ISSN 1744-1986

The Open University

***Technical Report N° 2007/14***

# "Generating Parenthetical Constructions"

***Eva Banik***

*6th November 2007*

---

***Department of Computing***
**Faculty of Mathematics and Computing**
**The Open University**
**Walton Hall,**
**Milton Keynes**
**MK7 6AA**
**United Kingdom**

***http://computing.open.ac.uk***

**Abstract**

This paper is a research proposal for a dissertation in Computational Linguistics, in particular Natural Language Generation. The purpose of the 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 (they help the reader to differentiate between more and less important information). They have been much studied in the linguistics literature but have received no attention so far in computational linguistics. While the ability to signal the relative importance of various items in a sentence is clearly an important contributor to the effectiveness of the text, existing natural language generation systems currently do not have a principled way of handling parentheticals.

The aim of the research proposed here is to create a framework to model the rhetorical properties of different types of parentheticals and the contexts that license their usage. We will develop a unified natural language generation architecture which integrates syntax, semantics, rhetorical structure and document structure into a complex representation in order to give a principled account of constraints on where and when parenthetical constructions are appropriate to generate. The system uses constraint based reasoning to reduce computational complexity and rank the output texts.

The expected results of this research will enable NLG systems to generate stylistically better output and give developers more control over the generation process and the user's interpretation of the generated text.

# Contents

# 1 Introduction

Natural language generation can be defined as the automatic production of natural language output from a form of non-linguistic representation. There are many reasons for building such systems, for example, generating reports, paraphrases and summaries; providing natural language conversational partners in dialogues to support applications such as intelligent help, intelligent tutorial systems, and database access systems; producing translations of texts in another language. The purpose of such systems is to convey information that would otherwise not be intelligible for the user (e.g., information captured in a database or a text written in another language) and thereby to achieve a specific communicative goal. The most important criterion for judging the text produced by a natural language generation (NLG) system goes beyond questions of whether it conveys the information captured in the database or foreign language text: *how efficiently the information is conveyed and the communicative goal achieved* (Scott and Souza, 1990). Most current NLG systems can certainly satisfy the first criterion, conveying the information represented in their input, but in terms of effectiveness they are far from the efficiency of human writers.

The most common problem with the output of natural language generation systems is that they tend to produce a succession of short, simple clauses connected by conjunctions. Texts like these appear monotonous and staccato. Avoiding such sequences of simple sentences is a fundamental principle of good writing, commonly mentioned in style manuals. For example, Rule 14 of "Strunk and White", one of the most influential and best-known prescriptive treatments of English grammar and usage (Strunk, 1918; Strunk and White, 1979), formalizes this principle as follows:

**14. Avoid a succession of loose sentences.**

This rule refers especially to loose sentences of a particular type, those consisting of two co-ordinate clauses, the second introduced by a conjunction or relative. Although single sentences of this type may be unexceptionable (see under Rule 4), a series soon becomes monotonous and tedious.
An unskilful writer will sometimes construct a whole paragraph of sentences of this kind, using as connectives and, but, and less frequently, who, which, when, where, and while, these last in non-restrictive senses (see under Rule 3).

> The third concert of the subscription series was given last evening, and a large audience was in attendance. Mr. Edward Appleton was the soloist, and the Boston Symphony Orchestra furnished the instrumental music. The former showed himself to be an artist of the first rank, while the latter proved itself fully deserving of its high reputation. The interest aroused by the series has been very gratifying to the Committee, and it is planned to give a similar series annually hereafter. The fourth concert will be given on Tuesday, May 10, when an equally attractive programme will be presented.

Apart from its triteness and emptiness, the paragraph above is bad because of the structure of its sentences, with their mechanical symmetry and singsong. [...]
If the writer finds that he has written a series of sentences of the type described, he should recast enough of them to remove the monotony, replacing them by simple sentences, by sentences of two clauses joined by a semicolon, by periodic sentences of two clauses, by sentences, loose or periodic, of three clauses–whichever best represent the real relations of the thought.

Strunk (1918), p.25

If Professor Strunk could see the output of natural language generation systems today, he would no doubt call them "unskillful writers" – as they all seem to violate his Rule 14. Consider a few recent examples:

```
the dose of the patient's medicine is taken twice a day.  it is
two grams.  if the overdose of the patient's medicine is taken
consult the doctor.

the patient takes the two-gram dose of the patient's medicine
twice a day.  if the patient takes the overdose of the patient's
medicine the patient should consult the doctor.

the two-gram dose of the patient's medicine is taken twice a day.
if the overdose of the patient's medicine is taken the patient
should ask the doctor for the advice.
```

Paiva (2004)

```
To access call functions
You have different possibilities:

    * Make a call.
      How?

    * Answer a call.
      Note:
      An incoming call will automatically be
      transferred to your  message box if you
      do not answer it.

    * Adjust the earpiece volume.
      Warning:
      If the volume is too high, others might
      be able to hear what the caller says.
      In addition, excessively high volumes may
      damage your ears.

    * Redial the last number.
      Note:
      Your phone will store the 5 previously
      dialed numbers.

    * Access in-call functions.
```

ISOLDE, Paris et al. (2003)

```
To schedule the appointment:
Before starting, open the Appointment Editor window by choosing
the Appointment option from the Edit menu.
Then proceed as follows:
1 Choose the start time of the appointment.
2 Enter the description of the appointment in the What field.
3 Click on the Insert button.
```

Drafter-2 (Power et al., 1998; Power and Scott, 1998)

```
 -- Behrens's principal activities were architecture and
industrial design.  -- He made electrical appliances and
prototype flasks.  -- He built the high tension plant and the
turbine factory for AEG in 1908-1910.  -- He built a housing
for the workers of AEG in Henningsdorf.  -- He created a number
of monumental buildings, such as the administration building
of Mannesmann in Duesseldorf and the German embassy in St.
Petersburg.
```

KOMET (Bateman and Teich, 1995; Teich and Bateman, 1994)

Strunk and White (1979) contrast the paragraph in Rule 14 (see the above quote) with the following example of good writing, an excerpt from "What I Believe" in *Two Cheers for Democracy*, by E. M. Forster (my emphasis):

> I believe in aristocracy, though – *if that is the right word, and if a democrat may use it.* Not an aristocracy of power, *based upon rank and influence*, but an aristocracy of the sensitive, the considerate and the plucky. Its members are to be found in all nations and classes, *and all through the ages*, and there is a secret understanding between them when they meet. They represent the true human tradition, *the one permanent victory of our queer race over cruelty and chaos.* Thousands of them perish in obscurity, a few are great names. They are sensitive for others as well as for themselves, they are considerate without being fussy, their pluck is not swankiness but the power to endure, and they can take a joke.

One of the key differences between the examples of 'unskillful' writing that we have just seen and the example of 'skillful' writing shown in the paragraph above is the variety of ways the sentences are combined in the latter. In addition to simple conjunctions (*and* and *but*) and sequences of sentences,

we can also find embeddings or parentheticals (which I have emphasised here with italics). Clearly one of the characteristics of good writing is the frequent occurrence of parenthetical constructions.

In addition to contributing to a clear writing style, parenthetical constructions have another important function: they make the comprehension of the generated text easier for the reader by explicitly signalling the relative importance (or relevance) of the component propositions to the overall 'message' being conveyed, by making some parts of the propositional content more 'salient' than others.

According to the prevailing theories of text comprehension, reading of a text involves processing at three different cognitive levels (Snowling and Hulme, 2005; Kintsch, 1998): first, the reader must decode the meaning of the words on the page and assign them to their roles in the individual sentences or phrases. Next, word meanings must be combined to form idea units or propositions and the reader must determine the coherence relations between these propositions. The reader must also organize propositions into higher order units and recognize the global topics of the text and their interrelationships[1]. Finally, readers must integrate the information conveyed by the text with their mental model of a relevant situation and their prior knowledge.

This model of reading comprehension originates from Thorndike (1917) who considers reading comprehension as a form of reasoning. According to Thorndike, "[u]nderstanding a paragraph is like solving a problem in mathematics. It consists in selecting the right elements of the situation and putting them together in the right relations, and also with the right amount of weight or influence or force for each. The mind is assailed as it were by every word in the paragraph. It must select, repress, soften, emphasize, correlate and organize, all under the influence of the right mental set or purpose or demand." (Thorndike (1917), p.431)

Studies have shown that the explicit signalling of the topic structure of a text (e.g., by the use of headings, titles, overviews, etc.) contributes to its reading comprehension and recall by selecting and emphasising the salient points (Lorch et al., 1993).

Parenthetical constructions play a similar role in text understanding by 'repressing' and 'softening' ideas rather than emphasizing them. Whereas headings and overviews help the readers distinguish important information by explicitly signalling them, parenthetical constructions contribute to the

[1] This higher order structure is called *macrostructure* and is frequently modelled with rhetorical schemata such as Rhetorical Structure Theory (see section 2.2 for an explanation)

same goal by explicitly signalling what is *not* (so) important and therefore should not be considered as the main topic of the text.

Consider for example the italicized non-restrictive relative clause[2], a frequently occurring example of parenthetical constructions, in (1):

(1) Kofi Annan, *who is the current U.N. Secretary General*, has spent much of his tenure working to promote peace in the Third World.

The main point that sentence (1) is trying to get across to the reader is that Kofi Annan has spent a lot of time promoting peace in the Third World. The sentence also mentions that Kofi Annan is the current U.N. Secretary General but this point is clearly not as important (or salient) as the meaning conveyed by the main clause. If instead the writer had intended the fact of Kofi Annan's position within the hierarchy of the UN to be the most important point in the sentence, this would be missed by the reader. To achieve this other reading, the writer/speaker would have to construct a sentence like (2):

(2) Kofi Annan, *who has spent much of his tenure working to promote peace in the Third World*, is the current U.N. Secretary General.

From a natural language generation point of view it is very important therefore to make the right choice when generating embedded constructions in order to ensure that the rhetorical properties of the input are preserved in the output text.

Consider, for example, the following semantic propositions as a possible input to a natural language generation system, corresponding to the simple sentences in (4):

(3) i age(Maddy, 4)
ii kidnapped(Maddy, 5_months_ago),
iii nationality(Maddy, English)

(4) Maddy is English. Maddy is 4 years old. Maddy was kidnapped five months ago.

When the choice of parenthetical constructions is available to the generator there are several possible outputs to choose from, all expressing the above semantic propositions. Some of these are illustrated in (5).

[2]In all the example sentences in this paper I use italics to show parenthetical constructions.

(5) a Maddy, an English 4-year-old, was kidnapped five months ago.
b Maddy - that four year old kidnapped five months ago - is English.
c Maddy (an English 4-year-old) was kidnapped five months ago.
d Maddy - that English four-year-old - was kidnapped five months ago.
e Maddy - who is four years old - is English and she was kidnapped five months ago.
f Maddy - who is English - is four years old and she was kidnapped five months ago.
g Maddy, who is English and 4 years old, was kidnapped five months ago.
h An English 4-year-old (Maddy) was kidnapped five months ago.
i An English 4-year-old, Maddy, was kidnapped five months ago.
j Maddy, who was kidnapped five months ago, is English and 4 years old.

The important point to be made here is that although all the sentence versions shown in examples (4) and (5) express the same propositional semantics, they are not equivalent with respect to their rhetorical properties. In (5a) the main point of the sentence is that Maddy was kidnapped five months ago. A sentence like this would be an appropriate headline, for example, for a tabloid article about Maddy's disappearance. The same holds for sentences (5c,d,f,g), where proposition (3ii) is realized as a main clause. (5b) on the other hand would sound rather odd as a headline about Maddy's disappearance, because – even though it conveys the same propositional content – it suggests that the fact that Maddy is English is more relevant than the fact that she was kidnapped.

While the ability to signal the relative importance of various items of propositional content in a sentence is clearly a very important contributor to the effectiveness of the text, current NLG systems are not able to provide a proper treatment of this phenomenon (see examples of system output shown earlier in this section). The aim of the proposed research is to give a principled account of when and how to generate parenthetical constructions. More specifically, the proposed body of work aims to create a framework to model the rhetorical properties of different types of parentheticals and the contexts that license their usage. The expected results of this research will enable NLG systems to generate stylistically better output and give developers more control over the generation process and the user's interpretation of the generated text.

# 2 Background

This section gives a survey of the relevant research context for my proposed PhD project. First the basic concepts for my research are defined: section 2.1 introduces parenthetical constructions and outlines the main research questions about parentheticals in the linguistic literature. Sections 2.2 and 2.3 introduce basic concepts of Rhetorical Structure Theory and abstract document structure. Section 2.4 reviews the application of Tree Adjoining Grammars to various linguistic phenomena and section 2.5 gives an overview of some of the related research in Natural Language Generation.

## 2.1 Parenthetical Constructions

Parenthetical constructions are strings of words that are embedded within a host sentence but seem structurally unrelated to it at the same time.

Dehe and Kavalova (2007) introduce parentheticals as constituents with many different defining properties: they can interrupt the prosodic flow of the host utterance, they are outside of the focus-background structure of the host, typically function as modifiers, additions or comments and they often convey the speaker's attitude towards the host utterance or provide background information.

### 2.1.1 The Linguistics of Parentheticals

In recent years there has been a renewed interest in parenthetical constructions in the theoretical linguistic community. Researchers have been trying to explain the relation between host utterance and the parenthetical and have been investigating the question of what level of representation such a relation might be best integrated into. Should parentheticals be represented at the level of syntax, are they licensed by pragmatics, discourse relations or conversational implicature or are they merely a prosodic or performance phenomena?

There are many different types of parenthetical constructions. On the one hand there are clearly non-syntactic examples like (6):

(6) a The main point – *why not have a seat?* – is outlined in the middle paragraph. (Burton-Roberts, 2005)

b What Iraq needs is education. We do not need to begin with the children – *they will follow* – but with the adults. (Blakemore, 2006)

Parentheticals of this kind illustrate most clearly a problem for syntactic theory where linear order is seen as the function of hierarchical structure and the order of words is determined by (and within) constituent domains. According to this commonly held basic tenet of syntactic theory (and fully developed in Kayne (1994)'s Linear Correspondence Axiom), when an expression P is linearly embedded within another expression H, the same expression H should dominate the embedded string P in the hierarchical syntactic structure (i.e., P should be a constituent of H). In the above examples the parenthetical is linearly contained in the host utterance on the one hand (and it's position is quite strictly limited) , but on the other hand there is clearly no syntactic relation between the two expressions.

The examples in (6) could plausibly be accounted for in the domains of prosody or linguistic performance. There are however many more examples of parentheticals that seem to be structurally integrated into the host clause at first sight, but on a closer look they are clearly not constituents of the sentence. Vocatives and appositive relative clauses are two prime examples.

The non-constituent status of **vocatives** (like the italicised element in (7))

(7) If Mary had tutored him, *John*, Bill would have passed.

is supported by the fact that they do not seem to be able to participate in VP-ellipsis (McCawley, 1982):

(8) A: Didn't you claim, John, that Bill would pass?
B: I didn't.
= I didn't claim that Bill would pass.
≠ *I didn't claim, John, that Bill would pass.

**Appositives** are problematic for syntactic analysis because they are not proper relative clauses (they are not introduced by relative pronouns – 'who' or 'which' – which is obligatory in restrictive clauses). They can't be taken to be coordinated with the subject either because coordination can only be successfully applied to non-coreferential elements. Therefore it is not clear what syntactic function the appositive elements should play in the clause[3].

(9) a The whole family – *John, Mary and the kids* – just disappeared.
b They disposed of – *fired or killed* – everyone they thought obstructive.
c It was dawn, *about quarter to six*, when they arrived.

[3]For a full description of appositives see Burton-Roberts (1975) and Burton-Roberts (1999a)

Another argument for the claim that appositives are syntactically unrelated to their host comes from the contrast in the possibilities for the interpretation of pronouns in (10), first noticed by Haegeman (1988):

(10) a John$_i$ always works better while his$_i$ /*John$_i$'s children are asleep.
b John studies mathematics, while his/John's wife studies physics.

Referential terms (here *John*) cannot be bound by a c-commanding NP in argument position, therefore in the while clause in (10a) 'his' cannot be replaced by 'John'. In (10b) however this is allowed, because the parenthetical while-clause is not syntactically related to the host and therefore the host subject does not c-command the pronoun in the parenthetical.

