

Fit it in but say it well!

**Cécile Paris, Nathalie Colineau,
Andrew Lampert**
CSIRO – ICT Centre
Locked Bag 17
North Ryde, NSW 1670, Australia
FirstName.LastName@csiro.au

Joan Giralt Duran
Barcelona School of Informatics
Technical University of Catalonia
Barcelona, Spain
joangi@gmail.com

Abstract

Reasoning about how much content to generate when space is limited presents an increasingly important challenge for generation systems, as the diversity of potential delivery channels continues to grow. The problem is multi-faceted: the generated text must fit into the allocated space, the space available must be well utilised, and the resulting text must convey its intended message, and be coherent, well structured and balanced. To address this problem, we use a discourse planning approach. Our system reasons about the discourse structure to decide how much content to realise. In this paper, we present two algorithms that perform this reasoning and analyse their effectiveness.

1 Introduction

The ability to reason about how much content to realise in order to convey a message when the allocated space is fixed is an important consideration for Natural Language Generation (NLG) systems. It will become even more pressing as the amount of available information increases (e.g., via the web or content management systems) and the space constraints on the delivery media become more diverse (e.g., via web browsers, email, PDAs, cell phones).

We, as humans, address this problem by shortening our sentences or by restricting the content we include. We can achieve the former by manipulating vocabulary and syntax. This, however, is of limited value in reclaiming significant amounts of space and requires careful attention to sentence-level grammar and vocabulary choice. We can achieve the latter by dropping those pieces of content whose contribution to the communicative goal is most limited. For instance, we might question the need for a long elaboration or an

example to illustrate our point. This approach can reclaim significant amounts of space but requires an understanding of the text's discourse structure.

Many organisations today store information in content management systems or other types of organisational databases. This information is typically stored at the level of paragraphs, along with images and database entries (e.g., directories of contact information). Such information systems provide an interesting opportunity for NLG systems, by reducing the cost of acquiring the underlying semantic knowledge base and removing the need to construct text from first principles. However, they also present challenges: how can a system still perform the type of reasoning typically expected of NLG systems when it has no control over the text at the sentence level? In particular, how can a system produce a coherent and well structured text that meets some specific space limitations? In these cases, a system only has one method available to ensure it produces an appropriate amount of text: it must reason about the text at the discourse level.

Many of our application domains are such that information is stored at the paragraph level in one or more data repositories. Our system answers people's information needs by retrieving appropriate pre-authored text fragments from such repositories and delivering that content via a variety of media, each with their own space requirements. The specific application we discuss in this paper is SciFly, a system that generates brochures about the work carried out in our organisation. The output is delivered as a two-page flyer, as shown in Figure 1¹, and can also be displayed as a web output and as a plain text summary in the body of an email. As our underlying text fragments are pre-authored, we cannot manipulate the realisation of their sentences. We thus concentrate on reasoning about the discourse structure to generate text fitting specific space constraints. Importantly, the brochures we

¹ Though the text is too small to read, the figure gives an idea of the document and its layout.



Figure 1: A brochure generated by our system

generate need to look professional, as if manually written. Coherence and good structure is paramount.

In the remainder of this paper, we discuss other work which has looked at controlling the amount of generated text. We then present our approach to generation when a text has to fit into some specific space and the system has no control over the text at the sentence level. We show how we can exploit the discourse structure to decide how much content to realise. We present two algorithms that perform this reasoning and analyse their comparative performance.

2 Related work

Generation systems have often exploited the discourse structure for a number of reasoning tasks. To achieve this, systems build a discourse tree during discourse planning. The tree includes the communicative goals that were achieved and the rhetorical relations holding between text spans, frequently represented using Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). The systems then reason about this discourse tree to allow them to participate in a dialogue (e.g., Moore and Swartout, 1989), generate appropriate cue phrases to link two spans of text (e.g., Scott and de Souza, 1990) or reason about layout (e.g., Bateman et al., 2001). See Taboada and Mann (2006) for other applications.

Description

VICCU is designed so that all information required by the specialist to make judgments on patient treatment is available in real time, as if he or she were present at the remote hospital. The communication - based on technology developed by the Centre for Networking Technologies for the Information Economy (IC-NEI) (Liu) - transmits multiple high quality digital video channels, high quality audio, vital signs data, written notes and medical images. Two-way high bandwidth video permits natural, low-latency, telepresence interaction with the specialist.

