Punctuation in a Lexicalized Grammar

Christine Doran*

The MITRE Corporation
202 Burlington Road
Bedford, MA 01730

Abstract

Just as people make use of information from punctuation to structure and understand text, NLP systems can use information from punctuation in processing texts automatically. The aim of the research presented here was to explore the feasibility of treating a sizable core of punctuation phenomena at the level of the sentence grammar. A large set of punctuation rules were manually derived from naturally occurring data, and added to the XTAG English grammar. Our results confirm that punctuation can be used in analyzing sentences to increase the coverage of the grammar, reduce the ambiguity of certain word sequences and facilitate later processing of larger text units, without either adversely impacting the existing grammar or deriving analyses which would be incompatible with later levels of processing.

1. Motivation

Punctuation helps us to structure, and thus to understand, texts. Many uses of punctuation straddle the line between syntax and discourse, because they serve to combine multiple propositions within a single orthographic sentence. They allow us to insert discourse-level relations at the level of a single sentence. Just as people make use of information from punctuation in processing what they read, natural language processing systems can use information from punctuation in processing texts automatically.

Most current NLP systems fail to take punctuation into account at all, losing a valuable source of information about the text. Those which do mostly do so in a superficial way, again failing to fully exploit the information conveyed by punctuation. To be able to make use of such information in a computational system, we must first characterize its uses and find a suitable representation for encoding them.

Previous work on punctuation was mostly of the descriptive variety, of which Quirk et al. (1985) and Sampson (1995) are particularly good instances. Some linguistic work has been done by Chafe (1988), Schmidt (1995), Jones (1996b) and Meyer (1987). Nunberg (1990) offers the most comprehensive linguistic discussion of punctuation to date, with an extensive analysis of the interactions of different punctuation marks. He is primarily interested in characterizing punctuation as a formal system, independent from syntax. Briscoe (1994) presents an treatment of punctuation within the Alvey Natural Language Tools grammar. He and Carroll (1995) show that this analysis considerably reduces ambiguity in parsing the SUSANNE corpus (a subset of the Brown corpus) and Jones shows similar results.

The work discussed here differs from previous work in a number of ways. It includes an analysis of the syntax of punctuation which has been implemented and integrated into a large English

This work was done while the author was a graduate student at the University of Pennsylvania, and was partially supported by NSF Grant SBR8920230 and ARO Grant DAAH0404-94-G-0426. Thanks to Aravind Joshi and Ted Briscoe for their helpful comments at all stages of this work, and to the members of the XTAG Project.

grammar that is being used on an everyday basis. In addition, the analysis differs considerably from those of Jones and Briscoe in treating punctuation within a framework which allows for more concise characterization of the non-local aspects of certain uses of punctuation. Furthermore, neither of their implementations cover the range of punctuated constructions our treatment does.

2. Analysis

Many parsers require that punctuation be stripped out of the input. Where punctuation is optional, as is often the case, this may have no effect. However, there are a number of constructions where punctuation is obligatory. Adding analyses of these to the grammar without the punctuation can lead to severe over-generation, possibly to the point where it is better to not add the constructions at all.

The work here focuses on extending a lexicalized syntactic grammar to handle phenomena occurring within a single sentence which have punctuation as an integral component. The main job of the sentence grammar, then, is to produce a structure that makes the appropriate units easily accessible to later levels of processing–not just basic grammatical elements like subject noun and verb group, but more complex relations like nominal apposition as well. Punctuation marks are treated as full-fledged lexical items in a Feature-Based Lexicalized Tree Adjoining Grammar (Joshi, 1985; Schabes, 1990; Vijay-Shanker & Joshi, 1991). The localization of both syntactic and semantic dependencies provides an elegant framework for encoding punctuation in the sentence grammar. The elementary units of LTAG are of a suitable size for stating most of the constraints we are interested in, and the derivation histories it produces contain information that later stages of processing will need about which elementary units have been used and how they have been combined. Each punctuation mark or pair of marks anchors its own elementary trees and imposes constraints on the surrounding lexical items. The TAG adjunction operation is advantageous in handling paired punctuation marks, because it allows us to keep both pieces of the complex object, e.g. a pair of parentheses or commas, in the same elementary tree, regardless of the size of the constituent they enclose.

