RST-Tool: An RST Analysis Tool

Michael O'Donnell
Department of AI, University of Edinburgh,
80 South Bridge, Edinburgh. EH1 1HN, UK.
email: micko@aisb.ed.ac.uk

Abstract

RST-Tool is a graphical interface for marking up the rhetorical structure of a text. This tool aides in both the segmentation of text, and the graphical linking of these segments into an RST-tree. The user can intermix both mononuclear or multinucler structures. McKeon-style schemas can also be included. The analysed text can be saved either as postscript for inclusion in documents, or exported for statistical analysis.

1 Introduction

Rhetorical Structure Theory (RST: Mann & Thompson 1987, Mann *et al.* 1993) is a theory of discourse structure used widely through the text generation community, and the wider discourse community. RST analyses a text in terms of a dependency tree, with each node of the tree being a segment of text. Each branch of the tree represents the relationship between a node (a nucleus) and an item of text whose occurrence is dependent on that text (the satellite). Figure 1 show the RST analysis of a short text (from Mann & Thompson 1987).[1]

Performing an RST analysis of a text by hand is a messy business. One starts to draw lines between bits of text, changes ones mind, crosses out the lines, draws new lines, and before long, one has an unreadable mess. Putting the whole process on the computer can simplify the process, making the analysis quicker, and allowing analyses to be altered without too much mess.

In my present project, Ilex,[2], we need to RST-analyse a substantial body of text, for use in a generation system which intermixes generated and (annotated) canned text. The analysis also needs to be entered into the computer in a machine-readable form. Rather than using text-based entry methods, we have developed a graphical interface to facilitate the analysis and markup of RST structure. This paper describes this tool, which we call *RST-Tool*.

Using this tool, one simply drags the mouse between segments of text to establish a relation between segments, and is then offered a list of labels to apply to that relation. Complex text structures can thus be analysed quickly. The interface is easy and intuitive to use.

Another application of the tool revolves around its use in *variable-length document presentation* – on-line documents whose length can be adjusted to the user's demands. Text marked-up using the RST-Tool can then be presented on the web, by a program which knows how to summarise this text on the basis of its RST-structure. See my paper in this volume for details.

The tool is used in two stages. Firstly, the user marks the segment boundaries throughout the text (see section 2). Secondly, the user graphically links these segments together into an RST-tree (see section 3). Each of these tasks has a separate interface within the tool.

[1]The RST figures throughout this paper are screen-dumps from the RST-Tool. The RST-Tool can export figures as postscript, but the process is currently being debugged for inclusion in Latex documents.

[2]An EPSRC-funded project, Chris Mellish and Jon Oberlander principal investigators. See Knott *et al.* (1996) for details.