Highlight

By fusing existing technologies, the Virtual Critical Care Unit (VICCU) has become a tool of real clinical use which solves the problem of how to get specialist critical care into smaller, regional hospitals.



Working with us

CSIRO offers flexibility in structuring partnerships, addressing issues of confidentiality and IP ownership under negotiated agreements with our customers. CSIRO welcomes enquiries relating to acquiring or licensing our intellectual property. Visit our website, or contact us for more information.
<http://www.ict.csiro.au>

Contact CSIRO

Phone 1300 363 400
 Email enquiries@csiro.au
 Web www.csiro.au



For further information about the CSIRO ICT Centre, contact:

Trevor Bird ICT Centre Chief Scientist Cnr Vimera and Pembroke Roads Mansfield NSW 2122 Phone: +61 2 9372 4288 Fax: +61 2 9372 4400 Email: Trevor.Bird@csiro.au	Gary Doherty Business Development Cnr Vimera and Pembroke Roads Mansfield NSW 2122 Phone: +61 2 9372 4557 Fax: +61 2 9372 4400 Email: Gary.Doherty@csiro.au	Tom McGinness Communication Manager Building 8BB, Macquarie University Sydney NSW 2113 Phone: +61 2 9326 3227 Fax: +61 2 9326 3200 Email: Tom.McGinness@csiro.au
---	--	--

© CSIRO. Disclaimer: CSIRO should be liable for technical or editorial errors contained herein. The information provided in this form of HTML text is subject to change without notice.

² There are a few exceptions for multinuclear relations.

both approaches are similar, the purpose and challenges of the two tasks are different: in a summary, one wishes to convey only the most essential part of a message, leaving out all the content that could be considered optional. In our task, we want to produce a text that fits the available space. In some cases, the original content is only, let's say, one paragraph too long to fit into the required space. The issue is to find the least essential paragraph to convey the message and still obtain a balanced text, as opposed to summarising the text per se. This is akin to the distinctions between shortening a paper by ½ a page to fit into the required 8-page limit and writing its abstract. In addition, shortening a text (rather than summarising it) raises a new challenge: that of ensuring that the final text is still balanced. Consider for example cases in which there are bulleted items. When shortening the text, one needs to ensure that all the items remain at about the same length, or the text will appear odd.

Our approach exploits the semantics of the rhetorical relations and the notion that some relations are more important than others. O'Donnell (1997) used the same principle to assign relevance scores to text nodes in the discourse structure to produce documents of variable length. While our approach is similar to O'Donnell's in that respect, his approach required individual sentences to be manually marked up with rhetorical relations. This allowed his system to manipulate the text at or below the sentence level, although repair had to occur after the process to ensure coherence and grammaticality. O'Donnell's approach was thus close to traditional NLG systems that build text from first principles and are able to vary the amount of text at the lexico-grammatical level (e.g., Reiter, 2000).

Like most NLG-based approaches to constraining the length of a text, we use greedy algorithms to cut content. Vander Linden (2008) reports on an alternate approach that used dynamic programming. His work has not yet been evaluated, so it is unclear how valuable it could be in our context.

3 System Architecture

To achieve our goal of delivering tailored and coherent information, we build upon both NLG and document synthesis technology to retrieve and re-purpose information (Colineau *et al.*, 2004). The retrieval process is orchestrated by a discourse planner that, using discourse plans, builds a discourse tree specifying what to extract, for which purpose and how the various pieces of information relate to each other (Paris *et al.*, 2008).

We use SciFly to illustrate how our system reasons about space constraints. Given a query from a user (one or more topic(s) of interest), SciFly

consults a repository of text fragments to assemble the relevant information. The fragments, written by the marketing team of our organisation, are self contained and comprised of typically one paragraph, two at most.³ SciFly integrates all the relevant fragments into a coherent whole (see Figure 1) using meta-data describing each fragment. The meta-data chosen is that envisaged for a new content management system for our organisation's website. Note that this meta-data does not correspond to rhetorical relations. It is the discourse plans that later determine which relation holds between fragments given their role in a specific communicative goal. As a result, a text fragment can be used with different relations in different brochures.

SciFly also produces a web output, and a PDF version of the paper brochure is emailed to the user with a summary in the email body. This summary is also built by reasoning about the discourse tree. As in discourse-based approaches to summarisation, only the nuclei are kept leaving out all the satellites corresponding to content considered as optional.

