ONTOLOGY-BASED EMAIL CATEGORIZATION AND TASK INFERENCE USING A LEXICON-ENHANCED ONTOLOGY

Prashant Gandhi and Roger Tagg
School of Computer and Information Science, University of South Australia

Keywords: Information overload, Personal information management, Task inference, Ontology, Lexicon.

Abstract: Today's knowledge workers are increasingly faced with the problem of information overload as they use current IT systems for performing daily tasks and activities. This paper focuses on one source of overload, namely electronic mail. Email has evolved from being a basic communication tool to a resource used – and misused – for a wide variety of purposes. One possible approach is to wean the user away from the traditional, often cluttered, email inbox, toward an environment where sorted and prioritized lists of tasks are presented. This entails categorizing email messages around personal work topics, whilst also identifying implied tasks in messages that users need to act upon. A prototype email agent, based on the use of a personal ontology and a lexicon, has been developed to test these concepts in practice. During the work, an opportunistic user survey was undertaken to try to better understand the current task management practices of knowledge workers and to aid in the identification of potential future improvements to our prototype.

1 INTRODUCTION

The problem of information overload is a multi-faceted one, covering a wide range of technical and social issues (Spira and Goldes, 2007). As a result, research into easing it is complex and wide-ranging.

The work described in this paper is part of a larger project named Virtual Private Secretary (VPS). The theme in VPS is to apply IT in a way much like a human secretary would organize the work of her principal, i.e. through a knowledge of the principal's work structures and task types.

This paper focuses on the growing problem of email overload that affects most knowledge workers today. Though overload arises from a number of different sources, it was considered imperative to focus on email, since email is widely considered as one of main contributors to information overload incidents, 60% according to a recent study (Mulder et al, 2006).

The current work, therefore, specifically explores the use of ontology concepts and lexicons for categorizing email messages around personal work topics, whilst also inferring, from the emails, any tasks that users need to act upon. An objective here is to wean the user away from the traditional (often cluttered) email inbox toward an environment where sorted and prioritized lists of tasks can be presented. A prototype email agent has been developed to test these concepts in practice. In addition, a user survey was undertaken to understand the current task management practices of knowledge workers and to aid in identifying future improvements to the prototype system.

The remainder of the paper is organized as follows. Section 2 gives more detailed motivation for the work. Section 3 discusses the different research approaches possible. Section 4 describes the prototype itself. We conclude with a brief evaluation, some reflections and some ideas for future work.

2 MOTIVATION

Current opinion, e.g. (Hall, 2004; Dabbish et al, 2005; Spira and Goldes, 2007) suggests that there are serious weaknesses in today's commercial personal information management (PIM) tools such as Microsoft's Outlook, IBM's Lotus Notes package, and Qualcomm's Eudora Pro. Although these tools typically provide spam-filtering mechanisms and means for manually organizing email messages into folders, and features for creating rule-based filters, these are tedious and cognitively demanding to use (Ducheneaut and Bellotti, 2001).

Gandhi P. and Tagg R. (2009).
ONTOLOGY-BASED EMAIL CATEGORIZATION AND TASK INFERENCE USING A LEXICON-ENHANCED ONTOLOGY.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Software Agents and Internet Computing*, pages 101-107
DOI: 10.5220/0001996101010107

It is also often observed that these tools have evolved from email clients into fuller PIMs, without having been designed as such. In particular, the integration of tasks with email is anything but seamless.

At the same time, prototype systems developed as a part of research initiatives have introduced interesting strategies and techniques for handling email workload. However, as of the time of writing, none have led to widely adopted PIM improvements.

Our main research question is therefore as follows: *How can we develop a practical system that can more accurately and reliably classify and prioritize the task implications of email messages around a user's work activities in order to overcome email overload?*

Our methodology has been the development and evaluation of a proof-of-concept prototype for exploring the potential of using a personal ontology (together with a set of lexical clues that indicate the relevance of each ontology concept) for categorizing email messages around user preferences and identifying implied tasks from message content. We then hope to understand the strengths and weaknesses of this approach as a potential component of next generation PIMs.

3 DIFFERENT APPROACHES TO CATEGORIZING MESSAGES

The three main approaches appear to be machine learning-based, ontology-based, and sender-assisted techniques. Two other approaches are those based on sender identity and those using threads.

3.1 Machine Learning Based

Categorization solutions based on machine learning techniques have been dominant in the research community (Sebastiani, 2002). This involves the application of artificial intelligence (AI) theories to build 'intelligent' software agents that can be trained to make categorization decisions on a user's behalf. A learning-based classifier for a category can be built through an inductive process where the system observes the characteristics of a training set of messages. Manual intervention is limited to deciding whether or not messages have been sensibly categorized.