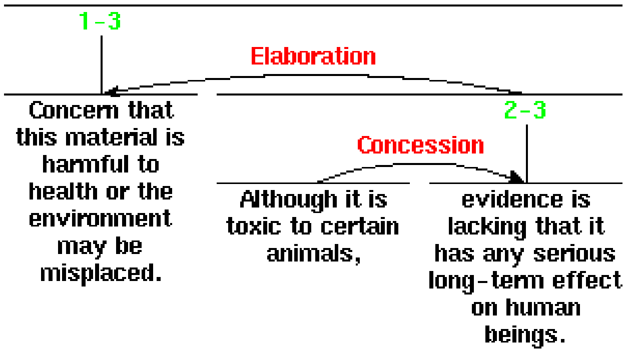

Figure 1: RST Analysis of a Short Text

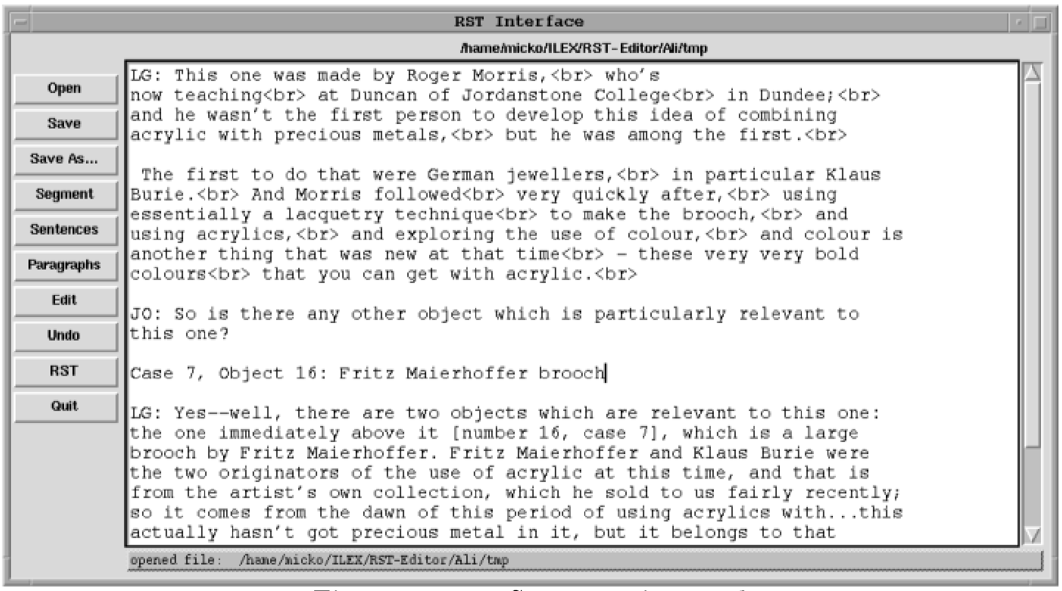

Figure 2: Text Segmentation Tool

The RST-Tool is written in Tcl-Tk, and is freely available for Unix, Mac and PC platforms. See http://www.dai.ed.ac.uk/staff/personal_pages/micko/RSTTool/ for details.

2 Text Segmentation

The text segmentation interface is shown in figure 2. Using this interface, the user inserts '
' throughout the text at each point where a segment boundary should occur. In "Segment" mode, simply clicking on a desired segment boundary inserts this marker. However, if the user desires to edit the text (modifying the text, correcting spelling errors, etc.), clicking on the "Edit" button shifts to edit-mode, allowing the text to be modified.

To reduce the number of clicks, the RST-Tool can automatically detect sentence boundaries. Clicking on the "Sentences" button automatically inserts a break mark between each sentence. Unfortunately, detecting clause boundaries within a sentence is more complex, and is not yet supported.

This system doesn't yet allow *embedded elements* – rhetorically dependent text-segments which occur within the dominating text segment. For instance, we might wish to treat the embedded clause in the following as dependent on the main clause: *John, – I think you know him – is here for two weeks.* At present, the interface does not handle such cases. A simple solution is for the user to move the embedded text outside of the enclosing text.

