Integration of Email and Task Lists

Simon Corston-Oliver, Eric Ringger, Michael Gamon, Richard Campbell

Microsoft Research; One Microsoft Way, Redmond, Washington 98052; USA
{simonco, ringger, mgamon, richcamp}@microsoft.com

Email has evolved from a mere communication system to a general tool for organizing workflow (Whittaker and Sidner, 1996; Cadiz et al., 2001; Bellotti et al., 2003). Widely used email clients address this issue by providing additional functionality, including contact management, calendaring and task lists (i.e., “to do” lists). However, some of these functions are poorly integrated. For example, to create a task, a user might have to switch from an email view to manually fill in a form.

We describe SmartMail (Corston-Oliver et al., 2004), a system that improves the integration of the various functions of an email client by identifying the speech act of each sentence in an email message and performing actions appropriate to the speech act. We consider fifteen application-specific speech acts, but our focus to date has been the processing of tasks; SmartMail helps users prioritize emails by identifying tasks.

We collected a corpus of 15,741 email messages sent by 3,098 distinct individuals, with no more than 50 messages from any one person. Three annotators labeled sentences as containing one of the fifteen speech acts. Pair-wise inter-annotator agreement was measured using Cohen’s Kappa on a common set of 146 messages, and ranged from 82.3% to 85.8%. Annotators marked a sentence as a task if it looked like an appropriate item to add to an on-going task list. The following are examples of tasks in our data:

> …*you could possibly generate the same sequence of numbers to select the same cases.*
> …*it’d be great if you could add some stuff re MSRDPS.*
> *Could you please remote desktop in and try running it on my machine.*

SmartMail segments an email message into header, message body (containing the new message content), and forwarded sections and then identifies tasks in each sentence in the message body using a machine-learned classifier. The tasks are reformulated and presented to the user. For example the sentence *On the H-1 visa issue, I am positive that you need to go to the Embassy in London to get your visa stamped into your passport* is reformulated as *Go to the Embassy in London to get your visa stamped into your passport*. The reformulation consists of analyzing the sentence to produce a logical form (LF) (Campbell & Suzuki, 2001), performing knowledge-engineered steps that isolate the part of the LF containing the task, removing extraneous material, and modifying the LF to disambiguate items that would be ambiguous out of context such as relative dates (e.g., *next Thursday*). The modified LF is passed to a sentence realization module (Aikawa et al., 2001) to produce a string that is displayed to the user.

We trained support vector machines (SVMs) (Vapnik, 1995) with linear kernels using SMO (Platt, 1999) to identify each speech act. Figure 1 illustrates the per-sentence and per-message precision/recall curves for the SVM classifier trained to distinguish task speech acts vs. non-tasks measured on a blind test set containing 699 messages. A message was marked as containing a task if it contained at least one sentence classified as a task. Since only one task need be found in order for the entire message to be classified as containing a task, accuracy is substantially higher than on a per-sentence basis. Three types of features were used: *properties of the message* such as the number of addressees, the size of the message, and the number of forwarded sections; *superficial features* such as capitalization, word unigrams, bigrams and trigrams; and *linguistic features* extracted using the NLPWIN system (Heidorn, 2000), including part-of-speech bigrams and trigrams, syntactic constituent structure features, and deeper linguistic features such as transitivity and tense. Feature ablation experiments demonstrated a modest improvement in classification accuracy when linguistic features were included (Corston-Oliver et al., 2004).

In one prototype user interface, messages that contain tasks are marked with a small flag icon. Users can sort mail to see all messages marked this way. When the user opens an email, the tasks that have been identified and reformulated are presented in one place, possibly with a check-box beside them, as illustrated in Figure 2. Checking the box adds the task to the task list, including the text of the original message. The regions of the message from which the tasks were extracted are also highlighted to provide context for the interpretation of the tasks.

