Natural Language Aggregation and Clarification using Cue Words

Hercules Dalianis

Department of Computer and Systems Sciences
The Royal Institute of Technology and
Stockholm University
Electrum 230, S-164 40 Kista
SWEDEN
ph (+46) 8 16 49 16
mob. ph. (+46) 70 568 13 59
fax. (+46) 8 703 90 25
email: hercules@dsv.su.se

Abstract

This paper describes how to clarify or avoid uninteded introduction of ambiguity in aggregated text. Aggregation is the process of removing redundant information in a text without losing any information. A side effect of aggregation is that the resulting text becomes ambiguous. Cue words is the solution on this problem. A method for automatically assigning the right cue word to the ambiguous and aggregated text is shown. This method is implemented in a prototype. One drawback of this method is that the cue words are not necessary in all cases.

1. Introduction

Aggregation is the process of removing redundant information in a text without losing any information. People perform aggregation all the time to make natural language expressions shorter, non-redundant, and easy to read, [Dalia97,96c]. Texts in all genres display evidence of aggregation as our corpus studies shows (see the Appendix). Aggregated texts sometimes need cue words e.g., *each, together, separately, both*, to clarify the aggregation. In the study we calculated the ratio cue words/sentences to be 2.0%, and the ratio (cue words)/(syntactic aggregation) to be 15% i.e., every seventh syntactic aggregation contains a cue word.

During the aggregation process an ambiguity can arise because of problems with quantifier scoping. This ambiguity can be resolved by using cue words For example:

John wrote an article.

Mary wrote an article

Aggregation: Predicate and direct object grouping =>
(Gives rise to four different interpretations).

John and Mary wrote an article.

OR

John and Mary wrote two articles.

OR

John and Mary wrote an article together.

OR

John and Mary wrote an article each.

The text can be aggregated in several ways. To determine the correct form the generator needs to be sensitive to the underlying semantics in order to insert cue words of the appropriate kind.

In this paper we analyze cue words associated to the *and* -coordination, their relationship to the underlying representation, and a method to clarify[1] or avoid the introduction of ambigiuities in the aggregated text is presented. The method is based on the use of cue words.

2. Previous research

The concept *cue words* is mentioned in [Quirk72], who calls them markers for coordination, for example, *both, each, either, neither*. According to [Quirk72] three types of coordinators are used for coordination : *and, or, but*. With each type of coordinator a set of cue words is used to clarify the coordination.

A parallel approach is the one within discourse structure theory, using cue words for signalling various discourse structures. [Hobbs90] recommends the use of conjunctions or sentential adverbs, to determine what type of discourse relation has been used. Hobbs says, though, that these conjunctions or sentential adverbs do not define the discourse relations. In [Hovy94], the authors go a step further and say that the use of a specific cue word signals that a specific discourse relation has been used, for example: *but, in order to, because, for example, first, second.*

3. Types of Cue Words in Aggregation

3.1. Overview

The cue words *each, together, separately, both* (*both* should be avoided because it may be ambiguous), are called by [Quirk72] markers of combined and segregatory coordination.

To solve the problem of ambiguity we need to define a set of cue primitives. Each cue primitive represents a possible interpretation of an aggregation in a text, i.e., a cue primitive clarifies an aggregated text. We use the term *cue primitive* instead of *cue word* to separate lexical choice from deep generation (lexical choice is not treated in this paper, however we still have to make some lexical choice and surface generation for demonstration and validation purposes).

The different classes of aggregation rules [Dalia96a,96b] are listed in Figure 1, below, with the rules discussed in the this paper listed in bold. (See [Dalia96b,96c] for definitions of the syntactic aggregation rules used in this paper).

In this paper we discuss only syntactic aggregation. Cue words are used with other types of aggregation but not always with the purpose of clarification. For example, in lexical aggregation, their purpose is to mark that a lexical aggregation is not complete but is missing one or more items [Dalia97].

---

[1] The term *disambiguation* seems more appropriate here, but it is already used by the natural language analysis community for parsing, therefore the term used here is *clarification*.

