semantics. Semantics are modelled only at the GTO level. Beyond that, at the level of individual thesauri, semantics are user defined in syntax extensions to relation types or pushed down into the application layer. We claim this as a strength, although it could be considered as a weakness from a theoretical language stance. But it is by this simplicity that we gain our generality — by concentrating on the high-level regularities and leaving low-level peculiarities to the syntax. This gives TML a tremendous advantage over languages understood only by computer scientists. We believe that the users and maintainers of document management systems should not need to have strong technical backgrounds to do their jobs.

Our aim with TML is to represent thesauri in a practical implementable way. The TML syntax is constrained through the use of a schema, but the schema does not fully specify the language; ie., we did not attempt to include all possible thesaural semantics or to prevent all representational errors within the syntax. The verification of compliance of a thesaurus instance to the model of the GTO requires some data validation to be carried out at the application level. This is also necessarily true for the class type and relation type extensions of a particular thesaurus. This is less of a danger than it might appear, because we expect such processing validation to occur in the TML authoring tools.

The choice of XML as the TML syntax is overwhelmingly pragmatic. XML is sufficient to our requirements, an open international standard, and is emerging as a software modelling standard. It is ubiquitous and can be understood and authored without great training. It allows TML maintenance and parsing tools to leverage the power of many off the shelf authoring products.

7 Conclusions

We have demonstrated that general thesaural support is feasible by designing a generic thesaural ontology and markup language that amalgamates different thesauri structure and allows us to represent the idiosyncrasies of specific thesauri in a common language. This permits general purpose thesaural tools such as our Thesaural Explorer to be built. These tools can work with many thesauri thereby leveraging development costs, providing a common user interface, and supporting flexible thesaural maintenance and evolution. Such tools permit document management systems to better organise and access repository content.

8 Acknowledgments

The work reported in this paper has been funded in part by the Research Data Networks (RDN) Co-operative Research Centre (CRC) program, Australia.

9 References

[AAT] Getty Art and Architecture Thesaurus, URL: http://www.gii.getty.edu/aat_browser

[Bradley 1998] Bradley, N. The XML Companion, Addison-Wesley, 1998.

[Dublin Core] An International recognised core set of metadata elements. URL: http://metadata.net.de/rdf/DC/

[Farquhar, Fikes and Rice 1996] Farquhar, A., Fikes, R., and Rice, J. The Ontolingua Server: a Tool for Collaborative Ontology Construction, Knowledge Acquisition Workshop, 1996.

[Fensel *et al* 1998] Fensel, D., Erdmann, M., Studer, R. Ontobroker: The Very High Idea, *Proceedings of the 11th International Flairs Conference, 1998*.

[Finin et al 1994] Finin, T., Fritzson, R., McKay, D. and McEntire, R. KQML as an agent communication language, CIKM'94 in the proceedings of the third international conference on information and knowledge management, 1994, 456-463.

[Glushko, Tenenbaum, and Meltzer 1999] Glushko, R., Tenenbaum, J., and Meltzer, B. An XML Framework for Agent-based E-Commerce, Communication of the ACM, March 1999, Vol 42, No 3, 106, 114.

[Harold 1998] Harold, E. XML: Extensible Markup Language, IDG Books Worldwide, 1998.

[ISO 2788] International Organisation for Standardisation (ISO) 2788, Documentation Guidelines for the establishment and development of monolingual thesauri, 1986.

[Keyword AAA] *Keyword AAA: A Thesaurus of General Terms*, Archives Authority of New South Wales, Sydney, 1995.

[Lancaster 1972] Lancaster, F., Vocabulary Control for Information Retrieval, Information Resources Press, 1972.

[LCSH] Library of Congress Subject Headings, URL at: http://www.loc.gov

[MeSH] Medical Subject Headings, URL at: http://www.nlm.nih.gov/mesh/filelist.html

[OCLC] OCLC Dewey Decimal Classification, URL at: http://www.oclc.org/oclc/fp/index.htm

[RDF] The Resource Description Framework for metadata syntax and interoperability, URL: http://www.w3.org/Metadata/RDF/

[TGN] Getty Thesaurus of Geographic Names, URL at: http://www.ahip.getty.edu/tgn_brower