SciFly follows a two-stage approach: during discourse planning, content and organisation are selected, and a discourse tree is built, as in (Moore and Paris, 1993). The discourse plans specify the RST that hold between text spans. They were written for SciFly, based on an analysis of a corpus of sample human-authored brochures. In the second stage, the presentation stage, the system reasons about the discourse tree and the characteristics of the delivery channel to decide how much content to include and how to present it.

The following section presents two algorithms that reason about how much to express based on the space available. The first one is fairly naïve, implementing the basic notions of how RST can help with this reasoning. The second algorithm addresses limitations discovered when we deployed the system as an information kiosk at a major IT fair.

4 Determining how much content to realise

During the discourse planning stage, SciFly retrieves all the information corresponding to the user's topics of interest and organises it into a coherent whole. As the output is to be a brochure-like document, presenting the work carried out in our organisation (e.g., projects and capabilities), the system includes more information than is strictly required. For example, an introduction about the organisation and relevant contact details are always

³ It is worth noting that it took one person in the marketing team just a few days to write the content.

included, and, in the case of a brochure about a project, a description of the research laboratory in which the project resides is also provided.

At the end of this stage, the system has produced a discourse tree. It includes the top level communicative goal, the intermediate goals and the rhetorical relations that exist between text spans. It thus encodes both the purpose of each retrieved text fragment and how these fragments relate to each other. This provides a basis to reason about which information to realise when there is too much content for some delivery medium (e.g., a double-sided A4 page).

As noted earlier, our two algorithms embody the principle of nuclearity. They also both exploit the notion that some relations are more important than others. For example, *context* (providing the context in which to understand the information in the nucleus) might be considered more important than *elaboration* (providing more details). By assigning an importance value to relations, it becomes possible to rank the relations based on their contribution to the communicative goal. Our assignment, presented in Table 1, is based on judgments from our marketing staff.⁴ The importance of each relation is defined in a declarative configuration file that can be easily modified to suit different preferences and application contexts.

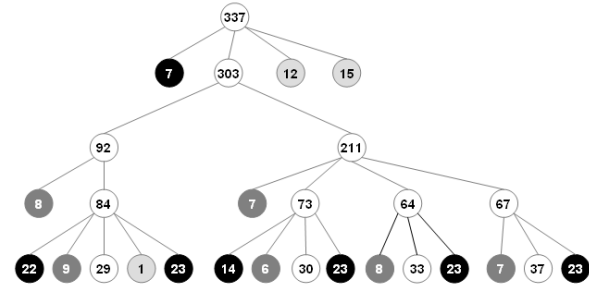
Shading	Discourse Relations	Importance
Black	Illustration, Background, Circumstance, Elaboration	Low
Dark Grey	Context, Motivation, Evidence, Summary, Justification,	Low-Medium
Light Grey	Preparation, Enablement	Medium-High
		High

Table 1: Some discourse relations and their ranking

To explain our algorithms, we represent the discourse tree using an abstract view, as shown in Figure 2. The communicative goals are represented as nodes. A white node indicates a nucleus; the other nodes, the satellites, are all shaded in grey corresponding to the importance of the rhetorical relation linking them to the nucleus.

Each node is the root of a subtree (empty if the node is a leaf) which generates some content. In both algorithms, the system computes for each node the approximate space required for that content in number of lines. This is computed bottom-up in an iterative manner by looking at the retrieved content at each node. Note, however, that the system can only compute an approximation of the number of

lines of content to be generated, as this depends on style, line-wrapping and other formatting attributes within the text fragments and global spacing decisions in the PDF rendering process. In Figure 2, the number inside each node indicates the approximate amount of content that node produces (in lines).



Device space: 200

Figure 2: Discourse tree annotated with how much content would be generated for each node.

4.1 Naïve algorithm

From the two features of RST mentioned above (nuclearity and relative importance of relations), we designed a straightforward algorithm to exploit the discourse structure in order to decide how much content to realise. This is our “naïve algorithm”. With this algorithm, the system examines the top level node to determine if the current structure will result in too much content, given the properties of the output medium (e.g., lines of content per page). If so, the system selects the relation with the lowest importance and traverses the tree, dropping all the satellite nodes (including their sub-trees) that are related to a nucleus with this relation. The algorithm repeats this process until the amount of content meets the device space constraint.

