as to the possible prosodic and pausal characteristics of the text [Markwardt, 1942]. Nunberg [1990] questions this claim, pointing out not only that the correlation between punctuation and intonation is not very good — just because the roles that intonation and punctuation respectively play in spoken and written language are similar, does not mean that one is simply a transcription of the other into another modality — but also that such a transcriptional account of punctuation is theoretically rather uninteresting.

[1]This work was carried out both under Esprit Acquilex-II, BRA 7315, and an ESRC Research Studentship, R00429334171. Grateful thanks are due for comments on the research and this paper to Ted Briscoe, John Carroll, Robert Dale, Henry Thompson and anonymous COLING reviewers. A condensed version of this paper is to appear in the Proceedings of the 15th International Conference on Computational Linguistics (COLING-94), Kyoto, Japan, August 1994

However, even if we recognise that punctuation fulfils a linguistic role of its own, it is by no means clear how this role is defined. Since there is still no concise linguistic account of the function of punctuation, we have to rely mainly on personal intuitions. This in turn introduces new problems, since there is a great deal of idiosyncrasy associated with the use of punctuation marks. Whilst most people may agree on core situations in which use of a given punctuation mark is desirable, or even necessary, there are still many situations where their use is less clear.

Of course, most languages formally define certain situations where punctuation is mandatory. The quantity of these situations varies depending on the language. Certain languages have a minimal level of formal definition, so that about the only necessary punctuation in English is a full-stop[2] or similar value symbol (e.g. exclamation mark) at the end of a sentence. Other languages have a rather more formal punctuation system though, so that in German, for example, a comma is mandatory before all finite subordinate clauses.

Although we have acknowledged that there is no coherent theory of punctuation, this is not to say that there are no accounts of the subject at all. Therefore in considering the role of punctuation within language we do not have to rely entirely on our own personal intuitions. In his recent review, Humphreys [1993] suggests that accounts of punctuation fall into three categories:

> "The first, prototypically authored by retired schoolmasters, is selflessly dedicated to the task of bringing Punctuation to the Peasantry...they usually contain lengthy pleas for the better treatment of the semi-colon, a stop which the author invariably considers to have been "wantonly neglected" of late. The second sort is the Style Guide, written by editors and printers for the private pleasure of fellow professionals and, in consequence, rarely seen in the open. The third, on the linguistics of the punctuation system, is much the rarest of all...Presumably this paucity reflects a widespread belief that, mere stipulation aside, there really is nothing much to be said of the topic."

So while publishers' style guides are by far the most numerous type of account of punctuation, there are many problems associated with their use. Since they are designed primarily for the production of language rather than for its analysis, they tend to be rather too prescriptive in their approach; many of the constructs that could be found in 'real' text are excluded on aesthetic grounds, or on a point of principle. Since we need to find out what information a given punctuation mark can tell us, such exclusions are unhelpful in the extreme. In addition, by their very nature, style guides only describe the use of punctuation, not its meaning, which is the more important aspect for us.

Thus whilst style guides are valuable sources of punctuation information, they are of limited use for the construction of any theory of the function of punctuation or for implementing any practical treatment. In order to investigate the role of punctuation within language, and hence its possible role within computational linguistics, it is the academic studies of punctuation that we must refer to. These, however, are far from numerous. In the work of Dale [1991], the potential of punctuation in the fields of discourse and natural language generation is explored. However, little mention is made anywhere of the role of punctuation within a syntactic framework.

[2]Throughout this paper I shall refer to sentence-final dots as full-stops rather than periods, to avoid confusion.

Therefore the current investigation tries to determine whether taking consideration of punctuation can further the goals of syntactic analysis of natural language, that is, to decrease the amount of ambiguity in the analysis without sacrificing the accuracy of the output, and therefore also to provide a truer and more accurate linguistic analysis.

2 Punctuation

Punctuation, as we consider it, can be defined as the central part of the range of non-lexical orthography. Although arguments could be made for including the sub-lexical marks (e.g. hyphens, apostrophes) and structural marks (e.g. bullets in itemisations, underlining), they are excluded since the former tend to be lexicalised or are very procedural in their application (e.g. pluralisation of nouns with an apostrophe) and the latter are rather difficult to represent. Indeed, it is difficult to imagine the representation of structural punctuation, other than through the use of some special structural description language such as SGML.

Within our definition of punctuation then, there are broadly three functions that a punctuation mark can perform: delimiting, separating and disambiguating, as described by Nunberg [1990] (although Nunberg only refers to the latter in a footnote). These categories are each distinct from one another, and perform separate linguistic functions. Some marks though, the comma especially, fall into multiple categories since they can have different roles, although each individual instance of such a multi-category punctuation mark will only have one particular role, and hence fall into a single category.

Delimiters (e.g. commas, dashes, parentheses) occur immediately before and after a particular lexical expression, and their effect is to remove that expression from the immediate syntactic context of the surrounding sentence. The delimited phrase then acts as a modifier to adjacent phrases in the original sentence instead. In the examples (1) *my best friend* serves to modify *John* and *that would be best* modifies *I will go to school.*

(1) John, my best friend, fell over and injured himself.
I will go to school — that would be best —- or you will have to come.

Separating punctuation marks (e.g. commas, semicolons) occur between similar grammatical items and indicate that the items form a list (2). Therefore there is a great deal of similarity in behaviour between separators and conjunctions, and the former seem to abbreviate for the latter in a certain sense since most lists of items, instead of being comma-separated, could be linked with conjunctions such as *and.* Sometimes the final conjunction of a list can be replaced by a separator too, but usually only non-final conjuncts are replaced. In addition to the above, sometimes a separating comma can also occur before the final conjunction of a list, although this is rare and tends to happen where the separated items are linguistically complex or long.

(2) I came; I saw; I conquered.
I need to buy butter, eggs, milk and flour.

Disambiguating marks, usually commas, occur in positions where an unintentional ambiguity could result if the marks were not there (3), and so perhaps illustrate best why the use of punctuation within natural language systems could be beneficial.

(3) Earlier, work was halted.
Those who can, contribute to the fund.

In addition to the consideration of the nature of different punctuation marks, it is necessary to understand the interaction of such marks in the various phenomena described by Nunberg [1990], since these phenomena will be fundamental to the implementation and treatment of punctuation in any framework:

Point absorption: Of the point symbols (i.e. all punctuation except bracket symbols), a stronger point will absorb a weaker point in accordance with a strict hierarchy if they are immediately adjacent. Commas are the weakest point symbols, and full-stops are the strongest, with dashes, colons and semicolons in ascending strength in between. In the examples (4) a full-stop has absorbed a comma, and a semi-colon has absorbed a dash;

(4) It was John, my friend.
We heard the sound of engines — that deep roar; the race had started.

Bracket absorption: If a point symbol occurs directly to the left of a final quotation mark or parenthesis then it is removed. Note that there are exceptions to this: for instance a full-stop before a final parenthesis will be retained. A comma and a full-stop are removed in the examples (5);

(5) ...(my brother, the teacher)...
He said, "The butler did it" and solved the case.