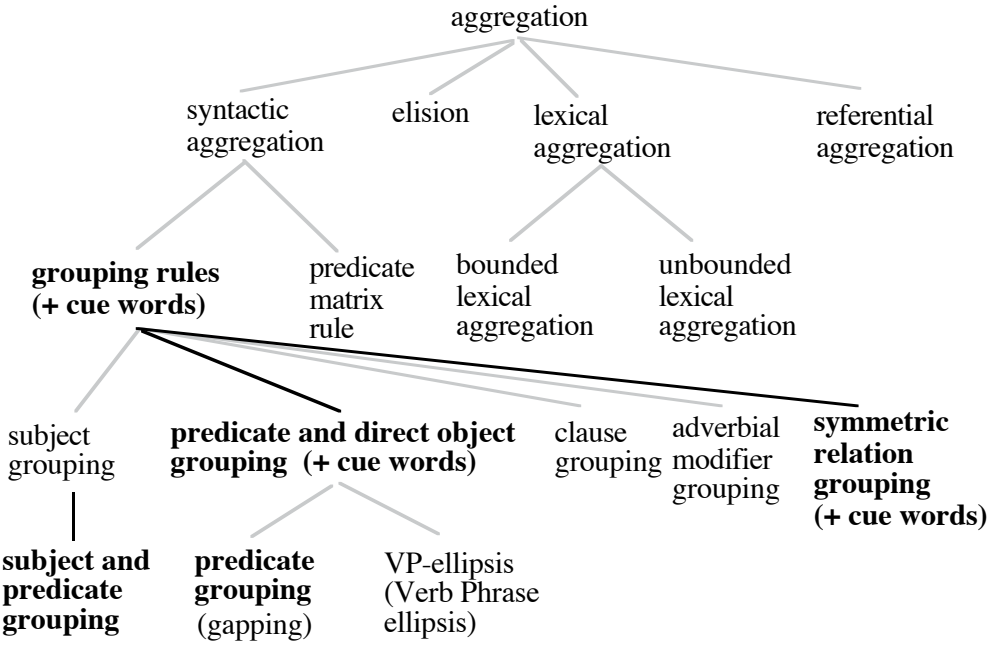


Figure 1. Hierarchical classification of aggregation types. The bold ones are discussed in this paper.

For syntactic aggregation we have defined a set of cue primitives to make it possible to analyse the different quantifier scopings. We differentiated two types of primitives, *joint* and *separate*, each with two subtypes, as shown in Figure 2 below.

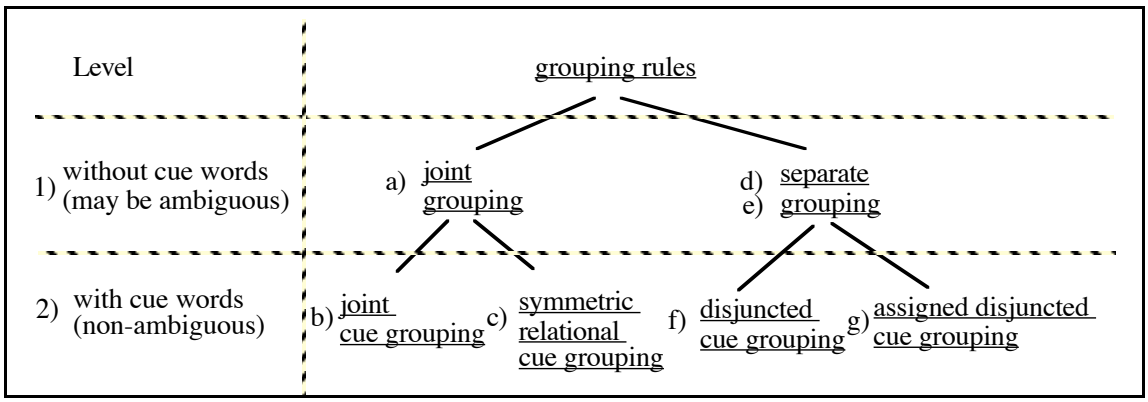


Figure 2. Cue primitives hierarchical tree. (The letter codes correspond to examples in Sections 3.2. and 3.3.)

Level 1 in Figure 2 above contains Joint and Separate grouping without cue words. Although at surface level the text is ambiguous, at deep level we can inspect whether the grouping is joint or separate. At level 2 we can see that the surface form has been augmented with cue words and is clarifed.

Figure 3 and 4 below contain instances of each type of cue primitive.