The machine learning based approach relies solely on 'endogenous' knowledge i.e. knowledge gained only from the documents themselves. The user does not stipulate the categorization scheme.

Example prototypes of this type include Maxims (Maes, 1994), MailCat (Segal and Kephart, 1999), and IEMS (Crawford et al, 2006). (Corston-Oliver et al, 2004) aimed to automatically identify tasks in email messages using machine-learning techniques.

Although they require lower maintenance, classifiers suffer from the 'slow start' problem, since they can only gradually build-up their competency as more examples are provided, over time. This problem becomes worse the more categories there are, as compared with the simple *spam/not spam* situation.

3.2 Ontology Based

The potential of leveraging ontology structures for classifying documents has been attracting increased research activity in recent years. For mail categorization, an ontology would contain the structure of a user's or a groups work topics, task types, priorities etc. This can be built and edited using an ontology editor.

Some representative prototypes include ECPIA (Li et al, 2006); CLIPS (Taghva et al, 2003), the latter being a hybrid approach. An earlier prototype within the VPS project, TaskMail (Punekar and Tagg, 2005), also used an ontology, albeit a simple flat one; hence this had number of shortcomings.

The key difference between an ontology and machine learning-based system is that the former relies on 'exogenous' information i.e. the category structure specified within the ontology, which someone has to create and maintain. However such an explicit definition of rules by a user or group of users could possibly lead to more accurate categorization, since human judgment is being indirectly leveraged. Additionally, the same ontology could be re-used across multiple applications (e.g. general document filing, personal bookmarks, classifying events received from a workflow system).

However, ontology alone is not enough. The system needs a lexicon of words and phrases, which, if they appear in a message, indicate that a category applies to the message.

3.3 Sender Assisted

Since it is usually the sender of a message that wants something done, it seems reasonable to expect him or her to explicitly specify one or more categories or tags for each document. This could be applied to email by using an XML-style tagging protocol, or by requiring senders to complete a recipient-dependent pop-up form containing a set of drop down choices that need to be made before the message can be sent.

The latter approach was the basis for another earlier VPS prototype named NatMail (Tagg and Mahalingam, 2005). Senders were required to submit messages through a "Contact Me" web page. The options in the drop down boxes were created via a Wizard based on the recipient's personal ontology.

The concept of sender-assistance might be expected to incur resistance from senders, since it demands a fundamental shift in work culture. However we are all getting more used to filling in web forms for airlines, banks, insurance companies etc. Even some university professors ask their students to do so.

The younger generation is already half way there, since categorization is an essential part of tools like Flickr and del.icio.us. In the long run the approach might just need to gradually become part of the work culture.

3.4 Other Classification Approaches

Another approach is the SimOverlap system of (Dredze et al, 2006), who match people in an email message with pre-defined activity participants. This is a valid simplification in many work structures, but not if the working roles are highly volatile.

The best known of other approaches is probably that of detecting threads or 'thrasks', as in TaskVista (Bellotti et al, 2003). In their TV-ACTA prototype, (Bellotti et al, 2007) introduced a distinct strategy based on the integration of 'to-do' lists with email. Users can drag-and-drop email messages into a system for creating to-dos, which can be then be sorted according to properties such as deadline and task type.

4 DESCRIPTION OF THE PROTOTYPE

Regarding the technologies and systems to be used to support the implementation of the prototype, it was decided that:

- Personal ontologies would be represented in the XML OWL (RDF) format using the output of this university's own EzOntoEdit ontology editor (Einig et al, 2006).
- A tool named SnipCat (Srinivasan Kumaar, 2008) would also be used to insert lexical clues into the ontology.
- The system should be developed in Java, and the output task lists should be maintained in a relational database, in our case Oracle.

Figure 1 outlines the steps involved. The process can be broken down into three key phases, namely, *initiation*, *work topic categorization*, and *task identification*, as detailed below.

