Task-focused Summarization of Email

Simon Corston-Oliver, Eric Ringger, Michael Gamon and Richard Campbell
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
{simonco, ringger, mgamon, richcamp}@microsoft.com

Abstract

We describe SmartMail, a prototype system for automatically identifying action items (tasks) in email messages. SmartMail presents the user with a task-focused summary of a message. The summary consists of a list of action items extracted from the message. The user can add these action items to their "to do" list.

1 Introduction

Email for many users has evolved from a mere communication system to a means of organizing workflow, storing information and tracking tasks (i.e. "to do" items) (Bellotti et al., 2003; Cadiz et al., 2001). Tools available in email clients for managing this information are often cumbersome or even so difficult to discover that users are not aware that the functionality exists. For example, in one email client, Microsoft Outlook, a user must switch views and fill in a form in order to create a task corresponding to the current email message. By automatically identifying tasks that occur in the body of an email message, we hope to simplify the use of email as a tool for task creation and management.

In this paper we describe SmartMail, a prototype system that automatically identifies tasks in email, reformulates them, and presents them to the user in a convenient interface to facilitate adding them to a "to do" list.

SmartMail performs a superficial analysis of an email message to distinguish the header, message body (containing the new message content), and forwarded sections. [1] SmartMail breaks the message body into sentences, then determines the speech act of each sentence in the message body by consulting a machine-learned classifier. If the sentence is classified as a task, SmartMail performs additional linguistic processing to reformulate the sentence as a task description. This task description is then presented to the user.

2 Data

We collected a corpus of 15,741 email messages. The messages were divided into training, development test and blind test. The training set contained 106,700 sentences in message bodies from 14,535 messages. To avoid overtraining to individual writing styles, we limited the number of messages from a given sender to 50. To ensure that our evaluations are indicative of performance on messages from previously unencountered senders, we selected messages from 3,098 senders, assigning all messages from a given sender to either the training or the test sets.

Three human annotators labeled the message body sentences, selecting one tag from the following set: Salutation, Chit-chat (i.e., social discussion unrelated to the main purpose of the message), Task, Meeting (i.e., a proposal to meet), Promise, Farewell, various components of an email signature (Sig_Name, Sig_Title, Sig_Affiliation, Sig_Location, Sig_Phone, Sig_Email, Sig_URL, Sig_Other), and the default category "None of the above". The set of tags can be considered a set of application-specific speech acts analogous to the rather particular tags used in the Verbmobil project, such as "Suggest_exclude_date" and

[1] This simple division into header, message body, and forwarded sections was sufficient for the corpus of email messages we considered. Messages containing original messages interleaved with new content were extremely uncommon in our corpus. Most senders were using Microsoft Outlook, which places the insertion point for new content at the top of the message.

"Motivate_appointment" (Warnke et al., 1997; Mast et al., 1996) or the form-based tags of Stolcke et al. (1998).

All three annotators independently labeled sentences in a separate set of 146 messages not included in the training, development or blind test sets. We measured inter-annotator agreement for the assignment of tags to sentences in the message bodies using Cohen's Kappa. Annotator 1 and annotator 2 measured 85.8%; annotator 1 and annotator 3 measured 82.6%; annotator 2 and annotator 3 measured 82.3%. We consider this level of inter-annotator agreement good for a novel set of application-specific tags.

The development test and blind test sets of messages were tagged by all three annotators, and the majority tag for each sentence was taken. If any sentence did not have a majority tag, the entire message was discarded, leaving a total of 507 messages in the development test set and 699 messages in the blind test set.

The set of tags was intended for a series of related experiments concerning linguistic processing of email. For example, greetings and chit-chat could be omitted from messages displayed on cell phones, or the components of an email signature could be extracted and stored in a contact database. In the current paper we focus exclusively on the identification of tasks.