A similar pattern can be seen with the discourse adverbial 'therefore' in (11):

(11) a John works best for private firms who (*therefore) employ him often.
b John works best for private firms, who therefore employ him often.

Blakemore (1987) points out that 'therefore' establishes a discourse connection between independent clauses and so it cannot be used to link two clauses where one clause is the constituent of the other. This explains why the discourse adverb is not allowed in (11a). The acceptability of (11b) again suggests that the parenthetical is not a syntactic constituent of the host clause.

The above phenomena have been used as arguments for a '**radical orphanage approach**' an analysis where parentheticals are not generated by the grammar as part of any syntactic structure (Haegeman, 1988; Burton-Roberts, 1999b) but are accounted for at the level of pragmatics.

There are however other types of parentheticals which clearly have a structural relation to their host sentences, such as non-restrictive relative clauses (12a), parenthetical adverbial clauses (12b) and sentence adverbials ((12c):

(12) a Penn, *who last week received an Oscar for his role in Clint Eastwood's Mystic River*, may also have thought of Eastwood's previous picture, Bloodwork ...
b My idea, *if you really want to know*, was to treat the phenomenon as a conventional implicature.
c The students were, *unfortunately*, on holiday.

These kinds of parenthetical constructions have led to analyses where parentheticals are accounted for at the level of syntax, typically by extending the boundaries of phrase structure in some way. Thus Sag (1997) develops a framework in HPSG that allows multi-dimensional phrasal types and imposes constraints on linear ordering; Emonds (1979) introduces an extra syntactic node E which dominates the host CP and to which the parenthetical is adjoined; and Espinal (1991) argues for a three-dimensional phrase structure which allows a constituent to be part of another constituent but at the same time allows them to be independent and only intersect on the level of terminal symbols.

Whether or not parentheticals are related to the host utterance on the level of syntax, they clearly must have pragmatic functions that lead speakers to use these constructions even though they introduce processing difficulties for the hearer. In other words the benefit of conveying some additional meaning by using a parenthetical seems to justify the increase in the effort required on the hearer's part to process the utterance. So what exactly is this 'additional meaning' conveyed by the parenthetical? Do different types of parentheticals have different pragmatic implications? And are these implications dependent on the position of the parenthetical within the host sentence?

Although a coherent linguistic theory which answers all these questions hasn't emerged yet, some of the questions have been addressed by many researchers in different frameworks, more recently e.g., by Ziv (2002) and Kavalova (2007).

Ziv (2002) considers the function of parentheticals in a specific location: clause second position. He claims that parenthetical constructions in this position have the discourse function to link the immediately preceding element (called a 'marked theme') to the previous discourse context. This, according to Ziv, explains why parenthetical constructions are not allowed after elements that cannot function as a marked theme, like pleonastic elements (13a,b) or 'prototypical scene-setting elements' (13c,d)[4]:

(13) a * It, *I believe*, is important to generate parenthetical constructions.
b * There, *I assume*, are a million theories about parentheticals.
c *? Once upon a time, *I believe*, there was a king. He lived in Africa.

[4]Ziv also points out an interesting parallel between parentheticals in this context and discourse adverbs like 'however': neither of them can occur after pleonastic elements.

i * It, however, will not rain.

d *? One bright evening, *I believe*, John decided to go for a walk.

Kavalova (2007) describes the properties of *and*-parentheticals in detail. She establishes the following possible places of interpolation for *and*-parentheticals:

- between a syntactic head (N, V, Adj) and its phrasal or clausal dependent:

(14) In fact I [VP was very candidly [V told] *and I repeat my acknowledgement of the candour* [OD/CL that it was placed before him in January last]]

- between copula and predicate:

(15) Well our question [ V was] *and I've asked this before* [CS/CL is this necessary for a tragedy]

- at the edge of a core clause (between an adverbial clause and the main clause):

A/CL Because on this theory *and it's very deeply held* uh [CL good educational news is by definition inadmissible as evidence]

- between core clausal constituents (e.g., subject and VP):

SU Mr Heath's government *and I'm not complaining because I'd advocated this at a previous time* [ introduced the threshold system]...

- and between an auxiliary and the main verb:

(16) He [VP [Aux was] *and I think you would agree with me* at the outset [V looking]] at expanding his business at that time not selling it

Kavalova establishes two subtypes of *and*-parentheticals based on how restricted their placement is within the clause: floating parentheticals and anchored parentheticals.

**Anchored parentheticals** occur to the immediate right of an element in the host clause (called the anchor and underlined in the examples in (17)) to which they are directly linked and to which they exhibit a proximity loyalty. Therefore if the parenthetical is placed at a distance from the anchor, the sentence becomes less felicitous or even ungrammatical, as illustrated in (17b) and (17c):

(17) a Uh uh whilst I would nationally fund the system *and it's ninety percent funded from the Exchequer now* we must recognise that I would retain the office of Chief Constable

b # Whilst I would nationally fund the system , we, *and it's ninety percent funded from the Exchequer now*, must recognise that I would retain the office of Chief Constable.

c #* Whilst I would nationally fund the system , we must recognise that I would retain the office of Chief Constable, *and it's ninety percent funded from the Exchequer now.*

**Floating parentheticals** are generally independent and are not related to a particular element in the host. They can appear in multiple positions within the sentence, for example the possible positions for the italicised parenthetical in (18a) are the positions marked by the '@' sign in (18b) (Kavalova, 2007):

(18) a I personally take the view *and I've informed the Soviet Government of this* that that visit of the Ballet would be more acceptable to all of our people including myself

b I personally take the view @ that @ that visit @ of the Ballet @ would be @ more acceptable @ to all of our people @ including myself.

In a footnote Kavalova does note however that different positions within the sentence might give rise to different pragmatic implications. For example when the parenthetical appears in the position in (19) the scope of the parenthetical seems to be the expression 'more acceptable', rather than the entire proposition (that the visit of the Ballet would be more acceptable to all) as in (18a).

(19) I personally take the view that that visit of the Ballet would be, *and I have informed the Soviet Government of this*, more acceptable to all of our people.

### 2.1.2 Limiting the scope of this work

Fascinating as the above illustrated research questions about parentheticals may be, a thorough theoretical linguistic analysis is not the aim of this dissertation. Our aim in looking at existing linguistic analyses at this point is to better delimit the scope of this work and to provide a starting point for corpus study and grammar development.

For example, types of parentheticals that this dissertation is NOT aiming to consider include the following:

- Disfluency and performance phenomena, backtracking, or any other types of parentheticals that are only manifested in spoken utterances

(20) a But a different role <,> uh because when we get to the time of uh Ezra *as with the more classical Wellhausen uh hypothesis* when we get to the time of Ezra we have the further narrowing of the office of priest

b suppose you write the trend words as you check out the mistakes you might *God I hope not* but you might embody a whole new theory of syntax (Dehe and Kavalova, 2007)

- One-word parentheticals[5]

(21) a ... because it's surely quite difficult these days to persuade an actor or actress to commit themselves for *what* six to eight months when there is always the possibility isn't there of lucrative television work.

b I'd be far more upset if somebody *say* scratched one of my records than tore one of my books

c I got quite good at *like* heating up thermometers and stuff and give myself a temperature and things (Dehe and Kavalova, 2007)

- Question tags

(22) a They're called Gasser the people next door *are they*

b He suffered great mental distress *didn't he* after the war (Dehe and Kavalova, 2007)

## 2.2 Rhetorical Structure Theory

Rhetorical structure theory (RST) was developed mainly by William Mann and Sandra Thompson (at the Information Sciences Institute, Marina Del Rey, California) as a basis for text generation, more specifically as a framework for planning various large texts. The theory has been defined and described in several papers, for example in Mann and Thompson (1986), Mann and Thompson (1987), Mann and Thompson (1988) and has been widely adopted in the Natural Language Generation community.

[5] For a thorough analysis of parenthetical *what* see Dehe and Kavalova (2006)

Rhetorical Structure Theory has been used in many Natural Language Generation systems: PAULINE (as described in e.g., Hovy (1988), Hovy (1990)), PENMAN (Hovy, 1993), ILEX (Mellish et al., 1998; Hitzeman et al., 1997), GIRL (Williams, 2004), ICONOCLAST (N. Bouayad-Agha, 2000; Power et al., 2003; R. Power, 2000) just to name a few.

The basic concepts in RST are *rhetorical relations*. The coherence of texts is attributed to the presence of these relations which represent the relationship between two text spans. The rhetorical relations express the effect the Writer intends to achieve on the Reader by presenting the two spans of text side by side.

Another central notion of RST is the concept of *nuclearity*: the idea that one text span of a rhetorical relation is more central to the writer's purposes than the other. The more central span is called the **nucleus** and the less central one the **satellite**. The majority of the rhetorical relations defined by Mann and Thompson have a nucleus and a satellite, the exceptions are two multinuclear relations: (*sequence* and *contrast*).

The definition of rhetorical relations consists of four fields:

1. Constraints on the Nucleus
2. Constraints on the Satellite
3. Constraints on the combination of Nucleus and Satellite
4. The Effect of the relation on the Reader

Mann and Thompson define a set of 23 rhetorical relations, however the authors note that "[t]he set of relations is not closed – depending on one's purposes and the kind of text in view, relations can be added, subdivided and otherwise manipulated." (Mann and Thompson (1986) pp 7.). Indeed, most natural language generation systems make use of only a subset of these rhetorical relations, in some cases extended by relations particular to the system's purpose.

In this dissertation proposal I am going to use three of the 23 rhetorical relations: Evidence, Concession and Elaboration. Of the three, only Elaboration is central to my research. As Scott and Souza (1990) note, embedding is the only available textual marker for the elaboration relation, an therefore it should only be used to mark elaboration in generated text (see 2.5.1 for a summary of the relevant hypotheses in Scott and Souza (1990)).

The rhetorical relations mentioned in this document are defined by Mann and Thompson as follows:

**relation name:** CONCESSION

**constraints on N:** Writer has positive regard for N

**constraints on S:** Writer is not claiming that S does not hold;

**constraints on the N+S combination:** Writer acknowledges a potential or apparent incompatibility between N and S; recognising the compatibility between N and S increases Reader's positive regard for N

**effect:** Reader's positive regard for N is increased

**relation name:** EVIDENCE

**constraints on N:** Reader might not believe N to a degree satisfactory to Writer

**constraints on S:** Reader believes S or will find it credible

**constraints on the N+S combination:** Reader's comprehending S increases Reader's belief of N

**effect:** Reader's belief of N is increased.

**relation name:** ELABORATION

**constraints on N:** none

**constraints on S:** none

**constraints on the N+S combination:** S presents additional detail about the situation or some element of subject matter which is presented in N or inferentially accessible in N in one or more of the ways listed below. In the list, if N presents the first member of any pair, then S includes the second:

1. set : member
2. abstract : instance
3. whole : part
4. process : step
5. object : attribute
6. generalisation : specific

The above definition of elaboration allows for the possibility that instead of one relation of elaboration, one might want to propose each of the six subtypes as a distinct relation. We have chosen not to do this, but to regard them as subtypes. (Mann (1988) footnote 18, pp 52)

**the effect:** Reader recognises the situation presented in S as providing additional detail for N. Reader identifies the element of subject matter for which detail is provided.

## 2.3 Abstract Document Structure

Most NLG systems use Rhetorical Structure Theory (RST, Mann and Thompson, 1988) to represent discourse structure and map directly from discourse structure to formatted text. As in RST rhetorical relations hold between spans of text, in this framework rhetorical relations are treated as part of document structure.

Power et al. (2003) argue that this approach can be problematic. One reason is that it makes more sense to take rhetorical relations to hold between ideas (as pointed out by Scott and Souza, 1990), because the same argument can be realized in several different ways, and because the underlying rhetorical structure is not always isomorphic to document structure. Also, previous research (e.g., Moore and Pollack, 1992) has shown that an "orthodox" RST analysis of texts can often be problematic.

Power et al. (2003) claim that the principles of RST in these cases are violated because the theory is applied to the surface text instead of the underlying propositional structure and argue for a new descriptive level for the analysis and generation of written text, called abstract document structure.

The definition of abstract document structure builds on ideas from markup languages and Nunberg's text grammar (Nunberg, 1990). It describes the ways abstract document units can combine to form a document. The function of rhetorical structure on the other hand is to describe the argument or meaning of the text. From this perspective a level of document structure can be thought of as parallel to sentence-level syntax and rhetorical structure as parallel to sentence-level semantics.

The task of document structuring then is to create a document structure that satisfactorily realizes a given rhetorical relation.

Power et al. (2003) define a document structure unit of a given level as a combination of one or more units of the next level down. However, as "indented" document units may introduce apparent violations of this hierarchy they use two features when defining the compositional rule: document level and indentation.

$[L_N, I_M] \rightarrow [L_{N-1}, I_M]^+$
$[L_A, I_M] \rightarrow [L_B, I_{M+1}]^+$

## 2.4 Tree Adjoining Grammars

Lexicalized Tree Adjoining Grammars are very well suited as a theoretical framework for a unified generation system. Not only have previous researchers investigated the application of TAGs to various linguistic phenomena in syntax (Kroch and Joshi, 1985), semantics (Kallmeyer and Joshi, 2003, 1999) and discourse (Webber et al., 1999, 2003; Webber, 2004) but it has also been shown that these linguistic results can be integrated into an NLG system to provide an efficient way to represent the syntax, semantics and pragmatics modules (Stone and Doran, 1997).

This section gives a brief overview of the Tree Adjoining Grammar formalism and its various applications to linguistic phenomena. First the basic concepts are introduced, then we survey the application of TAG to natural language syntax (section 2.4.2), semantics (section 2.4.3) and discourse (section 2.4.4).

### 2.4.1 Lexicalized Tree Adjoining Grammar

Tree Adjoining Grammars (TAG) have been first introduced by Joshi et al. (1975). Good introductions to the formalism can be found in Joshi and Schabes (1997), Joshi (1983), or Abeille and Rambow (2000).

A TAG consists of a finite set of *elementary trees*, each associated with a lexical item. The elementary trees specify local configurations of the lexical item they are associated with (the lexical *anchor*).

Elementary trees can be combined using two operations: substitution and adjunction. Figures 1 and 2 illustrate the results of these operations.

We call a Tree Adjoining Grammar lexicalized if all words in the language are associated with a finite set of elementary trees and if all elementary trees are associated with a lexical item. The trees in an LTAG thus have at least one terminal node on their frontier called the *anchor* of the tree.