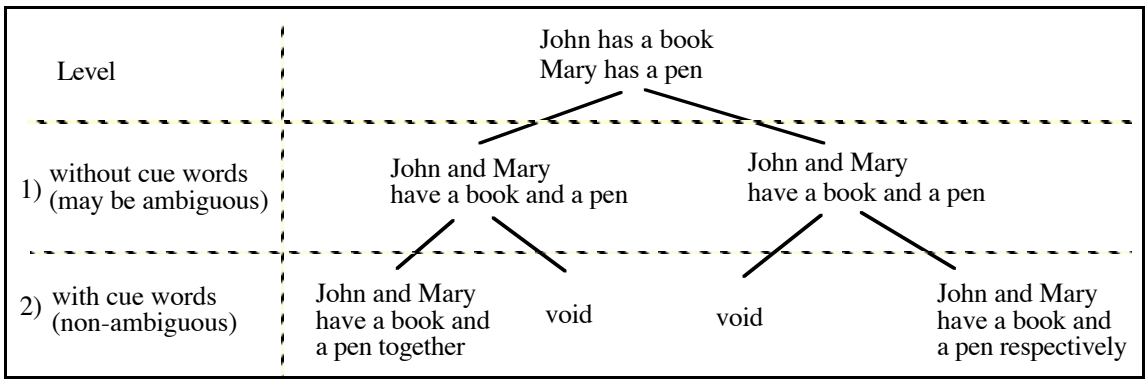


Figure 3. Instances of the cue primitives hierarchical tree (in the leaves notated "void" there is no instantiation of the example).

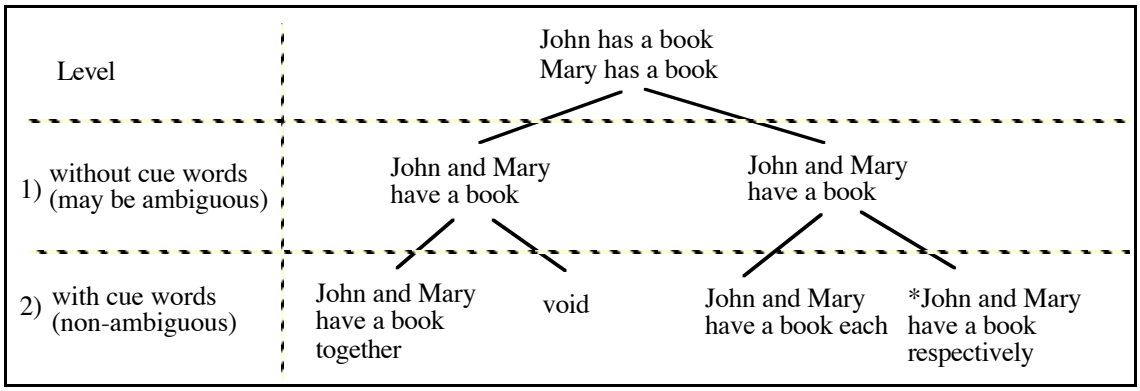


Figure 4. Instances of the cue primitives hierarchical tree (in the leaf notated "void" there is no instantiation of the example). The "respectively" example (marked with a "*") may not be correct English.

3.2. Joint grouping

Joint grouping occurs when two objects share a single instance. The aggregation rule: Predicate and Direct Object grouping (PDO grouping), triggers on *John has a book (s) and Mary has a book (s),* where they have the same book. (The grouping rules can apply over singular and plural instances)

Ex. **a)** *John and Mary have a book (s)*

(J+M) —poss→ book (s) Joint grouping

Clause a) is classified as separate grouping i.e. it has the meaning *John and Mary have a book (s) each.*, Joint grouping can be augmented with cue words in order to clarify the expression. Joint grouping is then divided into joint cue and relative cue grouping (see Level 2 in Figure 2).

3.2.1. Joint cue grouping

Joint cue grouping occurs when objects jointly have an instance and this is marked by a cue word. Aggregation rule: PDO grouping + cue primitive. Cue words: *together, jointly.*

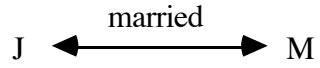
Ex **b)** *John and Mary have a book (s) together* , i.e., one interpretation of **a)** c.f. an other interpretation below in 3.3.1. Disjuncted Cue grouping. The choice of the cue word *together* could be replaced with a construction as *John and Mary share a book and a pen.*

(J+M) —poss→ book (s) Joint cue grouping

3.2.2. Symmetric relational cue grouping

Relative cue grouping occurs when two objects are related to each other symmetrically and this is marked with a cue phrase. Aggregation rule: Symmetric relation grouping + cue primitive. Triggers on *John is married to Mary and Mary is married to John*.

