AGGREGATION AS A SUBTASK OF TEXT AND SENTENCE PLANNING

Hercules Dalianis
Department of Computer and Systems Sciences
The Royal Institute of Technology and
Stockholm University
hercules@dsv.su.se

Abstract

Natural language generation is the technique of letting a computer automatically create natural language, e.g. English, Chinese or Greek, out of a computational representation. To generate natural language from computational representations, a number of processes must be carried out. Part of the process called sentence planning is the task of aggregation. Aggregation is the process which removes redundancies during generation of a natural language discourse without losing any information. Aggregation, which has been called ellipsis or coordination in Linguistics, makes text more fluent and easily read. While people do aggregation all the time without thinking about it, the contents of software engineering tools, data bases and expert systems, etc., is often highly redundant and needs aggregation before paraphrased to natural language.

This paper summarizes a larger work [Dalia96] which address various aspects of aggregation. When do we need to carry out aggregation ? What type of aggregations are there? Are there any general rules for how to aggregate? How are the rules related to each other? Aggregation may give rise to ambiguities: How can we solve them? How is aggregation related to the other generation processes?

1. INTRODUCTION

Aggregation, which is a subtask of Text and Sentence planning in Natural Language Generation (NLG), has received very little attention to date. We define aggregation to be the process that removes redundancies during generation of a natural language discourse without (ideally) losing any information. While people do aggregation all the time without thinking about it, the contents of software engineering tools, data bases and expert systems, etc., are often highly redundant, and therefore need aggregation in order to deliver high quality natural language. In this paper we develop the concept of aggregation and describe how and when it should be done.

2. WHAT IS AGGREGATION?

Aggregation[1] is the process of removing redundant information in a text without, losing any information. People do aggregation all the time to make natural language expressions shorter, non-redundant and easy to read.

For example

John has a book (a)

Mary has a book (b)

aggregation =>

John and Mary have a book (c)

We can see in the above example that in the two sentences (a) and (b) the objects which are different are, i.e. *John* and *Mary* , the rest of the sentences (a) and (b), are the same, i.e. *has a book* . Therefore we can aggregate the parts which are the same into one unit and then use the coordinator *and* between *John* and *Mary* and we obtain sentence (c).

In newspapers, books, articles you find various types of aggregation. For example the ratio (syntactic aggregation cases)/(total sentences) is approximately 33 % i.e. one third of the sentences has syntactic aggregation. The aggregation makes texts 10-20% shorter than it should have been without grouping as well as it is easier to read

We have studied aggregation from a "generation" view where we "generate ellipsis". The term ellipsis originates from the Greek word *ellípsis*, meaning missing or omission. We are using the term ellipsis in its more original general form.

[1] The term *aggregation* used in this thesis and in the Natural Language generation community is not he same as the term *aggregation* used in the conceptual modelling community.

Aggregation has many different aspects. Consider the example below:

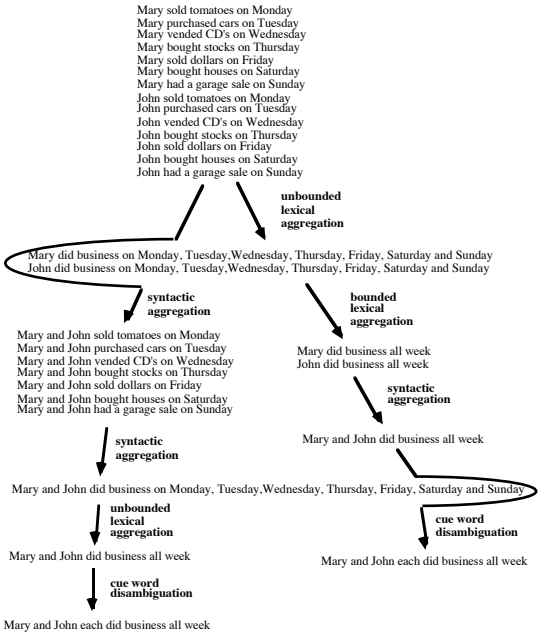

Figure 1 A text which is aggregated by syntactic and lexical aggregation rules and disambiguated by a cue word.