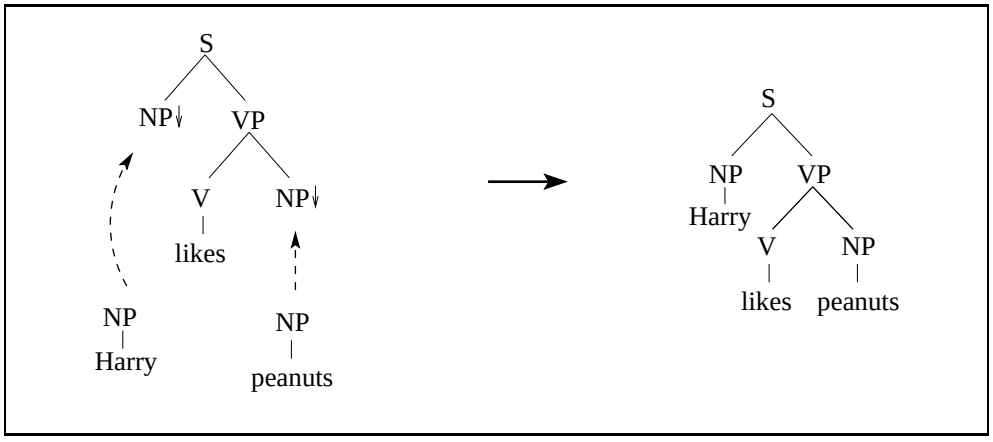



Figure 1: Substitution

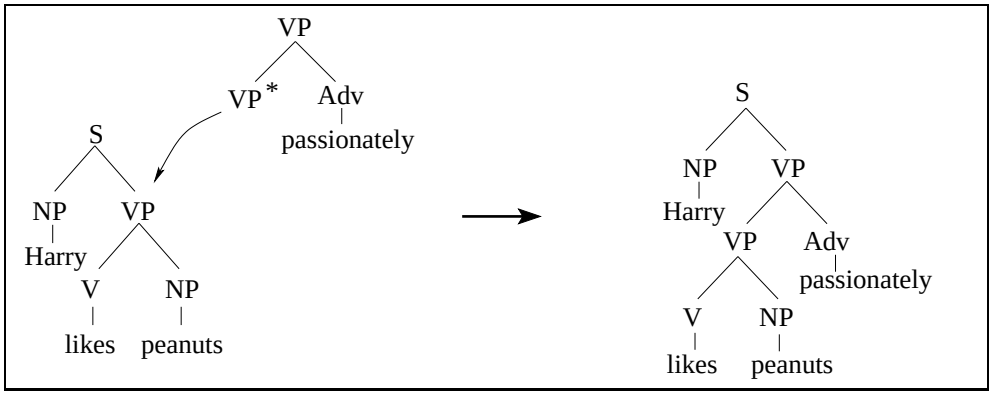



Figure 2: Adjoining

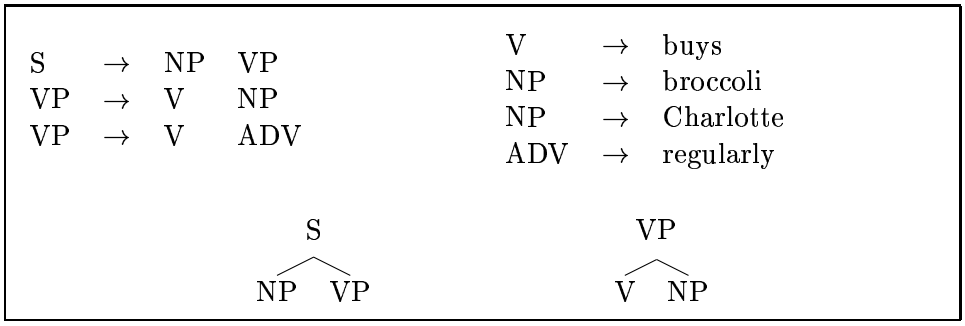



Figure 3: Domain of Locality of CFG

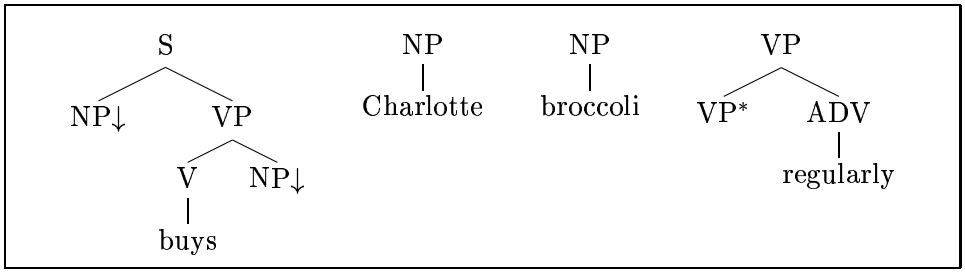



Figure 4: Domain of Locality of TAG

**Extended Domain of Locality**

Every grammar formalism has a domain of locality, which is defined as the domain over which grammatical dependencies can be specified.

For example, in a context free grammar the domains of locality are the rules of the grammar, illustrated by the one-level trees on Fig 3. It is easy to see that in Fig 3 the two arguments of the predicate (V) occur in two different rules, and therefore they belong to two different domains of locality.

A formalism A is said to provide an *extended domain of locality* with respect to another formalism B if there is a grammatical constraint which is not specifiable in the local domains of B but can be specified in the local domains of A (Joshi, 2004).

Consider the TAG on Fig 4. Here the two arguments of the verb are encapsulated in the same elementary tree, i.e., they belong to the same domain of locality, which is extended as compared to the domain of locality of CFGs.

Specifying a grammatical dependency between the verb and its subject in the CFG G involves recursive application of the grammar rules since V and

NP are spread across more than one local domain (the rules S $\rightarrow$ NP VP and VP $\rightarrow$ V NP). This recursion is eliminated from the domain of locality of the TAG on Fig 2.4. where a dependency between subject and verb can be specified directly, within the same domain of locality.

### 2.4.2 Using TAG to express natural language Syntax

The extended domain of locality of LTAG and the factoring of recursion from the elementary trees are two properties of LTAG that make this formalism appealing from a linguistic perspective. These properties lead researchers to investigate how TAG can be applied to natural language syntax (e.g., Kroch and Joshi (1985), Kroch and Joshi (1987), Kroch (1989), Frank (1992), Frank et al. (1999)). Most findings of these investigations have been implemented in the XTAG grammar and parser (XTAG-Group, 2001).

The following linguistic assumptions about the well-formedness of elementary trees have been adopted in the above literature:

- An elementary tree represents the maximal syntactic projection of a lexical item and encapsulates all (and only) the syntactic/semantic arguments associated with it.
- auxiliary trees are used for modifiers, auxiliaries, raising verbs and other predicates with verbal complements
- a predicative lexical item has a substitution or foot node in its tree for each of its subcategorized (syntactic and semantic) argument. Adjuncts associated with auxiliary trees have a footnode for the category they modify
- predicates are associated with different elementary trees for each construction they can occur in (e.g., transitivity alternations - passive, middle; clause types - relative, interrogative, etc.)

### 2.4.3 Semantics of TAG

The semantic framework for LTAGs was first defined by Joshi and Vijay-Shanker (1999), Kallmeyer and Joshi (1999), and Kallmeyer and Joshi (2003).

In accordance with the principle of Elementary Tree Minimality, (the idea that elementary trees are basic, minimal, non-decomposable units of syntax, Frank (1992)), basic semantic representations are associated with individual elementary trees in the lexicon. They consist of a set of formulas, and a set of variables which can be individual or propositional variables.A label is associated with each formula.

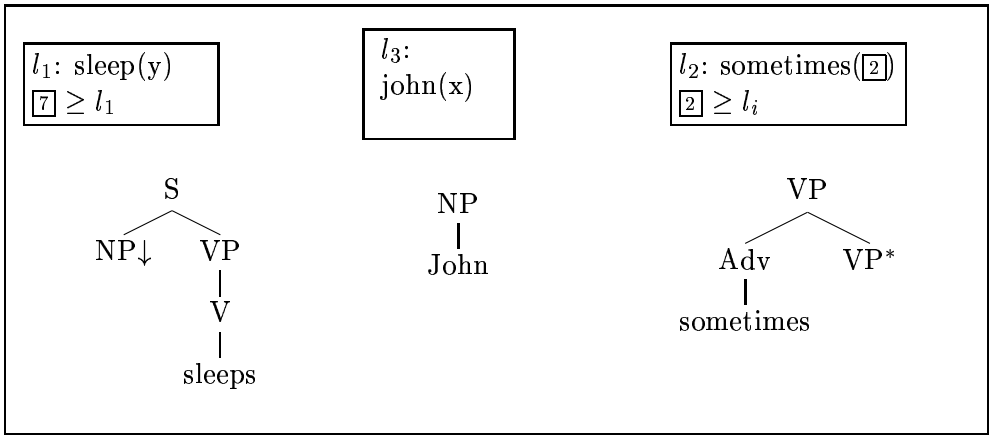



Figure 5: Elementary semantic representations

There are a set of scope constraints of the form $x \geq y$ (where x,y are propositional labels or propositional variables) imposed on the semantic representations which determine the relative scope of the individual semantic formulas. Figure 5 gives an example of the elementary semantic representations.

### 2.4.4 Lexicalized Tree Adjoining Grammar for Discourse

Bonnie Webber and her colleagues (Webber, 2004; Webber et al., 2003) extend a lexicalized TAG to discourse by introducing elementary tree structures for discourse connectives.

Initial Trees (Figure 6) convey discourse-level predicate-argument relations and are used for predicates whose local dependencies can be stretched long distance. Initial trees are used for subordinators, (*if, although, since, so that, in order for, in order to, to, by*), parallel constructions (*on the one hand ... on the other hand, either... or, not only... but also ..., admittedly... but*), coordinate constructions (*so*) and certain specific verb forms, like the imperative form of *suppose*.

Auxiliary Trees allow elementary trees to be modified or elaborated and can have two kinds of anchors: i) punctuation, coordinate conjunctions or null connectives (to express discourse units that continue a description) and ii) discourse adverbs (*instead, otherwise, then, in contrast, therefore, for example, nevertheless*).

In Webber et al. (2003) discourse adverbs express predicate-argument re-

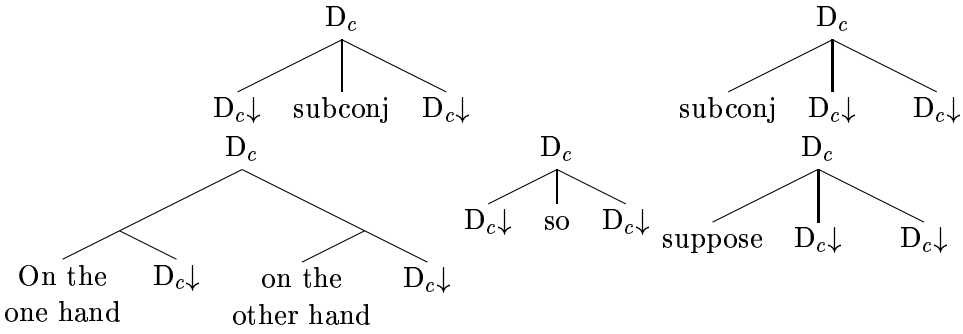



Figure 6: Elementary Discourse TAG trees

lations distinct from those conveyed by structural connectives. In their representation the elementary tree for discourse adverbs contains only one argument position because the other argument is recovered anaphorically from the previous discourse.

Because lexical items often only play discourse roles in specific structural configurations, the DLTAG perspective is to think of "anchored LTAG trees anchoring DLTAG-trees". So for example discourse adverbs are associated with sentence-level LTAG trees but have an interpretation that projects up to the discourse level.

## 2.5 Related work in Natural Language Generation

Work in the field of Natural Language Generation is very multi-faceted. Researchers have developed many different systems for specific purposes and a variety of different techniques and formalisms have been used over the years[6].

Implementations also differ in terms of the input expected by the system. The range varies between texts, a formal model or ontology created by the user with the help of an authoring tool and input from a knowledge base or database.

Reiter (1994) considers 6 NLG systems that generate English technical texts and implement the entire generation process, from some form of non-linguistic input data to English sentence outputs. The survey showed that

[6]Paiva (1998) gives an overview of 21 applied NLG systems. For the most recent list of existing NLG systems and what they generate see http://www.fb10.uni-bremen.de/anglistik/langpro/nlg-table/nlg-table-date-sort.html

all systems seem to conform to a modular architecture despite their widely ranging theoretical approaches. The modules are defined as Content Determination, Sentence Planning, Surface Generation, Morphology and Formatting and they are connected by a **pipeline architecture**.

The characteristics of a pipeline architecture is that the modules are linearly ordered and that information only flows one way from a certain module to the next, with no feedback between later stages and earlier modules in the pipeline.

While a modular architecture clearly has some advantages (e.g., for system development) and also seems to have at least some psycholinguistic relevance, it also restricts the number of solutions the system generates (as decisions made early in the pipeline are carried along to subsequent modules) and makes it very difficult if not impossible to represent phenomena that require information from more than one module simultaneously.

Moreover, a more recent survey of applied NLG systems (Cahill and Reape, 1998) has shown that although many systems do appear to follow the three stage consensus model of Content Determination, Sentence Planning and Surface Generation, the modules clearly have a very different function across NLG systems. Though there were many functional submodules that were shared by the systems, the order of execution of submodules and their assignment to one of the three stages did not show a consistent pattern. The survey also found that there was no definition of data interfaces between the consensus modules or even the lower level submodules that were shared by the system.

Several researchers have argued against a pipeline architecture, claiming that making choices in a predetermined order cannot lead to efficient generation of optimal texts (e.g., Danlos (1984), Mellish et al. (1998), Wanner and Hovy (1996)). More importantly, Koenraad De Smedt (1996) shows that the task of natural language generation can be thought of as simultaneous constraint satisfaction problem that involves information from several different linguistic levels.

The most recent attempt to define a **R**eference **A**rchitecture for natural language **G**eneration **S**ystems has been made by the **RAGS** project (Mellish et al., 2006) which defines a two-level model for describing NLG data structures. A high level model distinguishes six basic data types and defines how they relate to each other; and a low-level model specifies what data states can be passed between NLG modules.

The RAGS framework aims to provide a flexible definition of a *reference architecture* which can be adopted fully or partially by researchers work-

ing on developing NLG systems. Rather than proposing a standard or a *consensus* architecture, the purpose of RAGS is to serve as a well-specified reference point that researchers can choose to conform to in order to encourage comparison between systems. RAGS specifies abstract data types and a generic model of interactions between modules, but leaves decisions about processing flow to actual implementations.

RAGS introduces the following six high level abstract data types.

**Conceptual representations** typically serve as the input to NLG systems. They are not semantic representations and may not have a linguistic interpretation but they will have to be related to linguistic concepts later in the generation process.

**Rhetorical representations** specify how propositions are related to each other within a text. They define underlying rhetorical relations between parts of a discourse, and are distinguished here from document representations which specify how the individual relations might be realized by layout in written text (Scott and Souza, 1990; Power et al., 2003). The RAGS data specification assumes that rhetorical structure is hierarchical, and that it can be represented with a tree which has relations at the nodes and propositional content at the leaves, but which doesn't specify surface ordering of propositions.

**Document representations** specify information about the graphical presentation of written text, including its organisations into abstract document units, their relative position and layout. Parallel to rhetorical structure, document structure can also be represented as a tree, with feature structures at its nodes. The leaf nodes of a document structure tree are often associated with syntactic representations.

**Semantic representations** describe the meaning of individual propositions. They are structured in terms of lexical semantic predicates and semantic roles. Lexical semantic predicates are meant to be sufficiently abstracted away from specific lexical items so that the mapping between the two could be non-trivial. The RAGS specification does not require that quantifier scope be part of the representation, it can either be specified or left unspecified.

An example of a semantic representation that conforms to the RAGS specification is Minimal Recursion Semantics (MRS, introduced by Copestake et al. (1999) and used e.g., by Kay (1996), Stone et al. (2003)). A simple MRS representation (with an empty set of scope constraints) is illustrated in (23). In the second part of this paper MRS representations with no scope constraints will be used as an input to the proposed NLG system.

(23) $< h_0, \{h_1 : sneeze(x), h_2 : john(x), h_3 : past(h_1)\}, \{\} >$

The atomic units of a **Syntactic representation** in the RAGS framework is specified as a feature value matrix which can contain four main types of information: morpholexical class of the lexical item (e.g., part of speech); morphological information (e.g., third person, singular); root, stem and orthography; and other relevant information for realization (e.g., active or passive).

Non-atomic syntactic representations consist of a tuple <Head, Spec, Args, Adjs> where Head is the syntactic head, Spec is the specifier of the phrase, Args is the list of arguments and their grammatical functions and Adjs is a tuple of adjunct syntactic representations.

Finally, a **Quote representation** is a primitive type that allows the insertion of canned text in the output of an NLG system.

### 2.5.1 Generation as a Constraint Satisfaction Problem

The task of natural language generation has often been characterised as a *mapping problem* from non-linguistic representations to an expression in a natural language. Regarding generation as a constraint satisfaction problem redefines this basic task as finding natural language expressions that satisfy a set of constraints (given a non-linguistic input representation).

There are several advantages to this new perspective.

As pointed out by Piwek and van Deemter (2006), the idea that NLG systems have to take into account the fact that natural language expressions are very complex and are designed to satisfy several linguistic goals simultaneously goes back to at least Appelt (1985):

"One must constantly bear in mind that language behavior is part of a coherent plan and is directed toward satisfying the speaker's goals. Furthermore, sentences are not straightforward actions that satisfy only a single goal. The utterances that people produce are crafted with great sophistication to satisfy multiple goals at different communicative levels." (Appelt (1985), pp 1-2)

Constraint based approaches give a much more natural model of human languages than systems that regard generation simply as a mapping problem and they acknowledge the fact that verbatim mapping of an input to a natural language expression may not be the most important constraint to satisfy. This is the case for example if the purpose of the system is to achieve a communicative goal, to entertain people, or to have a conversation with a user.

