Content Aggregation in Natural Language Hypertext Summarization of OLAP and Data Mining Discoveries

Jacques Robin
Universidade Federal de Pernambuco (UFPE)
Centro de Informática (CIn)
Caixa Postal 7851
50732-970 – Recife, Brazil
jr@di.ufpe.br

Eloi L. Favero
Universidade Federal do Pará (UFPA)
Departamento de Informática (DI)
Campus do Guama
66075-900 – Belém, Pará
ellf@di.ufpe.br

Abstract

We present a new approach to paratactic content aggregation in the context of generating hypertext summaries of OLAP and data mining discoveries. Two key properties make this approach innovative and interesting: (1) it encapsulates aggregation inside the sentence planning component, and (2) it relies on a domain independent algorithm working on a data structure that abstracts from lexical and syntactic knowledge.

1 Research context: hypertext executive summary generation for intelligent decision-support

In this paper, we present a new approach to content aggregation in Natural Nanguage Generation (NLG). This approach has been developed for the NLG system HYSSOP (HYpertext Summary System of On-line analytical Processing) which summarizes OLAP (On-Line Analytical Processing) and Data Mining discoveries into an hypertext report. HYSSOP is itself part of the Intelligent Decision-Support System (IDSS) MATRIKS (Multidimensional Analysis and Textual Reporting for Insight Knowledge Search), which aims to provide a comprehensive knowledge discovery environment through seamless integration of data warehousing, OLAP, data mining, expert system and NLG technologies.

1.1 The MATRIKS intelligent decision-support system

The architecture of MATRIKS is given in Fig. 1. It extends previous cutting-edge environments for Knowledge Discovery in Databases (KDD) such as DBMiner (Han et al. 1997) by the integration of:

- a *data warehouse hypercube exploration expert system* allowing automation and expertise legacy of dimensional data warehouse exploration strategies developed by human data analyst using OLAP queries and data mining tools;
- an *hypertext executive summary generator* reporting data hypercube exploration insights in the most concise and familiar way: a few web pages of natural language.

These two extensions allow an IDSS to be used *directly by decision makers* without constant mediation of a data analyst.

1.2 The HYSSOP natural language hypertext summary generator

To our knowledge, the development of HYSSOP is pioneer work in coupling OLAP and data mining with natural language generation, Fig. 2. We view such coupling as a synergetic fit with tremendous potential for a wide range of practical applications. In a nutshell[1], while NLG is the only technology able to completely fulfill the reporting needs of

[1] See Favero (2000) for further justification for this view, as well as for details on the motivation and technology underlying MATRIKS.

OLAP and data mining, these two technologies are reciprocally the only ones able to completely fulfill the content determination needs of a key NLG application sub-class: textual summarization of quantitative data.

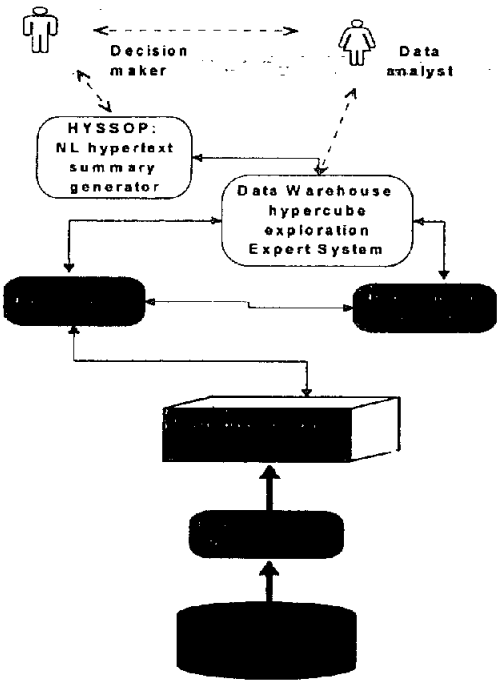

Fig. 1 – The architecture of MATRIKS

Generators that summarize large amount of quantitative data by a short natural language text (such as ANA (Kukich 1988), GOSSIP (Carcagno and Iordanskaja 1993), PLANDoc (McKeown, Kukich and Shaw 1994) among others) generally perform content determination by relying on a fixed set of domain-dependent heuristic rules. Such an approach suffers from two severe limitations that prevent it from reporting the most interesting content from an underlying database:

- it does not scale up for analytical contexts with high dimensionality and which take into account the historical evolution of data through time; such complex context would require a combinatorially explosive number of summary content determination heuristic rules;
- it can only select facts whose class have been thought ahead by the rule base author, while in most cases, it is its very unexpectedness that makes a fact interesting to report;

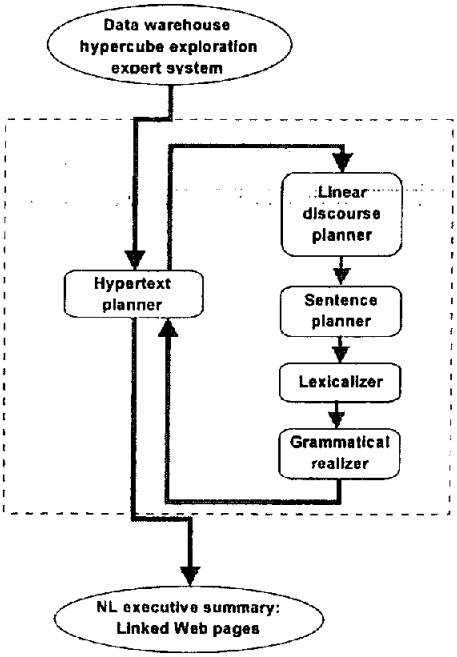

Fig. 2 – The architecture of HYSSOP

OLAP and data mining are the two technologies that emerged to tackle precisely these two issues: for OLAP, efficient search in a high dimensionality, historical data search space, and for data mining, automatic discovery in such spaces, of hitherto unsuspected regularities or singularities. In the MATRIKS architecture, heuristic rules are not used to define content worth reporting in a data warehouse executive summary. Instead, they are used to guide the process of searching the warehouse for unexpected facts using OLAP and data mining operators.

A data warehouse hypercube exploration expert system encapsulates such rules in its knowledge base to perform content determination. An example output of such expert system, and input to HYSSOP, is given in Fig. 3: the data cells selected for inclusion in the output textual summary are passed along with their OLAP context and the data mining annotations that justify their relevance. One output generated by HYSSOP from this input is given in Fig. 4. and Fig. 5.

	OLAP context				Data mining annotations				
	Dimensions			Meas.	Discovery	roll up context (avg%)			Drilldown
Cell	product	place	time	Δ	exception	prod	place	time	place
1c	Birch Beer	nation	Nov	-10	low	3	2	4	nation
2c	Jolt Cola	nation	Aug	+6	low	0	3	-7	nation
3c	Birch Beer	nation	Jun	-12	low	2	5	3	nation
4c	Birch Beer	nation	Sep	+42	high	-2	1	1	nation
5c	Cola	central	Aug	-30	low	7	-5	-1	region
6c	Diet Soda	east	Aug	+10	low	-5	7	-8	region
7c	Diet Soda	east	Sep	-33	medium	-1	0	7	region
8c	Diet Soda	east	Jul	-40	high	-1	5	8	region
9c	Diet Soda	south	Jul	+19	low	1	-1	-11	region
10c	Diet Soda	west	Aug	-17	low	2	4	1	region
11c	Cola	Colorado	Sep	-32	medium	-2	2	2	state
12c	Cola	Colorado	Jul	-40	medium	-1	4	0	state
13c	Cola	Wisconsin	Jul	-11	low	0	3	7	state

Fig. 3 – An example input of HYSSOP, derived from an example retailing database taken from, (Sarawagi, Agrawal and Megiddo, 1998). The part inside the bold sub-frame is the input to the sentence planner

Last year, the most atypical sales variations from one month to the next occurred for:
- *Birch Beer with a 42% national increase from September to October;*
- *Diet Soda with a 40% decrease in the Eastern region from July to August.*

At the next level of idiosyncrasy came:
- *Cola's Colorado sales, falling 40% from July to August and then a further 32% from September to October;*
- *again Diet Soda Eastern sales, falling 33% from September to October.*

Less aberrant but still notably atypical were:
- *again nationwide Birch Beer sales' -12% from June to July and -10% from November to December;*
- *Cola's 11% fall from July to August in the Central region and 30% dive in Wisconsin from August to September;*
- *Diet Soda sales' 19% increase in the Southern region from July to August, followed by its two opposite regional variations from August to September, +10% in the East but -17% in the West;*
- *national Jolt Cola sales' +6% from August to September.*

To know what makes one of these variations unusual in the context of this year's sales, click on it.

