On the Collective Classification of Email "Speech Acts"

Vitor R. Carvalho
Language Technologies Institute
Carnegie Mellon University
vitor@cs.cmu.edu

William W. Cohen
Center for Automated Learning and Discovery
Carnegie Mellon University
wcohen@cs.cmu.edu

ABSTRACT

We consider classification of email messages as to whether or not they contain certain "email acts", such as a request or a commitment. We show that exploiting the sequential correlation among email messages in the same thread can improve email-act classification. More specifically, we describe a new text-classification algorithm based on a dependency-network based collective classification method, in which the local classifiers are maximum entropy models based on words and certain relational features. We show that statistically significant improvements over a bag-of-words baseline classifier can be obtained for some, but not all, email-act classes. Performance improvement obtained by collective classification is appears to be consistent across email acts suggested by prior speech-act theory.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; H.4.1 [**Information Systems Applications**]: Office Automation; I.5.4 [**Pattern Recognition**]: Applications.

General Terms

Algorithms, Management, Measurement, Performance, Experimentation, Human Factors.

Keywords

Text Classification, Email, Speech Acts, Machine Learning, Collective Classification.

1. INTRODUCTION

One important use of work-related email is negotiating and delegating shared tasks and subtasks. To provide intelligent automated assistance for this use of email, it is desirable to be able to automatically detect the *purpose* of an email message—for example, to determine if the email contains a request, a commitment by the sender to perform some task, or an amendment to an earlier proposal.

In previous work, we presented experimental results on using text classification methods to detect such "speech acts" in email [2]. Based on theories of speech acts, and guided by analysis of several email corpora, we defined a set of "email verbs" (e.g., *Request*, *Deliver*, *Propose*, *Commit*) and considered the problem of classifying emails as to whether or not they contain a specific verb. Thus each verb becomes a binary text classification problem. (Note however that an email may contain several verbs, so the binary classes are not mutually exclusive.) We also defined a set of "email nouns", which are the objects of these verbs (for instance one might *Request* either *Data*, an *Opinion*, or an *Activity*), which were treated analogously.

In our previous work, [2] messages were classified using traditional text classification methods—methods that used features based only on the content of the message. However, it seems reasonable that the *context* of a message is also informative. Specifically, in a sequence of messages, the intent of a reply to a message M will be related to the intent of M: for instance, an email containing a *Request* for a *Meeting* might well be answered by an email that *Commits* to a *Meeting*. More generally, because negotiations are inherently sequential, one would expect strong sequential correlation in the "email-acts" associated with a thread of task-related email messages, and one might hope that exploiting this sequential correlation among email messages in the same thread would improve email-act classification.

The sequential aspects of work-related interactions and negotiations have been investigated by many previous researchers [10, 14]. For example, Winograd and Flores [14] proposed the highly influential idea of a*ction-oriented conversations* based on a particular taxonomy of linguistic acts; an illustration of one of their structures can be seen in Figure 1. However, it is not immediately obvious to what extent prior models of negotiation apply to email. One problem is that email is non-synchronous, so multiple acts are often embedded in a single email. Another problem is that email can be used to actually *perform* certain acts—notably, acts that require the delivery of files or information—as well as being a medium for negotiation. In our previous work, we also noted that certain speech acts that are theoretically possible are either extremely rare or absent, at least in the corpora we analyzed. In short, it cannot be taken for granted that prior linguistic theories apply directly to email.

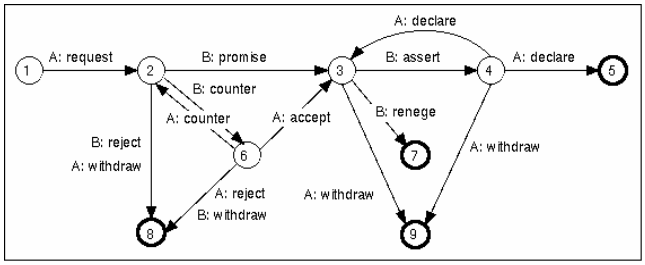

Figure 1. *"Conversation for Action"* Structure from Winograd, T., & Flores, F. (1986).

In this paper we study the use of the sequential information contained in email threads, and more specifically, whether it can improve performance for email-act classification. We first show that sequential correlations do exist; further, that they can be encoded as "relational features", and used to predict the intent of email messages *without* using textual features. We then combine these relational features with textual features, using an iterative collective classification procedure**.** We show that this procedure produces a consistent improvement on some, but not all, email acts.