Since constraints can not only apply to the surface properties of generated text but also to internal syntactic, semantic, pragmatic or document structure representations, this approach is very well suited to addressing problems that involve interactions between different levels of representations.

Constraint based systems also make it possible for system designers to separate a clear formulation of *optimal solutions* from *heuristics-based algorithms* for finding them.

For a detailed overview of constraint-based approaches in NLG see Piwek and van Deemter (2006).

An example of a constraint-based, implemented NLG architecture is the ICONOCLAST system[7] which builds on the idea of abstract document structure in Power et al. (2003). The implemented document structuring module imposes constraints on the set of permissible document structures and then ranks possible solutions according to a set of preferences specified by the user (Power, 2000).

Constraints are implemented by associating four variables with each node in the rhetorical structure: LEVEL (L0: text phrase, L1: text clause, etc.), INDENTATION (I0...Imax), CONNECTIVE (0 or a word from the lexicon), POSITION ($i \in$ 1..S where S is the number of the node's sisters. $i$ represents the position the node should be realized in with respect to its sisters.) The document structurer first determines possible values (the domain) of variables then it imposes constraints on combinations of these values and finally it enumerates combinations that satisfy the constraints.

There are two kinds of constraints that are applied to the set of possible solutions. Hard constraints filter out incorrect solutions and soft constraints are used to rank the set of remaining texts. The soft constraints assign each document a defect if:

- a nucleus appears before its satellite
- the generated structure is left branching
- a rhetorical grouping is lost
- a single sentence paragraph is generated
- oversimple text clauses are generated
- a discourse connective is repeated.

Kibble and Power (2004) extend the ICONOCLAST system by adding constraints that reformulate the transitions defined by Centering Theory (Grosz et al., 1995). This approach is unique in that it applies the tenets of Cen-

[7]Integrating Constraints on Content, Layout and Style
http://mcs.open.ac.uk/nlg/old_projects/iconoclast/

tering Theory simultaneously and in an integrated fashion to text planning, sentence planning and pronominalisation.

Another example of researchers designing linguistically motivated constraints to guide natural language generation are the heuristics defined by Scott and Souza (1990).

Scott and Souza (1990) describes phsycholinguistically informed hypotheses about the way rhetorical structure should be realized by a text in order to make automatically generated messages easily retrievable.

They assume that basic elements of input messages are verb-based, clause-sized propositions, each of which can be expressed as a single sentence. Of the three ways of combining the sentences that realize the input propositions – embedding, paratactic coordination, hypotactic coordination – Scott and Souza (1990) consider the first two. Since the syntax of complex sentences strongly correlates with their perceived rhetorical structure the hypotheses attempt to give an account of when the different types of combination should be used in order to faithfully express the input rhetorical structure.

The most relevant hypotheses to my research are the ones on embedding (for a full explanation and examples see Scott and Souza (1990) and references therein):

**Hypothesis 4.** Embedding is the only available textual marker for the elaboration relation, therefore embedding should only be applied to express elaboration.

**Hypothesis 5.** Embedding provides a strong syntactic cue that the embedded constituent is semantically subordinated to the matrix proposition. Therefore when embedding, nucleus of the relation must form the matrix of the sentence and the satellite the embedded clause.

**Hypothesis 6.** When the nucleus of an elaboration relation is more complex, there may be more than one candidate matrix proposition. In this case, the matrix proposition must be the earliest occurring candidate in the nucleus.

**Hypothesis 7.** Complex satellites can occur with *list* relations and we have to make sure that embedding doesn't destroy the relation or create any dangling sentences. Propositions of a list relation therefore should not be embedded if only one proposition remains in the relation after embedding.

**Hypothesis 8.** Syntactically simple expressions of embedding are to be preferred over more complex ones. (lexicalized modifiers are syntactically simpler than clausal/phrasal modifiers).

**Hypothesis 9.** Self embedding is only allowed in cases where the proposition that is the deeper of the two embeddings is expressed as an adjective or adverb.

### 2.5.2 Generation with TAG - Existing NLG systems

The earliest approach to generation using Tree Adjoining Grammar that I am aware of is McDonald and Pustejovsky (1985) who combine TAG with a systemic functional grammar in a generation system called MUMBLE-86.

MUMBLE-86 generates text from "L-Spec" representations which contain information about communicative goals, rhetorical force and semantic information. A dictionary-like lookup process transforms pieces of L-Spec into TAG-trees. Once a tree is picked, grammar routines traverse the tree to check grammaticality and they record contextual and pragmatic information that can be used to pick TAG trees for the remaining pieces of the input L-Spec. As TAG trees are traversed by grammar routines they are expanded by substitution and adjunction if the L-Spec contains an item whose realization can be used to expand the current TAG-tree.

McCoy et al. (1992) point out several problems with Mumble, of which two are relevant to the current context of discussion: 1) Mumble always starts generation from an initial tree and expand it by adjoining or substituting other trees. When generating an embedded structure, generation therefore must start with the innermost embedded clause. As a consequence, a separate module is needed in order to bring the input into a special logical form and to mark the place where generation can begin. 2) Mumble generates input strictly left-to-right which can be problematic at the generation of connectives that require the system to have access to both arguments of the connective at the same time.

In the system described by McCoy et al. (1992) generation starts by traversing a systemic grammar. This identifies the head and argument structure and a set of functional features that can be used to select a TAG tree. Arguments are then realized by a recursive traversal of the systemic grammar and tree adjoining grammar. The results of these network (grammar) traversals are stored in separate "regions", i.e., there is a separate region data structure for each piece of input (as opposed to Mumble's constantly expanding surface structure). Functional and syntactic selectional restrictions are passed down from mother to daughter regions using feature passing. McCoy et al. (1992) call this phase of generation the "descent process".

Once the system found a TAG tree for each region, the trees are combined using the standard TAG operations of substitution and adjunction. This

phase of generation is called the "ascent process" and results in the realization of the whole input.

More recent systems that use TAG for generation include Gardent and Thater (2001) who present a generator that uses tree descriptions and a flat semantic representation combined with constraints to eliminate invalid linguistic structures; Becker (2002) who uses a TAG to describe templates used for concept-to-speech synthesis in the SMARTKOM project and Bangalore and Rambow (2000) who combine a statistical approach to sentence realization with a tree-based representation for syntax.

Stone and Doran (1997) and Shieber and Schabes (1990) describe work that is most closely related to my research proposal.

Stone and Doran (1997) describes a sentence planner that uses a unified formalism to model the interaction of pragmatic and syntactic constraints on descriptions in a sentence. The system plays the role of a librarian answering patrons' queries about books.

The unique idea behind the system is that the generation of sentences is treated parallel to the construction of referring expressions, where "given a set of entities to describe and a set of intentions to achieve in describing them, a plan is constructed by applying operators that enrich the content of a description until all intentions are satisfied." These operators are represented as LTAG elementary trees that are associated with discourse constraints and a flat semantic representation. Pragmatic constraints model contextual conditions under which use of the LTAG tree is licensed. These constraints refer to four kinds of pragmatic notions that are frequently used in the linguistic literature: "newness" (Hearer-new, Hearer-old, Discourse-new, Discourse-old), salience, membership in Partially Ordered Set (POSET) relations with other discourse entities and open propositions under discussion. Lexical items are associated with the semantics of the word, pragmatic constraints on its use and the set of trees that describe the combinatory possibilities for realizing the word (possibly specifying additional pragmatic constraints).

The work reported in Shieber and Schabes (1990) is another example of a generation algorithm where syntactic and semantic composition is performed in parallel. Shieber and Schabes apply the framework of Synchronous TAGs (previously used for semantic interpretation and machine translation) to natural language generation. An STAG consists of pairs of TAG trees where one tree characterizes the natural language and the other the logical form language. In this framework the generation process is considered to be a reverse problem to "semantic parsing", i.e., . parsing natural language to derive a semantic interpretation. Here the input to the system is a semantic

formula which is parsed using the synchronous grammar to yield a natural language sentence.

### 2.5.3 Reducing the Complexity of Generation with TAG

The complexity of surface realization from a flat semantic representation is known to be exponential in the length of the input. As Kay (1996) shows, one reason for this is that the input to surface realization is a set of literals, rather than an ordered set of strings as in parsing. If each literal selects just one element in the lexicon then these lexical elements can be combined in $2^n$ (number of subsets of a set of size $n$) ways. Most existing realisers impose constraints on combination to reduce this complexity (e.g., constituents can be combined only if they have non-overlapping semantics and compatible indexes), but the processing of intersective modifiers cannot be so reduced. Given a set of $n$ intersective modifiers, all possible intermediate structures will be constructed by the realiser ($2^{n+1}$).

Another phenomenon that increases complexity of surface realization is lexical ambiguity. In any decent size grammar each semantic representation will be associated with more than one lexical item. So for an input representation of $n$ semantic formula, if we represent the number of lexical entries for the $i$th formula with $Lex_i$, the number of possible lexical sets for the input is $\prod_{i=1}^{i=n} Lex_i$.

So the final complexity given the above two sources (Gardent and Kow, 2006) is

$$2^n \times \prod_{i=1}^{i=n} Lex_i$$

As Gardent and Kow (2006) shows, Tree Adjoining Grammars naturally support at least three ways to reduce this complexity.

**Polarity Filtering** is a technique introduced by Perrier (2003) to reduce the impact of lexical ambiguity on parsing efficiency (see also Bonfante et al. (2004)). The technique is based on the observation that when a set of lexical resources cannot be combined it is most often because either a lexical resource is missing or because a syntactic requirement is not fulfilled[8]. Polarity filtering eliminates these sets of lexical items by associating each lexical entry with a set of polarities that reflect its syntactic requirements and resources. For each set of lexical items selected for a given input, the sum of polarities is computed and where the sum of polarities is not zero the sets are eliminated.

[8] This idea is similar to proof nets in type logical grammar

TAG provides a natural way to associate polarities with elementary trees: each elementary tree is associated with a polarity $+C$ where $C$ is the category of the root node and for each substitution or foot node in a tree, a polarity of $-C$ is added where $C$ is the category of the substitution or foot node. As the experiments of Gardent and Kow (2006) show, polarity filtering can substantially reduce the initial search space, for example for an input size of 14-16 semantic representations it divides the search space by 441.6 .

So far Polarity Filtering has not been implemented either in HPSG or the CCG approach. Although the method could presumably be adapted for these grammar formalisms as well, as Gardent and Kow (2006) points out, TAG provides a much more natural framework for this than either HPSG or CCG.

The assignment of polarities to types in Combinatory Categorial Grammar is not as straightforward, since each category can function both as a resource and a requirement, depending on the type of combinatory rule that is applied. So the computation of polarities would need to take into account the category structure as well as the several combination rules in the grammar. Similarly in HPSG one would need to take into account the interaction of lexical categories and lexical and phrasal rules.

**Delayed Adjunction** is a technique that has been proposed to reduce the computational complexity introduced by modifiers (Carroll et al., 1999; Carroll and Oepen, 2005). The idea of the strategy is that modifiers should be handled only after all predicate-argument requirements have been fulfilled or before the constituent to be modified has been combined with other elements. This reduces the computational complexity from $2^n$ for $n$ modifiers to $2^k + 2^{n-k}$ where $k$ is the number of semantic representations to be realized as modifiers.

Gardent and Kow (2006) shows that the two combination operations of Tree Adjoining Grammars provide an intuitive basis for the implementation of delayed adjunction: substitutions can be handled in the first phase of generation, and adjunctions in the second. This two-phase approach can be combined with polarity filtering to further improve performance as it provides an opportunity to discard trees that will not lead to a valid derivation already after the first phase, in particular all trees can be eliminated that have an unfilled substitution site and all saturated trees whose root node is not labelled S (assuming that the purpose is to generate sentences).

**Empty Semantic Items** are function words like 'to' or the complementizer 'that' which are required in the generated string but are not represented in the input semantic representation. A common way to reduce the compu-

tational complexity associated with handling these empty semantic items is to have a set of rules that associate empty elements with certain input semantic representations (Carroll et al., 1999). However, as Gardent and Kow (2006) point out, this approach fails to represent the fact that empty semantic items tend to be functional words, governed by syntax rather than semantic constraints. The extended locality of TAG elementary trees on the other hand provide the necessary syntactic environment for function words, which can be incorporated as co-anchors in elementary trees. This means that empty semantic items do not need to be postulated at all in a TAG-based generator and therefore have no impact on complexity.

## 2.6 Discussion

Lexicalized Tree Adjoining grammars have several properties that make them appealing as a formalism for natural language generation. The application of the formalism to natural language phenomena is well researched and the extended domain of locality of TAGs makes it possible to represent the context of a lexical item on several linguistic levels.

The Tree Adjoining Grammar that I propose for my research builds on the syntactic trees defined by the XTAG grammar (XTAG-Group, 2001) and adopts the main ideas from the literature on the semantics of TAG. However it tries to formulate discourse level as the semantics of document structure which is a departure from the ideas in the literature on Tree Adjoining Grammars for Discourse (DLTAG).

The DLTAG formalism models the syntax and compositional semantics of discourse connectives on the discourse level by specifying elementary trees for the syntax of discourse. It does not give an account of the textual units that can be used to express the arguments of discourse connectives. It also does not say anything about the syntactic realization of arguments on the discourse level, although this information is recoverable if we look at the syntactic tree that anchors the discourse tree.

My approach to generation is similar to Shieber and Schabes (1990) as it uses a parsing algorithm to associate syntactic and document structure representations with input semantic formulas. It is also similar to the approach taken in the SPUD system in that TAG elementary trees encode semantic and discourse-level information.

But none of the above mentioned generation systems have applied TAG to integrate information from all four levels of representation: syntax, semantics, discourse and document structure. This is one of the goals of my research project.

My generation system adopts ideas from previous constraint-based generation systems, particularly the ICONOCLAST document planner. It reformulates the constraints and heuristics specified by previous researchers as constraints on tree selection (or tree sets). The constraints imposed on tree sets will guide the generation process in an integrated fashion from semantic representations to syntax and document structure. A novelty of this approach is that not only document planning but also sentence planning will be reduced to a constraint satisfaction problem.

# 3 Research Proposal

The primary aim of this research is to *design a unified architecture that will allow the generation of parenthetical constructions in a generalized way.*

The approach adopted for this research is a unified generation architecture where every linguistic level is represented with the same theoretical formalism — a Lexicalized Tree Adjoining Grammar (LTAG Joshi et al., 1975) — and is treated using the same computational technique — constraint-based reasoning (Van Hentenryck, 1989).

The idea of an integrated generation architecture goes back to at least Appelt (1985) and has been most recently implemented by Kantrowitz and Bates (1992) and Harbusch and Woch (2002).As opposed to a traditional pipeline model where several modules are defined using different theoretical formalisms, a unified architecture makes use of a central knowledge base (expressed in a complex theoretical formalism) and a single processing unit that combines pieces of information from the central knowledge base to yield natural language output.

There are several advantages to an integrated generation architecture:

- the central knowledge base can be easily extended without restructuring the system's processing unit, therefore the system is easier to maintain

- there is no need to define an explicit communication language between modules because there is an implicit communication between concepts and their natural language counterparts in the central knowledge base

- the system is easier to parameterize, which allows the adaptation of the generation process e.g., to different psycholinguistic or linguistic models, or different styles of text

- there is no *generation gap* (Meteer, 1990), where one module of the system can possibly make a decision that conflicts with the decision made by another module further down the line.

Integrated generation architectures are a prime example of the "complicate locally, simplify globally" (CLSG) approach developed originally for grammar formalisms (Joshi, 2004). The main idea of the CLSG approach is to associate complex primitive structures with lexical items which directly capture their relevant linguistic properties and introduce general operations to compose these complex primitives.