Fig. 4 – Example of HYSSOP front-page output

The 40% decrease in Diet Soda sales was very atypical mostly due to the combination of the two following facts:
- *across the rest of the regions, the July to August average variation for that product was 9% increase;*
- *over the rest of the year, the average monthly decrease in Eastern sales for that product was only 7%."*
- *across the rest of the product line, the Eastern sales variations from July to August was 2%*

Fig. 5 – Example of HYSSOP follow-up page output (behind the 40% front page anchor link)

The architecture of HYSSOP is given in Fig. 2. HYSSOP is entirely implemented in LIFE (Ait-Kaci and Lincoln 1989), a language that extends Prolog with functional programming, arityless feature structure unification and hierarchical type constraint inheritance. For content realization, HYSSOP relies on feature structure unification. Lexicalization is inspired from the approach described in (Elhadad, McKeown and Robin 1997), while surface syntactic realization follows the approach described in (Favero and Robin 2000b). HYSSOP makes two innovative contributions to NLG research: one to hypertext content planning presented in (Favero and Robin 2000a) and one to content aggregation presented in the rest of this paper.

2 Research focus: content aggregation

in natural language generation

Natural language generation system is traditionally decomposed in the following subtasks: content determination, discourse-level content organization, sentence-level content organization, lexical content realization and grammatical content realization. The first three subtasks together are often referred to as content planning, and the last two together as linguistic realization. This separation is now fairly standard and most implementations encapsulate each task in a separate module (Robin 1995), (Reiter 1994).

Another generation subtask that has recently received much attention is content aggregation. However, there is still no consensus on the exact scope of aggregation and on its precise relation with the five standard generation tasks listed above. To avoid ambiguity, we define aggregation here as: *grouping several content units, sharing various semantic features, inside a single linguistic structure, in such a way that the shared features are maximally factored out and minimally repeated in the generated text.* Defined as above, aggregation is essentially a key subtask of sentence planning. As such, aggregation choices are constrained by discourse planning decisions and they in turn constrain lexical choices.

In HYSSOP, aggregation is carried out by the sentence planner in three steps:

1. content factorization, which is performed on a tabular data structure called a *Factorization Matrix* (FM) ;
2. generation from the FM of a discourse tree representing the hypertext plan to pass down to the lexicalizer;
3. top-down traversal of the discourse tree to detect content units with shared features occurring in non-adjacent sentences and annotate them as anaphora.

Such annotations are then used by the lexicalizer to choose the appropriate cue word to insert near or in place of the anaphoric item.

2.1 Content factorization in HYSSOP

The key properties of the factorization matrix that sets it apart from previously proposed data structures on which to perform aggregation are that:

- it fully abstracts from lexical and syntactic information;
- it focuses on two types of information kept separate in most generators, (1) the semantic features of each sentence constituent (generally represented only before lexicalization), and (2) the linear precedence constraints between them (generally represented only late during syntactic realization);
- it visually captures the interaction between the two, which underlies the factorization phenomenon at the core of aggregation.

In HYSSOP, the sentence planner receives as input from the discourse planner an FM representing the yet unaggregated content to be conveyed, together with an ordered list of candidate semantic dimensions to consider for outermost factoring. The pseudo-code of HYSSOP's aggregation algorithm is given in Fig. 10. We now illustrate this algorithm on the input example FM that appears inside the bold sub-frame of the overall HYSSOP input given in Fig. 3. For this example, we assume that the discourse planner directive is to factor out first the exception dimension, followed by the product dimension, *i.e., FactoringStrategy = [except,product]*. This example illustrates the mixed initiative choice of the aggregation strategy: part of it is dictated by the discourse planner to ensure that aggregation will not adversely affect the high-level textual organization that it carefully planned.

The remaining part, in our example factoring along the place and time dimensions, is left to the initiative of the sentence planner. The first step of HYSSOP's aggregation algorithm is to shift the priority dimension D of the factoring strategy to the second leftmost column of the FM. The second step is to sort the FM rows in (increasing or decreasing) order of their D cell values. The third step is to horizontally slice the

FM into row groups with identical D cell values. The fourth step is to merge these identical cells and annotate the merged cell with the number of cells that it replaced. The FM resulting from these four first steps on the input FM inside the bold sub-frame of Fig. 3 using *exception* as factoring dimension is given in Fig. 6.