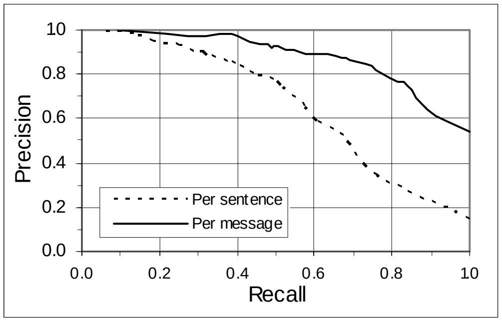

Figure 1: Precision/Recall curve for identifying tasks

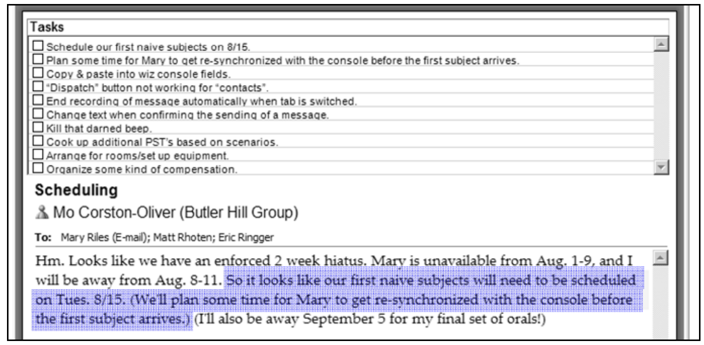

Figure 2: SmartMail prototype showing message with extracted task

In the scenarios we envisage, SmartMail is not intended to replace the user's reading email, but rather to suggest the order in which it should be read and as a mechanism for proposing actions to take. Therefore, false negatives (actual tasks that aren't flagged as such) will be caught by the user, and do not pose as serious a usability problem as false positives (non-tasks flagged as tasks), too many of which would lead the user to ignore flags altogether. Therefore we are more interested in maximizing precision than recall.

We plan to conduct user studies by distributing a SmartMail prototype to volunteers who would use it to read and process their own email over an extended period. We intend to observe how many tasks users actually add to their task list and to survey user satisfaction.

Acknowledgments

Many of the ideas presented here were formulated in discussion with Bob Atkinson, Dave Reed, Malcolm Pearson, Hank Leukart and Glenn Frankel. Our thanks go to Jeff Stevenson, Margaret Salome and Kevin Gaughen for annotating the data.

References

Aikawa, T., M. Melero, L. Schwartz and A. Wu. 2001. Multilingual Natural Language Generation. *Proceedings of EAMT*.

Bellotti, V., N. Ducheneaut, M. Howard, I. Smith. 2003. Taking email to task: the design and evaluation of a task management centered email tool. *Proceedings of the conference on human factors in computing systems*. pp. 345-352.

Cadiz, J. J., Dabbish, L., Gupta, A., & Venolia, G. D. 2001. Supporting Email Workflow. *MSR-TR-2001-88*: Microsoft Research.

Campbell, R. and H. Suzuki. 2002. Language neutral representation of syntactic structure. *Proceedings of SCANALU 2002*.

Corston-Oliver, S., E. Ringger, M. Gamon and R. Campbell. 2004. Task-focused summarization of email. *Proceedings of the Text Summarization Branches Out* workshop, ACL 2004.

Heidorn, G. 2000. Intelligent Writing Assistance. R. Dale, H. Moisl and H. Somers, (eds.), *Handbook of Natural Language Processing*. Marcel Dekker.

Platt, J. 1999. Fast training of SVMs using sequential minimal optimization. B. Schoelkopf, C. Burges and A. Smola (eds.) *Advances in Kernel Methods: Support Vector Learning*, MIT Press, Cambridge, MA. pp. 185-208.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Whittaker, S. and C. Sidner. 1996. Email overload: exploring personal information management of email. *Proceedings of CHI'96*, Conference on Human Factors in Computing Systems, ACM, NY. pp. 276-283.