Information that needs to be captured (or localized) by each primitive structure includes for example the number of arguments the lexical item has and

the syntactic, semantic, pragmatic properties and relative location of these arguments (with respect to the lexical item). In an alternative approach that starts with simple primitive structures but introduces complex combining operations such information will be distributed across the elementary structures, i.e., the dependencies between pieces of information associated with a lexical item will become non-local to the lexical item. The CLSG approach pushes these dependencies to become local, so that they are specified within individual primitive structures. (For a discussion of the domain of locality of grammar formalisms see section 2.4.1.)

In a CLSG approach, although the primitives are more complex, the combining operations are simple and they are also language independent (Joshi, 2004). An integrated natural language generation architecture arising from the CLSG approach therefore also has important linguistic and psycholinguistic implications as well as being computationally elegant.

Although in principle it is possible to develop an integrated architecture with a number of grammar formalisms (e.g., HPSG (Pollard and Sag, 1994), LFG (Kaplan and Bresnan, 1982), CCG (Steedman, 1996)), the class of grammar formalisms which take the CLSG approach to the extreme is the class of Lexicalized Tree Adjoining Grammars (Joshi, 2004), by virtue of their extended domain of locality[9]. For this reason, we have used LTAG as the underlying formalism for our integrated generation architecture and the research proposed here will explore how information about rhetorical structure and document structure can be integrated into the local domain of elementary trees.

In addition to the combining operations of Tree Adjoining Grammar (substitution and adjunction), our generation architecture also uses constraint-based reasoning (Van Hentenryck, 1989) to further reduce computational complexity, control the generator and rank the output texts.

The main contributions of the thesis are expected to be the following:

- extending a Tree Adjoining Grammar to represent abstract document structure (necessary for representing many parenthetical constructions)
- extending previous constraint-based approaches to represent interactions between linguistic levels (by formulating constraints on elementary TAG trees)
- developing an integrated natural language generation system that handles parenthetical constructions in a principled way and generates many different solutions for the same semantic representation. This includes extending constraint satisfaction techniques to sentence planning

[9] see section 2.4.1

- developing an evaluation method suitable for evaluating the generated text and whether parentheticals have been generated appropriately (reflecting the input rhetorical structure).

The rest of this paper explains the chosen research approach in detail. The next section explores how Tree Adjoining Grammars can be extended to give an account of document structure and section 3.3 describes the proposed system architecture. Section 3.4 discusses the expected research questions and section 4 gives a work plan and estimated time line.

### 3.1 A Tree Adjoining Grammar for Document Structure

Power et al. (2003) take the view that document structure and rhetorical structure should have the same relation as sentence-level syntax and semantics: document structure (just like syntax) describes the form of a linguistic expression, and rhetorical structure (just like semantics) describes the meaning, or the semantic organisation of a text.

For example, a semantic formula $approve(fda, elixir)$, meaning that the Food and drug Administration (FDA) approves the medicine Elixir can be realized in English by a syntactic form like

$[_S[_{NP}[_{Det}the]\ FDA]]\ \ [_{VP}\ \ approves][_{NP}Elixir]]]$

If a second semantic formula, $contain(elixir, gestodene)$, is added to the representation and we assume that the author knows that gestodene is a controversial ingredient, the CONCESSION rhetorical relation might be used to link these two semantic representations:

$concession(contain(elixir, gestodene), approve(fda, elixir))$

where the second argument of CONCESSION is the central one (the nucleus) and it is supported by the first argument (the satellite).

To realize this more complex message, Power et al. (2003) argue, we need more information than represented by a syntactic phrase-marker – we need to consider document structure as well as syntax. For example in (24a) and (b) the complex message is realized as a single syntactic sentence, but in (24c, d) and (e) the arguments of the rhetorical relation are expressed as separate sentences (i.e., the rhetorical relation is realized by both document structure and syntax).

(24) a Although Elixir contains gestodene, it is approved by the FDA.
b The FDA approves Elixir although it contains gestodene.

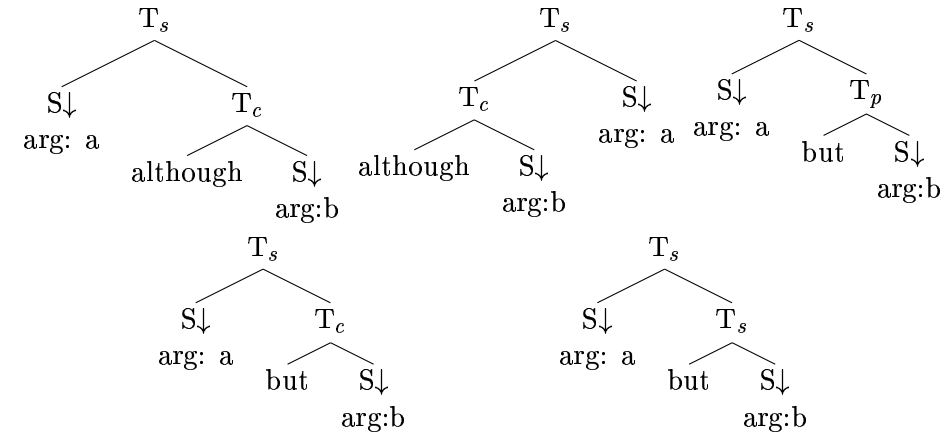



Figure 7: Elementary Document Structure trees for the CONCESSION relation

c Elixir contains gestodene; however, it is approved by the FDA.

d Elixir contains gestodene. However, it is approved by the FDA.

e Elixir contains gestodene. However, it is approved by the FDA.

Joshi and Vijay-Shanker (1999) define a compositional semantics for Tree Adjoining grammars where semantic propositions are associated with elementary trees. These elementary semantic propositions model the meaning of a lexical item in the syntactic context of the elementary tree and their arguments are co-indexed with the syntactic argument positions of the tree.

This tight coupling of syntax and semantics within elementary trees is parallel to the tight coupling of rhetorical structure and document structure in the analysis adopted here.

In a Tree Adjoining Grammar for Document Structure elementary trees are anchored by discourse connectives and they express relations between document structure units. Each elementary tree is associated with a semantic proposition expressing a rhetorical structure. For example, figure 7 illustrates five elementary trees for the CONCESSION relation.

Each rhetorical relation is associated with several document structure trees, one for each document structure context that can realize the relation. The elementary trees are anchored by the connectives that can be used to express the discourse relation.

The definition of the node labels S, Tp, Tc, Ts in the Document Structure trees above are adopted from Power et al. (2003) and can be described as follows:

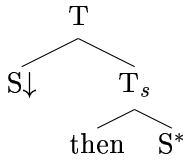
- S: syntactically saturated sentence
- Tp: text-phrase, span of text that ends with a comma
- Tc: text-clause, span of text that ends with a colon or semicolon
- Ts: text-sentence, span of text beginning with a capital letter and ending in a full stop

### Discourse Adverbs

Webber et al. (2003) define elementary TAG trees for discourse, anchored by discourse connectives, that are very similar to the document structure trees introduced in the previous sections. However, document structure trees cannot be based entirely on DLTAG[10] trees so it is not enough to just change the node labels into document units. More research is needed to establish specific elementary trees for document structure.

For example, in Webber et al. (2003) discourse adverbs take only one argument syntactically, the other is recovered anaphorically from the discourse context. However, discourse adverbs also require that the arguments of the rhetorical relation they express be realized with a certain document unit (Ts, Tc, etc.). So how can we make sure when generating a sentence that the anaphoric argument is assigned the right document structure when they are not localized in the adverb's elementary tree?

A safe working hypothesis is to generate the anaphoric argument immediately preceding the structural argument. This can be achieved by including a substitution site for the anaphoric argument in the connective's elementary tree:




This working hypothesis could be revised or confirmed after a corpus study investigating constraints on how much text and what kind of discourse context/rhetorical relations can intervene without changing the interpretation of the adverb. For example it could turn out that sentences realizing certain rhetorical relations (e.g., ELABORATION) are allowed to be adjoined to the "anaphoric" argument of a discourse adverb (realized by the substitution site

[10] Lexicalized Tree Adjoining Grammar for Discourse

in the above elementary tree), but not sentences realizing e.g., the EVIDENCE or CONCESSION relation. We leave this as a topic for further research.

### Modelling Embedding

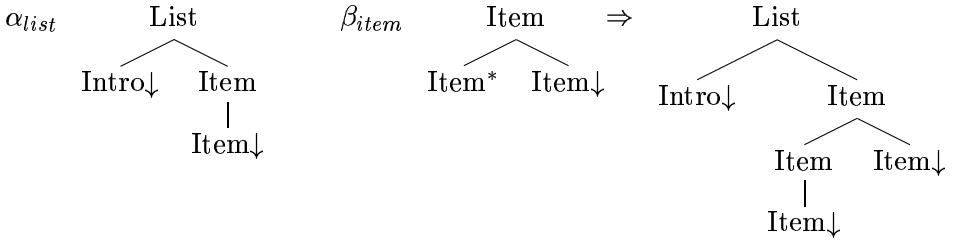
In some contexts that involve embedding it is not possible to associate elementary trees with rhetorical relations because the two arguments of the relation combine with each other (in most cases by adjoining). Embedded constructions can be modelled instead by imposing constraints on the types of elementary trees selected by the arguments.

A possible constraint (based on the heuristics of Scott and Souza (1990)) is the following: if a rhetorical structure is to be realized with interpolation, the embedded constituent (usually the satellite) is allowed to select either an initial tree or an auxiliary tree that adjoins to the 'host' constituent containing the embedding. The 'host' tree (usually the nucleus) on the other hand has to select an initial tree.

For example, if the satellite (*new(car)*) of the rhetorical relation ELABORATION[honda(car), new(car)] is to be realized as an embedded constituent (as in '*The new car is a honda*') then it is not allowed to select an initial tree whereas the nucleus (*honda(car)*) will be required to be an initial tree. A more detailed walk through of this example will be provided in section 3.3.

### Document Structure for Lists

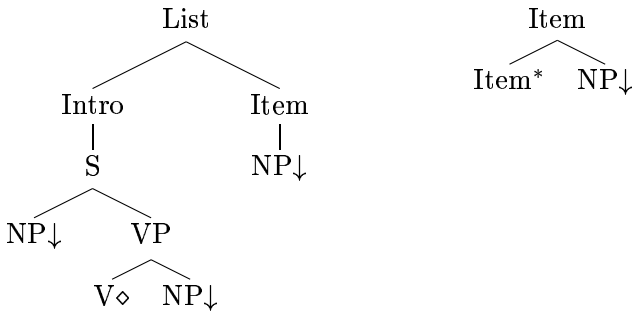
Another example of a rhetorical relation that is not associated with discourse connectives is the list relation. Lists can be associated with the following generic representation:




$\alpha_{list}$ is the elementary tree for a simple list with only one list item. The list introduction is substituted into the node labelled Intro↓ and the single list item is attached to the Item↓ node. Each additional list item selects an auxiliary tree with a Item root node that adjoins onto $\alpha_{list}$ so the multi-

nuclear rhetorical relation list(a,b,c,d) is broken down into three separate parts: list(a,b), bullet(b,c), bullet(c.d).

In order to model different types of lists the Intro and Item nodes should be refined to reflect the syntax specific to lists, for example the following tree shows a refinement for NP-lists:




## 3.2 Corpus Study

Since this dissertation aims to carry out research in the area of Natural Language Generation the proposed linguistic analysis on parentheticals will be focusing on the rhetorical, semantic and syntactic properties that licence the generation of certain kinds of parentheticals. The purpose of NLG is after all to generate text that clearly conveys the underlying message so in order not to mislead or confuse users even further by generating parentheticals, we need to answer the question: in what contexts are these constructions unambiguously signalling a particular rhetorical relation?

In order to best answer this question it is necessary to perform a corpus analysis and look at the rhetorical contexts where parentheticals occur. There have been few corpus studies on parentheticals and the existing ones have focused on the prosodic or syntactic aspects of the construction as well as considering types of parentheticals that are excluded from the current research (e.g., Dehe and Kavalova (2006), Kavalova (2007)).

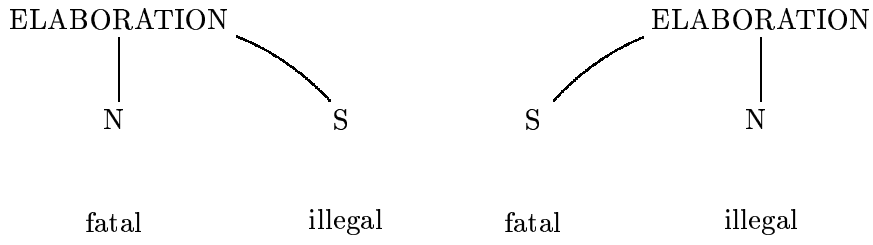
As a starting point for the corpus study we will consider at least three kinds of constructions: a subclass of adverbials, adjectives in specific contexts and non-restrictive relative clauses.

**Adverbials** can serve as parentheticals (in the sense of Haegeman (1991)) as in (25):

(25) John, *unfortunately*, has been delayed.

Adjectives and relative clauses can signal the elaboration rhetorical rela-

tion[11] in certain contexts. An example for the latter two kinds, given by Scott and Souza (1990), are the two propositions in (26a) which form the nucleus and satellite of an elaboration relation, schematically illustrated in Figure 8a.




(a) Rhetorical structure for a,b,c (b) Rhetorical structure for d

Figure 8: Rhetorical structures for (26)

This rhetorical relation can obviously be generated as the two consecutive sentences in (26a), but a more sophisticated system should also be able to realize it in a more complex sentence where the satellite of the elaboration is embedded in the nucleus. This embedding can take the form of an adjective (b) or a **relative clause** (c).

(26) a The substance is fatal. The substance is illegal.
b The illegal substance is fatal.
c The substance, which is illegal, is fatal.
d ## The fatal substance is illegal.

(26a,b and c) are semantically equivalent given the rhetorical relation in Figure 8a., and therefore they are all possible realisations of the input. Notice however that the representation in (26d) corresponds to the rhetorical structure illustrated in Figure 8b, therefore it is not semantically equivalent to (26a,b,c) and in a more sophisticated system it should not be a possible realization of the input. Scott and Souza (1990) mention several heuristics based on psycholinguistic evidence that can be used to decide when to use embedding and how to choose between possible syntactic realisations of the elaboration relation[12]. These heuristics will be implemented (and evaluated) in the first phase of grammar development and then the results augmented by implementing additional rules based on the corpus study.

The second goal of the corpus study is to look at parenthetical constructions

[11] See section 2.2 for a brief introduction to Rhetorical Structure Theory
[12] For a summary of these heuristics see section 2.5.1

at the interface of document structure and syntax, i.e., cases of syntactically independent interpolation that are licensed by some form of punctuation or layout, like dash-interpolations (27a), footnotes (27b) or sentence-embedded lists (28 and 29):

(27) a My car - you won't believe this - has been destroyed by a drunken idiot.

b My car[13] has been destroyed by a drunken idiot.

(28) Are you taking any of the following:

a Anticoagulants?

b Lithium?

c Methotrexate?

d Any other medicines which your doctor does not know about?

(Power et al., 2003)

(29) Are you taking anticoagulants, lithium, methotrexate, or any other medicines which your doctor does not know about?

These constructions are more problematic for generation because existing computational grammars are not equipped to handle them. There are two reasons for this.

*First*, these parentheticals are licensed by certain types of **abstract document structure** (as introduced by Power et al. (2003)), realized by punctuation or layout, which is typically not part of existing grammar implementations.

So far this theory of abstract document structure has only been implemented in the ICONOCLAST system which does not generate parentheticals. Other computational grammars do of course have a representation of *concrete* realisations of punctuation but have no notion of abstract document structure.

*Second*, these types of parentheticals are in most cases syntactically independent of their host clause therefore they will not be generated in clause medial positions by existing computational grammars.

The proposed corpus for this study is the **P**enn **D**iscourse **T**reebank (Miltsakaki et al., 2004a,b). In the **PDTB** both syntactic and discourse structure are annotated so the corpus provides a good context for studying the relationships between rhetorical role, syntax and punctuation. I propose to

[13] a beautiful green Ford Escort

study types of embedded constructions that are already tagged in the treebank and if time permits also tag interpolations that are not annotated (e.g., dash interpolations, parentheticals, comma-bounded interpolations).