The fifth step consists of recursively calling the entire aggregation algorithm inside each row group on the sub-FM to the right of D, using the remaining dimensions of the factoring strategy. Let us now follow one such recursive call: the one on the sub-FM inside a bold sub-frame in Fig. 6 to the right of the *exception* column in the third row group. The result of the first four aggregation steps of this recursive call is given in Fig. 7. This time it is the product dimension that has been left-shifted and that provided the basis for row sorting, row grouping and cell merging. Further recursive calls are now triggered. These calls are different from the preceding ones, however, in that at this point all the input constraints provided by the discourse planner have already been satisfied. It is thus now up to the sentence planner to choose along which dimension to perform the next factorization step. In the current implementation, the column with the lowest number of distinct values is always chosen. In our example, this translates as factoring along the time dimension for some row groups and along the space dimension for the others. The result of the recursive aggregation call on the sub-FM inside the bold frame of Fig. 7 is given in Fig. 8. In this case, factoring occurred along the time dimension. The fully aggregated FM resulting from all the recursive calls is given in Fig. 9. Note how the left to right embedding of its cells reflects exactly the left to right embedding of the phrases in the natural language summary of Fig. 4 generated from it.

2.2 Cue word generation in HYSSOP

Once content factorization is completed, the sentence planner builds in two passes the discourse tree that the lexicalizer expects as input. In the first pass, the sentence planner patterns the recursive structure of the tree (that itself prefigures the output text linguistic constituent structure) after the left to right and narrowing embedding of sub-matrices inside the FM.

cell	except	product	place	time	Δ
4c	high	Birch Beer	nation	Sep	+42
8c	*2	Diet Soda	east	Jul	-40
7c	med	Diet Soda	east	Sep	-33
11c	*3	Cola	Colora.	Sep	-32
12c		Cola	Colora.	Jul	-40
1c	low	Birch Beer	nation	Nov	-10
2c	*8	Jolt Cola	nation	Aug	+6
3c		Birch Beer	nation	Jun	-12
5c		Cola	central	Aug	-30
6c		Diet Soda	east	Aug	+10
9c		Diet Soda	south	Jul	+19
10c		Diet Soda	west	Aug	-17
13c		Cola	Wiscon	Jul	-11

Fig. 6 – Left shift, row grouping and cell merging along the exception dimension

Cell	product	place	time	Δ
1c	Birch Beer	nation	Nov	-10
3c	*2	nation	Jun	-12
5c	Cola	central	Aug	-30
13c	*2	Wisconsin	Jul	-11
6c	Diet Soda	east	Aug	+10
9c	*3	south	Jul	+19
10c		west	Aug	-17
2c	Jolt Cola	nation	Aug	+6

Fig. 7 – Recursion along the product dimension

Cell	time	place	Δ
9c	Jul	south	+19
6c	Aug	east	+10
10c	*2	west	-17

Fig. 8 – Recursion along the time dimension

cell	except	product	place X time / time X place		Δ
4c	high	Birch Beer	nation	Sep	+42
8c	*2	Diet Soda	east	Jul	-40
11c	med.	Cola	Colorado	Sep	-32
12c	*3	*2	*2	Jul	-40
7c		Diet Soda	east	Sep	-33
1c	low	Birch Beer	nation	Nov	-10
3c	*8	*2	*2	Jun	-12
5c		Cola	central	Aug	-30
13c		*2	Wiscon	Jul	-11
9c		Diet Soda	Jul	south	+19
6c			Aug	east	+10
10c		*3	*2	west	-17
2c		Jolt Cola	nation	Aug	+6

Fig. 9 – Final, fully aggregated FM after all recursive calls

In the second pass, the sentence planner traverses this initial discourse tree to enrich it with anaphoric annotations that the lexicalizer needs to generated cue words such as "again", "both", "neither", "except" etc. Planning cue words can be considered part of aggregation since it makes the aggregation structures explicit to the reader and prevents ambiguities that may otherwise be introduced by aggressive content factorization. A fragment of the sentence planner output discourse tree built form the aggregated FM of Fig. 9 is given in Fig. 12. The discourse tree spans horizontally with its root to the left of the feature structure and its leaves to the right. Note in Fig. 12 the cue word directive: ***[anaph=[occur=2nd, repeated=[product, region]]]***. It indicates that this is the second mention in the text of a content unit with *product = "Birch Beer"* and *region = nation*. The lexicalizer uses this annotation to generate the cue word *"again"* before the second reference to *"nationwide Birch Beer sales"*.