2. THE CORPUS

Although email is ubiquitous, large and realistic email corpora are rarely available for research purposes due to privacy considerations. The *CSpace* email corpus used in this paper contains approximately 15,000 email messages collected from a management course at Carnegie Mellon University. The email used in our experiments originated from working groups who signed agreements to make certain parts of their email accessible to researchers. In this course, 277 MBA students, organized in approximately 50 teams of four to six members, ran simulated companies in different market scenarios over a 14-week period [8]. The email tends to be very task-oriented, with many instances of task delegation and negotiation.

Messages were mostly exchanged with members of the same team. Accordingly, we partitioned the corpus into subsets according to the teams for many of the experiments. The 1F3 team dataset has 351 messages total, while the 2F2 team has 341, the 3F2 team has 443 and the 4F4 team has 403. In our experiments, we considered only the subset of messages that were in threads (as defined by the reply-To field of the email message), which reduced our actual dataset to 249 emails from 3F2, 170 from 1F3, 137 from 2F2 and 165 messages from 4F4. More precisely, all messages in the original *CSpace* database of monitored email messages contained a *parentID* field, indicating the identity of the message to which the current one is a reply. Using this information, we generated a list of *children* messages (or messages generated in-reply-to this one) to every message. A thread thus consists of a root message and all descendent messages, and in general has the form of a tree, rather than a linear sequence. However, the majority of the threads are short, containing 2 or 3 emails, and most messages have only a single child.

Compared to common datasets used in the relational learning literature, such as IMBd, WebKB or Cora [11], our dataset has a much smaller amount of linkage. A message is linked only to its children and its parent, and there are no relationships between two different threads, or among messages belonging to different threads. However, the relatively small amount of linkage simplified one technical issue in performing experiments with relational learning techniques: ensuring that all test set instances are unrelated to the training set instances. In most of our experiments, we split messages into training and testing sets by teams. Since each of the teams worked largely in isolation from the others, most of their relational information is contained in the same subset.

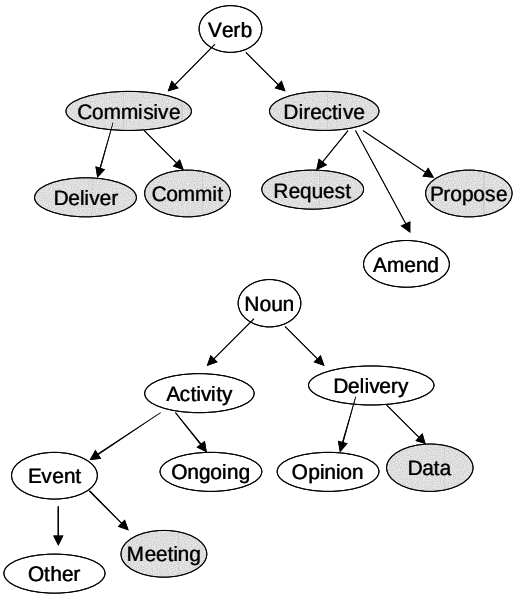

Figure 2. Taxonomy of email-acts used in experiments. Shaded nodes are the ones for which a classifier was constructed.

A taxonomy of speech acts applied to email communication (email-acts) has been described and motivated elsewhere [2]. As noted above, the taxonomy was divided into *verbs* and *nouns*, and each email message is represented by one or more verb-noun pairs: for example, an email proposing a meeting would have the labels *Propose*, *Meeting*. The relevant part of the taxonomy is shown in Figure 2. Very briefly, a *Request* asks the recipient to perform some activity; a *Propose* message proposes a joint activity (i.e., asks the recipient to perform some activity and commits the sender); a *Commit* message commits the sender to some future course of action; *Data* is information, or a pointer to information, delivered to the recipient; and a *Meeting* is a joint activity that is constrained in time and (usually) space. Several other possible verbs/nouns were not considered here (such as *Refuse*, *Greet*, and *Remind)*, either because they occurred very infrequently in our corpus, or because they did not appear to be important for task-tracking. The most common verbs found in the labeled datasets were *Deliver*, *Request*, *Commit*, and *Propose*, and the most common nouns were *Meeting* and *deliveredData* (abbreviated *dData* below). We also consider two aggregations of verbs: the set of *Commissive* acts is the union of *Deliver* and *Commit*, and the set of *Directive* acts is the union of *Request*, *Propose* and *Amend*. (*Amend* is not considered separately here.)