### 3.3 System Architecture

In this section we describe a system architecture that will be used to generate parenthetical constructions. The system uses the Tree Adjoining Grammar formalism to represent syntax, semantics, rhetorical structure and document structure in the integrated fashion described above.

The generation process involves the following four subtasks: tree selection; applying constraints to tree sets to eliminate inappropriate solutions; combining the remaining tree sets with the usual TAG operations; and finally, ranking the output using ranking constraints.

The input to the system is a set of semantic formulas and rhetorical relations, in a representation similar to minimal recursion semantics (Copestake et al. (1999), see also section 2.5). Optional information about the desired document structure can also be specified the same way.

The output is a set of documents that are compatible with the rhetorical structure given in the input and that contain the appropriate form of embedded constructions where embedding is possible.

A preliminary version of the system has been implemented in Prolog, using a toy grammar that contains trees for two rhetorical relations (EVIDENCE and ELABORATION) and elementary trees for basic transitive verbs, nouns, determiners, modifier adjectives, predicative adjectives and relative clauses.

Elementary trees for lexical items (other than discourse connectives) have been adopted from the XTAG grammar developed by the XTAG group at the University of Pennsylvania (XTAG-Group, 2001). Trees for discourse connectives have been specifically designed to incorporate document structure nodes, based on information from Power et al. (2003) and related publications.

The remainder of this section gives an illustrated walk through of the implementation, using the example semantic representation in (30).

(30) $p_0$: my-car(x)
$p_1$: japanese(x)
$p_2$: honda(x)
$p_3$: new(x)
$r_1$: evidence($p_1$, $p_2$)

$r_2$: elaboration($p_0$, $p_3$)

### 3.3.1 Tree Selection

The first step in generation is to select a tree for each semantic formula in the input. As there are several trees associated with each formula, the result of *tree selection* will be many tree sets. The possible elementary trees for each semantic formula in the input in (30) are illustrated on Fig. 9 and the 54 logically possible tree sets are shown on Figure 10.

In the remainder of this section we will use the handle of the semantic formula combined with the letter a, b, or c to indicate the specific tree selected by the system for the semantic formula under discussion. For example, $p1_a$ refers to the first tree (cell a) in the table on figure 9, associated with the proposition "$p_1$: japanese(x)".

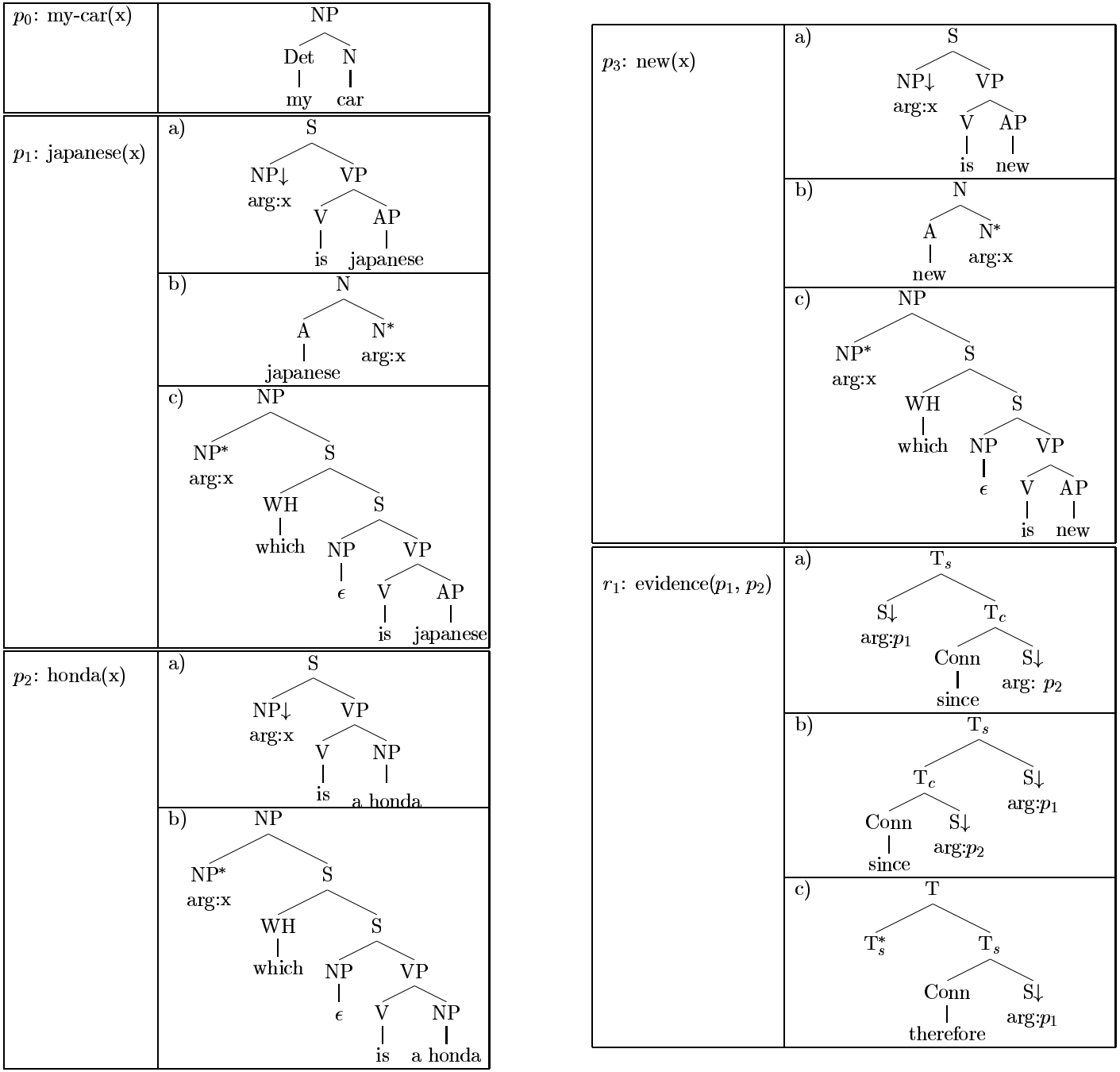



Figure 9: Tree selection

| | | |
|---|---|---|
| 1. $p0, p1_a, p2_a, p3_a, r1_a$ | 19. $p0, p1_b, p2_a, p3_a, r1_a$ | 37. $p0, p1_c, p2_a, p3_a, r1_a$ |
| 2. $p0, p1_a, p2_a, p3_a, r1_b$ | 20. $p0, p1_b, p2_a, p3_a, r1_b$ | 38. $p0, p1_c, p2_a, p3_a, r1_b$ |
| 3. $p0, p1_a, p2_a, p3_a, r1_c$ | 21. $p0, p1_b, p2_a, p3_a, r1_c$ | 39. $p0, p1_c, p2_a, p3_a, r1_c$ |
| 4. $p0, p1_a, p2_a, p3_b, r1_a$ | 22. $p0, p1_b, p2_a, p3_b, r1_a$ | 40. $p0, p1_c, p2_a, p3_b, r1_a$ |
| 5. $p0, p1_a, p2_a, p3_b, r1_b$ | 23. $p0, p1_b, p2_a, p3_b, r1_b$ | 41. $p0, p1_c, p2_a, p3_b, r1_b$ |
| 6. $p0, p1_a, p2_a, p3_b, r1_c$ | 24. $p0, p1_b, p2_a, p3_b, r1_c$ | 42. $p0, p1_c, p2_a, p3_b, r1_c$ |
| 7. $p0, p1_a, p2_a, p3_c, r1_a$ | 25. $p0, p1_b, p2_a, p3_c, r1_a$ | 43. $p0, p1_c, p2_a, p3_c, r1_a$ |
| 8. $p0, p1_a, p2_a, p3_c, r1_b$ | 26. $p0, p1_b, p2_a, p3_c, r1_b$ | 44. $p0, p1_c, p2_a, p3_c, r1_b$ |
| 9. $p0, p1_a, p2_a, p3_c, r1_c$ | 27. $p0, p1_b, p2_a, p3_c, r1_c$ | 45. $p0, p1_c, p2_a, p3_c, r1_c$ |
| 10. $p0, p1_a, p2_b, p3_a, r1_a$ | 28. $p0, p1_b, p2_b, p3_a, r1_a$ | 46. $p0, p1_c, p2_b, p3_a, r1_a$ |
| 11. $p0, p1_a, p2_b, p3_a, r1_b$ | 29. $p0, p1_b, p2_b, p3_a, r1_b$ | 47. $p0, p1_c, p2_b, p3_a, r1_b$ |
| 12. $p0, p1_a, p2_b, p3_a, r1_c$ | 30. $p0, p1_b, p2_b, p3_a, r1_c$ | 48. $p0, p1_c, p2_b, p3_a, r1_c$ |
| 13. $p0, p1_a, p2_b, p3_b, r1_a$ | 31. $p0, p1_b, p2_b, p3_b, r1_a$ | 49. $p0, p1_c, p2_b, p3_b, r1_a$ |
| 14. $p0, p1_a, p2_b, p3_b, r1_b$ | 32. $p0, p1_b, p2_b, p3_b, r1_b$ | 50. $p0, p1_c, p2_b, p3_b, r1_b$ |
| 15. $p0, p1_a, p2_b, p3_b, r1_c$ | 33. $p0, p1_b, p2_b, p3_b, r1_c$ | 51. $p0, p1_c, p2_b, p3_b, r1_c$ |
| 16. $p0, p1_a, p2_b, p3_c, r1_a$ | 34. $p0, p1_b, p2_b, p3_c, r1_a$ | 52. $p0, p1_c, p2_b, p3_c, r1_a$ |
| 17. $p0, p1_a, p2_b, p3_c, r1_b$ | 35. $p0, p1_b, p2_b, p3_c, r1_b$ | 53. $p0, p1_c, p2_b, p3_c, r1_b$ |
| 18. $p0, p1_a, p2_b, p3_c, r1_c$ | 36. $p0, p1_b, p2_b, p3_c, r1_c$ | 54. $p0, p1_c, p2_b, p3_c, r1_c$ |

Figure 10: 54 Logically possible tree sets

### 3.3.2 Constraints on Tree Sets

After all the possible tree sets have been selected for the input, the system applies constraints to reduce the number of solutions. The following constraints have been adopted from Scott and Souza (1990).

- When two clauses that realize arguments of a rhetorical relation are combined by embedding, the nucleus must form the matrix clause

This means, $p2$: honda(x) must select an initial tree, i.e., the relative clause tree in option b) for $p_2$ on figure 9 can't be a member of the possible tree sets. This constraint restricts the number of possible tree sets to 27:

1. $p0, p1_a, p2_a, p3_a, r1_a$
2. $p0, p1_a, p2_a, p3_a, r1_b$
3. $p0, p1_a, p2_a, p3_a, r1_c$
4. $p0, p1_a, p2_a, p3_b, r1_a$
5. $p0, p1_a, p2_a, p3_b, r1_b$
6. $p0, p1_a, p2_a, p3_b, r1_c$
7. $p0, p1_a, p2_a, p3_c, r1_a$
8. $p0, p1_a, p2_a, p3_c, r1_b$
9. $p0, p1_a, p2_a, p3_c, r1_c$
10. $p0, p1_b, p2_a, p3_a, r1_a$
11. $p0, p1_b, p2_a, p3_a, r1_b$
12. $p0, p1_b, p2_a, p3_a, r1_c$
13. $p0, p1_b, p2_a, p3_b, r1_a$
14. $p0, p1_b, p2_a, p3_b, r1_b$
15. $p0, p1_b, p2_a, p3_b, r1_c$
16. $p0, p1_b, p2_a, p3_c, r1_a$
17. $p0, p1_b, p2_a, p3_c, r1_b$
18. $p0, p1_b, p2_a, p3_c, r1_c$
19. $p0, p1_c, p2_a, p3_a, r1_a$
20. $p0, p1_c, p2_a, p3_a, r1_b$
21. $p0, p1_c, p2_a, p3_a, r1_c$
22. $p0, p1_c, p2_a, p3_b, r1_a$
23. $p0, p1_c, p2_a, p3_b, r1_b$
24. $p0, p1_c, p2_a, p3_b, r1_c$
25. $p0, p1_c, p2_a, p3_c, r1_a$
26. $p0, p1_c, p2_a, p3_c, r1_b$
27. $p0, p1_c, p2_a, p3_c, r1_c$

- Syntactically simpler expressions of embedding are preferred over more complex ones

There are two syntactic options available for embedding in this example: an adjective or a relative clause. Since adjectives are syntactically simpler than relative clauses we exclude tree sets that contain a relative clause for 'new' and keep the ones that contain adjectival trees. This leaves us with the following 18 tree sets:

1. $p0, p1_a, p2_a, p3_a, r1_a$
2. $p0, p1_a, p2_a, p3_a, r1_b$
3. $p0, p1_a, p2_a, p3_a, r1_c$
4. $p0, p1_a, p2_a, p3_b, r1_a$
5. $p0, p1_a, p2_a, p3_b, r1_b$
6. $p0, p1_a, p2_a, p3_b, r1_c$
7. $p0, p1_b, p2_a, p3_a, r1_a$
8. $p0, p1_b, p2_a, p3_a, r1_b$
9. $p0, p1_b, p2_a, p3_a, r1_c$
10. $p0, p1_b, p2_a, p3_b, r1_a$
11. $p0, p1_b, p2_a, p3_b, r1_b$
12. $p0, p1_b, p2_a, p3_b, r1_c$
13. $p0, p1_c, p2_a, p3_a, r1_a$
14. $p0, p1_c, p2_a, p3_a, r1_b$
15. $p0, p1_c, p2_a, p3_a, r1_c$
16. $p0, p1_c, p2_a, p3_b, r1_a$
17. $p0, p1_c, p2_a, p3_b, r1_b$
18. $p0, p1_c, p2_a, p3_b, r1_c$

This constraint will obviously need to be revised as it would prevent the

generation of relative clauses in many cases (whenever the proposition can also be expressed by an adjective), clearly an unwanted result.

- The matrix proposition has to be the earliest occurring candidate in the nucleus.

Elaboration is usually represented as a relation between two propositions: elaboration('my car is a honda', 'my car is new'). In the above representation elaboration is a relation between the shared element in the nucleus and satellite in the traditional RST representation ('my car') and the added information ('new'). But there are still two candidates for embedding, as there are two NP nodes where the NP "my car" can substitute. This corresponds to the choice between embedding 'new' in either 'My car is a Honda' or 'My car is Japanese'.

So we can reformulate the constraint as: substitute the shared element (the first argument of the elaboration relation) into the first available substitution site.

We may not be able to check whether this constraint is fulfilled until the full (or possibly a partial) derived tree is constructed.

### 3.3.3 Combining Trees

We assume that only sentential trees can substitute into a $T_s$ or $T_c$ node therefore $p1_b$ and $p1_c$ cannot combine with the elementary trees of the evidence relation ($r1_a$, $r1_b$, or $r1_c$), and the derivation of these tree sets will be eliminated by polarity filtering (see section 2.5.3).

This leaves the following 6 options:

1. $p0, p1_a, p2_a, p3_a, r1_a$
2. $p0, p1_a, p2_a, p3_a, r1_b$
3. $p0, p1_a, p2_a, p3_a, r1_c$
4. $p0, p1_a, p2_a, p3_b, r1_a$
5. $p0, p1_a, p2_a, p3_b, r1_b$
6. $p0, p1_a, p2_a, p3_b, r1_c$

As there are two NP substitution nodes that take the semantic variable $x$ as argument and we have only one NP "my car", one of the positions will be empty after tree combination. The work to fill in this empty argument position should be left for a module that handles referring expressions and picks an appropriate pronoun.

Assuming a simple module that prohibits pronominalisation of referring expressions on the first mention and simply inserts a pronoun for their subsequent mentions, the results of tree composition for each tree set are the following sentences (Fig. 11).

