On Lexical Aggregation and Ordering

Hercules Dalianis[1] and Eduard Hovy[2]

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

2) USC/Information Sciences Institute, 4676 Admiralty Way, Marina Del Rey , CA 90292-6695, USA, ph. (+1) 310-822-1511, fax (+1) 310-822-0751, email: hovy@isi.edu

1. Introduction: Lexical Aggregation

Aggregation is the process of removing redundant information during language generation while preserving the information to be conveyed. Aggregation is an important component of text or sentence planning. Without aggregation, automated language generation systems would not be able to produce fluent text from real-world databases and knowledge bases, since information is rarely stored in computers in forms directly supporting fluent expression.

Various types of aggregation (syntactic, lexical, referential) have been identified in [Hovy88, Cook84, Reinhart91, Horacek92, Dalianis&Hovy93,Wilkinson95,Dalianis95a, 95b,96a]. This paper investigates lexical aggregation, the process by which a set of items is replaced with a single new lexeme that encompasses the same meaning. We call the elements that will be aggregated the *aggregands* and the element (the lexeme) which is the result of the aggregation the *aggregator*.

Lexical aggregation can be divided into two major types, bounded and unbounded. With Bounded Lexical (BL) aggregation the aggregator lexeme covers a closed set of concepts and the redundancy is obvious, the aggregated information is recoverable, and the aggregation process must be carried out. In contrast, Unbounded Lexical (UL) aggregation is carried out over an open set of aggregands and consequently the aggregated information is not recoverable and has to be licensed by other factors, such as the hearer's goals. The example in Figure 1 contains both types of aggregation, where *fight* and *week* are the unbounded and bounded aggregators respectively.

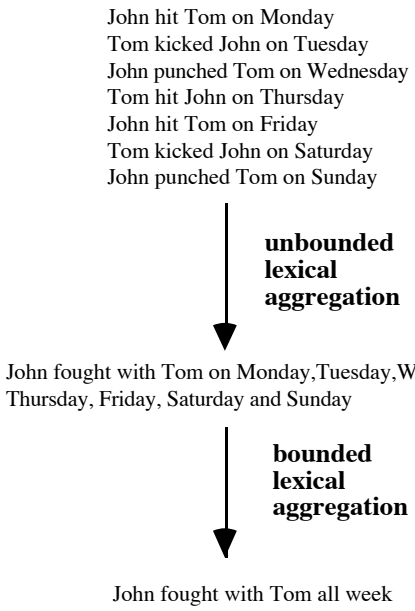

Figure 1. Example of Unbounded and Bounded lexical aggregation

2. Corpora Studies

Different subsets of an information collection may give rise to many and varied opportunities for aggregation. In fact, human-authored text contains aggregations throughout, as our corpus study shows. [Dalianis96b].

In the study we manually investigated in total 11 texts. The total amount of words in the first nine texts were 6.452 words and the ratio (syntactic aggregation cases)/(total words) was 1.8%. Including the two last texts, the ratio (syntactic aggregation cases)/(total sentences) was approximately 33%; i.e., one third of the sentences included 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.

Aggregated texts sometimes need cue words e.g., *each, together, separately, both*, to clarify the aggregation (see Example 1, next section). 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.

Some types of aggregation, such as Bounded lexical aggregation, refer to bounded sets, and are sometimes signalled by certain cue words, e.g., *except, all..except, exception(s) is/are, besides, excluding, exclusion, most...but, all...not, all...but*. An example of Bounded lexical aggregation with a cue word is:

Retail sales excluding auto dealers have remained practically unchanged since last June, Statistics Canada said.

Example taken from *Wall Street Journal 1992, March 24,* 60.862 words, which together with *Asiatisk Dagbok 1984,* 23.860 words contains 84.722 words and 5.807 sentences in both English and Swedish. The texts was scanned automatically for cue words and we found the ratio (Bounded Lexical aggregation cue words) / (total sentences) to be 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.

