

Relational Learning for Email Task Management*

Rinat Khoussainov and Nicholas Kushmerick

Department of Computer Science
University College Dublin, Ireland
{rinat, nick}@ucd.ie

1 Introduction and Background

Today’s email clients were designed for yesterday’s email. Originally, email was merely a communication medium. Today, email has become a “habitat” [Ducheneaut and Bellotti, 2001]—an environment where users engage in a variety of complex *activities*. Our goal is to develop automated techniques to help people manage complex activities or *tasks* in email. In many cases, such activities manifest the user’s participation in various structured processes or workflows. The central challenge is that most processes are distributed over multiple emails, yet email clients are designed mainly to manipulate individual messages.

A task-oriented email client would allow the user to manage activities rather than separate messages. For instance, the user would be able to quickly inquire about the current status of unfinished e-commerce transactions or check the outcome of recent project meetings. Some process steps could be automated, such as automatically sending reminders for earlier user’s requests. Similarly, the email client could remind the user when her/his input is required in some activity.

Previous work in this area has mainly focused on two distinct problems: finding related messages and semantic message analysis. The goal of *finding related messages* is to group emails according to tasks and possibly establish conversational links between emails in a task (e.g. extract a task from email given a seed message [Dredze, 2005]). Note that tasks need not correspond to folders (folders can be orthogonal to tasks); and conversations need not correspond to syntactic threads (users can use the “Reply” button or the same subject to start a semantically new conversation).

Semantic message analysis involves generating metadata for individual messages in a task that provides a link between the messages and the changes in the status of the underlying process, or the actions of the user in the underlying workflow. For example, [Cohen *et al.*, 2004] proposed machine learning methods to classify emails according to the intent of the sender, expressed in an ontology of “*email speech acts*”.

Our key innovation compared to related work is that we exploit the *relational structure* of these two tasks. The idea is that related messages in a task provide a valuable context that can be used for semantic message analysis. Similarly, the activity-related metadata in separate messages can pro-

vide relational clues that can be used to establish links between emails and group them into tasks. Instead of treating these two problems separately, we propose a *synergetic approach* where identifying related emails is used to assist semantic message analysis and vice versa.

Our contributions are as follows:

(1) We propose a new method for identifying relations between emails, based on pair-wise message similarity. We extend the similarity function to take into account available structured information in email.

(2) We propose a relational learning approach [Neville and Jensen, 2000] to email task management. We investigate how (a) features of related emails in the same task can assist with classification of speech acts, and how (b) information about speech acts can assist with finding related messages. Combining these two methods yields an iterative relational algorithm for speech act classification and relation identification.

(3) We evaluate our methods on a real-life email corpus.

2 Problem Decomposition and Email Corpora

In a non-relational approach, we would use the content of a message to assign speech acts, and some content similarity between messages to identify relations. In a relational approach to speech act classification, we can use both the message content and features of the related messages from the same task. For example, if a message is a response to a meeting proposal, then it is more likely to be a meeting confirmation or refusal. Similarly, we can use messages’ speech acts to improve relations identification, e.g. a request followed by a delivery are more likely to be related than two requests.

Therefore, we can identify four sub-problems: (P1) find relations in email using content similarity only (i.e. without using messages’ speech acts); (P2) classify messages into speech acts (semantic message analysis) using only the message content (i.e. without using information about the related messages); (P3) use the identified related emails to improve the quality of speech acts classification for a given message; (P4) use the messages’ speech acts to improve identification of relations (links) between emails. These four sub-problems can be combined into a synergetic approach to task management based on an iterative relational classification algorithm illustrated in Figure 1.

We used the PW CALO email corpus [Cohen *et al.*, 2004] for our study. It was generated during a 4-day exercise conducted at SRI specifically to generate an email corpus. During

*This research was funded by the Science Foundation Ireland and the US Office of Naval Research.

```

1: Identify initial relations (P1)
2: Generate initial speech acts (P2)
loop
3: Use related emails in the task to clarify speech acts (P3)
4: Use speech acts to clarify relations between emails (P4)
5: Update messages relations
6: Update messages speech acts
end loop

```

Figure 1: Iterative relational algorithm for task management

Table 1: Identifying relations		Precision	Recall	F1
P1	No time decay, thresh. prune	0.83	0.80	0.81
	Time decay, thresh. prune	0.84	0.80	0.82
	No time decay, threads prune	0.83	0.81	0.82
	Time decay, threads prune	0.84	0.82	0.83
P4	Using speech acts	0.91	0.80	0.85

this time a group of six people assumed different work roles (project leader, finance manager, researcher, etc) and performed a number of activities. Each email has been manually annotated with labels linking it to other emails and also with labels showing the intent of the sender, expressed in a verb-noun ontology of “*email speech acts*” [Cohen *et al.*, 2004]. Examples of speech acts are “Propose meeting”, “Deliver information”. For this study we only use 5 most frequent verbs (“Propose”, “Request”, “Deliver”, “Commit”, “Amend”) as speech acts. To perform experiments, we need to ensure our training and testing sets are unrelated. So we generated 2 non-overlapping corpora with messages received by 2 different users (“User 1”, 160 emails; and “User 2”, 33 emails).