We have analyzed naturally-occurring data (primarily from the Brown Corpus) representing a wide variety of constructions, and added treatments of them to the XTAG English grammar. The new trees are of two types. The first have the punctuation marks as anchors, reflecting the fact that they do not strongly constrain the lexical content of the constructions they participate in. For example, any NP except a pronoun can be an appositive, and this is reflected in the analysis by having the NP position as a substitution site in the NP appositive tree (Figure 1). The second type of tree has the punctuation marks as substitution sites, for instance the tree for parenthetical adverbs, where the lexical material may vary, some punctuation mark is required, but any of several types of punctuation mark is permissible. This is illustrated by the tree for a quoting clause shown in Figure 2. There are a total of 47 trees containing punctuation marks in the current implementation. Doran (1998) discusses all of the trees in more detail.

The full set of punctuation marks is divided into three classes: **balanced**, **structural** (term from (Meyer, 1987)) and **terminal**. The balanced punctuation marks are quotes and parentheses, structural are commas, dashes, semi-colons and colons, and terminal are periods, exclamation points and question marks. These three types of punctuation are essentially independent subsystems, and a given constituent will typically have only one of each type. Structural and terminal punctuation marks do not occur adjacent to other members of the same class, but may occasionally occur adjacent to members of the other class, e.g. a question mark on a clause which is separated by a dash from a second clause. Balanced punctuation marks are sometimes adjacent to one another, e.g. quotes immediately inside of parentheses as in example

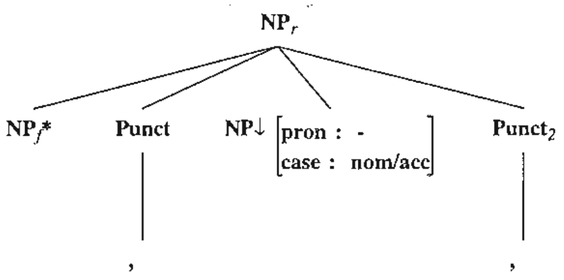

Figure 1: The non-peripheral NP appositive tree, showing relevant features.

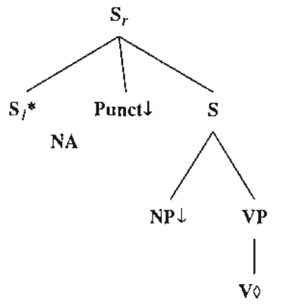

Figure 2: Tree for a quoting clause which follows the quote; the tree would be anchored by e.g. *mutter* in a sentence such as *Liver again, Mary muttered.*

(1). Features allow us to control these local interactions. We also use these features to control non-local phenomenon such as quote alternation, whereby single and double quotes alternate when embedded, and also to control the embedding of colons and semi-colons.

(1) Each enjoys seeing the other hit home runs ("I hope Roger hits 80", Mantle says), and each enjoys even more seeing himself hit home runs ("and I hope I hit 81"). [Brown:ca39]

2.1. How punctuation improves the grammar

There are two primary ways in which adding punctuation improved the coverage and performance of the XTAG English grammar. First, it allowed us to add some syntactically "exotic" constructions which would have previously been considered too unconstrained in their unpunctuated forms. Many such constructions occur with great frequency in naturally occurring texts. As an example, consider noun appositives, where an NP modifies another NP. Example (2) has two appositives. Without access to punctuation, the parser would derive every combinatorial possibility of NPs in noun sequences, which is obviously undesirable (especially since there is already unavoidable noun-noun compounding ambiguity). These phrases must be "bracketed" by punctuation, which provides precisely the sort of additional constraint we need to make the parsing task manageable. By adding a treatment of punctuation to the grammar, we can recognize and correctly analyze appositive constituents. Other similar such constructions include parenthetical elements, reported speech, compound sentences, comma coordination and vocatives. None of these constructions were handled by our English grammar before it was extended to treat punctuation.

(2) But Tony Robinson, *the current sheriff of Nottingham* – **a job that really exists** – rejected the theory, saying that "as far as we are concerned, Robin Hood was a Nottinghamshire lad." [clari.living.celebrities]

Second, punctuation provides additional constraints for parsing constructions already handled by the grammar. In developing a large grammar for any language, one of the fundamental concerns is the increase in ambiguity of derivations which invariably accompanies any increase in coverage of the language's constructions. Adding punctuation to the grammar reduces the ambiguity of analyses by marking the boundaries of clauses and phrases. Adding analyses of subordinate clauses, the majority of whose variants include punctuation, was found in (Doran, 1996) to improve the coverage of the XTAG English grammar by 6.6% on Brown corpus data.