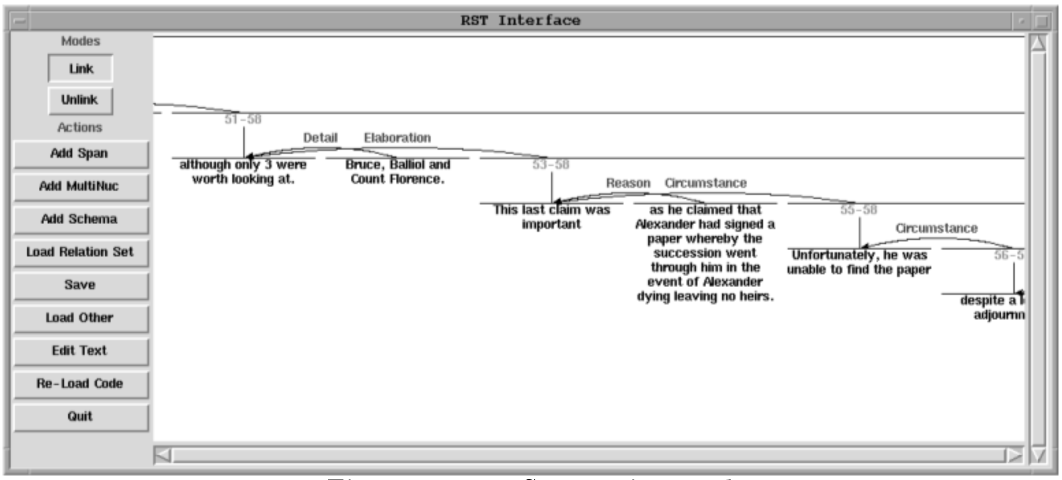

Figure 3: Text Structuring Tool

3 Text Structuring

The second step of RST Analysis involves structuring the text. The *text structuring* interface of the RST-Tool allows the user to connect the segments into a rhetorical structure tree, as shown in figure 3. For presentation, the RST-Tool follows the graphical style presented in Mann & Thompson (1987).

Initially, all segments are unconnected, ordered at the top of the window. The user can then drag the mouse from one segment (the nucleus) to another (the satellite). Upon releasing the mouse button, the system offers a menu of relations to choose from (the user can use the relation-sets provided with the system, or provide their own).

The user can switch freely between text segmentation and text structuring mode – to edit text, or to change segment boundaries. The system keeps track of the structure assigned so far. If the user, in editing the text, deletes a segment, the system forgets structuring information concerning that segment.

3.1 Types of Structure

Apart from simple nuclear-satellite relations, the tool allows a number of other options:

- **Multi-nuclear Relations**: such as JOINT and SEQUENCE (see figure 4).

- **Text Scoping**: sometimes we might want to treat a nucleus-satellite complex as itself the nucleus (or satellite) of a relation. The RST-Tool allows you to insert a *span* node over a nucleus and its sub-structure. This span-node can then fill either nucleus or satellite roles. For instance, the spans labelled "1-3" and "2-3" in figure 1 were created in this way.

- **Schemas**: Pure RST would seem to analyse a text into a single top-nucleus, with all the rest of the text being a satellite of that unit, or a satellite of a satellite, etc. Multi-nuclear relations alleviate this weak premise somewhat, but are not the whole solution. Being unhappy with this situation, we added the capability to use McKeon-style *schemas* in the RST-Tool, what are sometimes called "story grammars". These allow a sequence of named elements of structure, e.g., INTRODUCTION, BODY, CONCLUSIONS, BIBLIOGRAPHY, etc.(see figure 5). We find that the macro-structure of a text is best modelled with schemas, while the lower levels are captured through RST relations.

- **Clause-Internal Structure**: in my work on rst-based text summarisation, I have found that better quality summarisation can be achieved if we push RST-analysis into the clause,

