An Integrated Architecture for Generating Parenthetical Constructions

Eva Banik
Department of Computing
The Open University
Walton Hall, Milton Keynes,
e.banik@open.ac.uk

Abstract

The aim of this research is to provide a principled account of the generation of embedded constructions (called *parentheticals*) and to implement the results in a natural language generation system. Parenthetical constructions are frequently used in texts written in a good writing style and have an important role in text understanding. We propose a framework to model the rhetorical properties of parentheticals based on a corpus study and develop a unified natural language generation architecture which integrates syntax, semantics, rhetorical and document structure into a complex representation, which can be easily extended to handle parentheticals.

1 Introduction

Parentheticals are constructions that typically occur embedded in the middle of a clause. They are not part of the main predicate-argument structure of the sentence and are marked by special punctuation (e.g. parentheses, dashes, commas) in written texts, or by special intonation in speech.

Syntactically, parentheticals can be realized by many different constructions, e.g.: appositive relative clauses (1a), non-restrictive relative clauses (1b), participial clauses (1c) or subordinate clauses (1d).

(1) a The new goal of the Voting Rights Act [– more minorities in political office –] is laudable. (wsj1137)

b GE, [which vehemently denies the government's allegations,] denounced Mr. Greenfield's suit. (wsj0617)

c But most businesses in the Bay area, [including Silicon Valley,] weren't greatly affected. (wsj1930)

d So far, [instead of teaming up,] GE Capital staffers and Kidder investment bankers have bickered. (wsj0604)

A common characteristics of parentheticals is that they express information that is not central to the meaning of the overall message conveyed by a text or spoken utterance and since they are specifically marked by punctuation or intonation, they allow the reader to distinguish between more and less important parts of the message. By structuring information this way, parentheticals make it easier for readers to decode the message conveyed by a text. Consider for example the following message that has been expressed by two different texts: one without parentheticals (2a) and one that contains two parentheticals (2b).

(2) a Eprex is used by dialysis patients who are anaemic. Prepulsid is a gastro-intestinal drug. Eprex and Prepulsid did well overseas.

b Eprex, [used by dialysis patients who are anaemic,] and Prepulsid, [a gastro-intestinal drug,] did well overseas. (wsj1156)

Parentheticals have been much studied in linguistics (see (Dehe and Kavalova, 2007), (Burton-Roberts, 2005) for a recent overview) but so far they

Proceedings of the ACL-08: HLT Student Research Workshop (Companion Volume), pages 7–12,
Columbus, June 2008.

have received less attention in computational linguistics. Only a few studies have attempted a computational analysis of parentheticals, the most recent ones being (Bonami and Godard, 2007) who give an underspecified semantics account of evaluative adverbs in French and (Siddharthan, 2002) who develops a statistical tool for summarisation that separates parentheticals from the sentence they are embedded in. Both of these studies are limited in their scope as they focus on a very specific type of parentheticals.

From the perspective of natural language generation (NLG), as far as we know, nobody has attempted to give a principled account of parentheticals, even though these constructions contribute to the easy readability of generated texts, and therefore could significantly enhance the performance of NLG systems (Scott and Souza, 1990).

Most existing natural language generation systems use rhetorical structure to construct a text plan and map arguments of rhetorical relations onto individual sentences or clauses. As a result, the arguments of the same rhetorical relation will always occur immediately next to each other, although the surface realization of individual arguments may vary and a clause may appear *syntactically* embedded within the preceding clause. This linear succession of rhetorical relations and their arguments makes the generated text appear monotonous and staccato. As commonly mentioned by style manuals,[1] using different kinds of clause-combining strategies (e.g. semicolons, dash-interpolations, appositives) shows a clearer writing style.

The goal of this research is to give a principled account of parenthetical constructions and incorporate its findings into a natural language generation system.

2 System Architecture

We propose an integrated generation architecture for this purpose which uses a Tree Adjoining Grammar (Joshi, 1987) to represent linguistic information at all levels, including syntax, rhetorical structure and document structure.

Our approach is to make the elementary trees in the grammar as complex as possible, so that constraints on which trees can be combined with each other will be localized in the trees themselves. By incorporating information about rhetorical structure and document structure into the trees, we are extending the domain of locality of elementary trees as much as possible and this allows the generator to keep the global operations for combining trees as simple as possible. This approach has been referred to as the 'Complicate Locally, Simplify Globally' principle (Joshi, 2004).

