EASING THE ONTOLOGY DEVELOPMENT AND MAINTENANCE BURDEN FOR SMALL GROUPS AND INDIVIDUALS

Roger Tagg, Harshad Lalwani and Raaj Srinivasan Kumaar
School of Computer and Information Science, University of South Australia, Mawson Lakes SA 5095, Australia

Keywords: Ontologies, Group Work Support, Personal Information Management.

Abstract: Most attempts to aid overworked knowledge workers by changing to a task focus depend on the provision of computer support in categorizing incoming documents and messages. However such categorization depends, in turn, on creating - and maintaining - a categorization scheme (taxonomy, lexicon or ontology) for the user’s (or the group’s) work structure. This raises the problem that if users are suffering from overload, they are unlikely to have the time or expertise to build and maintain an ontology – a task that is recognized to be not a trivial one. This paper describes ongoing research into what options may exist to ease the ontology management burden, and what are the advantages and problems with these options.

1 INTRODUCTION

In a paper at the last ICEIS CSAC workshop (Tagg, 2007), we described a number of studies carried out by the authors’ research team, which addressed the difficulties, faced by many knowledge workers, of coping with an avalanche of unsorted and un-prioritized input information from a variety of sources and in a variety of applications and formats.

In that paper we described our approach as to re-focus the user’s interface to a single "to do" list, rather than multiple, disparate interfaces. We proposed achieving this with the aid of a personal ontology representing the user’s work structure.

We have developed an ontology editor that is able to include “indicator strings”, i.e. text strings which, if found in a document or message, indicate – with a certain subjective probability – that this document is relevant to a given ontology concept. We are now prototyping an email categorization tool that can take account of such relationships between text and concepts.

However if such an approach is ever to make a positive difference to the majority of real world users, we have to ensure that the burden placed on users to create and maintain their ontology – including the indicator strings – is not too onerous.

This paper describes some investigations that we have been recently undertaking into this last issue. Section 2 reviews related work on approaches to ontology creation and maintenance. Section 3 describes the actual experiments we have carried out, and our comments on the results. Section 4 introduces a range of theoretical models which we plan to test as the work proceeds. Section 5 contains reflections on the work done so far, and Section 6 outlines the work that still remains to be done.

2 RELATED WORK

2.1 Text Mining and Content Analysis

Our intention was to base our automated assistance to personal ontology creation on the text mining software Leximancer (Smith, 2003), developed at the University of Queensland. This tool “analyzes the content of collections of textual documents and displays a summary by means of a conceptual map that represents the main concepts contained within the text and how they are related”. It also has the “ability to automatically and efficiently learn which words predict which concepts”. Leximancer incorporates algorithms for the learning of concepts from frequent co-occurrences of words that appear near to each other.

Commercial text mining or content analysis tools are also available, such as Text Miner and Smart Discovery. Common to this class of tools is an orientation towards text appearing in the media (e.g.
newspapers) and the objective of finding out what
the main themes of that text passage are.

Text mining has also been proposed as an
approach to creating and improving ontologies, e.g.
(Dittenbach et al., 2004) (Cimiano and Völker,
2005). However, although this is directly relevant to
our goals, there is not so much emphasis on
identifying those text strings that most reliably
suggest the relevance of an incoming document to
an ontology concept.

2.2 Semi-automatic Ontology
Generation

This is a research topic closely related to text
mining, but with ontology generation as the primary
purpose. A representative example of a tool is
OntoGen (Fortuna et al., 2005). This works through
a dialogue in which the user is presented with a
number of windows, which show the current concept
hierarchy, a diagram of the ontology, and
suggestions for further sub-concepts that is based on
occurrence statistics of related keywords. A
“grounding” module allows validation by testing
how certain test documents are classified by the
system compared with their classification by domain
experts.

Other work in this area is being done by Wang et
al (Wang et al., 2007) of the AIFB group at the U.
Karlsruhe, where a mixed approach is proposed,
blending the Text2Onto text miner with the KASO
manual ontology development tool.