Our prior work [2] showed that machine learning algorithms can learn the proposed email-act categories reasonably accurately. It was also shown that there is an acceptable level of human agreement over the categories. In experiments using different human annotators, Kappa values between 0.72 and 0.85 were obtained. The Kappa statistic [1] is typically used to measure the human inter-rater agreement. Its values ranges from -1 (complete disagreement) to +1 (perfect agreement) and it is defined as *(A-R)/(1-R)*, where *A* is the empirical probability of agreement on a category, and *R* is the probability of agreement

for two annotators that label documents at random (with the empirically observed frequency of each label).

Error rate is a poor measure of performance for skewed classes, since low error rates can be obtained by simply guessing the majority class. Kappa controls for this, since in a highly a skewed class, randomly guessing classes according to the frequency of each class is very similar to always guessing the majority class; thus *R* in the formula will be very close to 1.0. Empirically, Kappa measurements on our datasets are usually closely correlated to the more widely used F1-measure.

3. EVIDENCE FOR SEQUENTIAL CORRELATION OF EMAIL ACTS

3.1 Pairwise correlation of adjacent acts

The sequential nature of email acts is illustrated by the regularities that exist between the acts associated with a message, and the acts associated with its children. The transition diagram in Figure 3 was obtained by computing, for the four most frequent verbs, the probability of the next message's email-act given the current message's act over all four datasets. In other words, an arc from A to B with label *p* indicates that *p* is the probability over all messages M that some child of M has label B, given than M has label A. It is important to notice that an email message may have one or more email-acts associated with it. A *Request*, for instance, may be followed by a message that contains a *Deliver* and also a *Commit*. Therefore, the transition diagram in Figure 3 is *not* a probabilistic DFA.

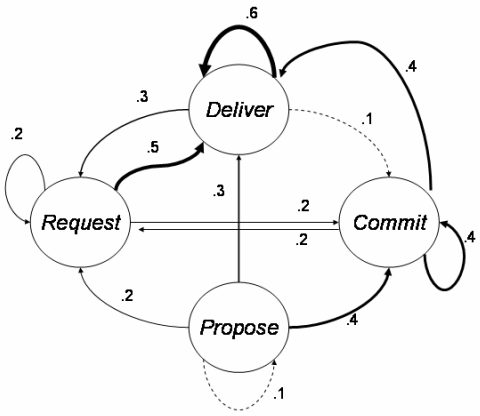

Figure 3. Transition Diagram for the four most common specific verbs.

Deliver and *Request* are the most frequent acts, and they are also closely coupled. Perhaps due to the non-synchronous nature of email and the relatively high frequency of *Deliver*, there is a tendency for almost anything to be followed by a *Deliver* message; however, *Deliver* is especially common after *Request* or another *Deliver*. In contrast, a *Commit* is most probable after a *Propose* or another *Commit*, which agrees with intuitive and theoretical ideas of a negotiation sequence. (Recall that an email thread may involve several people in an activity, all of whom may need to commit to a joint action.) A *Propose* is unlikely to follow anything, as they usually initiate a thread.

Very roughly one can view the graph above as encapsulating three likely types of verb sequences, which could be described with the regular expressions *(Request, Deliver+)*, (*Propose, Commit+, Deliver+)*, and *(Propose, Deliver+)*.

3.2 Predicting acts from surrounding acts

As another test of the degree of sequential correlation in the data, we considered the problem of *predicting* email acts using other acts in the same thread as features. We represented each message with the set of *relational features* shown in Table 1: for instance, the feature *Parent_Request* is true if the parent of contains a request; the feature *Child_Directive* is true if the first[1] child of a message contains a *Directive* speech act.

Table 1 - Set of Relational Features

Parent Features		*Child Features*
Parent_Request, Parent_Deliver, Parent_Commit, Parent_Propose, Parent_Directive, Parent_Commissive, Parent_Meeting, Parent_dData		Child_Request, Child_Deliver, Child_Commit, Child_Propose, Child_Directive, Child_Commissive, Child_Meeting, Child_dData

We performed the following experiment with these features. We trained eight different maximum entropy [9] classifiers, one for each email-act, using only the features from Table 1. (The implementation of the Maximum Entropy classifier was based on the Minorthird toolkit [3]; it uses limited-memory quasi-Newton optimization [13] and a Gaussian prior.) The classifiers were then evaluated in a different dataset. Figure 4 illustrates results using 3F2 as training set and 1F3 as test set, measured in terms of the Kappa statistic. Recall that a Kappa value of zero indicates random agreement, so the results of Figure 4 indicate that there is predictive value in these features. For comparison, we also show the Kappa value of a maximum-entropy classifier using only "content" (bag-of-words features).