Ex **c)** *John and Mary are married to each other*.



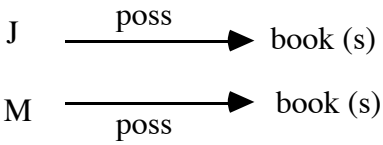
Symmetric relational cue grouping

The aggregated text without cue words may be ambiguous: *John and Mary are married* may either be joint or separate grouping. In most cases it would be joint grouping and pragmatically not ambiguous and therefore would not a cue word be needed, but in certain context with a lot of contrahents would it be necessary with a cue word to clarify the relation.

3.3. Separate grouping

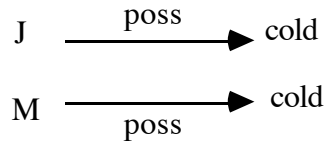
Separate grouping occurs when two objects have different instances of the same entity in common. The aggregation rule: PDO grouping + cue words triggers either on *John has a book (s) and Mary has a book (s)* or on *John has a cold and Mary has a cold*

Ex **d)** *John and Mary have a book (s)*



Separate grouping

Ex **e)** *John and Mary have a cold (s)* (no ambiguity is possible: you can't share a cold, therefore cue words are not necessary).



Separate grouping

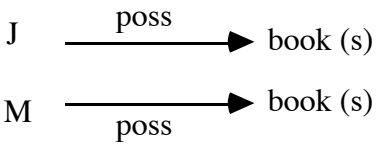
Separate grouping can be divided into disjuncted cue grouping and assigned disjuncted cue grouping

3.3.1. Disjuncted cue grouping

Disjuncted cue grouping occurs when two objects have two instances of one entity in common. Aggregation rule: PDO grouping + cue primitive. Cue word *each*.

Triggers on *John has a book (s) and Mary has a book (s)* .

Ex **f)** *John and Mary have a book (s) each* i.e. one other interpretation of **a)** compared to **b)** both above.

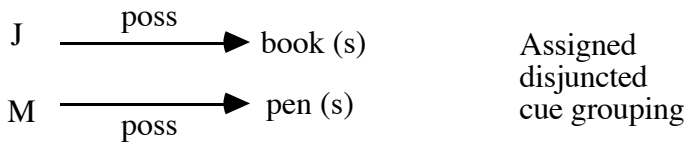


Disjuncted cue grouping

3.3.2. Assigned disjuncted cue grouping

Assigned disjuncted cue grouping occurs when two objects do not have any entity in common except a common relation. Aggregation rule: Predicate grouping + cue primitive. Cue word *respectively*.

Ex. **g)** *John and Mary have a book(s) and a pen(s) respectively .*



4. The Process of Selection of Cue Primitives for Clarification

4.1 Aggregation system architecture

Applying the aggregation rules and selecting cue primitive are interleaved processes. Once an aggregation rule triggers, the information used for the cue word selection disappears, unless we do not save this information.

To each aggregation rule a cue primitive selection is connected. The three aggregation rules with a cue primitive are the PDO-grouping rule, the Predicate grouping rule and the Symmetric relation grouping rule. We implement the aggregation rules to operate prior to surface form generation during the surface planning phase, as shown in Figure 4.

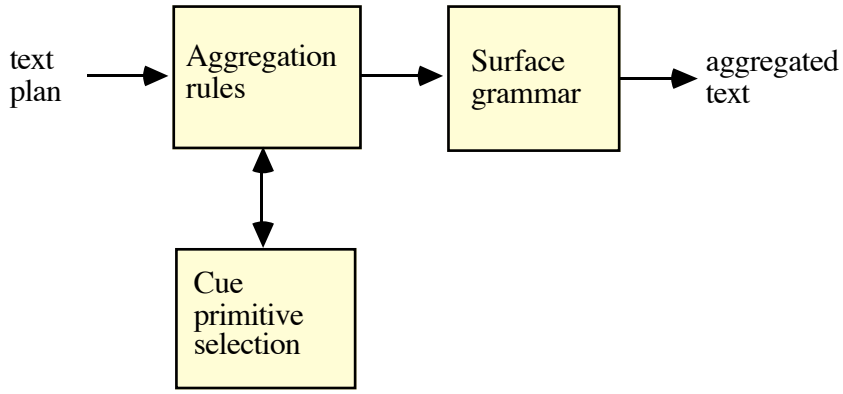


Figure 4. Aggregation system overview

As input to the Aggregation system a non-ambiguous input representation is needed. This means that every object and predicate in the knowledge base has to be augmented with a marker that distinguishes the different instances which has the same name. We assign a number to identify each instance. Certain combination of numbers are translated to one of the three types of cue primitive. The cue primitive selection must choose among the several different interpretations of each possible aggregation.