2.2. *Previous Work*

Information from punctuation has only recently been taken into consideration in parsing and grammar development (see (Briscoe, 1994; Jones, 1996b)). The only other such grammar to treat punctuation integrally is a POS-tag sequence grammar developed by (Briscoe & Carroll, 1995) using the Alvey Natural Language Tools as a starting point, which includes Briscoe's analysis of punctuation. Unlike the present work, they do not look at the particular lexical items in the input string, only the POS sequence. However, they do treat punctuation "lexically" to a certain extent, in the sense that each punctuation mark occurs in a range of (discourse) grammar rules.

3. Evaluation

Ideally, we would evaluate the punctuation rules using full parsing—take a corpus of sufficiently complex sentences, parse it both with and without the punctuation marks, and measure the improvements in coverage and accuracy when the punctuation is taken into consideration. However, such an experiment proved impossible for practical reasons because our current parser runs out of memory on sentences of any interesting length with their punctuation stripped.[1]

Another way to measure the improvement in the grammar is to use the supertagging technique developed by (Srinivas, 1997). Supertagging takes the trees of an LTAG, and uses them as complex part-of-speech tags. To evaluate the LTAG punctuation analysis, we used a supertagger trained on just over 1 million words of Wall Street Journal data whose supertags were derived by conversion from the (hand-corrected) Treebank parses. We first trained the tagger on the data with all punctuation stripped, and tested it on 2012 held-out sentences, also with punctuation stripped. We then retrained the tagger on the full million words, and tested it on the same test data with punctuation retained. The performance is shown in Table 1. The most important line is the middle one, showing performance of both sets of training data on exclusively non-punctuation tokens. We achieved an error reduction of 10.9% on non-punctuation tokens, showing that the presence of punctuation does indeed improve the accuracy of analysis of the surrounding texts. Our result reflects an increase in the number of non-punctuation tokens to which the correct structural tag was assigned only when punctuation was present. This figure is not directly comparable to the coverage improvement obtained by (Briscoe & Carroll, 1995) of 8%, which reflects an increase in the number of sentences for which some parse (not necessarily correct) was obtained. Nor can it be compared with their improved crossing brackets performance on SUSANNE sentences, which looks at the number of correct constituents. Supertagging accuracy is measured on a per word basis, and always assigns a tag to every word,

[1] Jones (1996a) encounters the same problem in attempting to evaluate his grammar. His chart-based parser cannot enumerate the number of parses possible for many of the unpunctuated sentences in his test set, and he has to turn to a special estimation process which interrupts the parser before it actually builds any parses.

so there is no notion of complete failure on a sentence. In that sense, supertagging does assign a structure to every sentence, but without assembling the supertag sequence assigned, you do not know what the hypothesized constituents are. The most appropriate comparison is with the evaluation presented in (Briscoe, 1994), where he finds a 2% improvement in "rule application" on SUSANNE sentences (i.e. the correct derivational step applied at a given point) since we can think of each LTAG tree as a rule (or possible several rules) to be applied.

	Trained and tested	
	on text without punctuation	with punctuation
% Correct		
Overall	87.1%	88.0%
On non-punct tokens	**87.1%**	**88.5%**
On punct tokens	—	83.7%

Table 1: Accuracy of supertagging with and without punctuation

One important thing to remember is that the supertagger has only a three-token window in assigning tags, and constructions involving punctuation often span a fairly large number of tokens (e.g. the comma around a relative clause, parentheses around sentences). This suggests that performance might be much more dramatically improved if we were able to use the full parser. The baseline performance for supertagging punctuation marks (i.e. assigning simply the most likely tag to each mark) is 65.9%. This is considerably lower than regular part-of-speech tagging at around 90% and supertagging overall at 77.2% for this corpus. The baseline for punctuation is lower because the average number of candidate supertags per token is higher: 6.5 supertags per punctuation mark compared with 1.5 parts-of-speech per word in standard part-of-speech tagging.