Notice that in order to compute the features for a message M, and therefore evaluate the classifiers that predict the email-acts, it is necessary to know what email-acts are contained in the surrounding messages. This circularity means that the experiment above does not suggest a practically useful classification method—although it does help confirm the

[1] The majority of the messages having children have only child, so instead of using features from all children messages, we consider only features from the first child. This restriction makes no significant difference in the results.

intuition that there is useful information in the sequence of classes observed in a thread. Also, it is still possible that the information derivable from the relational features is redundant with the information available in the text of the message; if so, then adding label-sequence information may not improve the overall email-act classification performance. In the next section we consider combining the relational and text features in a practically useful classification scheme.

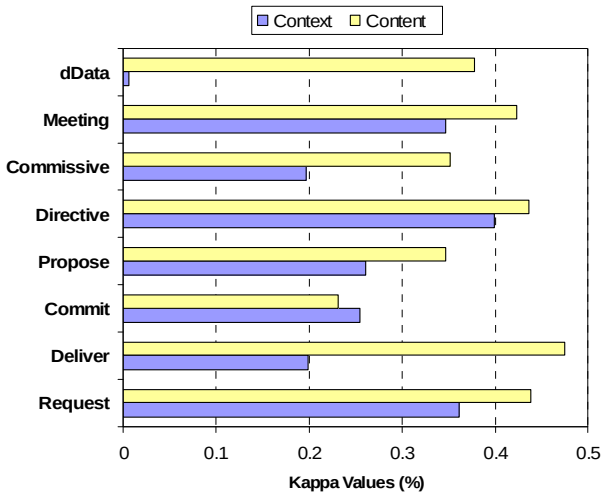

Figure 4. Kappa Values on 1F3 using Relational (Context) features and Textual (Content) features.

4. AN ITERATIVE CLASSIFICATION PROCEDURE

4.1 The Algorithm

In order to construct a practically useful classifier that combines the relational "context" features with the textual "content" features used in traditional bag-of-words text classification [2], it is necessary to break the cyclic dependency between the email acts in a message and the email acts in its parent and children messages. Such a scheme can not classify each message independently: instead classes must be simultaneously assigned to all messages in a thread. Such *collective classification* methods, applied to relationally-linked collections of data, have been an active area of research for several years, and several schemes have been proposed. For instance, using an iterative procedure on a web page dataset, Chakrabarti et al [4] achieved significant improvements in performance compared a non-relational baseline; also, in a dataset of corporate information, Neville and Jensen [11] used an iterative classification algorithm that updates the test set inferences based on classifier confidence. Overviews of recent relational classification papers can be found in elsewhere [7,12].

The scheme we use is dictated by the characteristics of the problem. Every message has multiple binary labels to assign, all of which are potentially interrelated. Further, although in the current paper we consider only parent-child relations implied by the reply-To field, the relational connections between messages are potentially quite rich—for example, it might be plausible to establish connections between messages based on social network connections between recipients as well. We thus adopted a fairly powerful model, based on iteratively re-assigning email-act labels through a process of statistical relaxation.

Initially, we train eight maximum entropy classifiers (one for each act) from a training set. The features used for training are the words on the email body, the words in the email subject, and the relational features listed in Table 1. These eight classifiers will be referred to as *local classifiers*.

The inference procedure used to assign email-act label with these classifiers is as follows. We begin by initializing the eight classes of each message randomly (or according to some other heuristic, as detailed below). We then perform this step iteratively: for each message we infer, using the local classifiers, the prediction confidence of each one of the eight email-acts, given the current labeling of the messages in the thread. (Recall that computing the relational features requires knowing the "context" of the message, represented by the email-act labels of its parent and child messages.) If, for a specific act, the confidence is larger than a *confidence threshold* θ, we accept (update) the act with the label suggested by the local classifier. Otherwise, no updates are made, and the message keeps its previous act.

The confidence threshold θ decreases linearly with the iteration number. Therefore, in the first iteration (j = 0), θ will be 100% and no classes will be updated at all, but after the 50^{th} iteration, θ will be set to 50%, and all messages will be updated. This policy first updates the acts that can be predicted with high confidence and delays the low confidence classifications to the end of the process.

The algorithm is summarized in Figure 5:

1- For each of the 8 email-acts, build a *local classifier* LC_{act} from the training set

2- Initialize the test set with email-act classes based on a content-only classifier.