Consider the example in Figure 2; the top node indicates 337 lines of content, while the space available for the device is 200 lines (specified in a device model). The *illustration* relation is the least important relation present. The algorithms thus drops all the satellites related by *illustration*. Since this is not enough, it picks the next least important relation (*background* in this case) and repeats the process until the space requirements are met. The resulting discourse tree and the ordered list of dropped nodes are shown in Figure 3. It is important to note that, with this approach, the resulting document is shorter but still coherent. This is because reasoning is done with the discourse tree and the rhetorical relations holding between subgoals.

We deployed the system with this algorithm at a trade fair in 2005 and 2006 and measured the general experience visitors had with the system. On average, people rated the system positively,

⁴ SciFly actually has 5 levels of importance. We have merged “low” with “low-medium” and “high” with “medium-high” here to avoid too many shades of grey.

emphasising the nice layout of the brochure, its conciseness and its informative content. The area identified as needing improvement was the amount of blank space in brochures (as seen in Figure 1), where it seems that more information could have

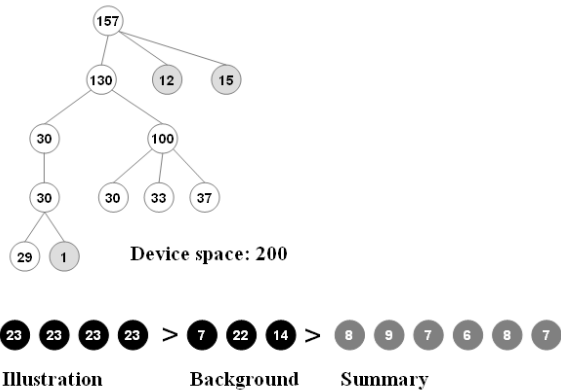


Figure 3: The resulting discourse tree after applying the naïve space constraint algorithm.

been included. This is because our naïve algorithm drops many sub-tree(s) at once, thus potentially deleting a lot of content at each step. For example, in its third pass, the algorithm deleted 45 lines in one go. This can be too much, and indeed, the resulting brochure can appear odd because of the excessive blank space. This led us to our enhanced algorithm.

4.2 The enhanced algorithm

We redesigned the algorithm to gain finer control over the text length. To do so, we take into account the depth of a node in addition to its rhetorical status. We first converted the *importance value* of a rhetorical relation (e.g., low, medium, high) into a *penalty score*. In our system, penalty scores range from 1 to 6, in increments of 1: from high importance relations with a score of 2, through to low importance relations with a score of 6. A nucleus is given a score of 1 to take the increment in tree depth into account.

We then assign each node an importance rating called its *weight*. It is computed by adding (1) the weight of the node’s parent, to take into account the depth of the node in the whole tree, and (2) the *penalty score* of the rhetorical relation which relates the node to its nucleus. A child node is thus *heavier* than its parent. The larger the weight, the less important the node is to the overall discourse structure.⁵

Once the weights have been computed, the system orders the nodes by their weight, and the

heaviest nodes get dropped first. Thus, nodes deeper in the tree and linked by a discourse relation with a high penalty score (low importance) get removed first. Nodes are now dropped one by one until the top level node has an amount of content that satisfies the space requirement. This provides finer control over the amount of realised content and avoids the limitation of the first algorithm.

We illustrate this process through the same example. We annotate the tree of Figure 2 with node weights, as shown in Figure 4 (the weights appear outside the nodes). The system can now order the satellite nodes, from *heaviest* to *lightest*. As we consider that nuclei are important, other nodes (i.e., satellites) are dropped preferentially at any given depth of the tree. This is why we do not include the nuclei in our ordered list (shown at the bottom of Figure 4). A nucleus will get dropped *only* if its parent is dropped. The system takes the nodes with the heaviest weight and drops them one by one, until the top node has the appropriate amount of content. In our example, the seven left most nodes of the ordered list will be dropped (as indicated in the Figure in the boxed area). This results in much less text being dropped than with the naïve algorithm to satisfy the same requirement (e.g., 137 lines dropped instead of 180). As before, pruning the tree does not affect the coherence of the resulting document.