- Each clause with the same predicate and direct object and whose filler instances are numbered identically is aggregated. The number assigned to the predicate is translated to the *joint* cue primitive.
- Each clause with the same predicate and direct object but with different filler instances numbers is aggregated. The number assigned to the predicate is translated to the *disjuncted* cue primitive.
- Each clause with the same predicate but with different filler instances numbers are aggregated. The number is translated to the *assigned* cue primitive.
- The remaining objects with assigned number are translated to the agree with number. (singular or plural).

During surface generation the cue primitives have to be translated to cue words, i.e., a lexical choice has to be made. The cue primitives correspond to the following cue words and many others:

Cue primitives	Cue words
Joint	*together, jointly*
Disjuncted	*each, separately*
Assigned	*respectively*

The *joint* cue primitive could give rise to a different lexical choice as e.g. John and Mary *share* a book and pen , than the construction with: John and Mary *have* a book and pen *together*.

(The cue word *both* should be avoided because it may be ambiguous, E.g. *John and Mary have both a book*, may mean that either they have a joint book or two separate books.

The exact selection of a particular cue word within each group of cue words is not treated in this paper since that is a lexical choice and surface generation problem, but it would be interesting carry out in future studies.

4.2 Implementation

A Prolog implementation of the aggregation and a cue primitive selection process is shown in Figure 5. "book/1" indicates instance 1 of a book, which is different instance from "book/2".

```
?- paraphrase(f(pres,poss/1,john,book/1)&f(pres,poss/2,mary,book/2)).

*Spy: | | |(6:7) Call: predicate_do(f(pres, poss / 1, john, book / 1) &
                                f(pres, poss / 2, mary, book / 2),
                                _17):
*Spy: | | | (7:8) Call: get_cue((poss / 1) & (poss / 2), _72):
*Spy: | | | (7:8) Exit: get_cue((poss / 1) & (poss / 2), poss / disj):
*Spy: | | |(6:7) Exit: predicate_do(f(pres, poss / 1, john, book / 1) &
                                f(pres, poss / 2, mary, book / 2),
                               f(pres, poss / disj, john & mary,
                                 book / sing)):
*Spy: | |(4:5) Call: surface(f(pres, poss / disj, john & mary,
                            book / sing),
                           _2):
*Spy: | |(4:5) Exit: surface(f(pres, poss / disj, john & mary,
                            book / sing),
                          [john, and, mary, have, a, book, each]):
john
  and mary have a book each.
    yes

?- paraphrase(f(pres,poss/1,john,book/1)&f(pres,poss/1,mary,book/1)).

*Spy: | | |(6:7) Call: predicate_do(f(pres, poss / 1, john, book / 1) &
                                f(pres, poss / 1, mary, book / 1),
                                _17):
*Spy: | | | (7:8) Call: get_cue((poss / 1) & (poss / 1), _72):
*Spy: | | | (7:8) Exit: get_cue((poss / 1) & (poss / 1), poss / joint):
*Spy: | | |(6:7) Exit: predicate_do(f(pres, poss / 1, john, book / 1) &
                                f(pres, poss / 1, mary, book / 1),
                               f(pres, poss / joint, john & mary,
                                 book / sing)):
*Spy: | |(4:5) Call: surface(f(pres, poss / joint, john & mary,
                            book / sing),
                          _2):
*Spy: | |(4:5) Exit: surface(f(pres, poss / joint, john & mary,
                            book / sing),
                          [john, and, mary, have, a, book, together]):
john
  and mary have a book together.
    yes
```

```
?- paraphrase(f(pres,poss/1,john,book/1)&f(pres,poss/2,mary,pen/1)).

*Spy: | | |(6:7) Call: predicate(f(pres, poss / 1, john, book / 1) &
                              f(pres, poss / 2, mary, pen / 1), _18):
*Spy: | | | (7:8) Call: get_ass_cue((poss / 1) & (poss / 2), _108):
*Spy: | | | (7:8) Exit: get_ass_cue((poss / 1) & (poss / 2), poss / ass):
*Spy: | | |(6:7) Exit: predicate(f(pres, poss / 1, john, book / 1) &
                              f(pres, poss / 2, mary, pen / 1),
                              f(pres, poss / ass, john & mary,
                                (book / 1) & (pen / 1))):
*Spy: | |(4:5) Call: surface(f(pres, poss / ass, john & mary,
                            (book / 1) & (pen / 1)),
                           _2):
*Spy: | |(4:5) Exit: surface(f(pres, poss / ass, john & mary,
                              (book / 1) & (pen / 1)),
                            [john, and, mary, have, a, book, and, a,
                             pen, respectively]):
john
  and mary  have  a  book
  and a  pen  respectively.
    yes
?-
```