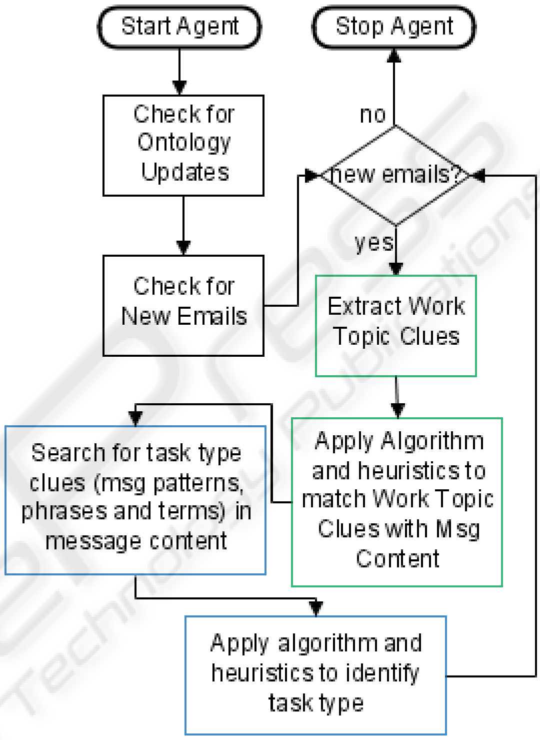


Figure 1: Prototype Process Overview.

4.1 Initiation Phase

We assume that the user has already created a personal ontology; this is imported into the agent as part of its internal database. When the agent is started, it checks (at 5-minute intervals) whether the ontology has changed from the previous session; if there are any changes the complete new ontology is imported.

4.2 Work Topic Categorization Phase

Having gained a copy of user's latest ontology, the agent then checks for new emails during the same 5-minute intervals. If the user has not received any new emails the process waits until the next 5-minute interval. However, if new emails are detected the message body is extracted for each email (identified by a distinct email ID) and stored in individual text files on a web server. Email attachments (if any) are also stripped off and stored on the web server.

Treating each email message as a separate document, the agent first tokenizes the contents (i.e. From, To, Cc, Subject and Body) of each email, removing punctuations.

Next, stop words present in the email message are removed in order to reduce the search space. This will hopefully improve classification performance, in terms of speed and accuracy.

Having defined a feature set for a new email, the agent then extracts the set of ontology clues from the database. These include upper case and lower case clue terms (stored separately) and phrases that were provided by the user. The extracted clue phrases are first compared with the content of the email message (i.e. performing a free text search). The matching phrases (if any) are identified and stored in a list. The clue terms are then compared with the message tokens and the matching terms (if any) are stored in the same list. If no matching phrases or terms were found at this stage, the email is labeled as 'unidentifiable' (i.e. the system was unable to identify the work topic) and the task identification phase commences.

However, if matching words or phrases were found the agent proceeds to identify the number of occurrences of each matched clue in the email. It seems logical to assume that an email with 5 occurrences of the clue *conference* would have a stronger association with the *conferences* work topic than an email with only 1 occurrence of conference.

To achieve this, we build on the concept in our ontology design of *indication strength*, which is the subjective probability that presence of the clue's string indicates relevance to the ontology concept. The weighted indication strength (WIS) for each matched clue is calculated based on multipliers that reflect the number of occurrences of the clue. The heuristic multipliers we applied in our tests ranged from 1.2 for 2 occurrences, to 1.66 for 10 or more.

Finally, the strengths of all the different clues indicating the same ontology category are added. If the total weighted indication strength for a category exceeds the threshold value (we have set the default at 1.0), the email is then labeled as belonging to that work topic.

4.3 Task Inference Phase

The task inference phase now follows in a similar but not identical fashion. The process begins with the removal of old messages (typically previous conversations between participants) within the body of the email. This is more important in task inference than with work topics, since the agent needs to be prevented from erroneously inferring tasks by scanning old messages in the thread.

In the ontology we have been using, we store a number of message patterns that the user receives regularly and which strongly indicate a particular task type. If such a match is found, the email is tagged with the appropriate task type (i.e. *ForInfoOnly*, *Reply* etc.) and the task identification process is brought to an end. This is because these message pattern clues always have an indication strength of 1.0.