Quote transposition: point punctuation directly to the right of a final quotation mark is moved to the left of that character (6). It is of course important that any procedure implementing quote transposition interacts in the proper order with that of bracket absorption, since otherwise a transposed character could subsequently be absorbed! It should be stressed that transposition only occurs with those quotes acting as 'speech marks', and not general 'accenting' quotes, such as those used in this sentence. The phenomenon of quote transposition occurs chiefly in American English, but it can occur generally too;

(6) He said, "I love you."
I said, "Do it now," but nobody listened.

Graphic absorption: orthographically, but not linguistically, similar coincident symbols are absorbed (7). Thus the dot marking an abbreviation will absorb an adjacent period whereas it would not absorb an adjacent comma. Since question marks and exclamation marks can co-occur with commas, there is reason to believe that they function not as types of full-stop, but as non-punctuational stress markers that simply graphically absorb full-stops at the end of a sentence.

(7) I work for the C.I.A., not the F.B.I.
You work for the C.I.A.?

Semicolon promotion: if in a list of comma-separated items one of the items itself contains a point punctuation mark, the commas may be replaced by semi-colons to prevent confusion or ambiguity (8). This process can only occur at the top-level of a sentence (the list may not occur in any further punctuational constructs in the sentence), and unlike a normal list a semi-colon is mandatory before the final conjunct of the list. If it is only the post-conjunct item that contains punctuational complexity, then the commas are not promoted to semi-colons.

(8) There were Mary; Arthur, her teacher; Bill, her dentist; and Fred.

...2003: vaccin contre le sida; 2006: guérison du sida;
2013: médicaments pour prévenir le cancer... (source: Le Figaro)

It should be noted that the phenomena described apply to English, and that even American and British English are claimed to differ in quote transposition. For other languages, the phenomena that apply to the interaction of punctuation need not be the same. In French, for example, the principle of point absorption is markedly different, since dashes and commas can occur adjacently without problem (9). Hence for every language a different set of phenomena will apply, just as a different grammar will apply for the lexical text of the language.

(9) Un travail plus sûr, plus qualifié — donc mieux
rémunéré —, des perspectives... (source: Le Figaro)

In addition to the phenomena associated with the interaction of punctuation, there are also distinct phenomena observable in the interaction of punctuation and lexical expressions. Thus delimited phrases cannot immediately contain delimited phrases of the same type, so that a dash interpolation could not immediately contain another dash interpolation. The sole exception to this may be with parentheticals, though many people object to nested parentheses in natural language. Similarly adjuncts such as the colon-expansion cannot contain further similar adjuncts. A consequence of this, in the context of colon and semicolon scoping, is that the sentence represented in (10) is ambiguous, but that represented in (11) is not. Examples of such sentences are shown in (12) and (13) respectively, taken from Nunberg [1990].

(10) words : words ; words .
(11) words : words ; words : words .

(12) The press secretary gave them the rules: they were not allowed to speak to the committee directly; all other members were forbidden to discuss what the committee had decided.

(13) The press secretary gave them the rules: they were not allowed to speak to the committee directly; all other members were forbidden to discuss what the committee had decided: a hiring freeze would take place.

3 The Grammar

Recognition of punctuational phenomena does not imply that they can be successfully encoded into a NL grammar, or whether the use of such a punctuated grammar will result in any analytical advantages.

Nunberg [1990] advocates the use of two separate grammars, operating at different levels. A lexical grammar is proposed for the lexical expressions occurring between punctuation marks, and a text grammar is proposed for the structure of the punctuation, and the relation of those marks to the lexical expressions they separate. The text grammar has within it distinct levels, such as phrasal and clausal, at which distinct punctuational phenomena can occur.

This should, in theory, make for a very neat system: the lexical syntactic processes being kept separate from those that handle punctuation. However, in practice, this system seems unlikely to succeed since in order to work, the lexical expressions that occur between punctuation marks must carry additional information about the syntactic categories occurring at their edges so that the text grammar can constrain the function of the punctuation marks.

For example, if a sentence includes an itemised noun phrase (14), the lexical expression before the comma must be marked as ending with a noun phrase, and the lexical expression after the comma must be marked as starting with a noun phrase. A rule in the text grammar could then process the separating comma as it clearly comes between two similar syntactic elements.

(14) He likes Willy, Andrew and Tom.
[end: np] [start: np]

However, as (15) shows, the separating comma concept could require information about the categories at arbitrarily deep levels occurring at the ends of lexical expressions surrounding punctuation marks.

(15) I like to walk, skip, and run.
I like to walk, to skip, and to run.
I like to walk, like to skip, but hate to run.

In addition to the problems of correctly processing a punctuation mark, thought must be given to the probable resulting linguistic analysis of the whole sentence. If lexical parsing is to occur before the punctuation stage, then some form of optionally-gapping grammar must be used, since in the examples in (15) the perfectly grammatical sentence *I like to walk* must be optionally gapped for the possibility of extra verbal objects. The only alternative to this is not to carry out a complete parse of the lexical expressions around the punctuation, and to unify lexical and punctuational components before carrying out the final parse.

Therefore even with the edge-category information discussed above, the parsing process is not necessarily made any easier (since often the full partial parses of all the separate expressions will have to be held from the lexical stage and joined at the punctuation stage). Therefore we seem to be at no advantage if we use this approach. In addition, it is difficult to imagine what linguistic or psychological motivation such a separation of punctuation from lexical text could hold, since it seems rather unlikely that people process punctuation at a separate level to the text it surrounds.

Hence it seems more sensible to use an integrated grammar, which handles both words and punctuation. This lets us describe the interaction of punctuation and lexical expressions far more logically and concisely than if the two were separated. Good examples of this are disambiguating commas; in a unified grammar we can simply write rules with an optional comma among the daughters (16) to mark that a comma could occur there if necessary.

(16) s → np (comma) vp.
s → pp (comma) s.

A feature-based tag grammar was written for this investigation (based loosely on one written by Briscoe and Waegner [1992]), and used in conjunction with the parser included in the Alvey Tools' Grammar Development Environment (GDE) [Briscoe et al, 1987], which allows for rapid prototyping and easy analysis of parses. It should be stressed that this grammar is solely one of tags, and so is not very detailed syntactically.

The principal modification to the grammar to specifically handle punctuation was the introduction of the notion of the *stoppedness* of a category. Through this, every category in the grammar has a *stop* feature which describes the punctuational character following it (17), and defaults to [st –] (unstopped) if there is no such character.

(17) the man, = [st c]
with the flowers. = [st f]

As a corollary of the stopping principle, the rules of the grammar further dictate that the mother category inherits the stop value of its rightmost daughter, and hence only rules to specifically add punctuation for categories which could be lexicalised are necessary. Thus a rule for the addition of a punctuation mark after a lexicalised noun would be as in (18). (The GDE terminology for a unification variable is the @-sign followed by an identifying letter.)