Figure 5. Trace of cue selection process

5. The coordinator "but" and affective cue words

In this paper we have discussed the *and*-coordinator and its cue words. The *or*- and *but*-coordinators also need to be studied. The *but* is a special coordinator, since it causes some sort of imbalance. It should be used when there exists a violated expectation in the knowledge representation or background information, i.e., something that is not normal (see also [Reinh91]).

E.g.

John studied hard but did not pass the exams.

in contrary to the ordinary expectation

John studied hard and passed the exams.

In order to perform *but* -aggregation the knowledge representation and inference system underlying the aggregation must be able to recognize and signal expectation violation

Further on we have discussed only affect-neutral cue words. However, as illustrated for example in [Hovy88], some cue words carry affective connotations, such as *not only...but,* or *however....*, even in the case of aggregation. For example, after subject and predicate aggregation, an aggregated sentence like:

John is a gentleman and a scientist.

may be augmented by using a cue word, for example:

John is equally well a gentleman and a scientist

The use of affective cue words, which requires a theory of affect, is left for later studies.

6. Conclusions

The method of aggregation makes text and easy to read conceptually attractive for the Reader, but aggregation may give rise to ambiguities, therefore it is important to use cue words to clarify the text. Aggregation and cue words tie together aggregated sentences into

one unit. Cue words clarify the meaning of the aggregated sentences. Cue words should only be used when aggregation has taken place and they should be used carefully.

To use the technique of cue word augmentation one needs a knowledge representation which differentiates among different instances of the same concept. This is important to consider when generating natural language from formal specifications which are not constructed for generation.

This study shows that there is a lot of interesting research problems to solve, both to study the connection between cue primitives and the selection of syntactic constructions and lexical choice, and to find the different exceptions for when not to use cue words. This work is not by any means complete it is just a beginning. e.g. cue words might be used for stressing a sentence without having the purpose of removing any ambiguity. Other interesting research tasks would be to make more corpora studies to see how cue words are used and also to find which cue words correspond to the coordinators *but* and *or* .

Acknowledgements

Many thanks to Eduard Hovy for advising me, for stimulating discussions, for all comments on the paper and on the English, and for fun both over the Internet as well as in person at ISI and in California.

7. References

Dalia96a H. Dalianis: Aggregation as a Subtask of Text and Sentence Planning, In the *Proceedings of Florida AI Research Symposium, FLAIRS-96*, Key West, Florida, May 20-22, pp 1-5, 1996.

Dalia96b H. Dalianis: Concise Natural Language Generation from Formal Specifications, Ph.D. dissertation, (Teknologie Doktorsavhandling), Department of Computer and Systems Sciences, Royal Institute of Technology/Stockholm University, June 1996, *Report Series No. 96-008, ISSN 1101-8526, ISRN SU-KTH/DSV/R--96/8--SE.*

Dalia96c H. Dalianis & E. Hovy: Aggregation in Natural Language Generation.*Trends in Natural Language Generation: an Artificial Intelligence Perspective*, EWNLG'93, Fourth European Workshop, Adorni, G. & Zock, M. (Eds), Springer Verlag Lecture Notes in Artificial Intelligence no 1036, pp 88-105, 1996

Dalia97 H. Dalianis & E. Hovy: On Lexical Aggregation and Ordering. In the *Proceedings of the 6th European Workshop on Natural Language Generation*, Gerhard Mercator University, Duisburg, Germany, March 24-26, 1997..