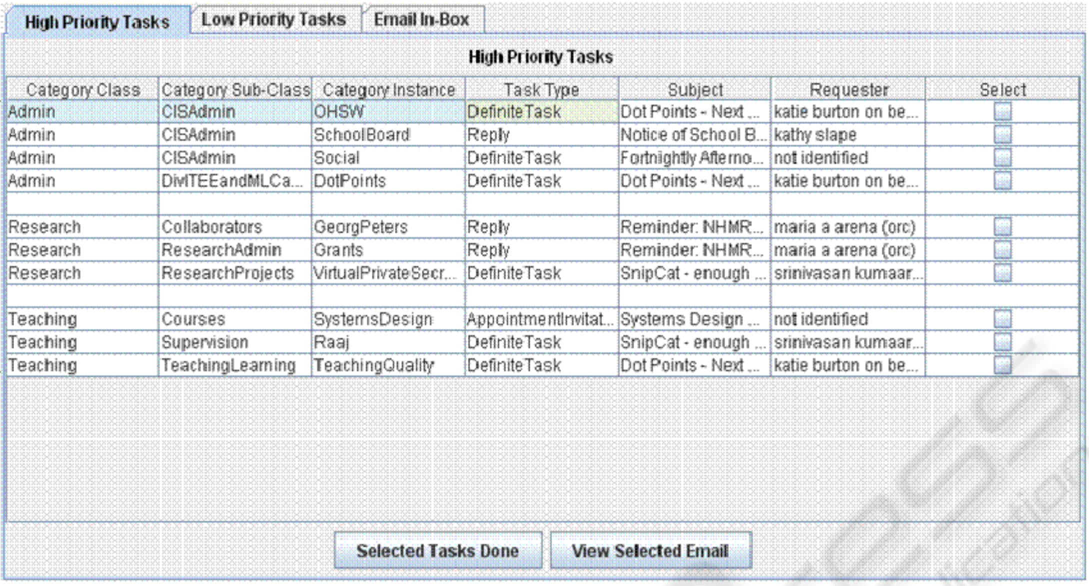
However, if a matching message pattern was not found, the list of remaining clue strings (words or phrases) that indicate each task type, together with their respective *locations* and *strengths*, are retrieved from the database. Thereafter, the clue strings are matched one task type at a time, upper case and lower case being taken into account.

4.4 Presentation of the Task Lists

To present the results of the above email classification activities, we have designed a prioritized "to do list" style interface. The idea is to encourage users to start by viewing only the messages that imply high priority tasks, and to group these by work topic; they can then view the less important ones later. Hopefully, this could help negate feelings of email overload. The interface consists of mailbox-like panels; a screenshot is shown in Figure 2 below.

Any one of three panels can be chosen. The default is 'High Priority Tasks', which includes those emails mapped to the task types *DefiniteTask*, *Reply*, *AppointmentInvitation*, *VoteApprove*, and *ConditionalTask.* The 'Low Priority Tasks' panel includes those mapped *ForInfoOnly*, *Questionnaire*, *ConfInvite*, and *PrivateCommunication.* Two buttons displayed at the bottom of both of the above panels facilitate user interaction with the interface.

The 'Email Inbox' panel, which displays all emails, was included since it was found, through the survey discussed below, that users would feel more comfortable with the system if they could switch back and forth between the 'new' interface and the traditional inbox view, without having to go back to the old email client. This relates to the issue of trust that arises when using a software agent.



Category Class	Category Sub-Class	Category Instance	Task Type	Subject	Requester	Select
Admin	CISAdmin	OHSW	DefiniteTask	Dot Points - Next ...	katie burton on be...	
Admin	CISAdmin	SchoolBoard	Reply	Notice of School B...	kathy slape	
Admin	CISAdmin	Social	DefiniteTask	Fortnightly Afterno...	not identified	
Admin	DivITEEandMLCa...	DotPoints	DefiniteTask	Dot Points - Next ...	katie burton on be...	
Research	Collaborators	GeorgPeters	Reply	Reminder: NHMR...	maria a arena (orc)	
Research	ResearchAdmin	Grants	Reply	Reminder: NHMR...	maria a arena (orc)	
Research	ResearchProjects	VirtualPrivateSecr...	DefiniteTask	SnipCat - enough ...	srinivasan kumaar...	
Teaching	Courses	SystemsDesign	AppointmentInvitat...	Systems Design ...	not identified	
Teaching	Supervision	Raaj	DefiniteTask	SnipCat - enough ...	srinivasan kumaar...	
Teaching	TeachingLearning	TeachingQuality	DefiniteTask	Dot Points - Next ...	katie burton on be...	

Figure 2: Example of a Task List from the Prototype.

5 EVALUATION

The system was built successfully as planned. Evaluation was in three parts.

- a first version was tested on one academic and then demonstrated at a project fair;
- a questionnaire survey was carried out on 40 people attending the project fair
- a revised version with improved performance was built and tested.

In the first version, categorization into work topics was quite successful, with an average precision of 80%. However the task type categorization was disappointing; the average precision was under 50%. Many messages that should have indicated definite tasks were only graded as low priority, or the agent failed to put them into any task category.

The survey showed positive responses from the majority of respondents, who included a mix of students, academics and people outside academia. They expressed interest in the idea of moving to a more structured, task list style interface. However, interest is a long way removed from changing one's everyday computing habits. Issues raised included ontology change management, the need to still refer to the inbox on a regular basis, and the lack of a feature for managing task deadlines and reminders.

In the tests of the second version, performance did improve, but the task inference was still not adequate for use in a real world environment. Our assessment was that the agent's inability to recognize deadline dates and times was a contributing factor, and some level of sender assistance (e.g. by tagging deadlines) might also be needed to make a significant difference.