2.3 Personal and Small Group
Ontologies

If ontologies are to be used by an individual or a
small group for categorizing documents and
messages, it may not be appropriate to use a
generalized ontology for the domains of interest.
The structure of the individual’s or group’s working
practices and objectives must also be included. Standard ontologies could be imported, but they will
not in general reflect the full range or right balance of
user interests. Individuals may, for example, be
involved in multiple groups (Tagg, 2006).

OntoPIM (Lepouras et al., 2006) (Katifori et al.,
2006) is an example of a system geared to
individuals and small groups. It is clearly task-
oriented, and is part of the DELOS TIM project
(Catarci et al., 2007). The OntoPIM concept of
Semantic Save is effectively an automatic tagging of
input documents across many applications and file
formats. It includes a system for the mapping of the
values of significant attributes into standard tags.

However this system is further "downstream" than
our current concern, which is how to generate and
maintain the ontology in the first place.

3 WORK DONE AND IDEAS
DEVELOPED

3.1 Overall Architecture

The work described here is part of an overall project
titled Virtual Private Secretary (VPS). The
motivation is to provide through software, for users
and groups that cannot afford a human PA (Personal
Assistant, or Secretary), some of a Secretary’s
functions in helping a boss or group to cope with a
heavy and diverse knowledge workload.

The overall architecture for the VPS project is
shown in Figure 1. It takes on board the concept of
many-to-many group membership (Tagg, 2006),
which recognizes that many users have to multi-task
work for multiple groups in the same time period.
as to correctly represent his/her current activity structure.

However as we suspect that most users may have neither the time nor the expertise to do this, we are looking at two approaches (see 3.2 and 3.3 below) to generating an ontology semi-automatically. A further advantage of an automated approach is that it could be run at given intervals (e.g. every 3 months) or whenever the user indicates that the nature and balance of his/her work has changed, thus helping maintenance as well as creation.

3.2 Identification of a User’s Work Categories through Text Analysis

In the first approach we have used Leximancer to analyze the patterns of words and concepts in a user’s email archives, a) where the archives have been pre-categorized and b) where no categorization has taken place and messages from all topics are intermixed.

a) Pre-categorized Email Archives
Archives were saved from one academic’s Outlook local folders into a set of text files. We ran Leximancer separately for each sub-category of his Teaching and Research categories. We then merged the results onto a single spreadsheet for each major category. Table 1 below shows part of the spreadsheet for the Teaching category. The columns represent the sub-categories. Only the top 20 words for each sub-category were included.

Although Leximancer offers a default stop word list, we decided to manually add to that list noise words that we judged not to be good indicators of the Teaching category; these are highlighted in yellow. We re-ran the analysis and there was some improvement, but a new set of noise words appeared, which were again added to the stop word list. We suspect that this process might have several iterations, and were concerned that we might finish up with different noise words for each major category, but when we repeated the Leximancer analysis for the Research category, only one word was different.

b) Uncategorized (Mixed Topic) Email Archives
Leximancer offers a facility to propose Themes (clusters) of concepts from an analysis of single text document, so we have tried this on a mixed-topic email inbox. The central part of the resultant map is shown in Figure 2 below.

The circles represent suggested clusters, with the concepts (in white) placed according to the closeness of their co-occurrence. Associated tables are available that show the actual co-occurrence statistics of each concept and of the actual words in the text.

Table 1: Partial Summary Spreadsheet for the Teaching Category.