3- For each iteration j=0 to T:
- 3.1- Update *Confidence Threshold* (%) $\theta = 100 - j$;
- 3.2- If ($\theta < 50$), make $\theta = 50$;
- 3.3- For every email msg in test set, in chronological order:
  - 3.3.1- For each email-act class:
    - 3.3.1.1- obtain confidence(act, msg) from LC_{act}(msg)
    - 3.3.1.2- if (confidence(act,msg) > θ), update email-act of msg
- 3.4- Calculate performance on this iteration

4- Output final inferences and calculate final performance

Figure 5. Iterative Collective Classification Procedure

The iterative collective classification algorithm proposed is in fact an implementation of a Dependency Network (DN) [6]. Dependency networks are probabilistic graphical models in which the full joint distribution of the network is approximated with a set of conditional distributions that can be learned independently. The conditional probability distributions in a DN are calculated for each node given its parent nodes (its *Markov blanket*). In our case, the nodes are the messages in an email

thread, and the Markov blanket is the parent message and the child messages. The confidence threshold represents a temperature-sensitive, annealing variant of Gibbs sampling [5]; after the first 50 iterations, then it reverts to pure Gibbs sampling. In our experiment below, instead of initializing the test set with random email-act classes, we always used a maximum entropy classifier previously trained only with the bag-of-words from a different dataset, and the number of iterations T was set to 60, ensuring 10 iterations of "pure" Gibbs sampling.

4.2 Initial Experiments

Initial experiments used for development were performed using 3F2 as the training set and 1F3 as the test set. Results of these experiments can be found in Table 2 and Table 3. The left part of Table 2 presents the results for when only the bag-of-words features are used (grey bars in Figure 4). The right part of Table 2 shows the performance when training and testing steps use bag-of-words features as well as the true labels of neighboring messages (yellow bars in Figure 4). It reflects the maximum gain that could be granted by using the relational features; therefore, it gives as an "upper bound" of what we should expect from the iterative algorithm.

In addition to Kappa, we report the more widely-used F1 statistic. We also give the improvement in Kappa over the baseline bag-of-words method, where it is relevant

For the *Deliver* act, this "upper bound" is negative: in other words, the presence of the relational features *degrades* the performance of the bag-of-words maximum entropy classifier, even when one assumes the classes of all other messages in a thread are known.

The left side of Table 3 presents the performance of the system if the test set used the estimated labels (instead of the true labels). Equivalently, it represents the performance of the iterative algorithm on its first iteration. The right side of Table 3 shows the performance obtained at the end of the iterative procedure. For every act, Kappa improves as a result of following the iterative procedure. Relative to the bag-of-words baseline, Kappa is improved for all but two acts, *Deliver* (which is again degraded in performance) and *Propose* (which is essentially unchanged.) The highest performance gains are for Commitment and Commissive.

Table 2 - Performance Baseline and Upper Bound

Train:3F2 Test:1F3	Bag-of-words only (Baseline)			Bag-of-words + True Relational Labels (Upper Bound)		
	F1 (%)	Kappa (%)		F1 (%)	Kappa (%)	ΔKappa (%)
Request	63.49	43.87		65.00	47.06	7.27
Deliver	77.08	47.47		74.07	41.67	-12.22
Commit	36.66	23.16		44.44	34.37	48.40
Propose	46.87	34.68		46.66	35.25	1.64
Directive	73.62	43.63		76.50	49.50	13.45
Commissive	77.47	35.19		81.41	44.58	26.68
Meeting	65.18	42.26		68.57	46.60	10.27
dData	41.66	37.76		41.66	37.76	0.00

Table 3 - Performance using Relational Features

Train:3F2 Test:1F3	Bag-of-words + Estimated Relational Labels			Bag-of-words + Estimated Relational Labels + Iterative		
	F1 (%)	Kappa (%)	ΔKappa (%)	F1 (%)	Kappa (%)	ΔKappa (%)
Request	62.29	42.65	-2.78	63.93	45.14	2.89
Deliver	70.65	36.03	-24.10	71.27	35.78	-24.63
Commit	41.50	31.25	34.93	44.06	32.42	39.98
Propose	40.67	28.26	-18.51	45.61	34.66	-0.06
Directive	76.08	48.34	10.80	75.82	48.32	10.75
Comm	80.00	43.47	23.53	82.96	47.83	35.92
Meeting	67.64	46.07	9.02	69.11	48.52	14.81
dData	41.66	37.76	0.00	43.47	40.04	6.04

Figure 6 illustrates the performance of three representative email-acts as the iterative procedure runs. In these curves we can see that two acts (*Commissive and Request*) have their performance improved considerably as the number of iteration increases. Another act, *Deliver*, has a slight deterioration in performance.