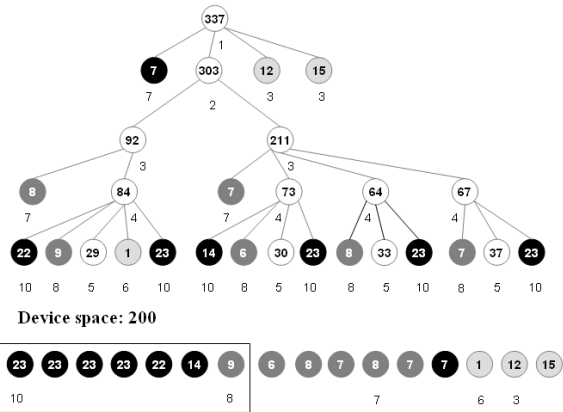


Figure 4: Tree annotated with weights and ordered list of (satellite) nodes

As mentioned previously, techniques used in discourse-based approaches to summarisation are quite similar to our algorithm. Like we do, they take advantage of the discourse structure and exploit the difference between nuclei and satellites to determine the most important units in a text. However, there are a number of differences that influence the ordering based on the importance of the nodes (units of text) and how much gets deleted. For example, the granularity of the units of text is different. This might have an impact on the type of discourse representation built (e.g., our satellites nodes

⁵ Note that this is similar in concept to the relevance rating proposed by O’Donnell (1997), but computed differently. O’Donnell’s rating penalises only nodes in a satellite scope, thus only partially taking the depth of the tree into account.

are rarely further decomposed). Moreover, although Marcu (1998) has suggested that the semantics of rhetorical relations may play a major role in determining the important units in a text, this was not integrated in his scoring function because of lack of empirical evidence in his data. We believe that these semantics not only influence the ordering of the nodes but also lead to a finer grained partial ordering, thus giving us finer grained control.

To illustrate this, we applied Marcu’s scoring function to the discourse tree of Figure 2. Figure 5 shows the partially ordered list of nodes we obtained (ranked from most to least important). While the two lists are not fundamentally different, the way nodes get dropped is different. With Marcu’s scoring function, we would have dropped 159 lines instead of 137 using a partial ordering with 4 levels instead of the 6 with our enhanced algorithm. Finally, our treatment of lists and itemised items is substantially different, as described below.

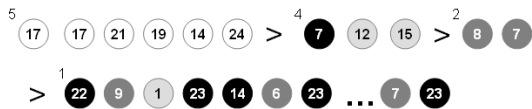


Figure 5: Ordered list of nodes when applying Marcu’s scoring function

This enhancement to the algorithm allows the system to have finer control over what is generated. However, as noted earlier, shortening a text, rather than summarising it, raises a new challenge: that of ensuring that the final text is still balanced. Sometimes, a discourse structure contains several parallel sub-structures that, if pruned unevenly, results in text that is unbalanced and appears odd. This happens for example in a paper, when we itemise or enumerate to provide a list of topics. We typically try to balance the amount of text each topic contains, avoiding having one topic with one sentence while another topic has several paragraphs. In our case, we have such parallel structures when describing a list of projects (belonging to a research laboratory, for example, or when the user has indicated interest in several projects). In these cases, the discourse structure contains several sub-trees which are likely to be of the same structure, as illustrated schematically in Table 2. This structure is generated during discourse planning by a plan containing a *foreach* statement, e.g., (*foreach project in project-list (describe project)*).

Project 1	Project 2	Project 3
• Introduction	• Introduction	• Introduction
• Description	• Description	• Description
• Illustration	• Illustration	• Illustration

Table 2. Example of parallel and balanced structures

To keep the overall structure balanced, the system annotates all sub-structures issued from such a *foreach* statement. Then, when constructing the ordered list of satellites, the system clusters nodes at the same level of depth in the sub-structures, taking into account their relationship to the nucleus, as shown in Figure 6. (Note that, since these sub-trees are generally identical, the nodes will often have similar weights.) When dropping nodes, the whole cluster will be deleted at the same time, rather than node by node.

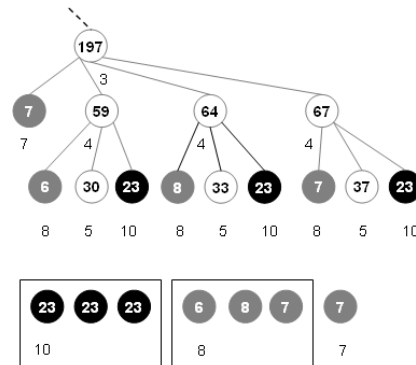


Figure 6: Ordered list of (satellite) cluster nodes

In the structure of Table 2, illustrated in Figure 6, for example, illustrations for all projects (cluster of weight 10) will be dropped together, then all introduction sections (cluster of weight 8). This prevents one sub-structure from being pruned more than its sibling structures and thus ensures the resulting brochure is balanced.