(18) n0[st @s] → n0[st –] [punc @s]

We can then specify that top level categories must be [st f] (full-stop), that items in a list should be [st c] (comma), etc. In rules where we want to force a particular punctuation mark to the right of a category, that mark can be included in the rule, with the preceding category unstopped: (19) illustrates the addition of a comma-delimited noun phrase to a normal noun phrase. Specifically mentioning the punctuation mark prevents the possibility of the mother category (i.e. the overall noun phrase) being unstopped, since a requirement of the delimitation is that the delimiting phrase ends with a comma, or, through point absorption, with another punctuation mark. Note that the phenomenon of point absorption has therefore been captured by unifying the value of the *stop* feature of the mother and the identity of the final punctuation mark. In the examples (20) the delimited phrase *Bill Clinton* is always unstopped, and the stop value of the whole noun phrase will therefore be whatever the punctuation mark following the delimited phrase is. Thus the possible values of *st* are all the possible values of *punc* in addition to [st –].

(19) np[st @s] → np[st c] np[st –] [punc @s].

(20) The President, *Bill Clinton* **,** entered.
The President, *Bill Clinton* **;** the Queen and...
More news about the President, *Bill Clinton* **:** he's...
...it's the President, *Bill Clinton* **.**

In a similar fashion the disambiguating (optional) comma as discussed above (16) can be introduced by specifically mentioning it in a rule, but stipulating that the preceding daughter-category of the rule be unstopped, i.e. [st –], as in (21). Although this usage of a comma between the subject noun phrase and predicate verb phrase seems wrong, it is valid for disambiguation as the second example of (3) shows.

(21) s[st @s] → np[st –] ([punc c]) vp[st @s].

An interesting problem associated with this method of inserting punctuation is that of the bracket characters. As discussed in the second section, the closing bracket characters (quotation marks and parentheticals) absorb internal point punctuation, but can co-occur with external point punctuation. In the case of quotation marks, the point punctuation can further be transposed inside the closing mark. Therefore constructions such as *quote comma, close-parenthesis full-stop* are perfectly correct, but seem to be blocked in the example rules previously, that force categories to be unstopped. This has been solved, somewhat inelegantly, by marking bracket-delimited phrases as unstopped, except in particular cases. When quote transposition occurs, for example, the stop value of the phrase inside the quotes should be the stop value of the whole quoted phrase. Also full-stop stopped phrases inside parentheses should make the whole bracket-delimited phrase full-stop stopped. The rules for a quote-delimited noun phrase look something like (22), in simplification.

(22) np[st –] → [punc q1] np [punc q2].
np[st @s] → [punc q1] np[st @s] [punc q2].

Thus the introduction of the *stop* feature seems sufficient to cope with the punctuational phenomena introduced above. In order to incorporate the phenomena of interaction between punctuation and lexical expressions (e.g. preventing immediate nesting of similar delimited phrases), we need to introduce a small number of additional features into the grammar. If, for example, we ensure that a comma-delimited noun phrase includes the feature-value [cm +], we can then stipulate that any noun phrase that includes a comma-delimited phrase must have the feature-value [cm –], so that the two cannot unify (23). Thus *my friend, who sings,* would be marked as [cm –], and hence would be allowed into the phrase *my friend, who sings, Luciano Pavarotti,* but not into *Luciano Pavarotti, my friend, who sings.* The latter is, of course, a valid expression, but has a different structure — both delimed expressions modifying *Luciano Pavarotti.* (Note that the unification of mother and right-most daughter stop values is omitted for clarity of presentation.)

(23) np[cm –] → np[st c] np[cm +, st –]

We can incorporate the relative scoping of colons and semicolons, as discussed in the previous section (examples (10) — (13)), into the grammar very easily too. The semicolon rule (24) can apply with any value of *co* in its arguments, but the colon rule (25) can only apply to categories with [co –]. The mother category of the colon rule bears the feature [co +] to prevent inclusion into further colon-bearing sentences. Note that there are more versions of the colon rule, which deal with different constituents to either side of the colon, and also that, since the GDE does not permit the disjunction of feature values, the semicolon rule is merely an abbreviation of the multiple rules required in the grammar. Stop unification is again omitted.

(24) $\quad$ s[co (@a $\vee$ @b)] $\rightarrow$ s[co @a, st sc] s[co @b].
(25) $\quad$ s[co +] $\rightarrow$ s[co −, st co] s[co −].

In this manner, the inclusion of a few simple extra features in a normal grammar achieves an acceptable treatment of punctuational phenomena. However, since this work only represents the initial steps of providing a full and proper account of the role of punctuation, no claims are made for the theoretical validity or completeness of this approach!

4 The Corpus

For the current investigation it was necessary to use a corpus sufficiently rich in punctuation to illustrate the possible advantages or disadvantages of utilising punctuation within the parsing process. Obviously a sentence which includes no punctuation will be equally difficult to parse with both punctuated and unpunctuated grammars. Similarly, for sentences including only one or two marks of punctuation, the use of punctuation is likely to be rather procedural, and hence not necessarily very revealing: a single punctuation mark in a sentence is almost certain to be a full-stop, which all grammars must cope with implicitly, since they are only fed a sentence at a time; a second punctuation mark is likely to be separating, something that unpunctuated grammars should be able to deal with in some unambiguous manner too.

Therefore the tagged Spoken English Corpus was chosen [Taylor & Knowles, 1988]. This features some very long sentences, and includes rich and varied punctuation. Since the corpus has been punctuated manually, by several different people, some idiosyncrasy occurs in the punctuational style, but there is little punctuation which would be deemed inappropriate to the position it occurs in. Indeed, it could be argued that a degree of idiosyncrasy is helpful, since such idiosyncrasy is likely to be present in most punctuated texts produced without rigid stylistic prescription or editing.

A subset of 50 sentences was chosen from the whole corpus. Between them these sentences include material taken from news broadcasts, poetry readings, weather forecasts and programme reviews, so a wide variety of language is covered.

The lengths of the sentences varied from 3 words to 63 words, the average being 31 words; and the punctuational complexity of the sentences varied from one mark (just a full-stop) to 16 marks, the average being 4 punctuation marks. A sample tagged sentence is shown in (26), where *fs* denotes a full-stop.

(26) Their_APP$ meeting_NN1 involves_VVZ
a_AT1 kind_NN1 of_IO life_NN1 swap_NN1
fs_FS

The punctuated grammar, developed with this subset of the corpus, was used to parse the corpus subset, and then an unpunctuated version of the same grammar was used to parse an unpunctuated version of the same subset. The unpunctuated grammar could have been produced so that it just ignored the punctuation present in the original corpus, but for minor technical reasons this was not done. The reason that testing was performed on the training corpus was that, in the absence of a complete treatment of punctuation, the punctuational phenomena in the training corpus were the only ones the grammar could work with, and although they included almost all of the core phenomena mentioned, slightly different instances of the same phenomena could cause a parse failure. For reference, a small set of novel sentences were also parsed with the grammars, to determine their coverage outside the closed test. It should be stressed, however, that the current investigation is not about the coverage of the grammar, or indeed particularly about its design, but about the problem of whether a grammar that takes account of punctuation performs better than a grammar that is identical, except that it ignores punctuation. The definition of better performance here is fewer analyses (retaining the correct one), since the hits on correct analyses should be identical for the grammars, given one is an unpunctuated version of the other.