The input to the generator is a set of rhetorical relations and semantic formulas. For each formula the system selects a set of trees from the grammar, resulting in a number of possible tree sets associated with the input.

The next step is to filter out sets of trees that will not lead to a possible realization. In the current implementation this is achieved by a version of polarity filtering where we associate not only the syntactic categories of root, substitution and foot nodes with a positive or negative value (Gardent and Kow, 2006) but also add the semantic variable associated with these nodes. The values summed up by polarity filtering are [node, semantic variable] pairs, which represent restrictions on possible syntactic realizations of semantic (or rhetorical) arguments.

Parentheticals often pose a problem for polarity filtering because in many cases there is a shared element between the parenthetical and its host, which normally occurs twice in non-parenthetical realizations of the same input, but only once when there is a parenthetical. (e.g., in (2a) the NP 'Eprex' occurs twice, but only once in (2b)). In order to allow for this variation, when summing up the values for substitution and root nodes we consider multiple occurrences of NP substitution nodes associated with the same semantic variable as if they were a single instance. This results in one or more NP substitution nodes left empty at the end of the derivation, which are then filled with a pronoun by a referring expression module at the final stage of the generation process.

3 Corpus Study

The generator is informed by a corpus study of embedded discourse units on two discourse annotated corpora: the RST Discourse Treebank (Carlson et al., 2001) and the Penn Discourse Treebank (PDTB-

[1] See for example, Rule 14 of (Strunk and White, 1979)

		Elab-add	Example	Elab-gen-spec	Restatement	Elab-set-mem	Attribution	Condition	Antithesis	Concession	Circumstance	Purpose	
NP-modifiers	relative clause	**143**		2		2							147
	participial clause	**96**	4			1	1				11	4	117
	NP	**34**		**8**	**22**								64
	NP-coord					6							6
	cue + NP	5	1						2	3	2		13
	Adj + cue	2											2
	number	2											2
	including + NP		**13**			5							18
VP- or S- modifiers	to-infinitive	4										**30**	34
	NP + V						**106**						106
	cue + S	5						**20**	**14**	**9**	**29**		77
	PP	11									9	1	21
	S	7	1	1									9
	according to NP						7						7
	V + NP						6						6
	as + S						4						4
	Adv + number	1									1		2
	cue + Adj										2		2
	cue + participial								2				2
	cue + V						1						1
		310	19	11	22	14	125	20	18	12	54	35	640

Table 1: Syntactic types of parentheticals in the RST corpus

Relation	Connective in parenthetical	Connective in host	distribution in corpus
TEMPORAL	101 (48.8%)	2	3434 (18.6%)
CONTINGENCY	53 (25.6%)	0	3286 (17.8%)
COMPARISON	38 (18.3%)	5	5490 (29.7%)
EXPANSION	15 (7.2%)	5	6239 (33.8%)
TOTAL:	207	12	18484

Table 2: Relations between parentheticals and their hosts in the PDTB

Group, 2008).[2] The aim of the study was to establish what rhetorical relations can hold between parentheticals and their hosts and whether individual rhetorical relations tend to correlate with specific syntactic types.

Table 1 illustrates the findings of the study on the RST corpus, showing the correlation between syntactic types of parentheticals and rhetorical relations between parentheticals and their hosts in the corpus. The majority of parentheticals in this study were syntactically related to their hosts and they can be divided into two main groups. The most frequently occurring type is ELABORATION/EXPANSION-type NP-modifiers which are realized by relative clauses, NPs or nominal postmodifiers with non-finite clauses and express some type of ELABORATION, EXAMPLE or RESTATEMENT relation. 73.4% of parentheticals belong to this group in the RST corpus.

The other type of parentheticals are NON-ELABORATION/EXPANSION-type VP- or S-modifiers, which are realized by subordinate clauses, to-infinitives and PPs and express CIRCUMSTANCE, PURPOSE, CONDITION, ANTITHESIS,or CONCESSION relations. 26.6% of parentheticals in the corpus belong to this group.