As we see there are several types of aggregation carried out during aggregation and there are also various orders to apply them. This example gives us a hint of the complexity of the aggregation processes.

In this paper we discuss aggregation during the sentence planning stage of NLG, in which we can distinguish four principal types:

1. *Syntactic* aggregation removes redundant information, but leaves (at least) one item in the text to carry the meaning explicitly. This is carried out at a pure syntactic level with no information loss about the content of the aggregated items.
2. *Elision* removes information that can be inferred and leaves no items in the text to carry the information explicitly, but the information remains there implicitly.
3. *Lexical aggregation* replaces a set of items with a new item, while the overall meaning is kept intact. This is carried out at a lexico-semantic level where information about the content of the aggregated items is needed.
   3.1*Bounded lexical aggregation* keeps the overall meaning intact and the aggregated information is retrievable. Bounded lexical aggregation requires a known set with a fixed number of elements.
   3.2*Unbounded lexical aggregation* may not keep the overall meaning intact and the aggregated information is not retrievable. Unbounded lexical aggregation requires an open set of elements.
4. *Referential aggregation* replaces redundant information with some sort of trace, such as a pronoun, to carry the information explicitly.

Syntactic aggregation have been discussed in [Dalia93], lexical aggregation in [Dalia95c] (in prep); referential aggregation have been discussed in [Wilki95]. Elision has been investigated in [McDon94]. Referential aggregation and elision lie outside the scope of this paper.

3. TYPES OF AGGREGATION

Linguists and Computational Linguists working with natural language analysis and parsing are interested in ellipsis or coordination because of finding the omitted pieces of a text, e.g., in [Dahl83,Sigur90]. They want to catch the real meaning of a text since they want to represent the meaning in some non-ambiguous representation. Computational Linguists working with Natural Language Generation are interested in "inverse ellipsis" or "inverse coordination" or more exactly what we here call aggregation. We want to generate ellipted sentences. The objective is to reduce redundancy and to avoid repetition.

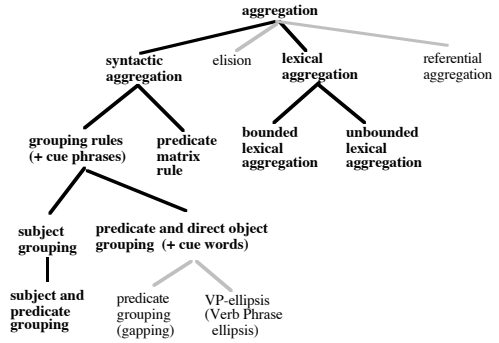

Figure 2 The tree describes the relation of the different aggregation types. The ones with the grey lines have not been investigated in this paper.

Figure 2, above, contains a tree clarifying the relation between various aggregation types. The bold text in the figure 2. shows the syntactic aggregation phenomena which have been treated previously in [Dalia93], where also a study of previous work on aggregation can befound. The syntactic aggregations found in [Dalia93] have been suggested to improve a NLG system and is described in [Dalia95a] and also implemented and described in [Dalia95b]. Syntactic aggregation has also been implemented in [Sigur92, Shaw95].

In linguistics, the results of aggregation is called ellipsis or coordination. [Quirk72] defines the strict sense of ellipsis when words are elided only if they are recoverable. Some type of ellipsis can function as ø-anaphora [Webbe79].

The motivation of ellipsis is to reduce redundancy and avoid repetition. [Quirk72] also includes a careful study of ellipsis; naming combined and segregatory coordination what we call (predicate and direct object grouping), as well

as ellipsis of subject and auxiliaries what we call (subject and predicate grouping). Neither Lexical aggregation nor Elision is mentioned at all. Authors who also have treated coordination, so-called syntactic aggregation, are [Oirso87,Gooda87].

The well-known term *gapping* corresponds to ellipsis of the first part of a predication in [Quirk72] which is redefined as *predicate grouping* according to this paper and not the predicate grouping defined in, [Dalia93].