The difference in performance can be seen on example (3). When the comma preceding the lexical conjunction *and* is removed, the supertagger incorrectly assigns a relative clause tag to the verb *gave*. With the comma present, the verb correctly gets a main verb tag for *gave*.

(3) He left his last two jobs at Republic Airlines and Flying Tiger with combined stock-option gains of about $22 million
and UAL gave_N0nx0Vnx1nx2
, and UAL gave_nx0Vnx1nx2
him a $15 million bonus when he was hired .

4. Conclusions

Our aim in undertaking this research was to find out how feasible it was to handle a sizable core of punctuation phenomena at the level of the sentence grammar, without either adversely impacting the existing grammar or deriving analyses which would be incompatible with later levels of processing, in particular at the discourse level. Our results confirm that punctuation can be used in analyzing sentences to increase the coverage of the grammar, reduce the ambiguity of certain word sequences and facilitate discourse-level processing of the texts.[2] We have implemented quite an extensive grammar for punctuation which has been incorporated into the XTAG English grammar, and found that the punctuation rules do indeed improve the coverage of the existing grammar with no negative impact on the rest of the grammar.

[2] In (Doran, 1998), we also show that the LTAG analysis of the text adjunct variant is fully compatible with a discourse grammar of the sort proposed by Webber and Joshi (1998).

References

BRISCOE T. (1994). *Parsing (with) Punctuation etc.* Technical Report MLTT-TR-002, Rank Xerox Research Centre, Grenoble, France.

BRISCOE T. & CARROLL J. (1995). Developing and Evaluating a Probabilistic LR Parser of Part-of-Speech and Punctuation Labels. In *Proceedings of the Fourth International Workshop on Parsing Technologies (IWPT'95)*, p. 48–57, Prague/Karlovy Vary, Czech Republic.

CHAFE W. (1988). Punctuation and the prosody of written language. *Written Communication*, **5** (4 p.), 395–426.

DORAN C. (1996). Punctuation in Quoted Speech. In *Proceedings of the SIGPARSE96*, Santa Cruz, California.

DORAN C. (1998). *Incorporating Punctuation into the Sentence Grammar: A Lexicalized Tree Adjoining Grammar Perspective*. PhD thesis, University of Pennsylvania.

JONES B. (1996)a. Towards a Syntactic Account of Punctuation. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, Copenhagen, Denmark.

JONES B. (1996)b. *What's the Point? A (Computational) Theory of Punctuation*. PhD thesis, University of Edinburgh.

JOSHI A. K. (1985). Tree Adjoining Grammars: How much context Sensitivity is required to provide a reasonable structural description. In D. DOWTY, I. KARTTUNEN & A. ZWICKY, Eds., *Natural Language Parsing*, p. 206–250. Cambridge, U.K.: Cambridge University Press.

MEYER C. F. (1987). *A Linguistic Study of American Punctuation*, volume 5 of *American University Studies, Series XIII - Linguistics*. New York: Peter Lang.

NUNBERG G. (1990). *The Linguistics of Punctuation*. CSLI Lecture Notes No. 18. Stanford: Center for the Study of Language and Information.

QUIRK R., GREENBAUM S., LEECH G. & SVARTVIK J. (1985). *A Comprehensive Grammar of the English Language*. London: Longman.

SAMPSON G. (1995). *English for the Computer:The SUSANNE Corpus and Analytic Scheme*. Oxford: Clarendon Press.

SCHABES Y. (1990). *Mathematical and Computational Aspects of Lexicalized Grammars*. PhD thesis, Computer Science Department, University of Pennsylvania.

SCHMIDT M. (1995). *Acoustic Correlates of Encoded Prosody in Written Conversation*. PhD thesis, The University of Edinburgh.

SRINIVAS B. (1997). *Complexity of Lexical Descriptions and its Relevance to Partial Parsing*. PhD thesis, Department of Computer and Information Sciences, University of Pennsylvania.

VIJAY-SHANKER K. & JOSHI A. K. (1991). Unification Based Tree Adjoining Grammars. In J. WEDEKIND, Ed., *Unification-based Grammars*. Cambridge, Massachusetts: MIT Press.

WEBBER B. & JOSHI A. (1998). Anchoring a Lexicalized Tree-Adjoining Grammar for Discourse. In *Proceedings of the Workshop on Discourse Relations and Discourse Markers (at COLING98)*, p. 86–92, Montreal, Canada.