[XML] Extensible Markup Language, URL: http://www.w3.org/XML/

[Zthes 1999] Z39.50 Profile for Thesaurus Navigation, URL: http://www.n-four.demon.co.uk/mirk/zthes-02.html

Building rich metadata from critical reviews for a scrutable filtering system

Sacha Groves

Basser Dept of Computer Science
University of Sydney
AUSTRALIA 2006

sacha@cs.usyd.edu.au

Judy Kay

Basser Dept of Computer Science
University of Sydney
AUSTRALIA 2006

judy@cs.usyd.edu.au

Abstract

We describe the Review Coder system for creating rich metadata for a scrutable filtering system. A scrutable system maintains explanations of the data and processes that drove the system operation. In the current paper we use Review Coder as part of a filtering systems for movies: the scrutability of the system means that a user can determine why the system recommended a particular movie or not.

The filtering process is based upon movie reviews and metadata built in association with them. These provide high quality information about the movie objects. From these, the filtering system is intended to build stereotypic models of reviewer's preferences for movies. These can drive the filtering process and the user can scrutinise both these models and the actual reviews which were used to construct them.

Keywords: Multimedia Resource Discovery, Multimedia Filtering, Scrutable Filtering, Extraction of Metadata

1. Introduction

As electronic objects become increasingly accessible, there is a growing need for tools that assist users in filtering large collections of objects so that the user can find objects of interest. The effectiveness of a filter depends significantly upon the quality of the metadata describing the objects. Accordingly, this establishes a critical role for tools which assist in the creation of high quality metadata. The importance of such tools is indicated by the vigorous efforts to create a range of tools. These tools support a range of tasks, including, for example: creation of metadata templates with tools such as Dublin Core Metadata Template (Koch, Borell, and Berggren, 1998) or the discipline specific tools like Medical Metadata Creator (, 1999); tools such as Mantis (Shafer, 1998) which manage templates and assist in production of metadata.

Filtering for large video objects such as movies is important since there is a large cost for 'browsing' such objects compared with simple text objects. This cost is both in terms of the user's time and in the bandwidth required to deliver a segment of the object suitable for browsing.

Because the effectiveness of filtering depends so heavily upon the quality of the metadata, there seems to be promise in developing collections of very rich metadata for movie objects. An indication of the interest in this area is the number of online resources about movies, such as the Internet Movies Database (Database, 1999, Guide, 1999, Finder, 1999).

Such resources can be regarded as metadata suitable for human analysis as a basis for manually selecting or filtering movies. Equally, there is considerable activity in automation of the processes, for example, development of schemas for representing movie metadata. (Hunter and Iannella, 1998, Hunter and Armstrong, 1999).

An important model for assisting in filtering involves a three stage process: define a metadata structure; allow various providers to create metadata; allow various providers to create filtering tools which operate by using the metadata. A good example of such a model is PICS (Resnick, 1998) which provides a specification for metadata intended for rating objects so parents and teachers can filter out objects which are unsuitable for their children. It

Proceedings of the Fourth Australasian Document Computing Symposium, Coffs Harbour, Australia, December 3, 1999.

enables specification of a rating service's vocabulary and scales, the format of the metadata labels and a format for filtering rules.

Note that in a PICS-style model, the filtering process will allow some objects to pass the filter because their metadata meets specified requirements. So, an explanation of the system's selection (or rejection) of an object would be in terms of three elements:

- the metadata for the object;
- the assumptions about metadata which are appropriate for a particular user;
- the actual process used by the filter to implement the intended filtering policy.

We want to build *scrutable* filtering systems, which can provide the user with access to all the elements of process so that they can determine the answer to high level questions like these. Why did the filter allow *Star Trek* through the filter? Why did it fail to allow *The King and I* through the filter? At a lower level the user should be able to find answers to questions: of these types:

A. What does the filter *believe* about me? which means: What is the filter's user model for me? and What does that model mean?

B. What does the filter *know* about *Star Trek*? which actually means: What is the metadata for this object? What does that metadata mean? What do its values mean?

C. How does the filter decide whether to filter out *Star Trekl* which means: How does the filter make use of the metadata and user model to accept or reject an object?

Our previous work on scrutable user models (Kay, 1995, 1998) provides a foundation for supporting answers to questions of type A. The current work is an extension of scrutability support. We wish to support users in understanding how the system's model of them is actually used in a system like a filter.