3 Solutions and Results

Identifying Relations without Using Speech Acts (P1). For each email in the corpus, we find the most similar preceding (in time) email using a pair-wise message similarity. Our similarity functions takes into account not only the textual similarity between messages, but also the available structured information in email, such as send dates, and message subjects. The textual similarity is defined as the TF/IDF cosine similarity between email texts. However, the terms appearing in the subject get a higher weight, since people often summarise email content in the subject making subject terms more important. Similarly, related messages tend to be sent around the same time. So, two messages with a large send time difference are less likely to be related. We use the following formula: $Sim(m_1, m_2) = Cosine_Sim(m_1, m_2) * \exp(-\alpha * Norm_time_diff(m_1, m_2))$, where $Cosine_Sim$ is the cosine message similarity, $Norm_time_diff(m_1, m_2)$ is the time difference between messages divided by the maximum time difference, and α is a time decay parameter.

There may be multiple pairs of messages with non-zero similarity in a corpus, however, not all are actually related. Hence, we would like to be able to prune the links suggested by the similarity function. One way is to use some threshold value: if the similarity is below the threshold, the messages are not related. Another way is to use email threads: messages from different threads are not related. Table 1 compares different methods on the “User 1” corpus.

Identifying Relations Using Speech Acts (P4). We treat the problem of finding related messages using speech acts as a supervised learning task. We assume that we have access to a training set, which provides the correct labels for both speech acts and message relations. The goal is to use this information to improve our performance on an unseen email corpus. From the given labelled email corpus, we produce a set of training instances as follows. For each message in the corpus (child), we identify the most similar preceding message (parent) using the previously defined similarity function. For each such pair of messages, we create one training instance with one numeric feature for the similarity between messages, and two subsets of binary features for each possible speech act (10 features in total). The first binary subset is filled with speech acts of the parent message: 1 if the message has this speech act, 0 otherwise. The second binary subset is filled with speech acts of the child message. The class label for the instance is positive if the corresponding messages are related and negative otherwise. The resulting classifier can then be used to identify links in an unseen email corpus.

To evaluate the potential for improvement from using speech acts, we tried to train and test a classifier on the same “User 1” corpus. We use the SMO as our classification algorithm [Platt, 1999]. As shown in Table 1, using speech acts worked here as a more effective pruning method resulting in the increase in precision with only marginal loss in recall.

Classifying Speech Acts without Related Messages (P2). As in the previous case, we treat the problem of email speech act classification as a supervised learning task. We use the standard text classification methods with bag-of-words document representations similar to [Cohen *et al.*, 2004], and SMO as the classification algorithm.

Classifying Speech Acts Using Related Messages (P3). We adopt here the relational learning terminology from [Neville and Jensen, 2000]. Each email message is represented by a set of features: *intrinsic features*, derived from the content of the given message; and *extrinsic features* derived from the properties of related messages in the same task. To represent the intrinsic features of a message, we use the raw term frequencies as in P2. To represent the extrinsic features of a message, we use the speech acts of related messages. We want to know whether speech acts of “surrounding” messages can help in classifying speech acts of a given message. For each speech act, we produce a separate binary classification problem where the goal is to identify whether the message has this act or not.

Each message can be viewed as a response to its parent message and as a cause for its children messages. In addition to looking at the immediate ancestors and descendants of a message, we can also include features from several “generations” of ancestors and descendants (e.g. parents, grandparents, children, grandchildren). For each “generation” of related ancestor and descendant messages, we use a separate set of extrinsic features with one feature per each possible speech act. The number of generations included into extrinsic features is regulated by the *depth of lookup* parameters: one for ancestor messages and one for descendant messages (0 lookup depth means we use only intrinsic features).