5 Comparative Performance Analysis

We performed an analysis of the two algorithms to assess their comparative effectiveness with respect to filling the space of a two-page brochure. In particular, we wanted to find out whether the enhanced algorithm reduced the amount of dropped content, filling up the available space more effectively.

We automatically generated 1605 brochures about randomly selected topics, using both algorithms. For each brochure, we stored data about the total amount of content initially assembled and the amount of content withheld from the generated brochure (none if nothing was dropped). For our analysis, we kept only the brochures for which content was withheld. This left us with 1507 brochures. We observed the following improvements, as shown in Figure 7:

- 82.5% of the brochures generated with the enhanced algorithm filled over 96% of the available space (leaving at most 8 lines of empty space). Only 29% of brochures

generated with the naïve algorithm achieved this performance.

- 96.5% of the brochures generated with the enhanced algorithm filled at least 90% of the space, compared with 44.5% of brochures generated using the naïve algorithm.

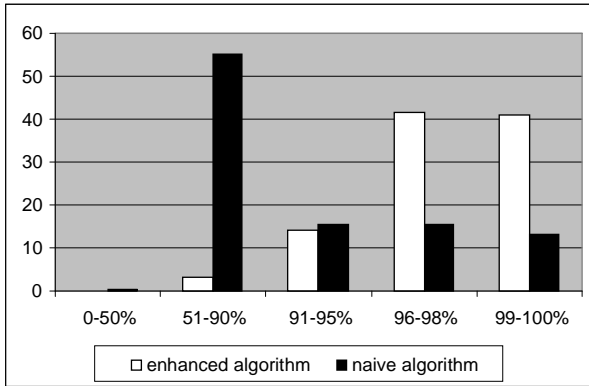


Figure 7: Percentage of fullness of the brochures (vertical axis is the number of brochures, in percentage)

Given the discourse planning approach employed by our system, we know that the generated text is coherent with both algorithms. The results on the amount of text generated show that our enhanced algorithm results in a much better use of the available space than the naïve algorithm. Finally, because the algorithm prunes parallel structures in a synchronised manner, we know that the resulting text is balanced. We have thus achieved our goal of producing a well structured coherent text that fills as much of the space available as possible, when fine-grained control over the text generated is not possible. We can also conclude that it is useful to exploit the discourse structure to reason about what to include when we have specific space requirements.

In further analysing the results, we found the following:

- 75% of brochures included more content using the enhanced algorithm, as we desired;
- 13% had the same amount of content regardless of the algorithm used. (Note that the same amount of content does not necessarily mean that the content is identical, as the algorithms select the content to drop differently.); and
- 12% of the brochures actually contained less content with the enhanced algorithm than with the naïve one.

We examined these results in more detail to understand what was happening.

In the brochures that gained content with the new algorithm, an average of 32 new lines of content was included, which represents about 15% of the whole brochure. More specifically, as shown in

Figure 8: 36% of the brochures gained between 1 and 20 lines of relevant content (representing 1-10% of the brochure); 28% gained between 21 and 40 lines (11-20% of the brochure); and 30% gained 41 to 60 lines (21-30% of the brochure).

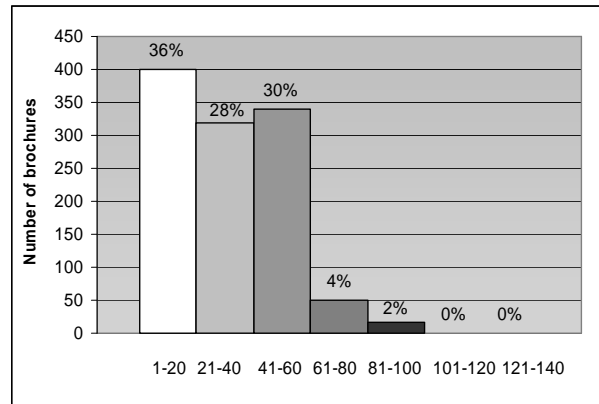


Figure 8: Percentage of gained lines with the enhanced algorithm

An example of brochure generated with the enhanced algorithm is shown in Figure 9. The content kept by the enhanced algorithm that would have been dropped by the naïve one is highlighted in grey.