The unpunctuated version of the grammar was prepared by removing all the features relating to specifically punctuational phenomena, and also removing explicit mention of punctuation marks from the rules. This, of course, left behind certain rules that were functionally identical, and so duplicate rules were removed from the grammar. Similarly for rules which performed the same function at different levels in the grammar (e.g. attachment of prepositions to the end of a sentence with a comma in the punctuated grammar is also catered for in the unpunctuated grammar by rules allowing prepositions to be attached to the ends of noun and verb phrases).

5 Results

Results of parsing with the punctuated grammar were very good, yielding, on average, a surprisingly small number of parses. The number of parses ranged from 1 to 520, with an average of 38. This average is unrepresentatively high, however, since only 4 sentences had over 50 parses. These were, in general, those with high numbers of punctuation marks, all containing at least 5, as in (27). Ignoring the four smallest and four largest results then, the average number of parses is reduced to just 15. Examples (28) and (29) are more representative of parsing. On examination, a great number of the ambiguities seem to be due to inaccuracies or over-generality in the lexical tags assigned to words in the corpus. The word *more*, for example, is triple ambiguous as determiner, adjective and noun, irrespective of where it occurs in a sentence.

(27) (The sunlit weeks between were full of maids:
Sarah, with orange wig and horsy teeth, was
so bad-tempered that she scarcely spoke;
Maud was my hateful nurse who smelled of
soap, and forced me to eat chewy bits of fish,
thrusting me back to babyhood with threats
of nappies, dummies, and the feeding bottle.)
520 punctuated parses

(28) (The castaway in 'Desert Island Discs', one
hour late, is silly-ass actor Jeremy Lloyd,
who's also known as a scriptwriter.)
4 punctuated parses

(29) (More news about the reverend Sun Myung Moon, founder of the Unification Church, who's currently in jail for tax evasion: he was awarded an honorary degree last week by the Roman Catholic University of la Plata in Buenos Aires, Argentina.)
18 punctuated parses

Besides the ambiguity of corpus tags, a problem arose with words that had been completely mistagged. If these caused the parse to fail completely, the tag was changed in the development phase of the grammar, but even so, the number of complete mistags was rather small in the sub-corpus used: around 10 words in the 50 sentences used.

The linguistic information that can be provided by the use of the punctuated grammar is shown in the representation of one of the possible parses of the sentence in example (29) in figure 1. The main features visible are the separation of the sentences into two main phrases, one to either side of the central colon, and the treatment of the two comma delimited noun phrases describing the subject of the first clause, *the Reverend Sun Myung Moon*. Whilst the correct unpunctuated parse of this sentence would also yield a similar structure as a possibility, there could be far greater ambiguity (more parses) and the unpunctuated parse will be unable to reflect such key punctuation-related linguistic facts as the colon indicating that the following phrase acts as an elaboration or explanation of the preceding one [Dale, 1991].

```
((More_DAR news_NN1
  (about_II
    (((the_AT Reverend_NNS1 Sun_NP1 Myung_NP1 Moon_NP1 cm_CM)
      (founder_NN1 (of_IO (the_AT Unification_NN1 church_NN1))) cm_CM)
     (who_PNQS
      (('s_VBZ)
       ((currently_RR) in_II
        (jail_NN1 (for_IF (tax_NN1 evasion_NN1)))))) co_CO)))
 ((he_PPHS1)
  ((was_VBDZ awarded_VVN) (an_AT1 honorary_JJ degree_NN1)
   (last_MD week_NNT1
    (by_II
     (the_AT (Roman_JJ Catholic_JJ) University_NNL1
      (of_IO
       (la_AT Plata_NP1
        (in_II ((Buenos_NP1 Aires_NP1 cm_CM)
                (Argentina_NP1) fs_FS))))))))))
```

Figure 1: One possible punctuated parse of the sentence in (29)

There did not seem to be any particular relationship between the length of a sentence and the number of parses, or between the quantity of punctuation marks and the number of parses. Indeed several sentences with high numbers of punctuation marks, such as (28) which

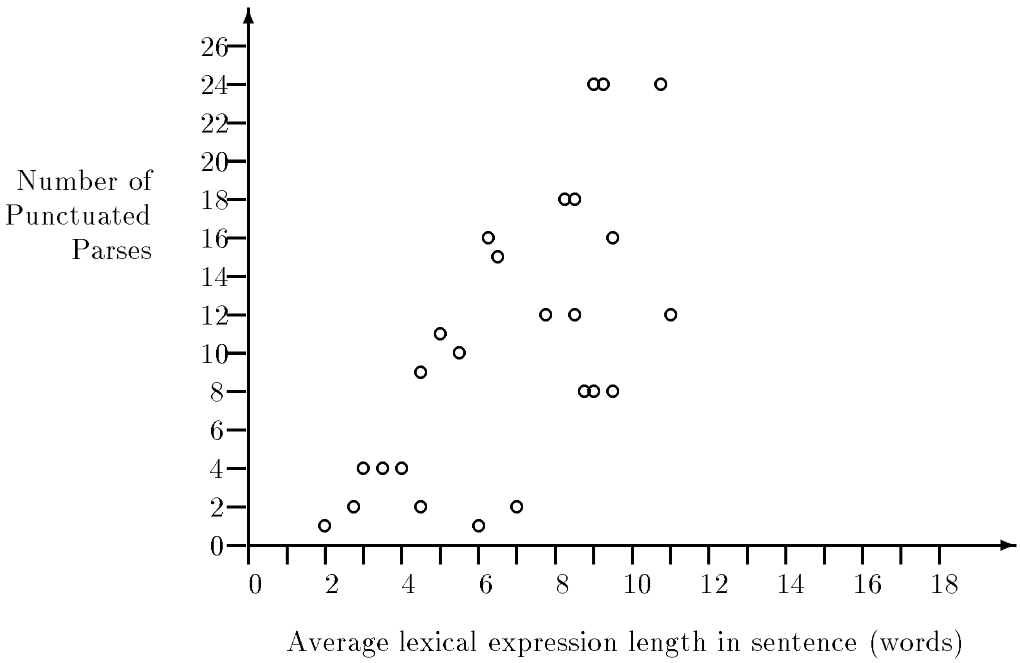

Figure 2: Plot of parse ambiguity against average lexical expression length with the punctuated corpus and grammar.

contains 6, and similarly several long sentences, such as (30), had low ambiguity. However if the two quantities are related, so that we obtain the average length in a sentence of the lexical expressions between punctuation marks, a better relationship emerges. Figure 2 shows a plot of the average expression length against parse ambiguity, for the central 25 sentences of the corpus subset, in terms of their ambiguity (so that very low or very high numbers of parses have been omitted). Whilst the plot is by no means an indication of any formal relationship, it does show the upward trend of parse ambiguity with length of lexical expressions. And in particular, it shows that there are no sentences that have short average lexical expression lengths but high ambiguity. It is interesting to compare this to the general results when parsing sentences conventionally (without punctuation), where greater sentence lengths tend to imply greater parse ambiguity, since it shows that the use of punctuation, in effect, breaks down the sentence into more manageable chunks.