Annotators were instructed to mark a sentence as containing a task if it looked like an appropriate item to add to an on-going "to do" list. By this criterion, simple factual questions would not usually be annotated as tasks; merely responding with an answer fulfills any obligation. Annotators were instructed to consider the context of an entire message when deciding whether formulaic endings to email such as *Let me know if you have any questions* were to be interpreted as mere social convention or as actual requests for review and comment. The following are examples of actual sentences annotated as tasks in our data:

```
Since Max uses a pseudo-
random number generator, you
could possibly generate the
same sequence of numbers to
select the same cases.

Sorry, yes, you would have to
retrain.

An even fast [sic] thing
would be to assign your own
ID as a categorical feature.

Michael, 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.

If CDDG has its own notion
of what makes for good
responses, then we should
use that.
```

3 Features

Each sentence in the message body is described by a vector of approximately 53,000 features. The features are of three types: properties of the message (such as the number of addressees, the total size of the message, and the number of forwarded sections in the email thread), superficial features and linguistic features.

The superficial features include word unigrams, bigrams and trigrams as well as counts of special punctuation symbols (e.g. @, /, #), whether the sentence contains words with so-called "camel caps" (e.g., *SmartMail*), whether the sentence appears to contain the sender's name or initials, and whether the sentence contains one of the addressees' names.

The linguistic features were obtained by analyzing the given sentence using the NLPWin system (Heidorn 2000). The linguistic features include abstract lexical features, such as part-of-speech bigrams and trigrams, and structural features that characterize the constituent structure in the form of context-free phrase structure rewrites (e.g., DECL:NP-VERB-NP; i.e., a declarative sentence consisting of a noun phrase followed by a verb and another noun phrase). Deeper linguistic analysis yielded features that describe part-of-speech information coupled with grammatical relations (e.g., Verb-Subject-Noun indicating a nominal subject of a verb) and features of the logical form analysis such as transitivity, tense and mood.

4 Results

We trained support vector machines (SVMs) (Vapnik, 1995) using an implementation of the sequential minimal optimization algorithm (Platt, 1999). We trained linear SVMs, which

have proven effective in text categorization with large feature vectors (Joachims, 1998; Dumais et al., 1998).

Figure 1 illustrates the precision-recall curve for the SVM classifier trained to distinguish tasks vs. non-tasks measured on the blind test set.

We conducted feature ablation experiments on the development test set to assess the contribution of categories of features to overall classification performance. In particular we were interested in the role of linguistic analysis features compared to using only surface features. Within the linguistic features, we distinguished deep linguistic features (phrase structure features and semantic features) from POS n-gram features. We conducted experiments with three feature sets:

1. all features (message level features + word unigram, bigram and trigram
2. features + POS bigram and trigram features + linguistic analysis features)
3. no deep linguistic features (no phrase structure or semantic features)
4. no linguistic features at all (no deep linguistic features and no POS n-gram features)

Based on these experiments on the development test set, we chose the feature set used for our run-time applications.

Figure 1 shows final results for these feature sets on the blind test set: for recall between approximately 0.2 and 0.4 and between approximately 0.5 and 0.6 the use of all features produces the best results. The distinction between the "no linguistic features" and "no deep linguistic features" scenarios is negligible; word n-grams appear to be highly predictive. Based on these results, we expect that for languages where we do not have an NLPWin parser, we can safely exclude the deeper linguistic features and still expect good classifier performance.

Figure 2 illustrates the accuracy of distinguishing messages that contain tasks from those that do not, using all features. A message was marked as containing a task if it contained at least one sentence classified as a task. Since only one task has to 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. In section 6, we discuss the scenarios motivating the distinction between sentence classification and message classification.