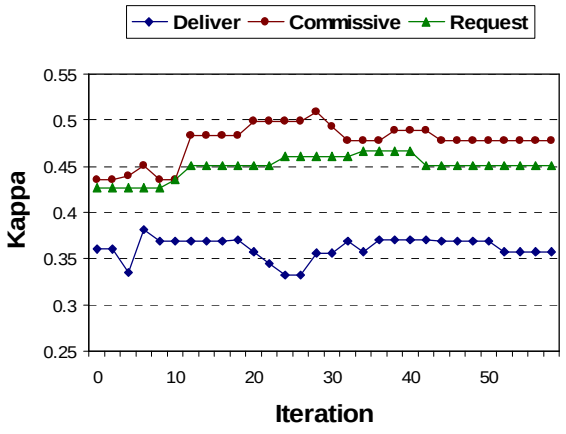

Figure 6. Kappa versus iteration on 1F3, using classifiers trained on 3F2.

4.3 Leave-one-team-out Experiments

In the initial experiments, 3F2 was used as the training set, and 1F3 was the test set. As an additional test, we performed four additional experiments in which data from three teams was used in training, and data from the fourth team was used for testing.

It should be emphasized that the choice to test on email from a team not seen in training makes the prediction problem more difficult, as the different teams tend to adopt slightly different styles of negotiation: for instance, proposals are more frequently used by some groups than others. Higher levels of performance would be expected if we trained and tested on an equivalent quantity of email generated by a single team (as we did in our earlier experiments).

Figure 7 shows a scatter plot, in which each point represents an email act, plotted so that its Kappa value for the bag-of-words baseline is the *x*-axis position, and the Kappa for the iterative procedure is the *y*-axis position. Thus points above the line $y=x$ (the dotted line in the figure) represent an improvement over the baseline. There are four points for each email-act—one for each test team in this "leave one team out" experiment.

As in the preliminary experiments, performance is usually improved. Importantly, performance is improved for six of the eight email acts for the team 4F4, the data for which was collected *after* all algorithm development was complete. Thus performance on 4F4 is a prospective test of the method.

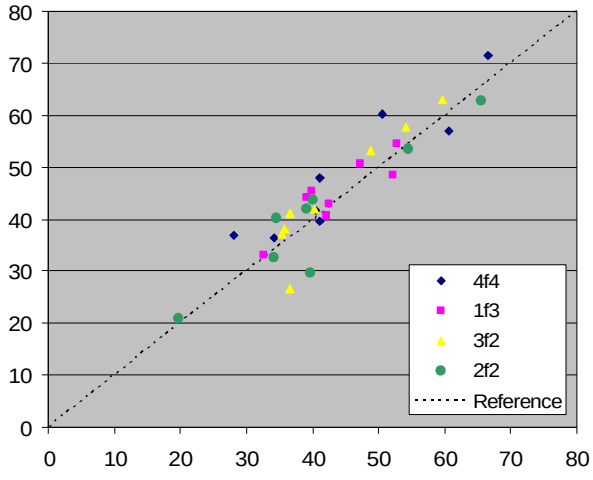

Figure 7. Plot of baseline Kappa (x-axis) versus Kappa after iterative collective classification was performed. Points above the dotted line represent an improvement.

Further analysis suggests that the variations in performance of the iterative scheme are determined largely by the specific email act involved. *Commissive*, *Commit*, and *Meet* were improved most in the preliminary experiments, and *Proposal* and *Delivery* were improved least. The graph of Figure 8 shows that the *Commissive*, *Commit*, and *Meet* are consistently improved by collective classification methods in the prospective tests as well. However, performance on the remaining classes is sometimes degraded.

Finally, Figure 9 shows the same results, with the speech acts broken into two classes: *Deliver* and *dData*, and all other classes. We note that *Deliver* is quite different type of "speech act" from those normally considered in the literature, as it represents use of email as a data-distribution tool, rather than as a medium for negotiation and communication. Figure 3 also shows that *Deliver*, has a fairly high probability of occurring after any speech act, unlike the other verbs. Based on these observations it is reasonable to conjecture that sequential correlations might be different for delivery-related email acts than for other email acts. Figure 9 shows that the collective classification method obtains a more consistent improvement for non-delivery email acts.