4. GENERAL ISSUES: AGGREGATION AND SENTENCE PLANNING

Why should one carry out aggregation and what types of aggregations are there?

In corpora studies, [Dalia93c], we have studied in total 11 texts. The total amount of words in the nine first texts were 6452 words and the ratio (syntactic aggregation cases)/(total words) is 1.8% if you include the two last texts, the ratio (syntactic aggregation cases)/(total sentences) is approximately 33%; i.e. one third of the sentences includes syntactic aggregation.

If each aggregation saves approximately six words, this will make the text 1.8% aggregations x 6 words = 11% shorter, in some cases up to 20% shorter, than it would have been without aggregation in addition the text becomes easier to read. Eight of ten are subject and predicate grouping and the rest are predicate direct object grouping.

In further analysis of two additional texts (*Wall Street Journal 1992, March 24,* 60862 words and *Asiatisk Dagbok 1984,* 23860 words, [Dalia93c) containing together 84722 word and 5807 sentences in both English and Swedish, the ratio (Bounded Lexical aggregation cue words)/(total sentences) is 0.5%. I.e., we have at least 0.5% BL-aggregations, because the ones with no BL aggregation cue word are not visible or easy to find, when scanning a text automatically.

We estimate that aggregation shortens texts by 10-20%.

Why should one not carry out aggregation?

Syntactic aggregation and bounded lexical aggregation should always be carried out since the resulting text is shorter and no information is lost during the aggregation. One exception occurs when the Hearer's goal says it specifically to obtain certain information, e.g., the Hearer asks if the Speaker knows if *Mary did business this Sunday*, to which the Speaker should answer:

Yes, Mary did business this Sunday.

and not

Mary and John did business all week.

When losing information which is not retrievable, in for example unbounded lexical aggregation, one requires a good reason to carry out that aggregation, e.g., the Hearer wants to know if the Speaker knows if Mary did business, then the Speaker should answer:

Yes, Mary did business all week.

and not

Mary sold tomatoes, purchased cars, vended CD's.....and had a garage sale all week.

Another case preventing syntactic aggregation is when there exists a temporal relation between two mutually exclusive states which may be broken:

The wall is red. Tom paints the wall. The wall is yellow.

should not be aggregated to

The wall is red and yellow. Tom paints the wall.

Where should aggregation be carried out during natural language generation?

According to [Wilki95], aggregation can take place during every phase of NLG except during content selection and surface form generation.

In this work we take a slightly simplified view of the text generation process as a pipeline of three stages. Text planning (which determines the content and overall discourse structure of the text material), is followed by sentence planning (which decides on the sentence structure and scope), which in turn is followed by surface form realization (which is based on syntax).

In our view and also in [Dalia93,95a,95b,96] aggregation takes place after text planning, but before sentence planning. Aggregation operates mainly at sentence clause level, but it may also operate at discourse structure level, where however it may distort the discourse structure, so that the text becomes incoherent; in such cases new text planning is required.

How should one order input propositions before aggregation?

As shown in [Dalia93], ordering the input propositions is essential before applying aggregation rules. Certain combinations of input propositions give the optimal aggregation (the most consumed input propositions per aggregation rule), while other orderings do not permit aggregation.

The aggregation rules themselves may be ordered in various ways as well. This is treated in [Dalia95c]. In general first one should use the unbounded lexical aggregation rule (since that is

the most powerful) followed by the syntactic aggregation rules, then bounded lexical aggregation, and finally pronominalization (which is not aggregation but must be carried out as well).

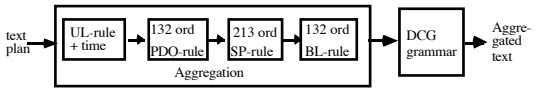

Figure 3. Overview of the input clauses ordering

What does a good text look like?