(30) (And in London expect long delays in
Chiswick, where the westbound elevated
section of the M4 is closed for most of today,
and only a single eastbound lane is open.)
2 punctuated parses

Initial attempts at parsing the corpus subset using the unpunctuated version of the grammar were unsuccessful on even the most powerful machine available. This was due to the failure of the machine to calculate the number of parses in the parse forest produced by the chart parser without enumerating them, which would have taken too much memory. A special section of code written for the GDE (grateful thanks are due to John Carroll for supplying this piece of code) to estimate the number of individual parses represented by the packed parse-forest showed that for all but the most basically punctuated sentences, the number of parses was ridiculously huge. The figure for the sentence in (29) was in excess of 6.3×10^{12} parses! Even though this estimate is an upper bound, since effects of feature value percolation during unpacking are ignored, it has been fairly accurate with most grammars in the past and still indicates that rather too many parses are being produced! Not all sentences produced such a massive number of parses: the sentence in (31) yielded only 192 parses with the unpunctuated grammar which was by far the smallest number of unpunctuated parses. Most sentences that managed to pass the estimation process produced between 10^6 and 10^9 parses.

(31) (Protestants, however, are a tiny minority in Argentina, and the delegation won't be including a Roman Catholic.)
9 punctuated parses

On examination of the grammar and the corpus, it is possible to understand why this has happened. The punctuated grammar had to allow for sentences including comma-delimited noun phrases adjacent to undelimited noun phrases, as illustrated by the rules (19) and (23). These are relatively easy to mark and recognise when the punctuation is available. However, without punctuational clues, and with the under-specific tagging system, any compound noun could appear as a set of delimited noun phrases with the unpunctuated grammar.

Therefore the unpunctuated grammar was further trimmed, to such an extent that parses no longer accurately reflected the linguistic structure of the sentences, since, for example, comma delimited noun phrases and compound nouns became indistinguishable. Some manual preparation of the sentences was also carried out to prevent the reoccurrance of simple, but costly, misparses.

The results of the parse now became much more tractable. For basic sentences, as predicted, there was little difference in the performance of punctuated and unpunctuated grammars. Results were within an order of magnitude, showing that no significant advantage was gained through the use of punctuation. The sentences in (32), (33) and (34) received 1, 11 and 84 parses respectively with the unpunctuated grammar.

(32) (Well, just recently, a day conference on miracles was convened by the research scientists, Christian Fellowship.)
4 punctuated parses

(33) (The assembly will also be discussing the UK immigration laws, Hong Kong, teenagers in the church, and of course, church unity schemes.)
2 punctuated parses

(34) (Next week a delegation of nine Protestant ministers from Argentina visits the Autumn assembly of the British Council of Churches; it's meant as a symbol of reconciliation between Christians, following the Falklands War.) 12 punctuated parses

For the most complex sentences, however, the number of parses with the unpunctuated grammar was typically more than two orders of magnitude higher than with the punctuated grammar. The sentence in (35) had 12,096 unpunctuated parses.

(35) (They want to know whether, for instance, in a scientific age, Christians can really believe in the story of the feeding of the five thousand as described, or was the miracle that those in the crowd with food shared it with those who had none?) 24 punctuated parses

Parsing a set of ten previously unseen punctuationally complex sentences with the punctuated grammar resulted in seven of the ten being unparsable. The other three parsed successfully, with the number of parses falling within the range of the results of the first part of the investigation. The parse failures, on examination, were due to novel punctuational constructions occurring in the sentences which the grammar had not been designed to handle. Parsing the unseen sentences with the unpunctuated grammar resulted in one parse failure, with the results for the other 9 sentences reflecting the previous results for complex sentences.

6 Conclusion

An explanation that should be made concerns the ultimate goals of the study of punctuation. The aim is not a genre-specific all-inclusive grammar of punctuation, but rather a general set of ideas that will indicate the function and meaning of a given mark of punctuation in any position that it occurs in. The current study forms a justification for such work. Since the goal is not genre-specific, the facts that the SEC is transcribed speech, and that it is idiosyncratic in punctuation use, according to the whims and linguistic intuitions of the transcribers, do not hinder the investigation in any way.

The current investigation seems to support the original premise — that inclusion and use of punctuational phenomena within natural language syntax can assist the general aims of natural language processing.

We have seen that for the simplest sentences, use of punctuation gives us little or no advantage over the more simple grammar, but, conversely, does no harm and can reflect the actual linguistic construction a little more accurately.

For the longer sentences of real language, however, a grammar which makes use of punctuation massively outperforms an otherwise similar grammar that ignores it. We also see that unlike unpunctuated grammars and texts, where there is some relation between sentence length and parse ambiguity, the ambiguity for punctuated sentences is better related to the lengths of the sequences of lexical expressions that occur between the punctuation marks. It is difficult to see how any grammar that takes no notice of punctuation could ever become successful at analysing such long, punctuated sentences unless some huge amount of semantic and pragmatic knowledge is used to disambiguate the analysis.

There are, however, several limitations to the current approach. In the grammar, there are several inelegant processes, such as in (22), which just served as working solutions to problems encountered. It is more than likely that deeper investigation into punctuational theory will yield better, more elgant ways of solving such problems. Similarly, adding punctuation to a grammar increases its complexity, which can compensate for the removal of other ambiguity. It would be helpful if the increase in complexity could be minimised.

As was shown by the attempt at parsing the novel sentences though, knowledge of the role of punctuation is still severely limited. The grammar only performed reliably on those punctuational phenomena it had been designed with. Unexpected constructs caused it to fail totally. Therefore, following the recognition that punctuation can play a crucial role in natural language syntax, what is needed is a thorough investigation into the theory of punctuation. Then theoretically based analyses of punctuation can play a full and important part in the analysis of language.

7 References

Briscoe, E J; C Grover; B Boguraev; and J Carroll (1987). "A Formalism and Environment for the Development of a Large Grammar of English." In *Proceedings, 10th International Joint Conference on Artificial Intelligence*, Milan, Italy, pp703-708

Briscoe, E J and N Waegner (1992). "Robust Stochastic Parsing Using the Inside-Outside Algorithm." In *Proceedings, AAAI Workshop on Statistically-based NLP Techniques*, San Jose, California.

Dale, R (1991). "Exploring the Role of Punctuation in the Signalling of Discourse Structure." In *Proceedings, Workshop on Text Representation and Domain Modelling*, Technical University Berlin, pp110-120.