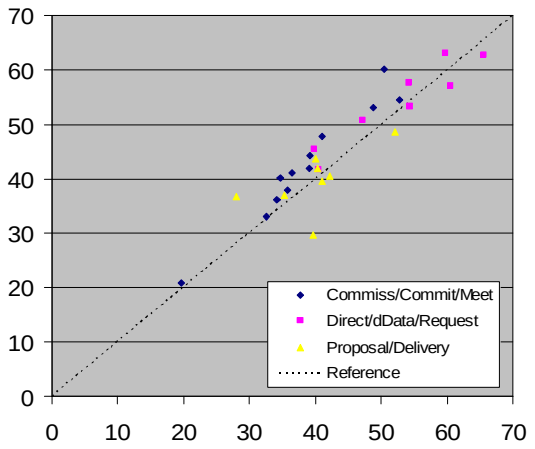

Figure 8. Performance improvement by groups of email-acts. Groups were selected based on performance in the preliminary tests.

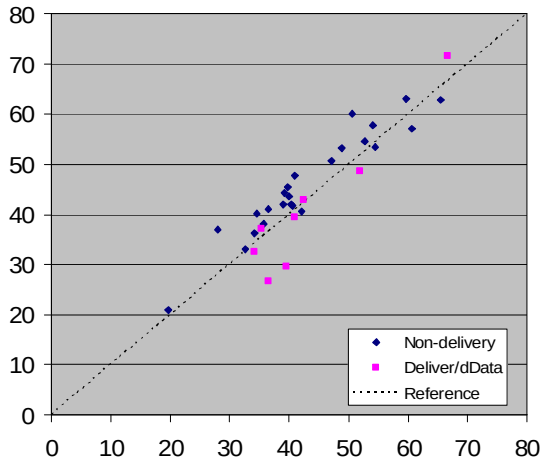

Figure 9. Performance improvement for delivery-related and non-delivery related email acts.

As a final summary of performance, Figure 10 shows, for each of the eight email acts, the Kappa value for each method,

averaged across the four separate test sets. Consistent with the more detailed analysis above, there is an average improvement in average Kappa values for all the non-delivery related acts, but an average loss for *Deliver* and *dData*.

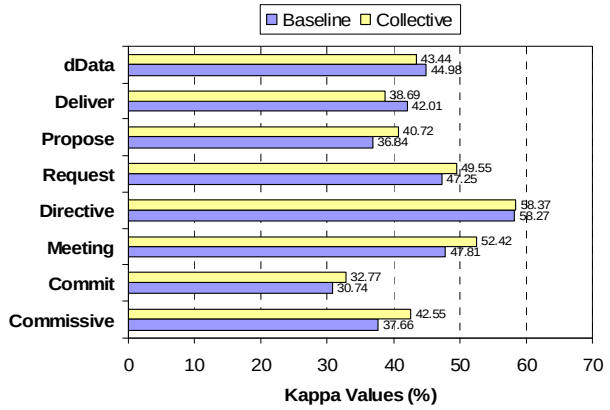

Figure 10. Kappa values with and without collective classification, averaged over the four test sets in the leave-one-team out experiment.

The improvement in average Kappa is statistically significant for the non-delivery related email acts (*p=0.01* on a two-tailed T-test); however, the improvement across all email acts is not statistically significant (*p=0.18).*

The preceding T-test considers significance of the improvement treating the data of Figure 10 as draws from a population of email-act classification problems. One could also take each act separately, and consider the four test values as draws from a population of working teams. This allows one to test the significance of the improvement for a particular email act—but unfortunately, one has only four samples with which to estimate significance. With this test, the improvement in *Commissive* is significant with a two-tailed test (*p=0.01)*, and the improvement in *Meeting* is significant with a one-tailed test (*p=0.04).* The improvement in *Commit* are not significant (*p=0.06* on a one-tailed test). In no case is the loss on performance statistically significant.

5. DISCUSSION

The experiments above demonstrate that a fairly straightforward scheme for collectively classifying email messages in a thread can improve performance. Our scheme is based on a dependency net (DN), in every email-act is predicted by a separate "local" maximum entropy (aka logistic regression) classifier that exploits features that examine the proposed classes of its parent and child email messages. Classification is performed by first proposing email-act labels using a bag-of-words classifier, and then iteratively updating labels using the predictions of the local classifiers—a form of Gibbs sampling.

The method improves performance for some, but not all email-act classes. On a four-fold cross validation test, performance is statistically significantly improved for *Commissive* acts, which include *Commit* and *Deliver*, and performance is very likely improved for *Meet* and *Commit*.

The consistent improvement *Meet* is encouraging, since in addition to recognizing intention, it is also important to recognize the specific task that an email "verb" is relevant to. Meeting arrangement is an easily-recognized task shared by all the teams in our study, and hence the *Meet* email "noun" served as a proxy for this sort of task-classification problem.