The concept *easily read text* has to be defined so it can be measured. Given a specific set of input propositions, to which a set of aggregations rules and a set of ordering rules in various combinations are applied, the result is a number of differently aggregated texts. Is it the case that the shortest text is best? Or is the Rhetorical Structure [Mann88] of the generated text the most important factor?

To answer these questions we performed a number of experiments on permuting and applying ordering rules and aggregation rules on a set of text plan clauses during generation, which is described in [Dalia95c]. We found that the Rhetorical Structure is a more important factor on deciding on which text is easily read than sentence length.

What happens after aggregation?

A side-effect as mentioned earlier is when applying aggregation at the discourse level the discourse structure may be distorted and new text planning is required. Also if aggregation occurs at sentence level the aggregated clauses may become unordered, i.e., incoherent, and reordering according to theme and focus may be required.

Aggregation may give rise to ambiguities: How can we prevent them?

One of the side-effects of aggregation is that an ambiguity can arise, because for example of problems with quantifier scoping. This is solved by using cue words such as*both, each, separately,* etc., which perform the disambiguation (see Fig. 1). The nature and use of cue words is discussed in [Dalia95d].

5. CONCLUDING REMARKS AND FUTURE DIRECTIONS

This paper have summarized some aspects of a longer work [Dalia96]. In that work the concept of aggregation has been defined and investigated. Aggregation contributes a novel part of the sentence planning phase of natural language generation, it clarifies the task of sentence planning and text realization in generation as well as discourse analysis and text linguistics.

The topic of aggregation is an important area since without aggregation automatically generated text is often very poor. This area has a direct application to real world problems in computational linguistics.

The work presented in this paper is a complement to research carried out by computational linguists and linguists working in the related areas as parsing and analysis of texts, since we have investigated different aspects of the ellipsis phenomena.

The concept of aggregation has been established in the Natural Language Generation community and the concept of aggregation has been used and referred to by other researchers [Kölln95,Shaw95,Wilki95]. Work summarized in this paper have been presented both at conferences for computational linguistics as well as for requirements engineering.

Implementation is important to reveal the nature of the aggregation phenomena and is a complement to empirical studies. Implementations in [Dalia95a,b,c,d] have been of great value to prove the findings in [Dalia93,95c,d] and have also lead to new discoveries in aggregation, such as the ordering problem of input text plan clauses, the distortion of discourse relations after aggregation [Dalia95c], and the problem of ambiguity after aggregation [Dalia95d].

Algorithms for carrying out syntactic and lexical aggregation during generation have been defined and implemented [Dalia95a,b,c,d]. A method for using cue words for disambiguation of aggregated text using the *and* -coordinator has been investigated and implemented [Dalia95d].

The solutions of the implementation problems provide guidelines for the architecture of an aggregation component and its relation to other generation components.

Aggregation in the sentence planning phase of natural language generation is required to make text non-redundant, short, and easily read. Aggregation is a fascinating research topic since it can be carried out at many different phases during natural language generation and it has various ways to be expressed, e.g., various syntactic and lexical aggregation types.

Many issues in the research area aggregation in natural language generation remain to be investigated. Our work is just the beginning. However, one problem is that research in aggregation is dependent on advances in other sentence planning areas such as sentence scoping, grouping and lexical choice.

In addition the use of cue words to disambiguate aggregated text is a topic which needs further investigation, for other coordinators, as for example, the *or* - and *but* - coordinators.

We have used the domain of requirements engineering for our aggregation generation. It would now be interesting to use other domains, as for example Medical Informatics [Ranki89a,89b, DiMar95] and Documentation and Technical manual generation [Rösne92, Svenb94].

Acknowledgements

Many thanks to Eduard Hovy for advising me, for stimulating discussions, and for fun both over the internet as well as in person at Information Sciences Institute/University of Southern California.

Reference

Dahl83 V. Dahl & M. McCord: Treating Coordination in Logic Grammars. In *American Journal of Computational Linguistics* Vol 9 No 2 April-June , 1983.