This paper describes a project which uses critical movie reviews as a foundation for a system which combines our goals for a source of high quality metadata for movie objects and scrutability. Essentially, our goal is that the typical critical reviews can be used to create conventional structured metadata which is suitable for use in a simple filter. At the same time, the review can serve as a basis for the explanation process associated with the metadata. So the answers to the questions of type B would take the form of: a summary of the structured metadata and the parts of the reviews which were the basis of the metadata.

Of course, for such a process to work, we need to enhance the reviews with the definition of metadata. We envisage that this might be best done by the reviewer. But it might equally be done by another person. It might even be done automatically be a natural language understanding system (although this would be a difficult problem and we are not exploring this possibility). We believe that the reviewer might be willing to invest a small amount of additional effort in creating the metadata a system might require. We also believe that there are advantages in maintaining the actual critical review in addition to the metadata. Firstly, the review is a creative work with its own value and it conveys more than the minimal information needed by a filter. Indeed, we see this project as building upon a foundation assumption that the critical review is a valuable piece of metadata for people to use. Our second reason for keeping the review as well as the structured metadata is that the review serves an invaluable role for scrutability of the system.

2. Overview of the system

Figure 1 illustrates the process involved in the first stage of the filtering system. An object such as a movie is indicated at the far left. Several reviewers write natural language free-form reviews such as those typically found in newspapers or on well known movies web sites (Ebert, 1999, Limited, 1999) as well as at our own site (Pak, 1999).

These reviews are represented in Figure 1 by the column of four reviews. The figure shows them linked from the actual movie since they are, essentially, metadata about that movie.

The next step illustrated is the definition of simple metadata tags for each review. These are shown in the middle column of the figure. Essentially, this is the production of metadata for the review: each review is the metadata for the movie as seen through the eyes of a particular reviewer. This metadata establishes each reviewer's perception of the central aspects of the movie: its positive and negative attributes, other properties and similarities with other movies. Note that one of

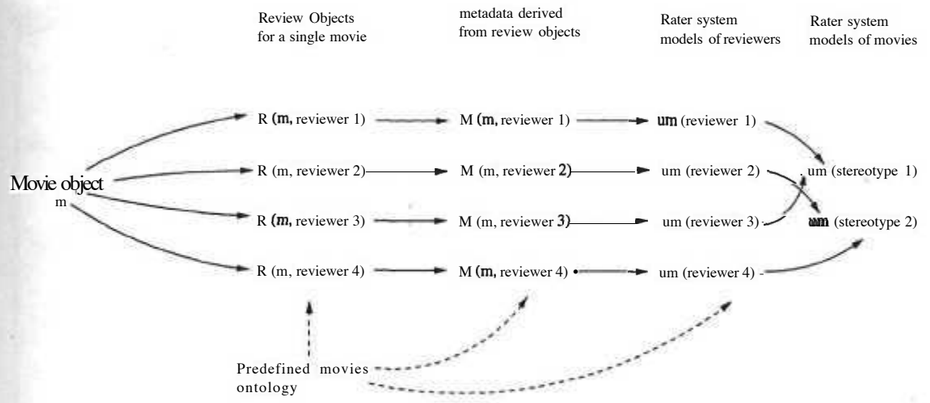

Figure 1. Overview of process to construct a stereotype for movie filtering

these reviewers might be a person involved in the production of the movie. Normally, the reviews and metadata for the reviews are produced by an independent critic.

The figure shows that the reviewer's construction of the metadata relies on a predefined movies ontology. This restricts the metadata to the set of terms which are to be used for the scrutable filter. This is important for three main reasons.

- As in the case of any filtering system, we need to ensure that the objects are coded using terms which will be usable for the subsequent filtering process and so they must be used consistently across movies and user models.
- The process is to be done by a person whose primary concern is the reviewing of a movie and we do not want to burden them with the task of deciding which terms to use.
- Finally, and critically for scrutability, we require explanations of the terms used: one of the requirements of a scrutable user model is that the user be able to access explanations of the meaning of the components of the model and these will be the terms used for the metadata. To ensure that such explanations will be available, we need to restrict the metadata to terms for which explanations have been supplied.