- ***factor(Matrix,FactoringStrategy)***
  variables: Matrix = a factorization matrix
  FactoringStrategy = a list of pairs (Dimension,Order) where Dimension ∈ dimensions(Matrix)
  and Order ∈ {increasing,decreasing}
  RowGroups = list of sub-matrices of Matrix
  begin
  if FactoringStrategy = emptyList
  *then FactoringStrategy <- **buildFactoringStrategy**(Matrix) ;*
  (Dim1,Order1) <- first(FactoringStrategy) ;
  RemainingFactoringStrategy <- rest(FactoringStrategy) ;
  *Matrix <- **leftShiftColumn**(Matrix,Dim1);*
  *Matrix <- **sortRows**(Matrix,Dim1,Order1) ;*
  *RowGroups <- **horizSlice**(Matrix,Dim1);*
  for each RowGroup in RowGroups do:
  *RowGroup <- **mergeCells**(RowGroup,Dim1) ;*
  *(LeftSubMatrix,RighSubMatrix) <- **cut**(RowGroup,Dim1) ;*
  *FactoredRightSubMatrix <- **factor**(RightSubMatrix, RemainingFactoringStrategy) ;*
  *RowGroup <- **paste**(LeftSubMatrix,FactoredRightSubMatrix,Dim1) ;*
  *Matrix <- **update**(Matrix,RowGroup);*
  endfor;
  return Matrix ;
  end.

- ***buildFactoringStrategy(Matrix)**: returns inside a list a pair (Dim,increasing) where Dim is the matrix's dimension (i.e., column) with the lowest number of distinct values.*
- ***leftShiftColumn (Matrix,Dim1)**: moves Dim1 to the second leftmost column next to the cell id column.*
- ***sortRows(Matrix,Dim1,Order)**: sorts the Matrix's rows in order of their Dim1 cell value; Order specifies whether the order should be increasing or decreasing.*
- ***horizSlice(Matrix,Dim1)**: horizontally slices the Matrix into row groups with equal value along Dim1.*
- ***mergeCells(RowGroup,Dim1)**: merges (by definition equal valued) cells of Dim1 in RowGroup.*
- ***cut(RowGroup,Dim1)**: cuts RowGroup into two sub-matrices, one to the left of Dim1 (including Dim1) and the other to the right of Dim1*
- ***paste(LeftSubMatrix,FactoredRightSubMatrix,Dim1)**: pastes together left and right sub-matrices.*
- ***update(Matrix,RowGroup)**: identifies the rows R_M of Matrix whose cell ids match those of RowGroup R_G and substitute those R_M by R_G inside Matrix*

Fig. 10 – HYSSOP's aggregation algorithm

A special class of aggregation-related cue phrases involves not only the sentence planner and the lexicalizer but also the discourse planner. One discourse strategy option that HYSSOP implements is to precede each aggregation group by a cue phrase explicitly mentioning the group's cardinal. An example summary front page generated using such a strategy is given in Fig. 11. The count annotation in the cell merging function of HYSSOP's aggregation algorithm are computed for that purpose. While the decision to use an explicit

count discourse strategy lies within the discourse planner, the counts are computed by the sentence planner and their realization as cue phrases are carried out by the lexicalizer.

Last year, there were 13 exceptions in the beverage product line.
The most striking was Birch Beer's 42% national fall from Sep to Oct.
The remaining exceptions clustered around four products were:

- *Again, Birch Beer's sales accounting for other two national exceptions, both decreasing mild values:*
  1. *a 12% from Jun to Jul;*
  2. *a 10% from Nov to Dec;*
- *Cola's sales accounting for four exceptions:*
  1. *two medium in Colorado, a 40% from Jul to Aug and a 32% from Aug to Sep;*
  2. *two mild, a 11% in Wisconsin from Jul to Aug and a 30% in Central region from Aug to Sep;*
- *Diet Soda accounting for 5 exceptions:*
  1. *one strong, a 40% slump in Eastern region from Jul to Aug;*
  2. *one medium, a 33% slump in Eastern region from Sep to Oct;*
  3. *three mild: two increasing, a 10 % in Eastern region from Aug to Sep and a 19% in Southern region from Jul to Aug; and one falling, a 17% in Western region from Aug to Sep;*
- *Finally, Jolt Cola's sales accounting for one mild exception, a 6% national fall from Aug to Sep.*