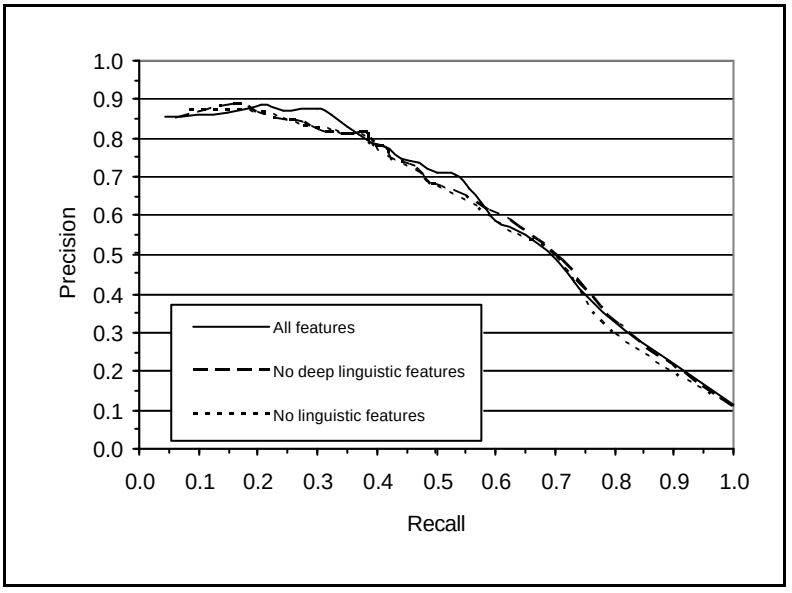

Figure 1: Precision-Recall curves for ablation experiments

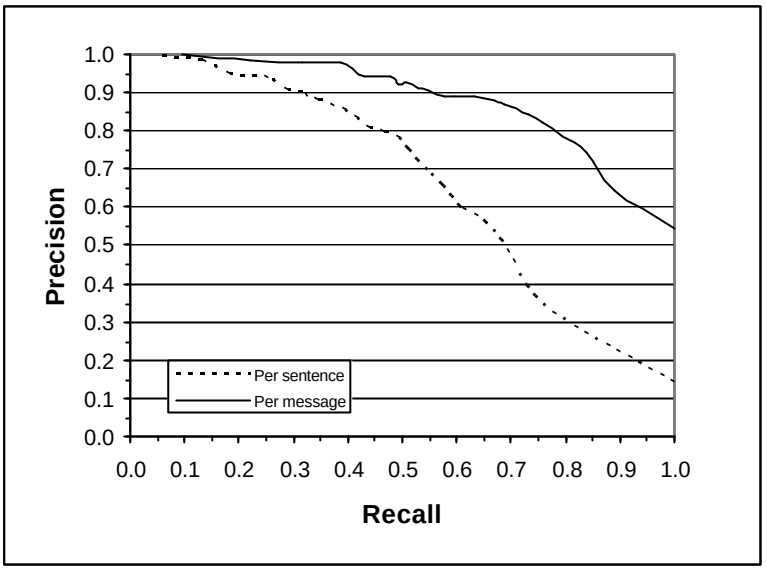

Figure 2: Precision-Recall curves comparing message classification and sentence classification

5 Reformulation of Tasks

SmartMail performs post-processing of sentences identified as containing a task to reformulate them as task-like imperatives. The process of reformulation involves four distinct knowledge-engineered steps:

1. Produce a logical form (LF) for the extracted sentence (Campbell and Suzuki, 2001). The nodes of the LF correspond to syntactic constituents. Edges in the LF represent semantic and deep syntactic relations among nodes. Nodes bear semantic features such as tense, number and mood.
2. Identify the clause in the logical form that contains the task; this may be the entire sentence or a subpart. We consider such linguistic properties as whether the clause is imperative, whether its subject is second person, and whether modality words such as *please* or a modal verb are used. All parts of the logical form not subsumed by the task clause are pruned.
3. Transform the task portion of the LF to exclude extraneous words (e.g. *please*, *must*, *could*), extraneous subordinate clauses, adverbial modifiers, and vocative phrases. We replace certain deictic elements (i.e., words or phrases whose denotation varies according to the writer or the time and place of utterance) with non-deictic expressions. For example, first person pronouns are replaced by either the name of the sender of the email or by a third person pronoun, if such a pronoun would unambiguously refer to the sender. Similarly, a temporal expression such as *Thursday*, which may refer to a different date depending on the week in which it is written, is replaced by an absolute date (e.g., *4/1/2004*).
4. Pass the transformed LF to a sentence realization module to yield a string (Aikawa et al., 2001).

Below we illustrate the reformulation of tasks with some examples from our corpus.

```
Example 1:
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.
Reformulation:
Go to the Embassy in London to
get your visa stamped into
your passport.
```

In this example, the embedded sentential complement, that is, the part of the sentence following *positive*, is selected as the part of the sentence containing the task, because of the modal verb *need* and the second person subject; only that part of the sentence gets reformulated. The modal verb and the second person subject are deleted to form an imperative sentence.

Example 2:
Can you please send me the follow up information for the demo(s) listed in this Email ASAP.
Reformulation:
Send Kendall the follow up information for the demo listed in this Email ASAP.