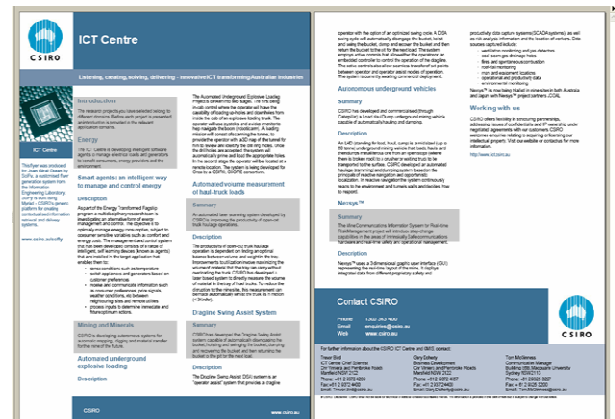


Figure 9: Example showing the differences in content between the two algorithms

In the 12% of cases where the enhanced algorithm drops more content than the naïve one, we noted that, on average, the naïve algorithm had produced a brochure which was about 3 lines away from a full brochure, while the enhanced one produced a brochure which was, on average, 7 lines away from such a brochure. Thus, these brochures were at least 96% filled for both algorithms. The reduction in realised content for the new algorithm was due to our treatment of parallel discourse structures, thus representing a desirable loss of content to create balanced brochures, as described earlier. This illustrates a limitation of this metric for comparative performance analysis: it only takes into account the space used rather than the overall

quality of the output. Clearly loss of content is desirable in some cases to maintain a well structured and balanced text. To measure the overall quality of the brochures produced by the two algorithms, we performed a user evaluation.

6 User evaluation

In our user evaluation, we asked users to compare pairs of brochures, where one brochure was produced by the naïve algorithm and the other by the enhanced algorithm. The users were asked to choose which brochure, if any, they preferred. The aim was to ensure that the improved use of space did not have any negative impact on the overall quality of the brochure and the users' satisfaction. The layout for all brochures was kept constant.

Seventeen users participated in the evaluation and were presented with seven pairs of brochures. The pairs of brochures were selected to represent the variety of output produced by both algorithms. As mentioned earlier, in the majority of the cases, the enhanced algorithm uses the available space more effectively by including more content in the brochure. However, in a number of cases, due to our treatment of parallel structures, and also because the two algorithms select the content to drop differently, the enhanced algorithm produces brochures with the same amount or less content than using the naïve one. To represent these cases and evaluate whether this has an impact on how users assess the quality of the brochures, we selected the brochures as follows: in three pairs, the brochures generated by the enhanced algorithm contained more content (cluster 1), in two pairs, both brochures had the same amount of content regardless of the algorithm used (cluster 2), and in two pairs, the brochures generated by the enhanced algorithm contained a bit less content (cluster 3). To control any order effect, the pairs were randomly presented from user to user, and in each pair, each brochure was randomly assigned a left-right configuration. The results are shown in Table 3.

Pairs	ENHANCED	NAIVE	EQUIV
Cluster 1	26	9	11
Cluster 2	8	14	12
Cluster 3	17	9	8
Total	51	37	31

Table 3: Users' ratings of preference

Table 3 shows that participants generally preferred the brochures generated with the enhanced algorithm over the ones produced with the naïve algorithm, or found them equivalent. If we group together the participants' preference for the brochures generated by the enhanced algorithm (51

votes) with the 31 cases where participants found the pair of brochures equivalent, we see that we have not lost any performance in terms of users' satisfaction with the enhanced algorithm, while we gain significantly in the amount of text included.

Interestingly, most users preferred the enhanced algorithm for cluster 3, where the enhanced algorithm produced less text but pruned parallel sub-structures in an even manner, resulting in more balanced documents. Also, users seem to prefer the naïve algorithm for cluster 2, when both algorithms produce the same amount of text. After further analysis, we found that users liked having text presented as *summary*, a relation which got dropped more often with the new algorithm. This can be addressed by changing the importance of this relation. We also asked the users to explain their choice. 'Good presentation' and 'good flow of information' were the reasons given for preferring the brochures generated with the enhanced algorithm.