This stage in the review process involves a shift of perspective. We now view the metadata created by a reviewer as a source of information for modelling that reviewer. This is represented with the accretion user model representation (Kay, 1995, 1998) which models attributes of users by keeping lists of the evidence available. Although the figure shows only one movie's review and metadata feeding into the reviewer's user model, in fact, all the reviews that they have done feed into the system's model for that reviewer. So, for example, suppose reviewer 1 has written many reviews in which violence is assessed as a positive attribute of movies and has rarely assessed it as a negative aspect. Their user model would have a long list of evidence that they like violent movies and a short list indicating that they dislike it. Evaluation of the full set of evidence is performed by a resolver process: for scrutability, there must be an explanation for its operation.

The last stage illustrated in Figure 1 is the construction a small number of default models, the stereotypes (Rich, 1989) which provide a quick model for the 'average user'.

3. Metadata extraction interface

So we define two forms of metadata, which we call *bland* and *judgemental* metadata. Bland metadata is the largely objective characteristics of the movie, aspects like the directors, actors, genre and the like. These appear in typical semi-structured information about movies. Indeed, since such information is already available and classified in movie databases. We have followed their choice of metadata elements.

From our perspective, the more interesting issues lie with the judgemental metadata. This is a structured representation of the reviewer's assessment of the movie. For example, they may

have liked a particular element described in the bland metadata as in the case of a reviewer who liked the performance by Keanu Reeves in the Matrix. Such a piece of judgemental metadata is then associated with the relevant actor metadata for that movie. Actual text from the review will serve an explanation for this piece of judgemental metadata.

We allow judgemental metadata to take one of the following forms:

1. metadata about bland metadata elements (as in the example of a reviewer who liked Keanu Reeves in the Matrix);
2. metadata about the movie as a whole (and supported by that part of the review which gave that assessment);
3. metadata which compares this movie with others (for example, indicating that it is better than another, similar movie)

We want the user of our interface to make a pass over the review and review the automatically generated bland metadata, possibly altering it by deleting aspects which the reviewer does not consider important enough to serve as metadata for the movie. The reviewer may also add new metadata. Usually, however, the reviewer will use this metadata as a basis for creating the first form of judgemental metadata in the above list.

In a later pass through the metadata, the user should link each element of the metadata to the relevant text in the review.

There are two main reasons for this design. First, our concern for scrutability means that we prefer to use simpler processes for filtering if possible: these seem to be the most promising foundations for providing easily understood explanations of the system operation. Secondly, and equally importantly, we want to leave reviewers quite free to write whatever they choose and then to decide on the metadata they want to define for that review.

The Review Coder interface is, in essence, an editor for creating metadata from movie reviews. The next three figures show example Review Coder screens. The interface has two main parts. At the left is a large text area for the input of reviews. The right part of the screen is for creating the structured metadata. At the top is a list of the structured elements for the metadata. For example, Figure 2 is a start screen for developing the metadata for the film "The Matrix" where the required elements of the structured metadata are:

Identifier:matrix_the_1999
Director:Larry Wachowski
DirectorAndy Wachowski

In addition, some examples of optional elements are:

Actor: Keanu Reeves (Neo)
Feature:special effects
Location:Sydney, Australia
Plot:A computer hacker suddenly learns from mysterious rebels that his whole reality is not what it appears to be at all
Similar:alice_in_wonderland
Similar:bound_1996
Similar:dark_city_1998
Similar:wizard_of_oz

Figures 2 and 3 show the interface with completed reviews. The list at the bottom right of the interface shows the metadata elements that were identified within the review with notation indicating whether the metadata element was considered

- good(+),
- bad (-),
- not as good as another film (<),
- better than another film (>)or
- as good as another film (=).

The * operator indicates that the metadata element was mentioned in the review, but that the review made no judgement about that element. For example, here is the metadata coded for a review:

+Feature: special effects
*Identifier:david_stratton
*Location:Sydney, Australia
-Plot:A computer hacker suddenly learns from mysterious rebels that his whole reality is not what it appears to be at all
*Similar:alice_in_wonderland
<Similar:bound_1996
=Similar:dark_city_1998
*Similar:wizard_of_oz

The reviewer types their review for the movie in the text box. They then select the metadata elements that they mentioned in their review from the list and code each element, one by one, using the '+','-' etc buttons.