Fig. 11 HYSSOP's front page output using discourse strategy with explicit counts

```
cat = aggr, level =1, ngroup =2, nmsg =2
common = | Exceptionallity = high %%The most atypical sales variations from one moth to the next occurred for

distinct = | cat =msg, attr =[product ="Birch beer", time =9, place =nation, var=+42]
           | %%Birch Beer with a 42% national increase from Sept to Oct

           | cat =msg, attr =[product ="Diet Soda", time =7, place =east, var=-40]
           | %%Diet Soda with a 40% decrease in the Eastern region from Jul to Aug

cat =aggr, level=1, ngroup=2, nmsg=3
common = | exceptionallity = medium %%At next level of idiosyncrasy came:

distinct = | cat =aggr, level =2, ngroup =2, nmsg=2,
           | common = | product=Cola, place=Colorado %% Cola's sales
           |
           | distinct = | cat=msg, attr=[time=7, var =-40] %% falling 40% from Jun to Jul
           |            | cat=msg, attr=[time=9 var =-32 %% and then a further 32 from Sep to Oct
           |
           | cat =msg, attr =[product ="Diet Soda", time =9, place =east, var=-33
           | anaph [occurr =2nd, repeated=[product, place]
           | %% again Diet Soda Eastern sales, falling 33% from Sep to Oct

| cat =aggr, ... %% Less aberrant but still notably atypical were: ...
```

Fig. 12 – Fragment of LIFE feature structure representing the discourse tree output of the sentence planner and input to the lexicalizer.

3 Related work in content aggregation

The main previous works on content aggregation are due to:

- (Dalianis 1995, 1996), whose ASTROGEN system generates natural language paraphrases of formal software specification for validation purposes;
- (Huang and Fiedler 1997), whose PROVERB system generates natural language mathematical proofs from a theorem prover reasoning trace;
- (Robin and McKeown, 1996), whose STREAK system generates basketball game summaries from a semantic network

representing the key game statistics and their historical context;

- (Shaw 1998), whose CASPER discourse and sentence planner has been used both in the PLANDoc system that generates telecommunication equipment installation plan documentation from an expert system trace and the MAGIC system that generates ICU patient status briefs from medical measurements.

In this section, we briefly compare these research efforts with ours along four dimensions: (1) the definition of aggregation and the scope of the aggregation task implemented in the generator, (2) the type of representation the generator takes as input and the type of output text that it produces, (3) the generator's architecture and the localization of the aggregation task within it, and (4) the data structures and algorithms used to implement aggregation.

3.1 Definition of the aggregation task

The definition of aggregation that we gave at the beginning of previous section is similar to those provided by Dalianis and Huang, although it focuses on common feature *factorization* to insure aggregation remains a *proper* subset of sentence planning. By viewing aggregation only as a process of combining *clauses*, Shaw's definition is more restrictive. In our view, aggregation is best handled prior to commit to specific syntactic categories and the same abstract process, such the algorithm of Fig. 10, can be used to aggregate content units inside linguistic constituents of any syntactic category (clause, nominal, prepositional phrases, adjectival phrases, etc.). In terms of aggregation task coverage, HYSSOP focuses on paratactic forms of aggregation. In contrast, ASTROGEN, CASPER, PROVERB and STREAK also perform hypotactic and paradigmatic aggregation.

3.2 Input representation and generated output text

A second characteristic that sets HYSSOP apart from other generators performing aggregation is the nature of its input: a set of data cells extracted from a dimensional data warehouse hypercube. In contrast, the other systems all take as input either a semantic network extracted from a knowledge base or a pre-linguistic representation of the text to generate such as Meteer's text structure (Meteer 1992) or Jackendoff's semantic structure (Jackendoff 1985). Such natural language processing oriented inputs tend to simplify the overall text generation task and hide important issues that come up in real life applications for which raw data is often the only available input. In terms of output, HYSSOP differs from most other systems in that it generates hypertext instead of linear text. It thus tackles the content aggregation problem in a particularly demanding application requiring the generator to simultaneously start from raw data, produce hypertext output and enforce conciseness constraints.

3.3 Generation architecture and aggregation localization

While its overall architecture is a conventional pipeline, HYSSOP is unique in encapsulating all aggregation processing in the sentence planner and carrying it out entirely on a deep semantic representation. In contrast, most other systems distribute aggregation over several processing components and across several levels of internal representations: deep semantic, thematic and even surface syntactic for some of them.