Dalia93 H. Dalianis & E. Hovy: Aggregation in Natural Language Generation. EWNLG-93, *Proceedings of the 4th European Workshop on Natural Language Generation*, Pisa, Italy 1993. Also in *Trends in Natural Language Generation: an Artificial Intelligence Perspective*, Adorni, G. & Zock, M. (eds.),Springer Verlag Lecture Notes in Computer Science (forthcoming 1996).

Dalia95a H. Dalianis: Aggregation in the NL-generator of the VIsual and Natural language Specification Tool. In *Proceedngs of The Seventh International Conference of the European Chapter of the Association for Computational Linguistics* (EACL-95), Student Session, pp 286-290, Dublin, Ireland, March 27-31, 1995.

Dalia95b H. Dalianis: Aggregation, Formal Specification and Natural Language Generation. In *Proceedings of the NLDB'95, First International Workshop on the Applications of Natural Language to Data Bases,*135-149, Versailles, France, June 28-29, 1995.

Dalia95c H. Dalianis & E. Hovy: On Lexical Aggregation and Ordering.(submitted to AAAI-96 and to INLG-96 workshop), 1995.

Dalia95d H. Dalianis: Natural Language Aggregation and Disambigation Using Cue Words. (submitted to ECAI-96), 1995.

Dalia96 H. Dalianis: Concise Natural Language Generation from Formal Specifications. (Forthcoming) Ph.D. dissertation,

DiMar95 C. DiMarco et al.: Healthdoc: Customizing patient information and health education by medical condition and personal characteristics. In *Proceedings of the Workshop on Patient Education*, Glasgow, 1995.

Gooda87 G. Goodall: *Parallel Structures in Syntax, Coordination, Causatives, and Restructuring*. Cambridge University Press, 1987.

Kölln95 M. Kölln: Employing user attitudes in text planning. In *Proceedings of the 5th European Workshop on Natural Language Generation*, pp. 163-179, Leiden, the Netherlands, 1995

Oirso87 R. R. van Oirsouw: *The Syntax of Coordination*, Croom Helm, 1987.

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

Ranki89a I. Rankin: Deep generation of a critique. In*Proceedings of the Second European on Natural Language Generation Workshop*, Edinburgh, April 6-8th 1989.

Ranki89b I. Rankin: The Deep Generation of Text in Expert Critiquing Systems, Licentiate Thesis No 184, Linköping University, 1989.

Rösne92 D. Rösner & M. Stede: Customizing RST for the Automatic Production of Technical Manuals. In *Aspects of Automated Natural Language Generation*, R. Dale et al. (eds.). Springer Verlag Lecture Notes in Artifical Intelligence no. 587, pp. 199-214, 1992.

Shaw95 J. Shaw: Conciseness through Aggregation in Text Generation. In *Proceeding of the 33rd Annual Meeting of Association of Computational Linguistics*, 26-30 June 1995, (ACL-95), Student Session, MIT, Camebridge, Massachussetts, USA, pp 329-331, 1995.

Sigur90 B. Sigurd & P. Warter: Understanding Coordination by Means of Prolog. Working Papers 36, pp 151-162, 1990, Department of Linguistics and Phonetics, Lunds University, 1990.

Sigur92 B. Sigurd et. al.: Automatic translation in specific domains: weather (Weathra) and stock market (Stocktra, Vectra). *Praktisk Lingvistik 15, 1992*, Department of Linguistics, Lund University, 1992.

Svenb94 S. Svenberg: Representing Conceptual and Lingusitic Knowledge for Multilingual Generation in a Technical Dom*ain, Proceedings of the 7th International Works*hop *on Natural Language Generation*, Kennebunkport,June, 1994.

Swart82 B.Swartout: GIST English Generator, *Procedings of AAAI-82*., American Association of Artificial Intelligence, Pittsburg, Pensylvania, 1982.

Webbe79 B. Webber: *A Formal Approach to Discourse Anaphora*. Garland Publishing Company, Inc, 1979.

Wilki95 J. Wilkinson: Aggregation in Natural Language Generation: Another Look, Computer Science Department, University of Waterloo, Canada, (unpublished M.Sc. thesis) 1995.