<table>
<thead>
<tr>
<th>Words</th>
<th>Workflow</th>
<th>IEO</th>
<th>ISM</th>
<th>Research Methods</th>
<th>Courses</th>
<th>Others</th>
<th>Systems Design</th>
<th>Projects</th>
<th>Teaching/Learning Programs</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>access</td>
<td>379</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>379</td>
</tr>
<tr>
<td>Andy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>approach</td>
<td></td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>apps</td>
<td></td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>assignment</td>
<td>1028</td>
<td>492</td>
<td>536</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>335</td>
</tr>
<tr>
<td>available</td>
<td>113</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>113</td>
</tr>
<tr>
<td>Blues</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>213</td>
</tr>
<tr>
<td>Bro</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>150</td>
</tr>
<tr>
<td>business</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>176</td>
</tr>
<tr>
<td>Ceo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>133</td>
</tr>
<tr>
<td>changes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>209</td>
</tr>
<tr>
<td>City_West</td>
<td></td>
<td>279</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>279</td>
</tr>
<tr>
<td>content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>ccs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>356</td>
</tr>
<tr>
<td>courses</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>218</td>
</tr>
<tr>
<td>current</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>783</td>
</tr>
<tr>
<td>data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>181</td>
</tr>
<tr>
<td>degree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>210</td>
</tr>
<tr>
<td>discussion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>208</td>
</tr>
<tr>
<td>document</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>164</td>
</tr>
<tr>
<td>email</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1304</td>
</tr>
<tr>
<td>exam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>227</td>
</tr>
<tr>
<td>external</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>440</td>
</tr>
<tr>
<td>Faculty</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68</td>
</tr>
<tr>
<td>field</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>66</td>
</tr>
<tr>
<td>finance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>351</td>
</tr>
<tr>
<td>give</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1100</td>
</tr>
<tr>
<td>group</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>570</td>
</tr>
<tr>
<td>help</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2768</td>
</tr>
<tr>
<td>hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>128</td>
</tr>
<tr>
<td>hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>38</td>
</tr>
</tbody>
</table>
While some clusters make sense for an IT academic, e.g. students, project, research and business, several others look less useful, e.g. website, review, paper and exam. One would naturally want to merge some of these, e.g. exam with students, and review with research. Similarly, one would almost certainly want to merge the concepts project and projects which have been mapped separately. This can be done using tools within Leximancer, but the user would need to understand these tools and to have the time to intervene - something one wants to minimize.

The same problem would arise in asking the user to decide which noise words should be excluded, for example in Figure 2 the words people, e-mail and website.

In our experiments, we first excluded the same set of noise words as for Teaching and Research, but we found we had to exclude more, to cater for the additional major work categories such as Administration. A separate noise word list for each user seems undesirable, but there may have to be different noise word lists for each different role or profession. Our thoughts on a solution to this have so far been limited to classing as stop words all those with low Specificity (see 4.1 below).

3.3 Identification of a User’s Work Categories through a Crawler

The second approach uses a crawler program to mine the user’s current folder structures in various places such as MyDocuments, Networked Drives and Places, Outlook Local Folders, Web Browser Bookmarks etc. A user's stored knowledge may be highly distributed, including USB drives, shared folders (e.g. MS Sharepoint folders containing minutes of meetings with Actions on individuals - these are often not read by the individuals!). However the more varieties of structures that are discovered, the more one has to reconcile possibly clashing work structures. We have not yet carried out trials with this approach.

3.4 Detection of Tasks

It has become clear in analyzing our results that emails, for example, vary widely in the extent to which they indicate a task or to-do for the user. We have termed this factor taskiness. The approach we have taken so far is to regard taskiness as an additional ontology concept, and to associate with it a set of text strings which (singly or in combination) suggest taskiness. Examples (which we selected manually from two users’ archives) include please, deadline, required by, at the latest, asap, earliest
However, it has proved difficult, with this approach, to separate important to-dos from hopeful requests and invitations (e.g. to buy something or take a questionnaire). Our analysis suggests that strings such as names in the Sender and Subject fields may be more significant.

Part of our taskiness detection method depends on the appearance of dates and times in certain text patterns. To this effect we have developed a program incorporating regular expression logic. One issue related to this, which we have yet to resolve, is whether we should include relative date/time expressions, e.g. next Tuesday, next January.

### 3.5 Identification of Task Instances

In the previous paper (Tagg, 2007) we discussed the need to recognize, and store in the to-do information, the names that identify business cases for detected tasks. One task we are currently looking at is how users of a semi-automated tool can be aided in nominating, for example, the source database tables – and columns - where these names can be looked up.

### 3.6 Identifying Other Priority Factors for the User’s To-Do List

Besides the use of text mining to feed an ontology and to detect taskiness, other factors need to be considered when setting up a system for generating a to-do list. These include the expected duration and complexity of an identified task; a user may have a limited time window in which to address his/her to-do list, and he/she may wish to give priority to tasks that can be completed quickly and easily. This would mean extending the ontology to include known task types and their attributes, possibly with some knowledge of inter-task dependencies.