Dale, R (1990). "A Rule-based approach to Computer-Assisted Copy Editing." *Computer Assisted Language Learning*, **2**, pp59–67.

Douglas, S and R Dale (1992). "Towards Robust PATR." In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, Nantes, France.

Humphreys, R L (1993). "Book Review: The Linguistics of Punctuation." *Machine Translation*, **7**.

Markwardt, A H (1942). *Introduction to the English Language*, Oxford University Press, New York.

Nunberg, G (1990). *The Linguistics of Punctuation*, CSLI Lecture Notes **18**, Stanford, California.

Taylor, L J and G Knowles (1988). *Manual of Information to Accompany the SEC Corpus*, University of Lancaster.

A A Punctuated Grammar

```
; Grammar for Tag Sequences (CLAWS2, SEC); non binary rules added cf x

FEATURE N{+, -}
FEATURE V{+, -}
FEATURE BAR{0, 1, 2}
FEATURE H{+, -}
FEATURE MINOR{DET, COMP, DEG, PRT, CONJ, INTERJ, NOT}
FEATURE PLU{+, -}
FEATURE POSS{+, -}
FEATURE NTYPE{PRO, PRONOM, NAME, NORM, PART, THERE, NUM, DIR, TIT, PRETIT, POSTTIT, TEMP, MEAS, POSS}
FEATURE ATYPE{ATT, PRD, POST, CAT, NUM, XCOMP, HOW}
FEATURE AFORM{ER, EST, NONE}
FEATURE SUBCAT{SFIN, SCOMP, SINF, NP, VPINF, VPING, NONE}
FEATURE PFORM{FOR, OF, WITH, WITHOUT, WH, THAN, AS, THAT}
FEATURE CJTYPE{BEGIN, END}
FEATURE WH{+, -}
FEATURE ADV{+, -}
FEATURE AUX{TO, BE, HAVE, DO, MODAL, CAT}
FEATURE VFORM{BSE, INF, ING, PPART, PAST, PRES}
FEATURE CONJ{+, -}
FEATURE HD{+, -}
FEATURE T{+}
FEATURE REL{+, -}
FEATURE RR{21, 22}
FEATURE REX{21, 22}
FEATURE II{21, 22, 31, 32, 33}
FEATURE VM{21, 22}
FEATURE DD1{21, 22}
FEATURE ST{C, F, H, G, IL, IR, O, D, S, -}
FEATURE D{+, -}
FEATURE P{+, -}
FEATURE A{+, -}
FEATURE AX{+, -}
FEATURE VB{+, -}
FEATURE F{A, R, D, N, Q}
FEATURE Q{S, D}
FEATURE Y{C, F, H, G, IL, IR, O, D, S}
```

Note all the possible types of punctuation (the Y feature), how two of those punctuations are types of quote (the Q feature) and how the stop feature (ST) can take all the punctuation values in addition to [ST –].

```
TOP T.

SET VHEAD = {N, V, PLU, VFORM}
SET NHEAD = {N, V, PLU, NTYPE, POSS}
SET PHEAD = {N, V, PFORM}
SET AHEAD = {N, V, AFORM, ATYPE, ADV}

ALIAS S = [V +, N -, BAR 2].
ALIAS V1 = [V +, N -, BAR 1].
ALIAS V0 = [V +, N -, BAR 0].
ALIAS N2 = [V -, N +, BAR 2].
ALIAS NP = [V -, N +, BAR 1].
ALIAS N0 = [V -, N +, BAR 0].
ALIAS P2 = [V -, N -, BAR 2].
ALIAS P1 = [V -, N -, BAR 1].
ALIAS P0 = [V -, N -, BAR 0].
ALIAS AP = [V +, N +, BAR 2].
ALIAS A1 = [V +, N +, BAR 1].
ALIAS A0 = [V +, N +, BAR 0].
ALIAS H2 = [H +, BAR 2].
ALIAS H1 = [H +, BAR 1].
```

```
ALIAS HO = [H +, BAR 0].
ALIAS DT = [MINOR DET].
ALIAS CP = [MINOR COMP].
ALIAS DG = [MINOR DEG].
ALIAS PT = [MINOR PRT].
ALIAS CJ = [MINOR CONJ].
ALIAS IJ = [MINOR INTERJ].
ALIAS NG = [MINOR NOT].
ALIAS T = [T +].

.......

PSRULE Ts   : T --> (CJ[CJTYPE END]) (PO[SUBCAT SFIN, ST -]) S[ST F].
PSRULE Tv   : T --> (AP[ADV -]) V1[ST F].
PSRULE Tn   : T --> NP[ST F].
PSRULE Tcon : T --> (CJ[CJTYPE END]) (PO[SUBCAT SFIN, ST -]) S[ST S] T.
```

The top category (i.e. complete parses) can be sentences (with possible propositionals or conjunctions preceding), verb phrases (with possible adjectivals) or noun phrases, provided they are all suitably stopped. There is also a rule for conjunction of sentences at the top level with semi-colons.

```
PSRULE Snv : S[ST @s] --> NP V1[ST @s].
PSRULE Spv : S[ST @s] --> P1[PFORM WH, SUBCAT NONE, ST -] V1[ST @s].

PSRULE Sas  : S[ST @s] --> A1[ADV +] S[D -, ST @s].
PSRULE Sts  : S[ST @s] --> NP[NTYPE TEMP] S[D -, ST @s].
PSRULE Sis  : S[ST @a] --> IJ ([Y C]) S[D -, ST @a].
PSRULE Sps  : S[ST @s] --> P1  S[D -, ST @s].

PSRULE Snos : S[D +, ST @s] --> NP[ST O] S[D -, ST @s].
PSRULE Sfos : S[D +, ST @s] --> (AP[ADV +]) [F N, ST O] S[D -, ST @s].
PSRULE Svos : S[D +, ST @s] --> V1[ST O] S[D -, ST @s].
PSRULE Snds : S[D +, ST @s] --> NP[ST D] S[D -, ST @s].
PSRULE Ssos : S[D +, ST @s] -->  S[ST O] S[D -, ST @s].

PSRULE Sscp : S[ST @s] --> S[D -, ST C] P1.
PSRULE Sscd : S[ST @s] --> S[D -, ST C] A1[ADV +, ST @s].
PSRULE Sacs : S[ST @s] --> [F A, ST C] S[D -, ST @s].
PSRULE Sgcs : S[ST @s] --> V1[VFORM ING, ST C] S[D -, ST @s].
PSRULE Ssca : S[ST @s] --> S[D -, ST C] [F A, ST @s].

PSRULE Tc : T --> (S[D +, ST C])* S[D +, ST -] ([Y C]) CJ[CJTYPE END] S[D +, ST @s].
```

At the sentence level we construct sentences, in the most simple instance, from nominal and verbal phrases. There are rules to augment sentences too, by adding adverbial, temporal, prepositional or interjectional constituents to their beginning. The D feature ensures that these applications occur in the correct order with respect to other sentential modifications.