Because of the decision taken in the PDTB to only annotate clausal arguments of discourse connectives, parentheticals found in this corpus are almost

[2]The details of this study are reported in (Banik and Lee, 2008)

all subordinate clauses, which is clearly an artifact of the annotation guidelines. This corpus only annotates parentheticals that contain a discourse connective and we have found that in almost all cases the connective occurs within the parenthetical. We have found only 12 discourse adverbs that occurred in the host sentence.

The present corpus study is missing several types of parentheticals because of the nature of the annotation guidelines of the corpora used. For example, in the RST corpus some phrasal elements that contain a discourse connective (3a) and adjectives or reduced relative clauses that contain an adjective without a verbal element are not annotated (3b):

(3) a But the technology, [while reliable,] is far slower than the widely used hard drives. (wsj1971)

b Each $5000 bond carries one warrant, [exercisable from Nov. 28, 1989, through Oct. 26, 1994] to buy shares at an expected premium of 2 1/2 % to the closing share price when terms are fixed Oct. 26. (wsj1161)

These constructions are clear examples of parentheticals and we would expect them to behave similarly to subordinating conjunctions and relative clauses respectively. As a test case we decided to allow adjectives to function as parentheticals in the grammar of the generator and if the results are evaluated as satisfactory, plan to extend this analysis to other constructions not covered by our corpus study.

4 Generating Parentheticals — An Example

We associate auxiliary trees with parenthetical occurrences of the most frequently embedded rhetorical relations based on the above corpus study.

The basic assumption behind assigning syntactic trees to parenthetical rhetorical relations is that the semantic type of the arguments of the relation should be mirrored by their syntax. Thus if one of the arguments of a rhetorical relation is an object then it must be represented by an NP in the syntax; if it is a proposition then it must be assigned an S- or VP-auxiliary tree. The satellite of the rhetorical relation is always substituted into the auxiliary tree,

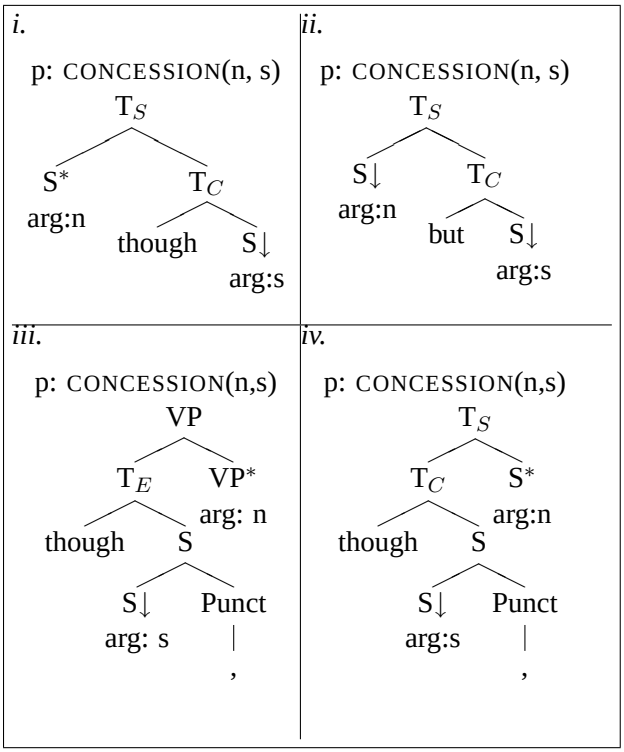

Figure 1: Elementary trees for CONCESSION

and the nucleus is associated with the footnode (this later gets unified with the semantic label of the tree that the auxiliary tree adjoins to).

Figure 1 illustrates four elementary trees for the CONCESSION relation. The trees in boxes *i.* and *ii.* correspond to regular uses of CONCESSION while the trees in *iii.* and *iv.* correspond to its parenthetical occurrences. Using these trees along with the elementary trees in Figure 3, and given the input below, the system generates the following five possible realizations:

```
   Input:  [[l3, concession, l1, l2], [l1,legal,x], [l2, fatal,
x], [x,substance]]

   Output:

1.  the substance, though it is fatal, is legal

2.  the substance is legal though it is fatal

3.  though it is fatal, the substance is legal

4.  though the substance is fatal, it is legal

5.  the substance is legal but it is fatal
```