(31) a $[p0, p1_a, p2_a, p3_a, r1_a]$: My new car is Japanese, since it is a Honda.

b $[p0, p1_a, p2_a, p3_a, r1_b]$: Since my new car is a Honda, it is Japanese.

c $[p0, p1_a, p2_a, p3_a, r1_c]$: My new car is a Honda. Therefore it is Japanese.

d $[p0, p1_a, p2_a, p3_b, r1_a]$:
i) My car is new. It is Japanese, since it is a Honda.
ii) My car is Japanese, since it is a Honda. It is new.

e $[p0, p1_a, p2_a, p3_b, r1_b]$
i) My car is new. Since it is a Honda, it is Japanese.
ii) Since my car is a Honda, it is Japanese. It is new.

f $[p0, p1_a, p2_a, p3_b, r1_c]$
i) My car is a Honda. Therefore it is Japanese. It is new.
ii) My car is new. It is a Honda. Therefore it is Japanese.

### 3.3.4 Ranking constraints for generated texts

Applying the relevant ranking constraints of Power et al. (2003) to the output we get the following ordering:

(32) i $[p0, p1_a, p2_a, p3_a, r1_b]$: Since my new car is a Honda, it is Japanese.

ii $[p0, p1_a, p2_a, p3_a, r1_c]$: My new car is a Honda. Therefore it is Japanese.

iii $[p0, p1_a, p2_a, p3_b, r1_b]$ (1 defect)
i) My car is new. Since it is a Honda, it is Japanese.
ii) Since my car is a Honda, it is Japanese. It is new.

iv $[p0, p1_a, p2_a, p3_b, r1_c]$ (1 defect)
i) My car is a Honda. Therefore it is Japanese. It is new.
ii) My car is new. It is a Honda. Therefore it is Japanese.

v $[p0, p1_a, p2_a, p3_a, r1_a]$: (2 defects) My new car is Japanese, since it is a Honda.

vi $[p0, p1_a, p2_a, p3_b, r1_a]$: (3 defects)
i) My car is new. It is Japanese, since it is a Honda.
ii) My car is Japanese, since it is a Honda. It is new.

Here each generated text is considered to have a defect if:

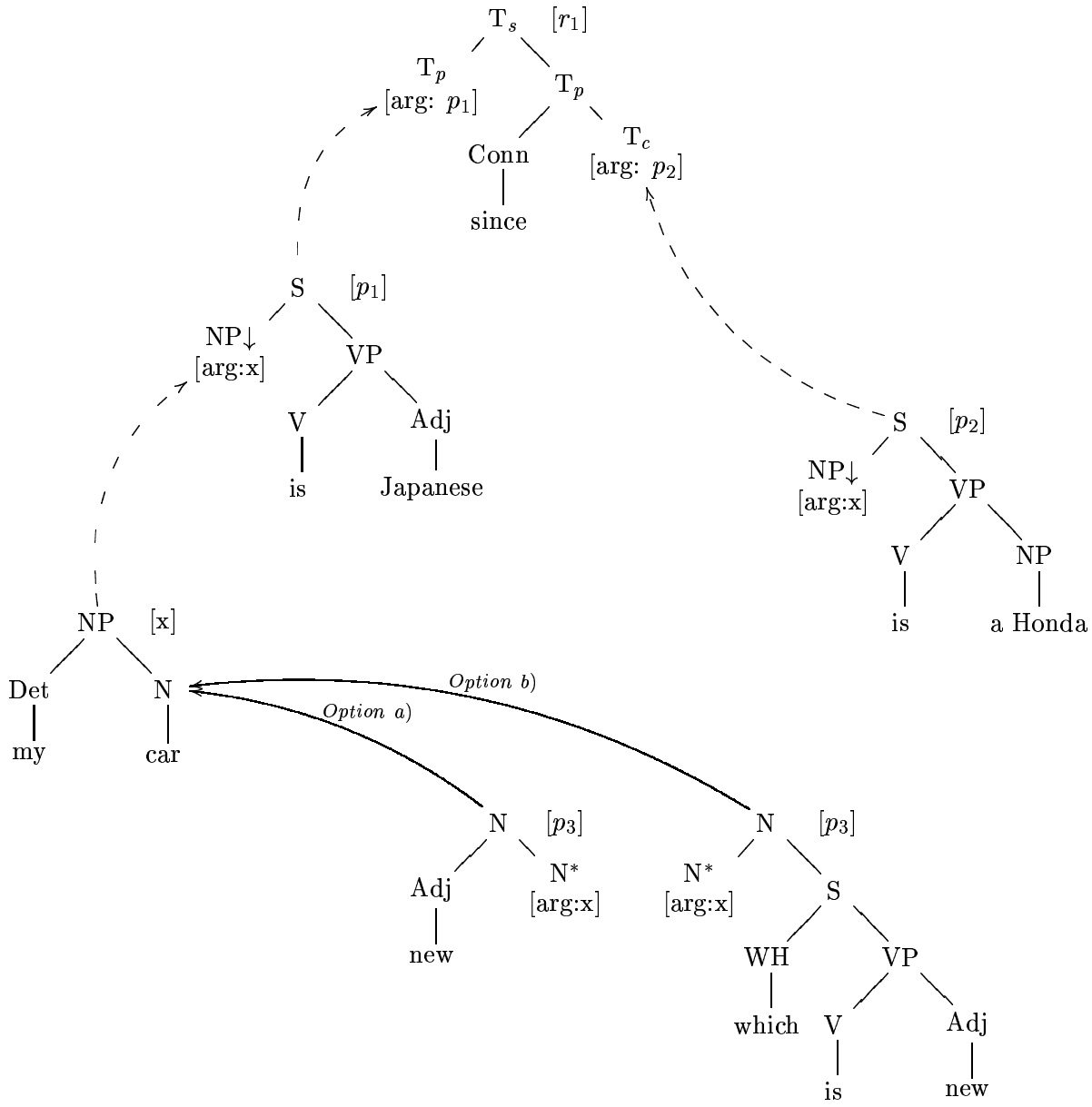



Figure 11: Combining tree sets $[p0, p1_a, p2_a, p3_a, r1_a]$ and $[p0, p1_a, p2_a, p3_b, r1_a]$

- a nucleus appears before its satellite: a, d i), d ii)
- the generated structure is left branching: a, d i), d ii)
- a single sentence paragraph is generated: d, e , f (i and ii)

## 3.4 Research questions:

### 3.4.1 Theoretical questions

- **Elementary Trees for Document Structure**
  What Document Structure elementary trees do we need for individual discourse connectives? What is the syntactic status of smaller document units (e.g., text sentence, text phrase)?
- **What are the properties of embedding?** What types of interpolations can we distinguish and what are their syntactic, semantic and rhetorical properties? Are there any constraints on what constituents can be embedded? Is there a limit on the size or document level of the unit? In what contexts is interpolation preferred? Is embedding allowed with multinuclear rhetorical relations or does it imply rhetorical subordination? What is the most common placement for embedded units?
- **What representation do we need for lists?** What elementary trees are required to represent lists? What types of list items do we need (i.e., what is the syntactic type of the bulleted constituents)? What's the syntax of the list introduction, the constituent that typically precedes the set of bullet points? What verbs can anchor list introduction trees? In the above representation, arguments of List relation are not localized within the elementary tree of the relation. Can this be a problem?
  Other related questions include: How to represent embedded lists? Is it enough to simply substitute another list initial tree into a list item and read off levels of embedding (indentation) the derivation tree?

The above theoretical questions will be answered by carrying out a corpus study on interpolations using the Penn Discourse Treebank, and consulting style guides.

### 3.4.2 NLG System Design

- **Constraints on tree selection:** The hypotheses described in Scott and Souza (1990) can be rephrased as constraints on tree selection. What additional constraints do we need? How can the results of the corpus study on embedded constructions be rephrased as constraints on possible tree sets?

- **Ranking constraints for generated texts:** Power et al. (2003) describe constraints to rank the output of a document structurer module. Some of these constraints are expected not to be violated at all because the requirements they impose on the document and rhetorical structure are actually built into the Document Structure TAG trees.Which constraints are applicable in this framework what additional constraints are needed based on the corpus study?

### 3.4.3 Evaluation

Evaluation of natural language generation systems is a challenging task and currently a hot topic in NLG. Evaluating the quality of human writing has always been a controversial task and since the generation goals and domains of different implementations are so varied, there are no "gold standard" input and output test sets that can be used for qualitative evaluation. The wide range of position papers at a recent Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation[14] illustrate the different opinions and the fact that this question is still a very much evolving central research objective in the field.

An expected contribution of this thesis is the design of an evaluation procedure for a system that generates many different versions of text from the same input semantic representation.

This is necessary because the focused goal of this PhD work relates to a problem that has to date, not been evaluated. For example, one of the most current, extensive and widely-cited NLG systems that focus on style (ICONOCLAST) has not been evaluated at all (Power et al., 2003).

Indeed, very few NLG systems have been evaluated (see papers from the above mentioned workshop) and those that have, focus on particular issues, e.g., whether readers stop smoking (Reiter et al., 2003) or whether users can successfully use the NLG system to compose queries (Hallett et al., 2007). Clearly, none of these are suitable for the task at hand.

Of course it is important to keep abreast of developments in evaluation of NLG systems and should any appropriate evaluation regimes for this research emerge, they will certainly be adopted.

In the meantime, several questions need to be answered: how do we evaluate the quality of the generated text other than using human judgement? how do we know that embedded constructions have been generated at the right place, in the right context? How do we know if the right type of embedding

[14] http://www.ling.ohio-state.edu/ mwhite/nlgeval07/accepted.html

has been generated? How do we evaluate the ranking of the output? How do we determine what is the best solution?

Human judgement will be an important part of evaluation, but additional methods will also be investigated. For example one way to evaluate the system would be to collect news from different sources that describe the same event, and create a logical form that represents the content of the news. This logical form can then be used as an input to the generation system and the output can be compared to the actual news stories.

For example given the semantic representation in (33) about the September 11th attacks on the U.S the output of the system could potentially be compared to the following news stories:

(33) e1: crashed(plane1, wtc_tower1),
e2: crashed(plane2, wtc_tower2),
e3: crashed(plane3, pentagon),
e4: crashed(plane4, near_pittsburgh)
p1: hijacked(plane1),
p2: hijacked(plane2),
p3: hijacked(plane3),
p4: hijacked(plane4),
p5: terrorist_attack(e1),
p6: terrorist_attack(e2),
p7: terrorist_attack(e3),
p8: terrorist_attack(e4),
p9: destroyed(e1, wtc),
p10: destroyed(e2, wtc)

**Boston Globe:** NEW YORK - The twin towers of the World Trade Center were destroyed this morning after two airplanes crashed into them, in what seems to have been the largest terrorist attack in history. The raid, followed by a plane that crashed into the Pentagon and another plane crash outside Pittsburgh, prompted comparisons to the Japanese attack on Pearl Harbor in 1941.

**US Dept.of Defense:** Between 9 and 10 a.m., two apparently hijacked commercial aircraft hit both World Trade Center towers in New York City, while a third airliner crashed into an outside wall of the Pentagon. Both 100-story trade center buildings collapsed within two hours. A fourth airliner, also believed to have been hijacked, crashed near Pittsburgh in the same time frame.

**ABC:** Four U.S. passenger planes were apparently hijacked and crashed - two jets flew into the twin towers of the World Trade Canter in New

York City, a plane struck the Pentagon and a plane crashed near Pittsburgh.

**WashingtonPost:** President Bush placed the military on "high alert" today and returned to Washington to address a nation deeply shaken by devastating terrorist attacks that destroyed the World Trade Center towers, seriously damaged the Pentagon and killed all those aboard four commercial airliners.

**Canada.com:** America under terrorist attack Several acts of terrorism have wreaked havoc on the United States.

* Two planes crash into World Trade Center
* Plane crashes into Pentagon
* Plane crashes near Pittsburgh

**CNN:** At 8:45 a.m. EDT, the first of two airliners crashed into the World Trade Center, opening a horrifying and apparently coordinated terrorist attack on the United States, which saw the collapse of the two 110-story towers into surrounding Manhattan streets and a later attack on the Pentagon.

**Yahoo!News:** Three hijacked planes crashed into major U.S. landmarks on Tuesday, destroying New York's World Trade Center and plunging the Pentagon in Washington into flames, in an unprecedented assault on key symbols of U.S. military and financial power.

**Reuters:** In the worst attack on American soil since Pearl Harbor, three hijacked planes slammed into the Pentagon and New York's landmark World Trade Center on Tuesday, demolishing the two 110-story towers that symbolize U.S. financial might.

**NYT on the Web:** In parallel attacks in New York City and Washington, planes crashed into each of the twin towers of the World Trade Center around 9 this morning and a plane later crashed into the Pentagon, causing smoke, fire and a sense of near panic in the streets.

This input representation is very rudimentary of course and only serves to illustrate the point. A lot more detail will need to be included in the input in order for the system to be able to generate news-quality text. We will also have to make sure that the exact same content is mentioned in the news, for example not all of the news items in this example mention the fourth plane crash near Pittsburgh.

Another way to evaluate the quality of the output is to parse the output text and look at the frequency of the generated syntactic constructions (in the given rhetorical context) in a corpus. The generation system performs well according to this evaluation if it is capable of generating the most frequently occurring constructions for the input rhetorical relation.

For the evaluation of ranking constraints, statistical tools could be used that show a measure of some linguistic or rhetorical property of the text, e.g., coherence. In this case the ranking constraints work well if the ordering of the generated text corresponds to the ordering of the values established by the statistical tool. A similar approach is taken by e.g., Mutton et al. (2007) when they define the GLUE fluency measure.

# 4 Work plan

| | | Work plan | Contingency |
|---|---|---|---|
| 2007 | May June July August September | **WP1:** *Objective:* To design trees for basic document units and implement a preliminary version of the system. *Milestone:* A preliminary implementation of an NLG system, which includes a grammar for basic document structure trees and a framework to define constraints on selection of TAG trees | 1 month |
| 2007 | October November December January February | **WP2:** *Objective:* To carry out a corpus study and consult style guides to define a grammar for interpolations. The task is to determine the syntactic, semantic and rhetorical properties of interpolations, establish subclasses and contexts in which they occur. *Milestone:* A set of grammatical rules which describe the usage of interpolations in the corpus | 2 weeks |
| 2008 | March April May | **WP3:** *Objective:* To analyse the results of the corpus study and come up with a TAG for interpolations *Milestone:* A grammar for interpolated constructions | 1 month |
| 2008 | June July August September | **WP4:** *Objective:* To implement the grammar and test the system. Refine the grammar and implementation if necessary until generated text is satisfactory *Milestone:* An enhanced NLG system which is now capable of generating interpolations | |
| 2008 | October November December | **WP5:** *Objective:* To evaluate the system. Design of an appropriate evaluation procedure will be an on-going process. *Milestone:* An evaluation regime which can be applied to a system that generates parenthetical constructions; a result of this evaluation. | 2 weeks |
| 2009 | January February March April May June July August September | **WP6:** *Objective:* To write up dissertation, submit to advisers and do revisions *Milestone:* A document which has the approval of the supervisors to be submitted as a dissertation | 1 month |
| | October | submit dissertation | |

## References


Abeille, Anne, and Owen Rambow. 2000. Tree Adjoining Grammars: An overview. In *Tree Adjoining Grammars: Formalisms, linguistic analysis and processing*, ed. Anne Abeille and Owen Rambow. CSLI Publications, Stanford, CA.

Appelt, D.E. 1985. *Planning English sentences.* Cambridge: Cambridge University Press.

Bangalore, Srinivas, and Owen Rambow. 2000. Using TAGs, a tree model, and a language model for generation. In *Proceedings of TAG+5*. Paris.

Bateman, John, and Elke Teich. 1995. Selective information presentation in an integrated publication system: an application of genre-driven text generation. *Information Processing Management* 31:753–767.

Becker, Tilman. 2002. Practical, template-based natural language generation with TAG. In *Proceedings of TAG+6*. Venice, Italy.

Blakemore, Diane. 1987. *Semantic constraints on relevance.* Blackwell, Oxford.

Blakemore, Diane. 2006. Divisions of labour. the analysis of parentheticals. *Lingua* 116:1670–1687.

Bonfante, Guillaume, Bruno Guillaume, and Guy Perrier. 2004. Polarization and abstraction of grammatical formalisms as methods for lexical disambiguation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, 303. Morristown, NJ, USA: Association for Computational Linguistics.

Burton-Roberts, Noel. 1975. Nominal apposition. *Foundations of Language* 13:391419.