Colons and dashes are added in the next batch of rules, and it should be noted that in this instance the colon rules take rather different forms to those described in the main text.

There are also rules for augmentation of sentences with comma-stopped phrases, and it should be noticed that the punctuation is forced both in these rules and the previous ones.

Finally, there is an example of the new conjunction rule, as applies in the new situation where items can be comma-conjoined. Note the possibility of the occurrence of a comma before the conjunction.

```
PSRULE FQn : [F Q, ST @s] --> NP[ST @s].
PSRULE FQi : [F Q, ST -]  --> (IJ)+.
```

```
PSRULE FRn : [F R, ST @s] --> N2[WH +, NTYPE PRO] V1[ST @s].
PSRULE FRp : [F R, ST @s] --> PO[PFORM THAT]V1[ST @s].

PSRULE FDp : [F D, ST @s] --> P1[ST @s].
PSRULE FDn : [F D, ST @s] --> NP[D -, ST @s].
PSRULE FDf : [F D, ST @s] --> [F R, ST @s].
PSRULE FDs : [F D, ST @s] --> NP[ST -] V1[ST @s].

PSRULE FNp : [F N, ST @s] --> PO[SUBCAT SFIN] (A1[ADV +]) (P1) NP[ST -] V1[ST @s].
PSRULE FNd : [F N, ST @s] --> DT[WH +] V1[ST @s].
PSRULE FNn : [F N, ST @s] --> NP[ST -] V1[ST @s].

PSRULE FAO  : [F A, ST @s] --> (A1[ADV +]) PO[PFORM WH] NP[ST -] V1[ST @s].
PSRULE FA1  : [F A, ST @s] --> (A1[ADV +]) P1[PFORM WH, SUBCAT NONE] NP[ST -] V1[ST @s].
PSRULE FAc  : [F A, ST @s] --> (A1[ADV +]) [N -, V -, PFORM WH, SUBCAT @b] CJ[CJTYPE END]
                                [N -, V -, PFORM WH, SUBCAT @a] NP[ST -] V1[ST @s].
```

This section contains rules for the construction of various clauses, both the specifically punctuational (quoted and delimited) and those used in the hand-parsed varsion of the SEC.

The quoted clauses (FQ) describe the two instances of quotation use in the corpus, and the delimited clauses - FD - (which will be attached to NPs in that section) are shown to be able to consist of sentences (unpunctuated), prepositional phrases, noun phrases or relative clauses.

```
PSRULE Vv   : VO --> A1[ADV +] HO[A -].
PSRULE Vva  : VO[A +] --> (A1[ADV +]) HO[AUX @a].
PSRULE Vnot : VO[AUX @a] --> HO[AUX @a] NG.

PSRULE V    : V1[A @a, ST -]  --> HO[A @a].
PSRULE Vm   : V1[A @a, ST @s] --> HO[A @a] [Y @s].
PSRULE Vp   : V1[ST @s] --> HO P1[ST @s].
PSRULE Vpa  : V1[ST @s] --> HO (P1) AP[ST @s].
PSRULE Vn   : V1[ST @s] --> HO NP[ST @s].
PSRULE Vnn  : V1[ST @s] --> HO NP[ST -] NP[NTYPE TEMP, ST @s].
PSRULE Vna  : V1[ST @s] --> HO NP AP[ST @s].
PSRULE Vnna : V1[ST @s] --> HO NP[ST -] NP[NTYPE TEMP] AP[ST @s].

PSRULE V1f : V1[D +, ST @s] --> H1[A -, D -, ST -] (A1[ADV +]) V1[VFORM INF, ST @s].
PSRULE V1g : V1[D +, ST @s] --> H1[A -, D -, ST -] (A1[ADV +]) V1[VFORM ING, ST @s].
PSRULE V1n : V1[D +, ST @s] --> H1[D -, ST -] (A1[ADV +]) [F N, ST @s].
PSRULE V1p : V1[D +, ST @s] --> H1[D -, ST -] (A1[ADV +]) [F N, ST C].
PSRULE V1a : V1[D +, ST @s] --> H1[D -, ST -] (A1[ADV +]) [F A, ST @s].

PSRULE Vnt : V1[A +, ST @s] --> H1[A +, ST -] NG NP[ST @s].

PSRULE Vc : V1[D +, ST -] --> (V1[D +, ST C])* V1[D +, ST -] ([Y C]) CJ[CJTYPE END] H1[D +, ST -].

PSRULE Vpar : VO[A @a] --> HO[AUX @x]     (NP[NTYPE TEMP]) VO[A @a, VFORM PPART].
PSRULE Vdo  : VO[A @a] --> HO[AUX DO]     (NP[NTYPE TEMP]) VO[A @a, VFORM BSE].
PSRULE Vto  : VO[A @a] --> HO[AUX TO]     (NP[NTYPE TEMP]) VO[A @a, VFORM BSE].
PSRULE Vhav : VO[A @a] --> HO[AUX HAVE]   (NP[NTYPE TEMP]) VO[A @a, VFORM PAST].
PSRULE Vmod : VO[A @a] --> HO[AUX MODAL]  (NP[NTYPE TEMP]) VO[A @a, VFORM BSE].
PSRULE Vbe  : VO[A @a] --> HO[AUX BE]     (NP[NTYPE TEMP]) VO[A @a, VFORM ING].
```

This is the rather complex range of verbal rules - extra features, such as A, are necessary to distinguish between verb phrases formed by auxiliary verbs and their participles, and those formed from the promotion of auxiliary verbs without participles. The distinction is necessary since otherwise ambiguity can arise in the cases where further infinitival or gerund phrases are added. Since the SEC does not distinguish between intransitive, transitive and ditransitive verbs, there is much scope for ambiguity in the addition of noun phrases and prepositions to the end of a verb phrase.

```
PSRULE NP  : NP[ST -]  --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) (NO)* HO.
PSRULE NPm : NP[ST @s] --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) (NO)* HO[] [Y @s].
PSRULE NPp : NP[ST @s] --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) (NO)* HO   P1[D -, ST @s].
PSRULE NPf : NP[ST @s] --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) (NO)* HO  (P1[D -, ST -]) [F R, ST @s].

PSRULE NP2  : NP[ST -]  --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) H2.
PSRULE NP2m : NP[ST @s] --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) H2[] [Y @s].
PSRULE NP2p : NP[ST @s] --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) H2   P1[D -, ST @s].
PSRULE NP2f : NP[ST @s] --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) H2  (P1[D -, ST -]) [F R, ST @s].

PSRULE NPq  : NP[NTYPE NORM, ST -]  --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) [Q @q] [F Q] [Q @q].
PSRULE NPqm : NP[NTYPE NORM, ST @s] --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) [Q @q] [F Q] [Q @q] [Y @s].
PSRULE NPqp : NP[NTYPE NORM, ST @s] --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) [Q @q] [F Q] [Q @q] P1[D -, ST @s].
PSRULE NPqt : NP[NTYPE NORM, ST @s] --> (DT) (N2[NTYPE NUM]) (AP[ADV -, ST -]) [Q @q] [F Q, ST @s] [Q @q].

PSRULE Nref : NP[ST @s] --> H1[ST -] N2[NTYPE PRO, POSS -, WH -, ST @s].
PSRULE Npos : NP[ST @s] --> NP[NTYPE POSS] H1[ST @s, D -].

PSRULE NPd : NP[D +, ST @s] --> H1[D +, ST C] [F D, ST -] [Y @s].
PSRULE NPe : NP[D +, ST @s]  --> H1[D +, ST C] [F D, ST -] [Y @y] P1[D -, ST @s].

PSRULE NPc : NP[D +, ST @s] --> (NP[D -, ST C])* NP[D -, ST -] ([Y C]) CJ[CJTYPE END] (A1[ADV +]) H1[D -, ST @s].
```