6 CONCLUSIONS AND FUTURE WORK

The main contribution of this work has been to demonstrate the concept of an ontology-driven email categorization agent. In regards to system performance, the results produced so far have been positive. The strong results achieved in classifying emails around work topics, is particularly encouraging.

Admittedly, limited user testing of the prototype system has been undertaken in this work due to the time constraints and the need to acquire ethics approval.

Future initiatives would need to focus on not only testing the system with a larger user base and sample data sets but also over a longer period of time. However, having said that, the user testing undertaken so far has been useful in terms of gauging system performance, user attitudes and acceptance, as well as for establishing future research direction.

This work does not entirely rule out use of a machine learning approach, and acknowledges that a combined ontology and machine learning-based model might be the way forward in the future, especially for overcoming the challenges of implicit task identification.

ACKNOWLEDGEMENTS

We would like to thank Harshad Lalwani for his technical inputs and help in developing the VPS email agent prototype.

REFERENCES

Bellotti, V., Ducheneaut, N., Howard, M., & Smith, I. (2003) Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool. *Proceedings of the computer-human interaction conference (CHI) 2003*, CHI Letters 5 (1). 297-304.

Bellotti, V., Thornton, J., Chin, A., Schiano, D., & Good, N. (2007). TV-ACTA: Embedding an Activity-Centered Interface for Task Management in Email. *CEAS, 2007*.

Corston-Oliver, S., Ringger, E., Gamon, M., & Campbell, R. (2004). Task-focused Summarization of email. *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 43–50.

Crawford, E., Kay, J., & McCreath, E. (2006). IEMS - the Intelligent Email Sorter. *Proceedings of the International Conference on Machine Learning*, 25-32

Dabbish, L.A., Kraut, R.E., Fussell, S., & Kiesler, S. (2005) Understanding Email Use: Predicting Action on a Message. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 691-700.

Dredze, M., Lau, T., & Kushmerick, N. (2006). Automatically Classifying emails into Activities. *Proceedings of the 11th International Conference on Intelligent User Interfaces*, 70-77.

Ducheneaut, N., & Bellotti, V. (2001). E-mail as Habitat: an exploration of embedded personal information management. Interactions 8, 5, 30-38.

Einig, M, Tagg, R. & Peters, G (2006), Managing the Knowledge Needed to Support an Electronic Personal Assistant, in *Proceedings of the ICEIS conference*, Paphos, Cyprus.

Hall, S. (2004). A Better Microsoft Outlook . Blog, http://nwvc.blogs.com/northwest_vc/2004/02/a_better_micros.html (Viewed 27 November 2008).

Li, W., Zhong, N., & Liu, C. (2006). ECPIA: An Email-Centric Personal Intelligent Assistant. *Lecture notes in computer science, 4062*, 502.

Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM, 37*, 31-40.

Mulder, I., de Poot, H., Verwij, C., Janssen, R., & Bijlsma, M. (2006). An Information Overload Study. *Proceedings of the CHI Conference SIG of Australia on Design: Activities, Artefacts and Environments*, 245-252.

Punekar & Tagg, R. (2005). Ontology Assisted Pre-processing of Incoming email: Working paper, University of South Australia.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys 34 1*, 1-47.

Segal, R., & Kephart, J. (1999). MailCat: an intelligent assistant for organizing e-mail. *Proceedings of the third annual conference on Autonomous Agents*, 276-282.

Spira, B., & Goldes, M. (2007). “Information Overload: We have met the enemy and he is us”, Basex Inc., http://freya.basex.com/web/webdownloads.nsf/e67dc0f5617d6e9c85256a99005ea0e7/ff7810b594932040852570b9008022ce/$FILE/BasexInformationOverloadWhitePaper.pdf (Viewed 18 May, 2008)

Srinivasan Kumaar, R. (2008), Helping Users to Create and Maintain a Personal Knowledge Base or Ontology to Support Task-oriented Work Management, Masterate Thesis, University of South Australia.

Tagg, R. (2007), Task Integration for Knowledge Workers, in *ICEIS Workshop on Computer Supported Activity Coordination*, Madeira, Portugal, May 2007.

Tagg, R. & Mahalingam, N. (2005). Improving Customer Service and Reducing Administrative Overload with Sender-assisted Message Categorization: Working paper, University of South Australia.

Taghva, K., Borsack, J., Coombs, J., Condit, A., Lumos, S. & Nartker, T. (2003). Ontology-based Classification of email. *Proceedings of ITCC 2003 Conference on Information Technology: Coding,* 194-198.