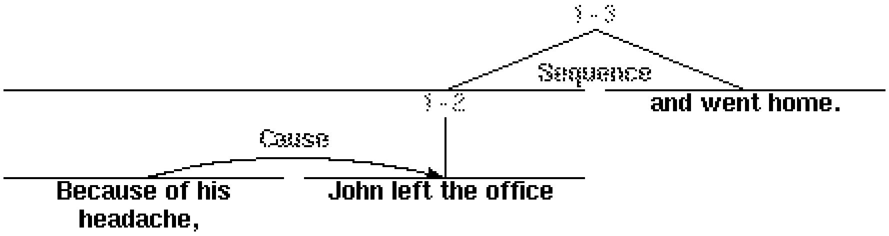

Figure 4: Scoping and Multi-nuclear Relations

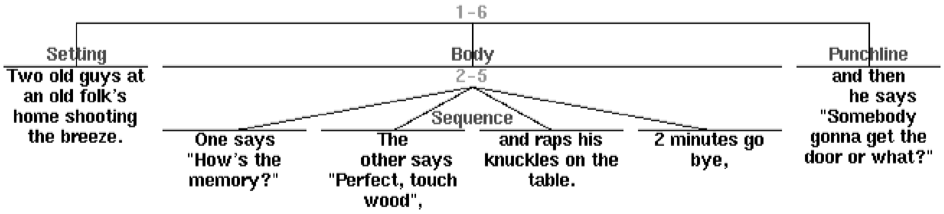

Figure 5: Constituent Structure

and allow the clausal adjuncts to be treated as satellites of the nuclear clause itself. For instance, (N: *Edward surrendered,*)(S: *in 1245*). Some of these adjuncts can be connected to the clause with standard RST relations, but many can not. A set of new relations, borrowed from the Systemic labelling of adjuncts (cf. Halliday 1985), has been added for this reason.

3.2 Defining Relations

When forming a relation between two text-nodes, a menu pops up prompting the user to specify the relation-type. These relations are not built into the system, but can be edited by the user. The user modifies a text file to add, delete or modify the relation set. A sample set derived from Mann & Thompson (1987) is included with the distribution.

3.3 Other Points

Because rst-structures can become very elaborate, the RST Tool allows the user to *collapse* sub-trees – hiding the substructure under a node, This makes it easier, for instance, to connect two nodes which normally would not appear on the same page of the editor.

The user can save the present state of the screen as postscript, for inclusion in Latex documents. Alternatively, a snapshot utility can be used to save selected parts of the structure in other formats. The structured text can be saved to a file, for later re-editing, or for use in variable-length document presentation.

4 Conclusions

This paper has presented the RST-Tool, a graphical interface for marking up the rhetorical structure of text. The tool is useful for several reasons:

1. **Reducing Analysis Time**: the tool reduces the amount of time taken to analyse text, compared with hand-analysis, or text-based entry. The tool allows easy creation of structures, and modification if needed.

2. **Figure Preparation**: The tool provides an easy way to prepare figures depicting RST structure for inclusion in documents.

The RST-Tool has also proved useful in its role as document preparation software for variable-length document presentation. While some summarisation systems attempt to use automatically recognition of RST structure, software has not yet been developed which provides reliable automatic recognition of RST structure over unrestricted text. However, good progress in this direction is being made by Ono *et al.* 1994; Marcu (1996); Hoffman (1996), etc.

5 References

Halliday, M.A.K. 1985 *Introduction to Functional Grammar*. London: Edward Arnold.

Hoffman, Beryl. 1996. "Summarization: an Application for NL Generation ". Proceedings of the 8th International Workshop on Natural Language Generation, Herstmonceux Castle, UK, 13-15 June.

Knott, Alistair, Chris Mellish, Jon Oberlander & Mick O'Donnell. 1996. "Sources of Flexibility in Dynamic Hypertext Generation". Proceedings of the 8th International Workshop on Natural Language Generation, Herstmonceux Castle, UK, 13-15 June.

Mann, William & Sandra Thompson, 1987. "Rhetorical Structure Theory: A Theory of Text Organization". Technical Report ISI/RS-87-190.

Mann, William, Christian Matthiessen and Sandra Thompson. 1993. "Rhetorical Structure Theory and Text Analysis". In Mann, William and Sandra Thompson (eds.) *Text Description: Diverse Analyses of a Fund Raising Text.* Amsterdam: John Benjamins.

Marcu, Daniel. 1996. "Building up rhetorical structure trees". Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96). August 4 - 8, 1996, Portland, Oregon.

O'Donnell, Michael. 1997. "Variable-Length On-Line Document Generation". Proceedings of the 6th European Workshop on Natural Language Generation March 24 - 26, 1997 Gerhard-Mercator University, Duisburg, Germany.

Ono, Kenji, Kazuo Sumita, & Seiji Miike. 1994 "Abstract generation based on rhetorical structure extraction". Proceedings of the 15th International Conference on Computational Linguistics (COLING-94), Vol. 1. August 5-9, Kyoto, Japan.