Since the use of noun phrases in the SEC is so very varied, with no guarantee of an article, I have somewhat non-standardly reclassified noun phrases as bar-level 1 and categorised those nominals that appear in the lexicon as phrasal as bar-level 2. Thus the nou phrases formed from these two types of nominal are shown in the first two sets of rules, which permit the addition of a relative clause to the end of the phrase. Then, since I am treating quotations as nominals here (which is not necessary the most valid process, but works in the present circumstances) there is a similar section treating quoted clauses.

Following rules for reflexive and possessive cases, there are two rules for delimited noun phrases, where the considerations in the main text concerning nesting of delimited phrases have been taken into account with the D feature. Finally, there is another implementation of the new conjunction rule.

```
PSRULE P1np : P1[ST @s, D @d] --> (A1[ADV +, ST -]) HO[SUBCAT NP] NP[ST @s, D @d].
PSRULE P4 : P1[ST @s] --> HO[SUBCAT VPING] V1[VFORM ING, ST @s].
PSRULE P5 : P1[ST @s] --> HO[SUBCAT VPINF] V1[VFORM INF, ST @s].
PSRULE Padj : P1[ST @s] --> HO[SUBCAT NONE, ST -] AP[ADV -, ST @s].
PSRULE Pvp : P1[ST @s] --> HO[SUBCAT NONE, ST -] V1[ST @s].
PSRULE P1prpn : PO[SUBCAT @c, ST @s] --> HO[SUBCAT @c, ST -] [Y @s].

PSRULE P1bcon : P1[CONJ -] --> CJ[CJTYPE BEGIN] H1[CONJ +].

PSRULE Cp1- : P1[ST @s] --> H1[ST -] H1[CONJ +, ST @s].
PSRULE Cp1+ : P1[CONJ +, ST @s] --> CJ[CJTYPE END] H1[ST @s].
PSRULE Cp-  : PO[ST @s] --> HO[ST -] HO[CONJ +, ST @s].
PSRULE Cp+  : PO[CONJ +, ST @s] --> CJ[CJTYPE END] HO[ST @s].

PSRULE Apun : A1[ST @s] --> H1[ST -] [Y @s].
PSRULE A1prom : A1[ST @s] --> HO[ST @s].
PSRULE Achn : A1[ADV -] --> AO[ADV -] [Y C] HO[ADV -].
PSRULE A2 : A1[ST @s] --> DG H1[ST @s].
PSRULE A3 : A1[ST @s] --> HO[ADV -] P1[ST @s].
PSRULE A5 : A1[ST @s] --> HO V1[VFORM INF, ST @s].

PSRULE APavp : AP[ST @s] --> A1[ADV +, ST @s].
PSRULE APadv : AP[ADV -, ST @s] --> A1[ADV +] H1[ST @s].
PSRULE APadn : AP[ADV +, ST @s] --> H1[ADV +] NP[ST @s].
PSRULE APprom : AP[ADV -, ST @s] --> H1[ADV -, ST @s].
PSRULE A1bcon : A1[CONJ -] --> CJ[CJTYPE BEGIN] H1[CONJ +].
PSRULE Acomp : AO[HD +, ST -] --> AO[HD -] HO[ST -].
```

```
PSRULE Ca1- : A1 --> H1 H1[CONJ +].
PSRULE Ca1+ : A1[CONJ +] --> CJ[CJTYPE END] H1.
PSRULE Ca-  : A0 --> H0 H0[CONJ +].
PSRULE Ca+  : A0[CONJ +] --> CJ[CJTYPE END] H0.
```

Rules for prepositional, adjectival and adverbial phrases are little changed from the original grammar. The changes that have been made have been as responses to specific syntactic circumstances found in the corpus. The biggest changes have probably been made as concerns adverbs, particular in connection with their interaction with punctuation, almost invariably commas.

```
PSRULE Aofc : A0[AFORM NONE, ADV +, ST -] --> [RR 21] [RR 22].
PSRULE Afri : A0[AFORM NONE, ATYPE XCOMP, ADV +, ST -] --> [REX 21] [REX 22].
PSRULE P2pl : P0[ST -] --> [II 21] [II 22].
PSRULE P3pl : P0[ST -] --> [II 31] [II 32] [II 33].
PSRULE VMpl : V0[AUX MODAL, CONJ -, ST -] --> [VM 21] [VM 22].

PSRULE DDa : DT[PLU -, WH -, POSS -] --> [DD1 21] [DD1 22].
PSRULE DDb : N2[HD -, NTYPE PRO, PLU -, POSS -, WH -, CONJ -, ST -] --> [DD1 21] [DD1 22].
```

Specific rules to handle phrasal lexemes in the corpus.

```
PROPRULE HFC_V : [V +, N -] --> [H +], U. F(0) = F(1), F in VHEAD.
PROPRULE HFC_N : [V -, N +] --> [H +], U. F(0) = F(1), F in NHEAD.
PROPRULE HFC_P : [V -, N -] --> [H +], U. F(0) = F(1), F in PHEAD.
PROPRULE HFC_A : [V +, N +] --> [H +], U. F(0) = F(1), F in AHEAD.
PROPRULE AGR1 : [V +, N -, BAR 2] --> [N +, V -], [H +]. PLU(1) = PLU(2).
PROPRULE WH1 : [V -, N +, BAR 1] --> [H +], U. WH(0) = WH(1).

DEFRULE CONJ1 : [N, V, ~CONJ] --> [], U. CONJ(0) = -.
DEFRULE CONJ2 : [] --> [N, V, ~CONJ], U. CONJ(1) = -.

.......

WORD FS : [Y F].
WORD CM : [Y C].
WORD YG : [Y G].
WORD YH : [Y H].
WORD YIL : [Y IL].
WORD YIR : [Y IR].
WORD CO : [Y O].
WORD SC : [Y S].
WORD QM : [Y F].
WORD EM : [Y F].
WORD DH : [Y D].
WORD DQ : [Q D].
WORD SQ : [Q D].
```

The lexical entries for punctuation marks.