Figure 2 gives the elementary trees assigned to

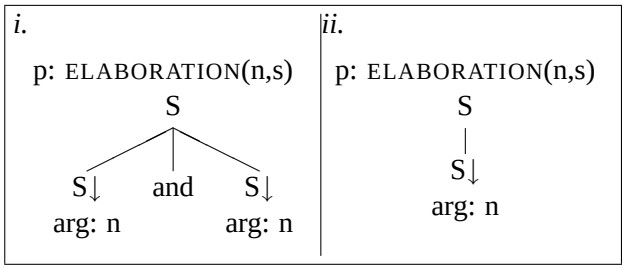

Figure 2: Elementary trees for ELABORATION

the most frequently occurring parenthetical rhetorical relation, ELABORATION-ADDITIONAL. The tree in box *i.* is associated with non-parenthetical uses of the relation, and box *ii.* shows the tree used for parenthetical ELABORATION. Since in parenthetical uses of ELABORATION the two arguments of the relation combine with each other and not with a third tree, as in the case of parenthetical CONCESSION, the role of the lexically empty parenthetical tree in box *ii.* is to restrict the type of tree selected for the nucleus of ELABORATION. Since the satellite has to end up as the parenthetical, the nucleus has to be restricted to the main clause, which is achieved by associating its semantic variable with an S substitution node in the tree.

To give an example, Figure 3. illustrates elementary trees for the input below:

```
   Input:  [[l3, elaboration, l1, l2], [l1,illegal,x], [l2,
fatal, x], [x,substance]]

   Output:

1.  the fatal substance is illegal

2.  the substance, which is fatal, is illegal

3.  the substance is illegal and it is fatal
```

The parenthetical ELABORATION tree is used for constructing outputs 1. and 2., which restricts the nucleus to select the initial tree in box *iii.* on Figure 3. As a result, the satellite of the relation has to select on of the auxiliary trees in box *i.* or *ii.* in order to be able to combine with the nucleus. The case where both satellite and nucleus are assigned initial trees is handled by the non-parenthetical tree in box *i.* on Figure 2.

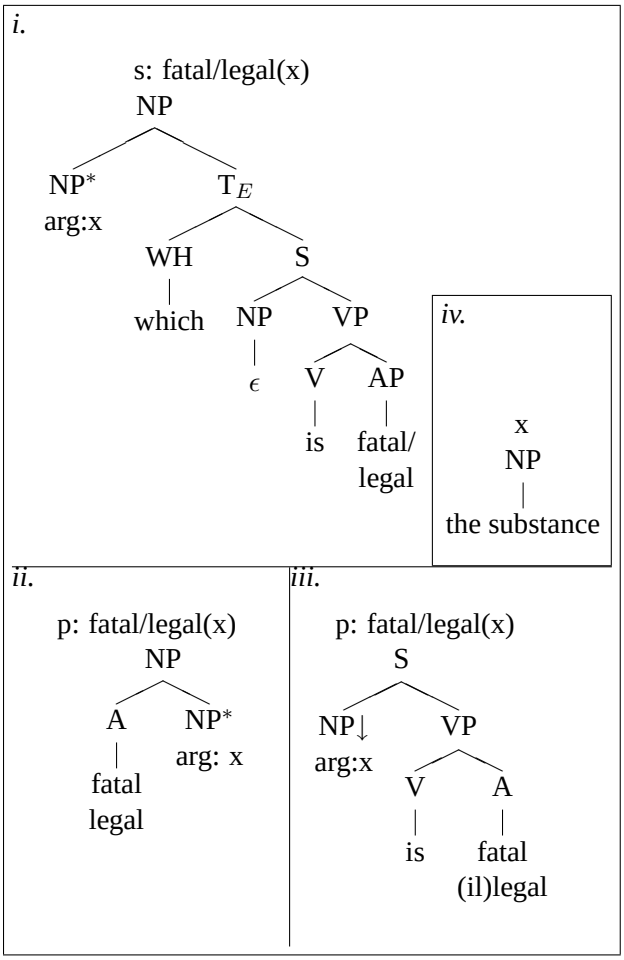

Figure 3: Elementary TAG trees for semantic formulas

5 Directions for further research

A possible way to control the generator is to enrich the input representation by adding restrictions on the types of trees that are allowed to be selected, similarly to (Gardent and Kow, 2007) (e.g., if a rhetorical relation is restricted to selecting initial trees for its satellite then it won't be generated as a parenthetical). Another way to select a single output is to establish ranking constraints (these could depend, e.g., on the genre of the text to be generated) and choose the top ranked candidate for output.