4. Evaluation

The interface we have shown in the paper was a prototype. We have performed a small, scale formative evaluation. It aimed to assess:

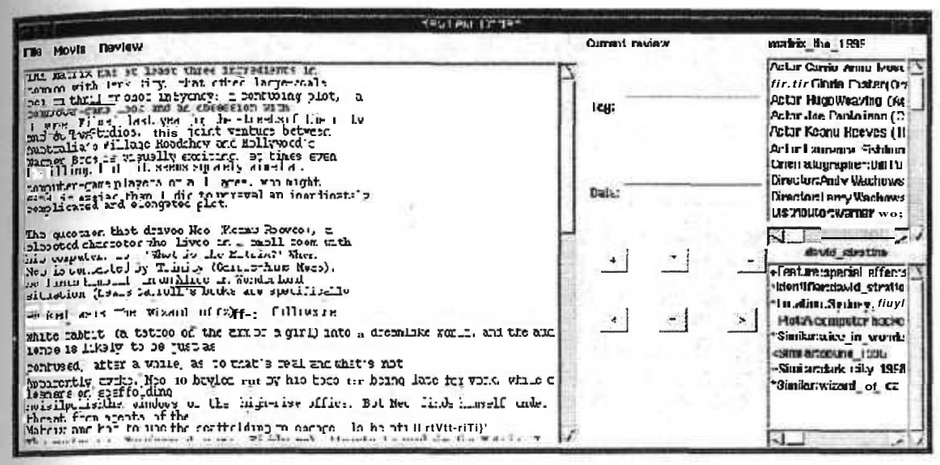

Figure 2. Interface with a review of Matrix by David Stratton

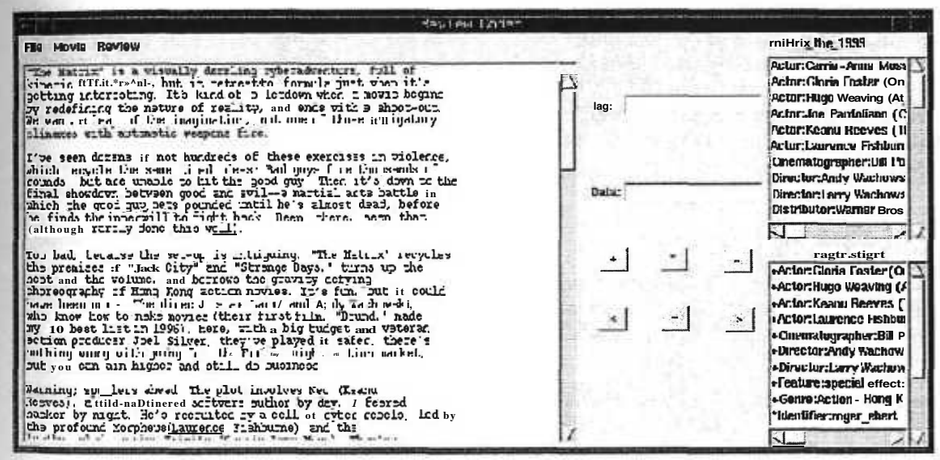

Figure 3. Interface with a review of Matrix by Roger Ebert

- the ease of use of the interface for coding a review, given a brief set of instructions;
- whether users could code a critic's review as a set of structured elements;
- the consistency in the coding between users.

The users (who we will also refer to as participants) were all asked to code a single movie review, Roger Ebert's Review of the movie "The Matrix".

The experiment took the users through the following steps.

1. Brief instructions in use of the interface.
2. The participant read the review carefully.
3. Then the participant went through the supplied metadata terms for the movie, coding the review.
4. During the process, the participants commented on what they were doing and, in keeping with the cooperative think-aloud approach (Monk, Wright, Haber, and Davenport, 1993) suggested improvements to the interface.

The participants were three Computer Science postgraduate students. All had completed a

course in user interface design and were avid movie buffs. This choice of participants means that we need to take care in interpreting this evaluation. The choice of movie buffs was desirable since we wanted the coding to be done by people who would be likely to appreciate the subtleties of the review and to take account of the broad range of elements that reviewers like Ebert mention. A side-effect of selecting movie buffs was that all had actually seen the movie before this evaluation experiment.

On the other hand, the strong computer science background of the participants makes their assessment of the interface closer to that of an expert. Since our main concern was to assess the effectiveness of the overall coding strategy, these users are a satisfactory population for a formative assessment.