Performance is not improved for two of the eight classes, *Deliver* and *dData.* It should be noted that many email *Requests* could plausibly be followed by a *Commit* (e.g., "I'll have the budget ready by Friday") or a *Deliver* (e.g., "I'm attaching the budget you asked for"), and context clues do not predict which type of response will be forthcoming; this may be why context is more useful for predicting *Commissive* acts than the narrower class *Deliver*. We also note that while the email act *Deliver* and its associated object *dData* do model a frequent use of email, they are not suggested by prior theoretical models of negotiation of speech acts. The performance improvement obtained by collective classification is consistent, and statistically significant, across all "non-delivery" acts—i.e., across all acts suggested by prior theory.

6. CONCLUSIONS

In this work we explored how the relational information in an email thread can be used help classifying email according to the user's intent (that is to recognize email-acts). While it can be addressed using traditional text classification methods, email-act classification has unique characteristics [2]. Here we showed that the sequence of email-acts in a thread of email messages contain information useful for classifying certain email acts. This idea is appealing and agrees with the general intuition that, for instance, a *Commit* message is likely to be preceded by a *Request* or *Propose*, or that a *Request* is likely to be followed by a *Deliver*.

Specifically, we showed that modest but statistically significant improvements for some email-act classes are obtained by applying a dependency-network based collective classification method, in which the local classifiers are maximum entropy models based on words and certain relational features. Statistical tests suggest that the method we proposed will improve most email-acts that are justified by prior speech-act theory.

These results are encouraging as the degree of linkage in our data is small, the data is highly variable. The variability arises in part because different teams adopt different task negotiation and delegation styles, and in our experiments to date, data from one set of teams is always used to learn email-act classifiers for another team. In future work we hope to study the relative value of training data obtained from other teams, and data obtained from the team whose email-acts are being predicted. This is an important question, because it clarifies the degree to which classifiers for email-acts are team- or person-dependent.

It may also be helpful to consider additional external features that might be useful in linking data—for instance, features that relate entities in email messages to a task, or features that relate the senders and receivers via social network

properties. Such features could be easily integrated into our model.

7. REFERENCES

[1] Carletta, J. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics, 22, 2* (1996), 249-254.

[2] Cohen, W.W., Carvalho, V. R., and Mitchell, T.M. Learning to Classify Email into "Speech Acts". *Proceedings of the EMNLP (Conference on Empirical Methods in Natural Language Processing),* (Jul. 2004).

[3] Cohen, W.W., Minorthird: Methods for Identifying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data, *http://minorthird.sourceforge.net, (2004)*

[4] Chakrabarti, S and Indyk P. Enhanced Hypertext Categorization Using Hypelinks. *Proceedings of the ACM SIGMOD* (1998), Seattle, Washington.

[5] Geman, S. and Geman, D. Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* (1984) 6: 721-741

[6] Heckerman, D., D. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering and data visualization. *Journal of Machine Learning Research*, 1, (2000), 49-75

[7] Jensen, D., Neville, J. and Gallagher,B. Why Collective Classification Inference Improves Relational Classification. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. (2004)

[8] Kraut,R.E., Fussell, S.R., Lerch, F.J. and Espinosa, A. Coordination in Teams: Evidence from a Simulated Management Game. *To appear in the Journal of Organizational Behavior*

[9] Berger, A., Della Pietra, S., Della Pietra, V. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), (1996) pp. 39 – 71.

[10] Murakoshi, H., Shimazu, A. and Ochimizu, K. Construction of Deliberation Structure in Email Communication. *Pacific Association for Computational Linguistics*. (1999). pp. 16-28.

[11] Neville, J. and Jensen, D. Iterative Classification in Relational Data. *In L. Getoor and D. Jensen (Eds). Papers of the AAAI-2000 Workshop on Learning Statistical Models from Relational Data.* (2000).AAAI Press.

[12] Neville, J., Rattigan M. and Jensen, D. Statistical Relational Learning: Four Claims and a Survey. *Proceedings of the Workshop on Learning Statistical Models from Relational Data, 18th International Joint Conference on Artificial Intelligence* (2003).

[13] Sha, F. and Pereira, F. Shallow Parsing with Conditional Random Fields. *HLT-NAACL, ACM,* (2003)

[14] Schoop,M. A Language-Action Approach to Electronic Negotiations. *LAP Proc. of the Eighth Annual Working Conference on Language-Action Perspective on Communication Modelling.* (2003)

[15] Winograd, T. and Flores, C.F. Understanding Computers and Cognition. *Ablex Publishing Corp., Norwood, NJ,* (1986).