3.4 Data structures and algorithms for aggregation

All previous approaches to aggregations relied on rules that included some domain-specific semantic or lexical information. In contrast, the aggregation algorithm used by HYSSOP is domain independent since it relies only on (1) generic matrix row and column shuffling operations, and (2) on a generic similarity measure between arbitrary data cells.

4 Conclusion

We presented a new approach to content aggregation in the context of a very challenging and practical generation application: summarizing OLAP and data mining discoveries

as a few linked web pages of fluent and concise natural language. We believe that the key contribution to our work is to show the feasibility to perform effective paratactic aggregation:

- encapsulated within a single generation component (the sentence planner)
- using a domain-independent algorithm and a simple data structure, the factorization matrix, that captures the key structural and ordering constraints on paratactic aggregation while completely abstracting from domain semantic idiosyncrasies as well as from lexical and syntactic details.

This is a first success towards the development of a plug-in content aggregation component for text generation, reusable across application domains. In future work, we intend to empirically evaluate the summaries generated by HYSSOP.

References

Aït-Kaci H. and Lincoln P. (1989) LIFE – A natural language for natural language. *T.A . Informations*, 30(1-2):37-67, Association pour le Traitement Automatique des Langues, Paris France.

Carcagno D. and Iordanskaja L. (1993) Content determination and text structuring; two interrelated processes. In H Horacek (ed.) *New concepts in NLG: Planning, realisation and systems.* London: Pinter Publishers, pp 10-26.

Dalianis H. (1995) Aggregation, Formal specification and Natural Language Generation. *In Proc. of the NLDB'95 First International Workshop on the application of NL to Databases*, 135-149, Versailles, France.

Dalianis H. (1996) Aggregation as a subtask of text and sentence planning. *In Proc. of Florida AI Research symposium*, FLAIRS-96, Florida, pp 1-5.

Elhadad M., McKeown K. and Robin J. (1997) Floating constraints in lexical choice. *Computational Linguistics*, 23(2).

Favero E. L. (2000). *Generating hypertext summaries of data mining discoveries in multidimensional databases*. PhD Thesis. Centro de Informática, UFPE, Recife, Brazil.

Favero E. L. and Robin J. (2000a). Using OLAP and data mining for content planning in natural language generation. Accepted for publication in *Proc. of 5th International. Conference on Applications of Natural Language to Information Systems*, NLDB'2000, 28-30 June, Versailles France.

Favero E. L. and Robin J. (2000b). Implementing Functional Unification Grammars for Text Generation as Featured Definite Clause Grammars. Submitted to *Natural Language Engineering.*

DBMiner (2000): http://db.cs.sfu.edu/DBMiner/index.html

Huang G. and Fiedler A (1996) Paraphrasing and aggregation argumentative text using text structure. *In Proc. of the 8th International NLG Workshop*, pages 21-3, Sussex, UK.

Jackendoff R. (1985) *Semantics and Cognition.* MIT Press, Cambridge, MA, June 15-17.

Kukich K. (1988) Fluency in Natural Language Reports in *Natural Language Generation Systems*, McDonald, D. & Bloc, L. (Eds.), Springer-Verlag.

McKeown K., Kukich, K. and Shaw J. (1994) Practical issues in automatic document generation. *In Proc. of ANLP'94*, pages 7-14, Stuttgart, Oct.

Meteer M. (1992) Expressibility and the problem of efficient text planning. *Communication in Artificial Intelligence*. Pinter Publisher Limited, London.

Reiter E. (1994) Has a Consensus NL Generation Architecture Appeared, and is it Psycholinguistically Plausible? *In Proc of the Seventh International Workshop on Natural Language Generation (INLGW-1994)*, pages 163-170. Kennebunkport, Maine, USA.

Robin J. (1995) *Revision-based generation of natural language summaries providing historical background: corpus-based analysis, design, implementation and evaluation.* Ph.D. Thesis. CUCS-034-94, Columbia University, Computer Science Department , New York, USA. 357p.

Robin J. and McKeown K. (1996) Empirically designing and evaluating a new revision-based model for summary generation. *Artificial Intelligence*, 85(1-2). 57p.

Sarawagi S. Agrawal R and Megiddo N. (1998) Discovery-driven exploration of MDDB data cubes. *In Proc. Int. Conf. of Extending Database Technology (EDBT'98)*, March.

Shaw J. (1998) Segregatory coordination and ellipsis in text generation. *In Proc. of the 17th COLING'98.*