At the moment the elementary trees in the grammar contain document structure nodes (Power et al., 2003) which are not used by the generator. We plan to extend the analysis of parentheticals to big-

ger structures like footnotes or a paragraph separated in a box from the rest of the text and the document structure nodes in the elementary trees will be used to generate these.

Given the small size of the grammar, currently polarity filtering is enough to filter out just the grammatical realizations from the set of possible treesets. As the grammar size increases we expect that we will need additional constraints to reduce the number of possible tree sets selected for a given input. Also, once the generator will be capable of handling longer inputs, we will need to avoid generating too many parentheticals. Both the number of possible tree sets and the number of parentheticals in the outputs could be reduced by allowing the generator to select parenthetical realizations for only a predefined percentage of each rhetorical relation in the input. This number can be first obtained from our corpus study, and fine-tuned based on evaluations of the generated output.

The current implementation uses a very simplistic referring expression module which inserts a pronoun in every NP position left open at the end of the derivation, unless it is in a sentence initial position. Parentheticals often involve the use of referring expressions and can sound more natural when the embedded constituent involves a reference to an element in the main clause, therefore a more sophisticated algorithm for referring expression generation will be used in the future.

Although our corpus study gives important information about which rhetorical relation to realize as a parenthetical, how often, and using which syntactic construction, there seem to be additional restrictions on the use of certain parentheticals. Consider for example the two realizations (4 a and b) of the CONCESSION relation below where the parenthetical in (4b) sounds very unnatural:

```
   concession:
n:  a few people may experience side-effects
s:  most people benefit from taking Elixir
```

(4) a Though most people benefit from taking Elixir, a few people may experience side-effects.

b ?? A few people, though most people benefit from taking Elixir, may experience side-effects.

References

E. Banik and A. Lee. 2008. A study of parentheticals in discourse corpora – implications for NLG systems. In *Proceedings of LREC 2008, Marrakesh*.

O. Bonami and D. Godard. 2007. Parentheticals in underspecified semantics: The case of evaluative adverbs. *Research on Language and Computation*, 5(4):391–413.

N. Burton-Roberts. 2005. Parentheticals. In E. K. Brown, editor, *Encyclopaedia of Language and Linguistics*. Elsevier Science, 2nd edition edition.

L. Carlson, D. Marcu, and M. E. Okurowski. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, pages 1–10, Morristown, NJ, USA. Association for Computational Linguistics.

N. Dehe and Y. Kavalova, editors, 2007. *Parentheticals*, chapter Parentheticals: An introduction, pages 1–22. Linguistik aktuell Linguistics today 106. Amsterdam Philadelphia: John Benjamins.

C. Gardent and E. Kow. 2006. Three reasons to adopt tag-based surface realisation. In *The Eighth International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+8)*, Sydney/Australia.

C. Gardent and E. Kow. 2007. A symbolic approach to near-deterministic surface realisation using tree adjoining grammar. In *In 45th Annual Meeting of the ACL*.

A. K. Joshi. 1987. The relevance of tree adjoining grammar to generation. In G. Kempen, editor, *Natural Language Generation*, pages 233–252. Martinus Nijhoff Press, Dordrect, The Netherlands.

A. K. Joshi. 2004. Starting with complex primitives pays off: complicate locally, simplify globally. *Cognitive Science: A Multidisciplinary Journal*, 28(5):637–668.

PDTB-Group. 2008. The Penn Discourse Treebank 2.0 Annotation Manual. Technical Report IRCS-08-01, Institute for Research in Cognitive Science, University of Pennsylvania.

R. Power, D. Scott, and N. Bouayad-Agha. 2003. Document structure. *Computational Linguistics*, 29(4):211–260.

D. Scott and C. S. Souza. 1990. Getting the message across in RST-based text generation. In C. Mellish R. Dale M. Zock, editor, *Current Research in Natural Language Generation*, pages 31–56. Academic Press.

A. Siddharthan. 2002. Resolving attachment and clause boundary ambiguities for simplifying relative clause constructs. In *Student Research Workshop, ACL*.

W. Jr. Strunk and E. B. White. 1979. *The Elements of Style*. Macmillan, third edition.