In this example, the whole sentence is selected as containing the task (modal verb, second person subject); modal elements including *please* are deleted along with the second person subject to form an imperative. In addition, the first person pronoun *me* is replaced by a reference to the sender, Kendall in this instance.

Example 3:
I've been Wednesday at the lecture on Amalgam you gave in the 113/1021 Room (which I really liked), and I've been wondering how feasible would it be to use Amalgam for learning requirements or code corpus structures and rules (and eventually rephrase them in some way).
Reformulation:
On June 5, 2002 Pablo wrote: 'I've been Wednesday at the lecture on Amalgam you gave in the 113/1021 Room (which I really liked), and I've been wondering how feasible would it be to use Amalgam for learning requirements or code corpus structures and rules (and eventually rephrase them in some way).'

This example illustrates what happens when NLPWin is unable to produce a spanning parse and hence a coherent LF; in this case NLPWin misanalyzed the clause following *wondering* as a main clause, instead of correctly analyzing it as a complement clause. SmartMail's back-off strategy for non-spanning parses is to enclose the entire original sentence in quotes, prefixed with a matrix sentence indicating the date and the name of the sender.

6 Task-Focused Summarization

We have considered several scenarios for presenting the tasks that SmartMail identifies. Under the most radical scenario, SmartMail would automatically add extracted tasks to the user's "to do" list. This scenario has received a fairly negative reception when we have suggested it to potential users of a prototype. From an application perspective, this scenario is "fail hard"; i.e., classification errors might result in garbage being added to the "to do" list, with the result that the user would have to manually remove items. Since our goal is to reduce the workload on the user, this outcome would seem to violate the maxim "First, do no harm".

Figure 3 and Figure 4 illustrate several ideas for presenting tasks to the user of Microsoft Outlook. Messages that contain tasks are flagged, using the existing flag icons in Outlook for proof of concept. Users can sort mail to see all messages containing tasks. This visualization amounts to summarizing the message down to one bit, i.e., +/- Task, and is conceptually equivalent to performing document classification.

The right-hand pane in Figure 3 is magnified as Figure 4 and shows two more visualizations. At the top of the pane, the tasks that have been identified are presented in one place, with a check box beside them. Checking the box adds the task to the Tasks or "to do" list, with a link back to the original message. This presentation is "fail soft": the user can ignore incorrectly classified tasks, or tasks that were correctly identified but which the user does not care to add to the "to do" list. This list of tasks amounts to a task-focused summary of the document. This summary is intended to be read as a series of disconnected sentences, thus side-stepping the issue of producing a coherent text from a series of extracted sentences. In the event that users prefer to view these extracted sentences as a coherent text, it may prove desirable to attempt to improve the textual cohesion by using anaphoric links, cue phrases and so on.

Finally, Figure 3 also shows tasks highlighted in context in the message, allowing the user to skim the document and read the surrounding text.

In the prototype we allow the user to vary the precision and recall of the classifier by adjusting a slider (not illustrated here) that sets the probability threshold on the probability of Task.

Figure 3 and Figure 4 illustrate a convention that we observed in a handful of emails: proper names

occur as section headings. These names have scope over the tasks enumerated beneath them, i.e. there is a list of tasks assigned to Matt, a list assigned to Eric or Mo, and a list assigned to Mo. SmartMail does not currently detect this explicit assignment of tasks to individuals.

Important properties of tasks beyond the text of the message could also be automatically extracted. For example, the schema for tasks in Outlook includes a field that specifies the due date of the task. This field could be filled with date and time information extracted from the sentence containing the task. Similarly the content of the sentence containing the task or inferences about social relationships of the email interlocutors could be used to mark the priority of tasks as High, Low, or Normal in the existing schema.

7 Conclusion

In this paper we have presented aspects of SmartMail, which provides a task-oriented summary of email messages. This summary is produced by identifying the task-related sentences in the message and then reformulating each task-related sentence as a brief (usually imperative) summation of the task. The set of tasks extracted and reformulated from a given email message is thus a task-focused summary of that message.

We plan to conduct user studies by distributing the prototype as an Outlook add-in to volunteers who would use it to read and process their own mail over a period of several weeks. We intend to measure more than the precision and recall of our classifier by observing how many identified tasks users actually add to their "to do" list and by administering qualitative surveys of user satisfaction.