### 4 THEORETICAL MODELS

#### 4.1 With no Pre-categorization

The theory underlying Leximancer seems suitable for our purpose, although it is recommended that some degree of seeding of the concepts is often required. Leximancer does cater for stop words – although as with seeding, some expertise is needed to choose a suitable list.

Some commonly occurring words and phrases appear in most documents, and their appearances are therefore of lower value in deciding to what category a document belongs. To try and reduce the noise, we plan to append to the stop word list any word that has too low a measure which we call Specificity.

Our simplistic Specificity percentage is defined as: \[100 \times \left(1 - \frac{N_i}{T}\right)\], where \(N_i\) is the number of messages/documents that word \(i\) appears in, and \(T\) is the total number of messages/documents. This measure, although extremely coarse, we believe to be adequate as long as the categories are fairly evenly balanced – unlike if, for example, 90% of the messages belong to one category.

#### 4.2 Categories have been Proposed, but without any Training Sets

This case is where we have a set of concepts, whether from Leximancer, from a crawler, or from a manual process. But we also need to know which words indicate which concept, and the probabilities. Leximancer can tell us which words were included in its proposed concepts, but without probabilities.

#### 4.3 Pre-categorized Training Sets

This is the simplest situation. For each training set \(j\) (which is specific to a category or sub-category) we record the count of times a word \(i\) appears as \(N_{ij}\). A word's Local Density is then \(N_{ij} / L_j\) where \(L_j\) is the total length of the training set \(j\) (in lines or characters).

The Discrimination Value of a word \(i\) to indicate category \(j\) can be measured as the ratio of the Local Density for \(j\) to the Global Density \(N_i / L\) across all training sets. If a word appears no more often in the training set than in the whole collection, then the ratio is 1 or less, so the word does not have much value. If it appears twice (or more) as often, then it could be selected as an “indicator string”.

### 5 REFLECTIONS

In discussions both within and outside the team it has become clear that the example ontology we have been using does not strictly differentiate is-a and part-of relationships. It may be that a part-of hierarchy is more appropriate to a user's work structure. However the graphical aspects of our ontology editor only represent is-a relationships - the rest have to be entered through property sheets. We
have been attempting to develop additional graphical support for part-of and process inter-dependency relationships, but the resulting interface may be too complex for our intended users.

A continuing obstacle in our work so far has been the density of noise words in our text archives. These skew the automatic analysis, and adding them to the stop word list often does little more than throw up a new set of noise words in the next iteration. The danger in this process is that what is noise to one user may be significant to another, and every user is forced to maintain his or her individual stop word list.

A full ontology approach may not in fact be the best solution. We only need to maintain a small and relatively simple structure of a person’s work. However the need remains to make it easy for the user to set up and maintain his/her work structure and means of recognizing context.

Additionally, for any solution to gain wide acceptance by users, issues of adoption and diffusion of software tools are critical. To stand any chance of adoption, a tool has to relieve the user’s overload – rather than add yet another straw to the camel’s back.

6 FUTURE WORK

We are continuing to test our theories and ideas on further collections of documents and email archives. Up to now we have only looked at email archives from one or two persons. Looking at more may impose ethical issues such as confidentiality.

Five particular areas of planned future work with our current investigations are:

a) Test how seriously the appearance of repeated original messages in email archives affects categorization;

b) Test different cut-off levels of specificity when classing words as stopwords;

c) Test concept sets determined by a crawler approach, including learning how we might align different ontologies that are suggested by different parts of a user’s folder structures (e.g. bookmarks);

d) Develop an approach to using available data that relates proper names appearing in text to the user’s work structure;

e) Develop a user-friendly wizard that leads the user through a variety of tools that help the ontology and lexicon construction and maintenance.

ACKNOWLEDGEMENTS

We would like to thank Paul Swatman (U of South Australia), Gerhard Schwabe (U of Zürich) and Wolfgang Maass (U of Furtwangen, Germany) for their helpful feedback; and Bharat Mordiya, Jayeshkumar Parmar and Sudarshan Patel for their work on the lexical and process extensions to the EzOntoEdit ontology editor.

REFERENCES