7 Conclusions

The ability to reason about how much content to generate in order to convey a message under space constraints is an important consideration for NLG systems. In our applications, we cannot resort to the traditional method of controlling the lexical-grammatical resources to that effect. We generate text by re-using existing text fragments over which we do not have any control, and we need to produce, at discourse planning stage, a discourse tree with all the appropriate available content, in order to realise the output on several delivery channels (e.g., full structure for a web output, subset of the structure to fit specific space constraints such as a double-sided A4 page and only nucleus content for the email summary). We thus had to find other ways to satisfy space requirements. To this end, we implemented two algorithms that reason about the discourse tree. The first naïve algorithm, embodying in a straightforward manner the notions of nuclearity and the rhetorical relations' relative importance, resulted in a sub-optimal use of space. The enhanced algorithm addressed this limitation and ensured a balanced text. Our comparative analysis showed that our enhanced algorithm produces documents filling most of the available space, while maintaining users' satisfaction.

Acknowledgements

We would like to thank Meriem Raji for her work on the user experiment, all the people who took part in our evaluation, our group and Keith Vander Linden for their input throughout the project.

References

- John Bateman, Thomas Kamps, Jörg Kleinz and Klaus Reichenberger (2001). Constructive text, diagram and layout generation for information presentation: the DArt_bio system. *Computational Linguistics*, 27(3):409-449.
- Nathalie Colineau, Cécile Paris and Mingfang Wu. (2004). Actionable Information Delivery. *Revue d'Intelligence Artificielle (RSTI – RIA)*, Special Issue on Tailored Information Delivery, 18(4), 549-576.
- William C. Mann and Sandra A. Thompson. (1988). Rhetorical Structure Theory: Toward a functional theory of text organisation. *Text* 8(3):243-281.
- Daniel Marcu. (1998). To build text summaries of high quality, nuclearity is not sufficient. In *Working Notes of the AAAI-98 Spring Symposium on Intelligent Text Summarization*, Stanford, CA, 1–8.
- Johanna D. Moore and William R. Swartout. (1989). A Reactive Approach to Explanation. In *Proceedings of the 1989 International Joint Conference on Artificial Intelligence (IJCAI 1989)*, 1504-1510.
- Johanna D. Moore and Cécile L. Paris. (1993). Planning Text for Advisory Dialogues: Capturing Intentional and Rhetorical Information, *Computational Linguistics*, 19 (4):651-694, Cambridge, MA.
- Kenji Ono, Kazuo Sumita and Seiji Miike (1994). Abstract generation based on rhetorical structure extraction. In *Proceedings of the 15th Conference on Computational Linguistics (CoLing 1994)*, Volume 1, Kyoto, Japan, 344 – 348.
- Mick O'Donnell (1997). Variable Length On-Line Document Presentation. *Proceedings of the 6th European Workshop on Natural Language Generation*. Gerhard-Mercator University, Duisburg, Germany.
- Cécile Paris, Nathalie Colineau, Keith Vander Linden and Andrew Lampert (2008). Myriad: A Reusable Platform for Tailored Information Delivery. CSIRO Technical Report 08/050.
- Ehud Reiter. (2000). Pipelines and Size Constraints. *Computational Linguistics*, 26:251-259.
- Lucia H.M. Rino and Donia R. Scott. (1994). Automatic generation of draft summaries: heuristics for content selection. In *Proceedings of the 3rd International Conference on the Cognitive Science of Natural Language Processing*, Dublin.
- Donia R. Scott and Clarisse S. de Souza. (1990). Getting the message across in RST-based text generation. In Dale, Mellish & Zock (eds). *Current Research in Natural Language Generation*. London: Academic Press. 119-128.
- Karen Spark Jones (1993). What might be in a summary? *Information Retrieval 93: Von der Modellierung zur Anwendung*. (Ed: Knorz, Krause and Womse-Hacker), Konstanz: Universitätsverlag Konstanz, 0-26.
- Maite Taboada and William C. Mann. (2006). Applications of Rhetorical Structure Theory. *Discourse Studies*, 8(4):567-588.
- Keith Vander Linden. (2008). A Dynamic Programming Approach to Document Length Constraints. In *Proceedings of the Fifth International Conference on Natural Language Generation*, June 12-14, 2008, 177-180.