Overall, the users found the interface straightforward to use. They completed the task with minimal awareness of the interface, being able to focus on the problems of deciding on the meaning of the review and how to code it. The experiment showed consistency for the '<' and '>' codings: there was no case where one user coded an aspect as '<' and another as '>'. However, different users chose to code different aspects.

5. Discussion and conclusions

The evaluation was purely formative. It indicates that users are able to attempt the task for which the interface was primarily designed: quick and simple creation of judegmental metadata.

Much remains to be done. The next phase of this research will improve the support for scrutability. The interface will be altered so that the creation of metadata will operate as follows:

- the user will be able to highlight text within the review that serves as the basis for a particular piece of metadata.
- they would then use parts of the current interface to create the metadata tag.

For example, the text "Keanu Reeves turned in a good performance" would be selected and the metadata created would set "Actor:Keanu Reeves" as "+".

Currently, the Review Coder system stores the metadata in a plain text file in the Tag:Data format. To accommodate the tagging of the evidence, we will explore use of the Resource Description Framework (W3C, 1999). Not only is RDF the current leading choice for metadata definitions, its use of XML enables the tagging of the review to be treated as a markup of the text. What is more, research into motion image metadata has suggested a Dublin Core (Core, 1999) and MPEG-7 hybrid (MPEG, 1998) using RDF as the framework to be suitable(Hunter and Iannella, 1998, Hunter and Armstrong, 1999). We are also exploring other metadata representations for objects similar to movies (CDWA, 1999). The current interface ontology is based upon analysis of metadata used by such sources as the Internet Movie Database (Database, 1999).

References

Medical Metadata Creator, http://medir.ohsu.edu/bicc-informatics/ebm/latest.htm, 1999.

CDWA, *Categories for the Description of Works of Art (CDWA),* http://hul.harvard.edu/ldi/html/metadata.html, 1999.

Dublin Core, *Dublin Core Metadata Initiative,* http://purl.org/DC, 1999.

Movie Database, *Internet Movie Database,* http://www.imdb.com, 1999.

R Ebert, *Roger Ebert on Movies,* http://www.suntimes.com/ebert/ebert.html, 1999.

Movie Finder, *Movie Finder,* http://www.moviefinder.com, 1999.

All Movie Guide, *All Movie Guide,* http://www.allmovie.com, 1999.

J Hunter and L Armstrong, *A Comparison of Schemas for Video Metadata Representation,* WWW8 http://www8.org/w8-papers/3c-hypermedia-video/comparison/comparison.html, 1999.

J Hunter and R Iannella, "The Application of Metadata Standards to Video Indexing," *Second European Conference on Research and Advanced Technology for Digital Libraries,* Crete, Greece, 1998.

J Kay, "The um toolkit for cooperative user modelling," *User Modeling and User-Adapted Interaction,* vol. 4, no. 3, pp. 149-196, Kluwer, 1995.

J Kay, *A scrutable user modelling shell for user-adapted interaction,* PhD Thesis, Basser Department of Computer Science, University of Sydney, Australia, 1998.

T Koch, M Borell, and M Berggren, *Dublin Core Metadata Template,* http://www.lub.lu.se/metadata/DC_creator.html, 1998.

News Limited, *News Limited Reviews,* http://entertainment.news.com.au/filmarc, 1999.

A Monk, P Wright, J Haber, and L Davenport, *Improving your human-computer interface: a practical approach,* Prentice Hall, 1993.

MPEG, *MPEG-7: Context and Objectives (version - 10 Atlantic City),* http://drogo.cselt.stet.it/mpeg/standards/mpeg-7/mpeg-7.htm, 1998.

J Pak, *Zoo/City Movie Reviews,* http://www.zoocity.com.au/review.html, 1999.

P Resnick, *Platform for Internet Content Selection (PICS),* http://www.w3.org/PICS/, 1998.

E Rich, "Stereotypes and user modeling," in *User models in dialog systems,* ed. by A Kobsa and W Wahlster, pp. 35-51, Springer-Verlag, Berlin, 1989.

K Shafer, *Mantis, A Flexible Cataloging Toolkit,* http://orc.rsch.oclc.org:6464/toolkit.html, 1998.

W3C, *http://www.w3.org/RDF,* 1999.