3. The Problem of Ordering

The following problem is described in [Dalianis&Hovy93]: Since aggregation rules operate only over adjacent clauses, a reordering of the input clauses is essential for effective aggregation to occur. Certain combinations of input clauses give rise to less redundant text (and hence more readable text, by the basic assumption underlying aggregation) than others. But what are the optimal ordering(s)? And do other criteria apply when measuring optimality? We call issues relating to the ordering of input clauses *the clause ordering problem* of aggregation.

A second ordering problem rears its head. We call this the *rule ordering problem* of aggregation. Given various kinds of aggregation rules — lexical (bounded and unbounded), syntactic (various rules), referential, etc. — does it matter in which order the rules are applied? Depending on how the lexical aggregation rules are written, it might indeed:

a. *Mariette bought the Christmas tree*
b. *Mariette carried it inside*
c. *Mariette mounted it*
d. *Ann fetched the decorations*
e. *Ann hung the decorations on the tree*

a . b . c : Syntactic-SP (Subject and Predicate) aggregation⇒ f

f. *Mariette bought, carried inside, and mounted the Christmas tree*

d . e : Syntactic-SP (Subject and Predicate) aggregation⇒ g

g. *Ann fetched and hung the decorations on the tree*

f . g : Syntactic-PDO (Predicate and Direct Object) aggregation⇒ h

h. *Mariette and Ann bought, carried inside, and mounted, and fetched and hung the decorations on the Christmas tree respectively*

h : UL-aggregation⇒ i

i. *Mariette and Ann put up the Christmas tree*

or alternative rule ordering:

a . b . c : UL-aggregation⇒ j

j. *Mariette installed the Christmas tree*

d . e : UL-aggregation⇒ k

k. *Ann decorated the Christmas tree*

j . k : Syntactic-SP-aggregation⇒ l

l. *Mariette and Ann installed and decorated the Christmas tree respectively*

l : *no more aggregation possible*: new BL-aggregation inference required

(Note: the cue word *respectively* is introduced by aggregation to clarify the aggregated text; for more about cue words see [Dalianis96c]).

In the first case, assuming the existence of a BL-aggregation inference rule that defines *put up a Christmas tree* as the sequence of events (a) to (e), this rule would produce (i). This rule would however not be able to produce (i) from (l), since (l) contains different actions altogether; here a new rule that decomposed *put up a Christmas tree* into the actions (j) *installed,* and (k) *decorated* would be required. Thus, unless the set of BL-aggregation rules were so crafted as to include all subdecompositions, different orderings of the aggregation rules will produce different results.

Furthermore, although lexical aggregation operates over lexis, interactions between syntactic and lexical aggregation necessitate the careful ordering of their respective rules. We performed an experiment to determine the optimal ordering(s) by applying several aggregation rules, in all permutations, to the clauses of a text plan. We implemented three aggregation rules (the Subject-Predicate and Predicate-Direct-Object (Syntactic) aggregation rules and the Bounded Lexical aggregation rule); also to control the order of input clauses, we created three ordering rules. An ordering rule orders the clauses in a text plan according to the weights of the ordering rule. The weights correspond to the predicate, subject, and object of the clause.

In order to determine the best order of applying aggregation rules and the ordering rules we performed the following experiment. We had a computer program cycle through all permutations of rules, and generate all possible texts for a given set of input clauses. We then analyzed these texts manually, trying to find a definition of (or failing that at least heuristics for) optimality.

Three aggregation rules and three ordering rules give 6! = 720 possible permutations (the 720 possible texts were generated automatically and came to 166 pages of A4 size). Some example permutation outputs are listed in [Dalianis96b]. To analyse the results (quite a job!), we had to make qualitative judgements. Our findings are as follows.

1. Somewhat surprisingly, text length (i.e., redundancy of words) is not the best measure of the readability of aggregated texts. Instead, a better measure is internal (structural) coherence, such as is the focus of, for example, Rhetorical Structure Theory [Mann&Thompson88].