Burton-Roberts, Noel. 1999a. Apposition. In *The concise encylcopaedia of syntactic categories*, ed. E. K. Brown and J. Miller. Elsevier.

Burton-Roberts, Noel. 1999b. Language, linear precedence and parentheticals. In *The clause in english*, ed. Peter Collins and David Lee, 33–52. Amsterdam/ Philadelphia: John Benjamins.

Burton-Roberts, Noel. 2005. Parentheticals. In *Encyclopaedia of language and linguistics*, ed. E. K. Brown. Elsevier Science, 2nd edition edition.

Cahill, Lynne, and Mike Reape. 1998. Component tasks in applied NLG systems. Technical Report ITRI-99-05, ITRI, University of Brighton.

Carroll, J., and S. Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. *2nd IJCNLP* .

Carroll, John, Ann Copestake, Dan Flickinger, and Victor Poznanski. 1999. An efficient chart generator for (semi-)lexicalist grammars. In *Proceedings of the 7th European Workshop on Natural Language Generation (EWNLG'99)*, 86–95. Toulouse, France.

Copestake, Ann, Dan Flickinger, and Ivan A. Sag. 1999. Minimal recursion semantics. an introduction. Draft, September 1999.

Danlos, Laurence. 1984. Conceptual and linguistic decisions in generation. In *Proceedings of the 10th International Conference on Computational Linguistics*, 319–25. Stanford, CA.

Dehe, Nicole, and Yordanka Kavalova. 2006. The syntax, pragmatics, and prosody of parenthetical what. *English Language and Linguistics* 10:289–320.

Dehe, Nicole, and Yordanka Kavalova, ed. 2007. *Parentheticals*, chapter Parentheticals: An introduction, 1–22. Linguistik aktuell Linguistics today 106. Amsterdam Philadelphia: John Benjamins.

Emonds, Joseph. 1979. Appositive relatives have no properties. *Linguistic Inquiry* 10:211–243.

Espinal, Teresa M. 1991. The representation of disjunct constituents. *Language* 67:726–762.

Frank, Robert. 1992. Syntactic locality and Tree Adjoining Grammar: Grammatical, acquisition and processing perspectives. Doctoral Dissertation, University of Pennsylvania.

Frank, Robert, Seth Kulick, and K. Vijay-Shanker. 1999. C-command and extraction in Tree Adjoining Grammar. In *Proceedings of the 6th Mathematics of Language Conference*. Orlando, USA.

Gardent, Claire, and Eric 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.

Gardent, Claire, and Stefan Thater. 2001. Generating with a grammar based on tree descriptions: a constraint-based approach. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, 212–219. Morristown, NJ, USA: Association for Computational Linguistics.

Grosz, B.J., A.K. Joshi, and S Weinstein. 1995. Centering: a framework for modelling the local coherence of discourse. *Computational Linguistics* 21:203–225.

Haegeman, J. 1991. *Introduction to government and binding theory*. Oxford (UK) and Cambridge (USA): Blackwell.

Haegeman, Liliane. 1988. Parenthetical adverbials: the radical orphanage approach. In *Aspects of modern linguistics: Papers presented to masatomo ukaji on his 60th birthday*, ed. S. Chiba et al, 232–254. Tokyo: Kaitakushi.

Hallett, Catalina, Donia Scott, and Richard Power. 2007. Composing questions through conceptual authoring. *Comput. Linguist.* 33:105–133.

Harbusch, Karin, and Jens Woch. 2002. Integrated natural language generation with schema-tree adjoining grammars. In *CICLing*, 304–313.

Hitzeman, Janet, Chris Mellish, and Jon Oberlander. 1997. Dynamic generation of museum web pages: the intelligent labelling explorer. *Archives and Museum Informatics* 11:105–112. Citeseer.ist.psu.edu/hitzeman97dynamic.html.

Hovy, E. 1988. *Generating Natural Language Under Pragmatic Constraints*. Hillsdale, New Jersey: Lawrence Erlbaum.

Hovy, E. 1990. Pragmatics and Natural Language Generation. *Artificial Intelligence* 43:153–197.

Hovy, Eduard H. 1993. Automated discourse generation using discourse structure relations. *Artificial Intelligence* 63:341–385.

Joshi, Aravind K. 1983. How much context-sensitivity is required to provide reasonable structural descriptions: Tree Adjoining Grammar. In *Natural language processing: Psycholinguistic, computational and theoretical perpectives*, ed. L. Karttunen D. Dowty and A. Zwicky. New York, Cambridge University Press.

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

Joshi, Aravind K., L. S. Levy, and M. Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences* 10:136–163.

Joshi, Aravind K., and Yves Schabes. 1997. Tree-Adjoining Grammars. In *Handbook of formal languages and automata*, ed. Grzegorz Rosenberg and Arto Salomaa, volume 3, 69–124. Springer-Verlag, Heidelberg.

Joshi, Aravind K., and K. Vijay-Shanker. 1999. Compositional semantics with Lexicalized Tree Adjoining Grammar (LTAG): How much underspecification is necessary? In *Proceedings of the Third International Workshop on Computational Semantics (IWCS-3)*, ed. H.C.Bunt and E.G.C. Thijsse, 131–145. Tilburg.

Kallmeyer, Laura, and Aravind Joshi. 1999. Factoring predicate argument and scope semantics: Underspecified semantics with LTAG. In *Proceedings of the twelfth Amsterdam colloquium*, ed. Paul Dekker, 169–174. Amsterdam: ILLC/Department of Philosophy, University of Amsterdam.

Kallmeyer, Laura, and Aravind Joshi. 2003. Factoring predicate argument and scope semantics: Underspecified semantics with LTAG. *Research on Language and Computation* 1:1-2:3–58.

Kantrowitz, M., and J. Bates. 1992. Integrated natural language generation systems. In *Aspects of automated natural language generation*, ed. R. Dale, E. Hovy, D. Rösner, and O. Stock, 13–28. Berlin: Springer.

Kaplan, R. M., and J. Bresnan. 1982. Lexical-functional grammar: A formal system for grammatical representation. In *The mental representation of grammatical relations*, ed. J. Bresnan, 173–281. Cambridge, MA: MIT Press.

Kavalova, Yordanka. 2007. And-parenthetical clauses. In *Parentheticals*, ed. Nicole Deh and Yordanka Kavalova, Linguistik aktuell Linguistics today 106, 1–22. Amsterdam Philadelphia: John Benjamins.

Kay, Martin. 1996. Chart generation. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, ed. Arivind Joshi and Martha Palmer, 200–204. San Francisco: Morgan Kaufmann Publishers.

Kayne, Richard. 1994. *The antisymmetry of syntax*. The MIT Press.

Kibble, Rodger, and Richard Power. 2004. Optimizing referential coherence in text generation. *Computational Linguistics* 30:401–416.

Kintsch, W. 1998. *Comprehension: A paradigm for cognition*. New York: Cambridge University Press.

Koenraad De Smedt, Michael Zock, Helmut Horacek. 1996. Architectures for natural language generation: Problems and perspectives. In *Trends in natural language generation –an artificial intelligence perspective*, ed. M. Zock G. Adorni, 17–46. Springer, Berlin, Heidelberg.

Kroch, Anthony. 1989. Asymmetries in long distance extraction in a Tree Adjoining Grammar. In *Alternative conceptions of phrase structure*, ed. M. Baltin and A. Kroch, 66–98. University of Chicago Press.

Kroch, Anthony, and Aravind K. Joshi. 1985. The linguistic relevance of Tree Adjoining Grammar. Technical Report MS-CS-85-16, Department of Computer and Information Sciences, University of Pennsylvania.

Kroch, Anthony, and Aravind K. Joshi. 1987. Analyzing extraposition in a Tree Adjoining Grammar. Technical Report MS-CS-85-16, Department of Computer and Information Sciences, University of Pennsylvania.

Lorch, R. F. J., E. P. Lorch, and W. E. Inman. 1993. Effects of signaling topic structure on text recall. *Journal of Educational Psychology* 85:281–290.

Mann, W. C. 1988. *Text generation: the problem of text structure*, 47–68. New York, NY, USA: Springer-Verlag New York, Inc.

Mann, William C., and Sandra A. Thompson. 1986. Rhetorical structure theory: Description and construction of text sructures. Technical Report ISI/RS-86-174, Information Sciences Institute.

Mann, William C., and Sandra A. Thompson. 1987. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, Information Sciences Institute.

Mann, William C., and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text* 8:243–281.

McCawley, James D. 1982. Parentheticals and discontinuous constituent structure. *Linguistic Inquiry* 13:91–106.

McCoy, Kathleen F., K. Vijay-Shanker, and Gijoo Yang. 1992. A functional approach to generation with TAG. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics*, 48–55. Morristown, NJ, USA: Association for Computational Linguistics.

McDonald, David D., and James D. Pustejovsky. 1985. TAGs as a grammatical formalism for generation. In *Proceedings of the 23rd annual meeting on Association for Computational Linguistics*, 94–103. Morristown, NJ, USA: Association for Computational Linguistics.

Mellish, C., A. Knott, J. Oberlander, and M. O'Donnell. 1998. Experiments using stochastic search for text planning. In *Proceedings of the Ninth International Workshop on Natural Language Generation*. Niagara-on-the-lake, Ontario.

Mellish, Chris, Donia Scott, Lynne Cahill, Daniel Paiva, Roger Evans, and Mike Reape. 2006. A reference architecture for natural language generation systems. *Natural Language Engineering* 12:1–34.

Meteer, Marie Wenzel. 1990. The "generation gap": the problem of expressibility in text planning. Doctoral Dissertation, University of Massachusetts, Amherst, MA, USA.

Miltsakaki, Eleni, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2004a. Annotating discourse connectives and their arguments. In *Proceedings of the HLT/NAACL Workshop on Frontiers in Corpus Annotation*. Boston, MA.

Miltsakaki, Eleni, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2004b. The penn discourse treebank. In *Proceedings of the Language Resources and Evaluation Conference*. Lisbon, Portugal.

Moore, Johanna D., and Martha E. Pollack. 1992. A problem for rst: the need for multi-level discourse analysis. *Comput. Linguist.* 18:537–544.

Mutton, Andrew, Mark Dras, Stephen Wan, and Robert Dale. 2007. Gleu: Automatic evaluation of sentence-level fluency. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 344–351. Prague, Czech Republic: Association for Computational Linguistics.

N. Bouayad-Agha, R. Power, D. Scott. 2000. Integrating content and style in documents: a case study of patient information leaflets. *Information Design Journal* 9:161–176.

Nunberg, Geoff. 1990. The linguistics of punctuation. Technical Report CSLI Lecture Notes, No. 18., Stanford: Center for the Study of Language and Information.

Paiva, Daniel. 1998. A survey of applied natural language generation systems. Technical Report ITRI-98-3, University of Brighton, Information Technology Research Institute.

Paiva, Daniel S. 2004. Using stylistic parameters to control a natural language generation system. Doctoral Dissertation, ITRI, University of Brighton.

Paris, C., K. Vander Linden, N. Colineau, and S. Lu. 2003. Producing instructions. In *Paris, C., Vander Linden, K., Colineau, N. and Lu, S. (2003). Producing Instructions. In the Proceedings of the Workshop on Technologies for Electronic Documents for Supporting Learning, AI-Ed 2003*, 653 – 663. Sydney, Australia.

Perrier, Guy. 2003. Les grammaires d'interaction. Doctoral Dissertation, Université Nancy2.

Piwek, Paul, and Kees van Deemter. 2006. Constraint-based natural language generation: A survey. Technical Report 2006/03, Department of Computing, The Open University.

Pollard, Carl J., and Ivan A. Sag. 1994. *Head-driven phrase structure grammar*. Chicago: University of Chicago Press.

Power, R. 2000. Planning texts by constraint satisfaction. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, 642–648. Saarbrücken, Germany.

Power, R., and D. Scott. 1998. Multilingual authoring using feedback texts. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, 1053–1059. Montreal, Canada.

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

Power, R., D. Scott, and R. Evans. 1998. What you see is what you meant: direct knowledge editings with natural language feedback. In *13th european conference on artificial intelligence (ECAI'98)*, ed. H. Prade, 677–681. Chichester, England: John Wiley and Sons.

R. Power, D. Scott, C. Doran. 2000. Generating embedded discourse markers from rhetorical structure. In *Proceedings of the 7th European Workshop on Natural Language Generation (EWNLG'99)*, 30–38.

Reiter, Ehud. 1994. Has a consensus NL generation architecture appeared, and is it psychologically plausible? In *Proceedings of the 7th. International Workshop on Natural Language generation (INLGW '94)*, ed. David McDonald and Marie Meteer, 163–170. Kennebunkport, Maine.

Reiter, Ehud, Roma Robertson, and Liesl M. Osman. 2003. Lessons from a failure: generating tailored smoking cessation letters. *Artificial Intelligence* 144:41–58.

Sag, Ivan A. 1997. English relative clause constructions. *Journal of Linguistics* 33:431–484.

Scott, D., and C. S. Souza. 1990. Getting the message across in rst-based text generation. In *Current research in natural language generation*, ed. C. Mellish R. Dale M. Zock, 31–56. Academic Press.

Shieber, Stuart M., and Yves Schabes. 1990. Generation and synchronous Tree-Adjoining Grammars. In *5th. International Workshop on Natural Language Generation, 3-6 June 1990*. Pittsburgh, PA.

Snowling, Margaret J., and Charles Hulme, ed. 2005. *The science of reading: A handbook*, chapter "Part III : Reading Comprehension. Blackwell Publishing,.

Steedman, Mark. 1996. *Surface structure and interpretation*, volume 30 of *Linguistic Inquiry Monographs*. Cambridge, Massachusetts: MIT Press.

Stone, Matthew, and Christine Doran. 1997. Sentence planning as description using Tree Adjoining Grammar. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ed. Philip R. Cohen and Wolfgang Wahlster, 198–205. Somerset, New Jersey: Association for Computational Linguistics.

Stone, Matthew, Christine Doran, Bonnie Webber, Tonia Bleam, and Martha Palmer. 2003. Microplanning with communicative intentions: The spud system. *Computational Intelligence* 19:311–381.

Strunk, William. 1918. *The elements of style*. Ithaca, N.Y.: Priv. print. [Geneva, N.Y.: Press of W.P. Humphrey].

Strunk, William, Jr., and E. B. White. 1979. *The elements of style*. Macmillan, third edition.

Teich, Elke, and John A. Bateman. 1994. Towards an application of text generation in an integrated publication system. In *Proceedings of the Seventh International Workshop on Natural Language Generation, Kennebunkport, Maine, USA, June 21-24, 1994*, 153–162. Kennebunkport, Maine, USA.

Thorndike, Edward L. 1917. Reading as reasoning: a study of mistakes in paragraph reading. *Journal of Educational Psychology* 8:232–332.

Van Hentenryck, P. 1989. *Constraint satisfaction in logic programming*. The MIT Press.

Wanner, L., and E. Hovy. 1996. The healthdoc sentence planner. In *INLG'96*, 1–10. Herstmonceux Castle, Sussex.

Webber, Bonnie. 2004. D-LTAG: extending lexicalized TAG to discourse. *Cognitive Science* 28:751–779.

Webber, Bonnie, Alistair Knott, Matthew Stone, and Aravind Joshi. 1999. Discourse relations: a structural and presuppositional account using lexicalized TAG. In *Proceedings of the 37th. Annual Meeting of the American Association for Computational Linguistics (ACL'99)*, 41–48. University of Maryland.

Webber, Bonnie, Matthew Stone, Aravind Joshi, and Alistair Knott. 2003. Anaphora and discourse structure. *Computational Linguistics* 29:545–587.

Williams, Sandra. 2004. Natural language generation (nlg) of discourse relations for different reading levels. Doctoral Dissertation, University of Aberdeen.

XTAG-Group, The. 2001. A Lexicalized Tree Adjoining Grammar for English. Technical report, University of Pennsylvania, Philadelphia.

Ziv, Yael. 2002. This, i believe, is a processing instruction: Discourse linking via parentheticals. In *Proceedings of Israel Association for Theoretical Linguistics 18*, ed. Yehuda N. Falk. Bar Ilan University.