Hobbs90 J. Hobbs: Literature and Cognition: *CSLI, Lecture Notes report*, Number 21, 1990.

Hovy88 E.H. Hovy: *Generating Natural Language under Pragmatic Constraints.* Hillsdale, New Jersey, Lawrence Erlbaum Associates Publishers, 1988.

Hovy94 E. Hovy & E. Maier: Parsimonius or Profligate: How may and which discourse relations? Submitted to *Discourse Processes*, 1994.

Reinh91 T. Reinhart: Elliptic Conjunctions-Non-Quantifictional LF. In *The Chomskyan Turn*, A.Kasher (ed.), Basil Blackwell, pp 360-384, 1991.

Quirk72 R. Quirk et al: *A grammar of contemporary English,* Longman Group Limited, 1972.

Appendix: Some empirical statistics on aggregation

In order to determine the occurrence and distribution of aggregation in naturally occurring texts in various genres we studied excerpts from the following: a road map, travel book, newspapers, a scientific book, and a diary.

1) *California Road Atlas (Thomas Bros. Maps)* pp. 213-220, in total 1000 words.

2) *Fielding's Mexico 1985* by L. & L. Foster, Introduction pp. 1-2, in total 400 words.

3) *Macintosh Programmer's Workbench*, by Joel West, pp. 335-336, in total 636 words.

4) *Los Angeles Times Calendar,* Monday Oct 31 1994, 907 words.

5) *MacWeek, Vol 8 No 41, 1994,* p. 3, in total 518 words.

6) *Time Vol 145, No. 8, February 1995,* pp 36-38, in total 672 words.

7) *The Art of Probability, for Scientist and Engineers,* by R.W. Hamming: Addison Wesley Publishing Company 1991, pp 1-2, in total 870 words.

8) *Wall Street Journal 1992, March 24*, sample of 716 words and 52 sentences.

9) *Asiatisk Dagbok 1984, (Asian Diary 1984)* (in Swedish) by Hercules Dalianis, sample of 733 words and 45 sentences.

10) *Wall Street Journal 1992, March 24*. Text contained 60862 words and 4548 sentences. In this text the ratio Words/Sentences = 13.4. In this text, 24 cue words for Bounded Lexical aggregation were found: except (4), exceptions are (2), exceptions is (1), besides (2), excluding (2), exclusion (1), most ... but (4), all.. not (2), all... but (2). The following Bounded Lexical cue words were not found: apart of, aside from, exclusive of, exception of.

11) *Asiatisk Dagbok 1984, (Asian Diary 1984)* (in Swedish) by Hercules Dalianis. Text contained 23860 words and 1259 sentences. The ratio Words/Sentences = 20.0. The following 5 Bounded Lexical aggregation cue words were found: förutom(2) (besides), alla .... utom (3) (all...but). Bounded Lexical aggregation cue words not found: flesta ....utom, (most ....but).

In summary: In the above texts we calculated the ratio (syntactic aggregation cases)/(total words) to be 1.8%, the ratio (cue words)/(total words) to be 0.02%, the ratio cue words/sentences to be 2.0%, and the ratio (cue words)/(syntactic aggregation) to be 15% i.e., every seventh syntactic aggregation contains a cue word

Text	Words	Sent.	Synt.Agg.	S/SP	PDO	cue word	BL cue
1	1000	65	29	26	3	2	
2	400		11	10	1	3	
3	636		9	9	0	0	
4	907		9	7	2	2	
5	518		5	4	0	0	
6	672		15	9	6	3	
7	870		9	5	4	4	
8[1]	716	52	10	8	2	2	0
9[1]	733	45	19	16	3	1	0
10	60862	4548				86	24
11	23860	1259				32	5
Total	91174	5969	116	94	21	135	29

Table 1. Table describing the results from the empirical study on texts.

Synt. Agg. = Syntactic Aggregation
S/SP = Subject/Subject Predicate Grouping
PDO = Predicate and Direct Object Grouping
BL = Bouned Lexical Aggregation Cue word

[1] Texts 8 and 9 are extracts from texts 10 and 11. Texts 10 and 11 have been computer scanned, while texts 8 and 9 have been manually read and elaborated on to investigate if they seem to follow the pattern of the other texts 1..7. Which they did.