2. One method to obtain good aggregation results is to perform pairwise application of one ordering and one aggregation rule at a time. A known good ordering rule should be applied on the input clauses and immediately followed by its corresponding aggregation rule, which can then be followed by another pair, etc. For example, the ordering 213 is best associated with the SP aggregation rule; the ordering 132 is best associated with the PDO aggregation rule; and the ordering 132 with the Bounded Lexical aggregation rule.

3. With respect to the rule ordering problem, the best order of aggregation rules is:

- first: Unbounded Lexical aggregation (this is the most powerful aggregation rule);
- next, the syntactic aggregations (preferably PDO followed by SP);
- next, Bounded Lexical aggregation;
- finally, other sentence planning tasks such as pronominalization.

4. Conclusions

While Unbounded and Bounded lexical aggregation are related to one another, UL-aggregation operates over an open set and loses some aggregated information, and BL-aggregation operates over a closed set and the aggregated information is retrievable. To select an appropriate UL-aggregation one may employ a hearer model. In both types of lexical aggregation one must check that the aggregands follow each other consecutively in time.

From the permutation experiment for obtaining the optimally aggregated text we conclude that one should not always select

the shortest text, but the one with the best discourse organization (which we model by the best RST structure). We found certain optimal orderings of text plan clauses before aggregation, each ordering associated with aggregation rule. Regarding the order of applying the aggregation rules, we propose first to use the most powerful rule (namely UL aggregation), then the predicate and direct object (PDO) grouping rule, then the subject and predicate (SP) grouping rule, and finally BL aggregation.

This paper is an extract of a longer work described in [Dalianis96b]. A great deal more work is required on the various aspects of lexical aggregation. Thoroughly studies of text corpora are necessary, as well as more fine-grained definitions of the various phenomena of lexical aggregation. The implementation of the finding will also be conditional upon the specific choices of knowledge representation system and inference support. This study is just a beginning of lot of exciting research !

5. References

[Cook84] Cook, M.E. et al. 1984.Conveying implicit content in narrative summaries *Proceedings of 10th International Conference on Computational Linguistics*, (COLING-84), pp 5-7, Stanford University.

[Dalianis&Hovy93] Dalianis, H. and E.H. Hovy. 1993. Aggregation in Natural Language Generation. In *Proceedings of the Fourth European Workshop on Natural Language Generation*. Pisa, Italy (67-78). 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)

[Dalianis95a] Dalianis, H. 1995. Aggreg ation in the NL-generator of the VIsual and Natural Language Specification Tool. In *Proceedings of The Seventh International Conference of the European Chapter of the Association for Computational Linguistics* (EACL-95), Student Session. Dublin, Ireland (286-290).

[Dalianis95b] Dalianis, H. 1995. 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.

[Dalianis96a] Dalianis, H. 1996. Aggregation as a Subtask of Text and Sentence Planning, To appear in the *Proceedings of Florida AI Research Symposium, FLAIRS-96*, Key West, Florida, May 20-22, 1996.

[Dalianis96b] Dalianis, H. 1996. 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.*

[Dalianis96c] Dalianis, H. 1996. Natural Language Aggregation and Clarification Using Cue Words, Department of Computer and Systems Sciences, Royal Institute of Technology/ Stockholm University, *Report Series No. 96-007, ISSN 1101-8526, ISRN SU-KTH/ DSV/R--96/7--SE.*

[Horacek92] Horacek, H. 1992. An Integrated View of Text Planning. *In Aspects of Automated Natural Language Generation.* Dale, R. et al. (eds.), Springer Verlag Lecture Notes in Artifical Intelligence no 587 (193-227).

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

[Mann&Thompson88] Mann, W.C. and S.A. Thompson. 1988. Rhetorical Structure Theory: Towards a Functional Theory of Text Organization. *TEXT* 8(3) (243-281).

[Reinhart91] Reinhart, T. 1991. Elliptic Conjunctions-Non-Quantificational LF. In *The Chomskyan Turn*. A.Kasher (ed.), Basil Blackwell (360-384).

[Wilkinson95] Wilkinson, J. 1995. Aggregation in Natural Language Generation: Another Look. Unpublished M.Sc. thesis, Computer Science Department, University of Waterloo, Canada.