The ability to reformulate tasks is in principle separate from the identification of tasks. In our planned usability study we will distribute variants of the prototype to determine the effect of reformulation. Do users prefer to be presented with the extracted sentences with no additional processing, the tasks reformulated as described in Section 5, or an even more radical reformulation to a telegraphic form consisting of a verb plus object, such as *Send information* or *Schedule subjects*?

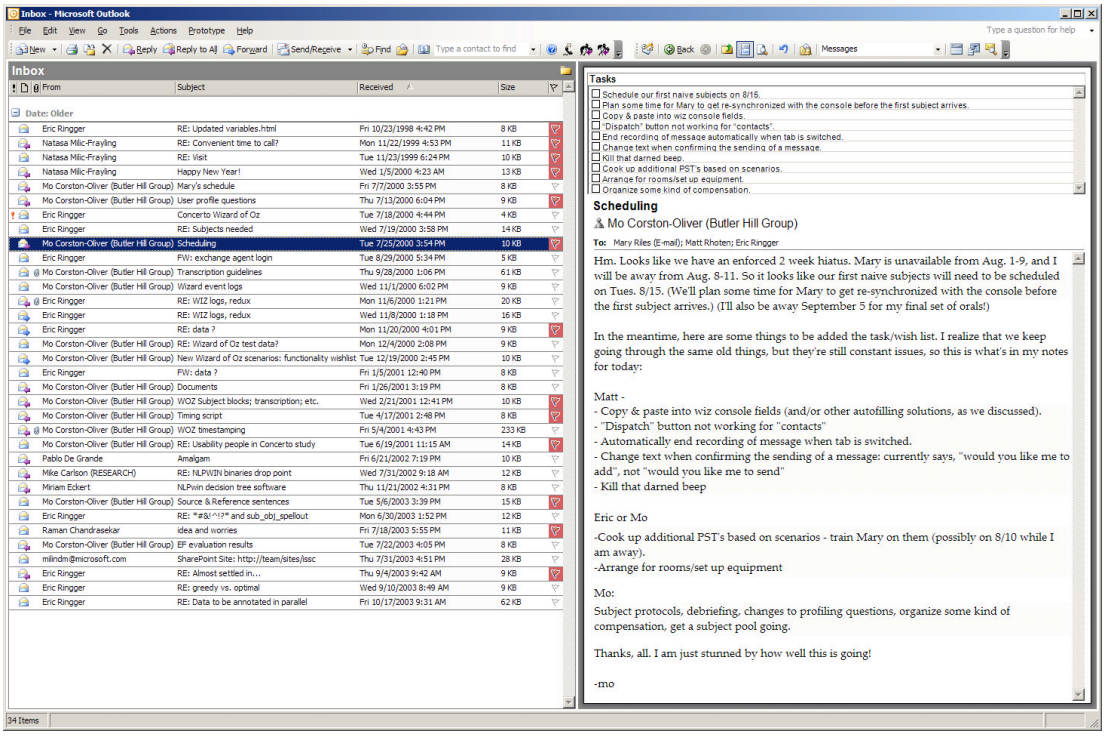

Figure 3: Prototype system showing ways of visualizing tasks

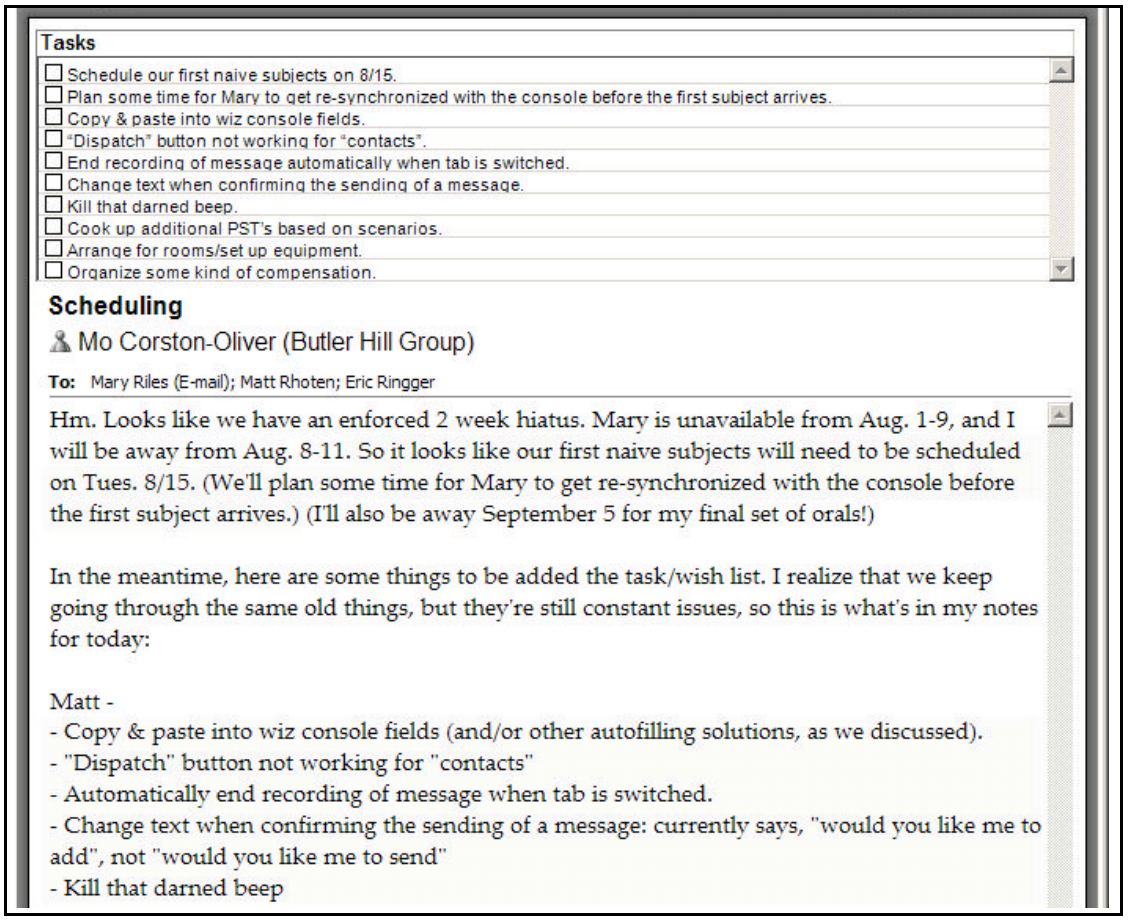

Figure 4: Magnified view of prototype system showing message with enumerated tasks

8 Acknowledgements

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

References

Aikawa, Takako, Maite Melero, Lee Schwartz and Andi Wu. 2001. Multilingual natural language generation. *EAMT.*

Bellotti, Victoria, Nicolas Ducheneaut, Mark Howard , Ian 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*, pages 345-352.

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

Campbell, Richard and Hisami Suzuki. 2002. Language neutral representation of syntactic structure. *Proceedings of SCANALU 2002*.