We evaluated speech act classification using the human-

Ancest./Descend. lookup	0/0	0/1	1/0	1/1
Amend (p=0.43)	0.40	0.36	0.45 v	0.40
Commit (p=0.05)	0.17	0.20	0.28	0.37 v
Deliver (p=0.24)	0.22	0.29 v	0.21	0.29 v
Propose (p=0.21)	0.08	0.11	0.13 v	0.15
Request (p=0.05)	0.09	0.17	0.14	0.31 v

for all speech acts a **do**

Train C_a on the training set to classify speech act a using only intrinsic features

Train R_a on the training set to classify speech act a using intrinsic+extrinsic features

end for

Train L on the training set to classify email links

*/*Problem 1*/*

Set relations in the test set using similarity function

*/*Iterative classification*/*

for $Iteration = 1 \dots I$ **do**

*/*Problem 2*/*

Use classifiers C_a to set speech acts in the test set

*/*Problem 3*/*

$Threshold = 1$

for $Subiteration = 1 \dots K$ **do**

for all messages m in the test set **do**

for all speech acts a **do**

Obtain confidence for “ m has a ” using R_a

Obtain confidence for “ m has no a ” using R_a

end for

end for

For all cases where confidence for “ m has/has no a ” is greater than $Threshold$ update speech acts of m

$Threshold = Threshold/2$

Evaluate performance for speech acts

end for

*/*Problem 4*/*

Use L to find links between emails in the test set

Evaluate performance for relations

end for

Figure 2: Iterative relational algorithm (detailed version)

annotated (correct) relations between messages and the correct speech acts for related messages on the “User 1” corpus. Notice that using the correct speech acts for related messages does not mean that we use the class label of an instance among its features. Each message uses only the speech acts of the related messages, but not its own speech acts. Classification accuracy is not a good measure of performance on imbalanced data sets, so we use the Kappa statistics instead [Cohen *et al.*, 2004]. The results in Table 2 are obtained in 5-fold cross-validations repeated 10 times for statistical significance testing (paired T-test). “V” marks statistically significant improvements over the *base line* (both lookup depths are 0).

Iterative Algorithm for Task Management (P1 + P2 + P3 + P4). The results for P3 and P4 have demonstrated promise for our synergetic approach to task management. Therefore, here we combine the described methods for solving P1–P4 into the algorithm shown in Figure 2 (which is a detailed version of the algorithm in Figure 1).

In our experiments, we used the “User 1” corpus as the training set and the “User 2” corpus as the test set. To ob-

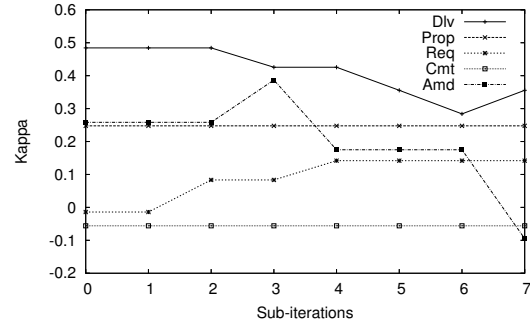


Figure 3: Speech acts classification, Iteration 1

tain confidence scores for SMO, we used the distance from the hyper-plane normalised over all test instances. We use the similarity function with time decay and threshold-based pruning to identify the initial links between messages (P1). We repeated the inner speech act classification loop 10 times ($K = 10$) and the outer iteration loop 2 times ($I = 2$).

The initial links identification resulted in precision = recall = F1 = 0.95. It improved after the first iteration to precision=1.0; recall=0.95; F1=0.98, and remained the same after the second iteration. Figure 3 shows how the speech acts classification performance was changing during the first main iteration. Once the links improved after the first iteration, we were able to further improve the performance for the “Request” speech act at the second iteration to Kappa=0.23.

Discussion. Our experiments demonstrated that: (1) structured features in email, such as message subject and send dates, can be very useful for identification of related messages and grouping them into email tasks; (2) the properties of related messages in the same task can be used to improve the semantic message analysis. In particular, the features of related messages in a task can improve the performance of the email speech acts classification; (3) the semantic metadata in messages can be used to improve the quality of task identification. In particular, taking into account speech acts of messages improves identification of links between emails.

Finally, our combined iterative classification algorithm was able to simultaneously improve performance on both speech acts and message relations. These results provide a good empirical evidence in favour of the proposed synergetic approach to email task management.

References

- [Cohen *et al.*, 2004] W. Cohen, V. Carvalho, and T. Mitchell. Learning to classify email into “speech acts”. In *Empirical Methods in Natural Language Processing*, 2004.
- [Dredze, 2005] M. Dredze. Taxie: Automatically identifying tasks in email. In *Manuscript avail. from the author*, 2005.
- [Ducheneaut and Bellotti, 2001] N. Ducheneaut and V. Bellotti. Email as habitat: An exploration of embedded personal information management. *ACM Interactions*, 8(1), 2001.
- [Neville and Jensen, 2000] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI Workshop on Learning Statistical Models from Relational Data*, 2000.
- [Platt, 1999] John C. Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. MIT Press, 1999.