Dumais, Susan, John Platt, David Heckerman, Mehran Sahami 1998: Inductive learning algorithms and representations for text categorization. *Proceedings of CIKM-98*, pages 148-155.

Heidorn, George. 2000. Intelligent writing assistance. In R. Dale, H. Moisl and H. Somers, (eds.), *Handbook of Natural Language Processing*. Marcel Dekker.

Joachims, Thorsten. 1998. Text categorization with support vector machines: Learning with many relevant features. *Proceedings of ECML* 1998, pages 137-142.

Mast, M., Kompe, R., Harbeck, S., Kiessling, A., Niemann, H., Nöth, E., Schukat-Talamazzini, E. G. and Warnke., V. 1996. Dialog act classification with the help of prosody. *ICSLP 96.*

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

Stolcke, A., E. Shriberg, R. Bates, N. Coccaro, D. Jurafsky, R. Martin, M. Meteer, K. Ries, P. Taylor and C. Van Ess-Dykema. 1998. Dialog act modeling for conversational speech. *Proceedings of the AAAI-98 Spring Symposium on Applying Machine Learning to Discourse Processing.*

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

Warnke, V., R. Kompe, H. Niemann and E. Nöth. 1997. Integrated dialog act segmentation and classification using prosodic features and language models. *Proc. European Conf. on Speech Communication and Technology*, vol 1, pages 207—210.