Detecting Emails Containing Requests for Action

Andrew Lampert †‡
†CSIRO ICT Centre
PO Box 76
Epping 1710
Australia
andrew.lampert@csiro.au

Robert Dale
‡Centre for Language Technology
Macquarie University 2109
Australia
rdale@science.mq.edu.au

Cecile Paris
CSIRO ICT Centre
PO Box 76
Epping 1710
Australia
cecile.paris@csiro.au

Abstract

Automatically finding email messages that contain requests for action can provide valuable assistance to users who otherwise struggle to give appropriate attention to the actionable tasks in their inbox. As a speech act classification task, however, automatically recognising requests in free text is particularly challenging. The problem is compounded by the fact that typical emails contain extraneous material that makes it difficult to isolate the content that is directed to the recipient of the email message. In this paper, we report on an email classification system which identifies messages containing requests; we then show how, by segmenting the content of email messages into different functional zones and then considering only content in a small number of message zones when detecting requests, we can improve the accuracy of message-level automated request classification to 83.76%, a relative increase of 15.9%. This represents an error reduction of 41% compared with the same request classifier deployed without email zoning.

1 Introduction

The variety of linguistic forms that can be used to express requests, and in particular the frequency with which indirect speech acts are used in email, is a major source of difficulty in determining whether an email message contains one or more requests. Another significant problem arises from the fact that whether or not a request is directed at the recipient of the email message depends on where in the message the request is found. Most obviously, if the request is part of a replied-to message that is contained within the current message, then it is perhaps more likely that this request was directed at the sender of the current message. However, separating out content intended for the recipient from other extraneous content is not as simple as it might appear. Segmenting email messages into their different functional parts is hampered by the lack of standard syntax used by different email clients to indicate different message parts, and by the *ad hoc* ways in which people vary the structure and layout of messages.

In this paper, we present our results in classifying messages according to whether or not they contain requests, and then show how a separate classifier that aims to determine the nature of the zones that make up an email message can improve upon these results. Section 2 contains some context and motivation for this work before we briefly review relevant related work in Section 3. Then, in Section 4, we describe a first experiment in request classification using data gathered from a manual annotation experiment. In analysing the errors made by this classifier, we found that a significant number of errors seemed to arise from the inclusion of content in parts of a message (e.g., quoted reply content) that were not authored by the current sender, and thus were not relevant other than as context for interpreting the current message content. Based on this analysis, we hypothesised that segmenting messages into their different functional parts, which we call **email zones**, and then using this information to consider only content from certain parts of a message for request classification, would improve request classification performance.

To test this hypothesis, we developed an SVM-based automated email zone classifier configured with graphic, orthographic and lexical features; this is described in more detail in (Lampert et al., 2009). Section 5 describes how we improve request classification performance using this email zone classifier. Section 6 summarises the performance of our request classifiers, with and without automated email zoning, along with an analysis of the contribution of lexical features to request classification, discussion of request classification learning curves, and a detailed error analysis that explores the sources of request classification errors. Finally, in Section 7, we offer pointers to future work and some concluding remarks.

2 Background and Motivation

Previous research has established that users routinely use email for managing requests in the workplace — e.g., (Mackay, 1988; Ducheneaut and Bellotti, 2001). Such studies have highlighted how managing multiple ongoing tasks through email leads to information overload (Whittaker and Sidner, 1996; Bellotti et al., 2003), especially in the face of an ever-increasing volume of email. The result is that many users have difficulty giving appropriate attention to requests hidden in their email which require action or response. A particularly lucid summary of the requirements placed on email users comes from work by Murray (1991), whose ethnographic research into the use of electronic messaging at IBM highlighted that:

> [Managers] would like to be able to track outstanding promises they have made, promises made to them, requests they've made that have not been met and requests made of them that they have not fulfilled.

This electronic exchange of requests and commitments has previously been identified as a fundamental basis of the way work is delegated and completed within organisations. Winograd and Flores were among the first to recognise and attempt to exploit this with their Coordinator system (Winograd and Flores, 1986). Their research into organisational communication concluded that "Organisations exist as networks of directives and commissives". It is on this basis that our research explores the use of requests (directive speech acts) and commitments (commissive speech acts) in email. In this paper, we focus on requests; feedback from users of the request and commitment classifier plug-in for Microsoft Outlook that we have under development suggests that, at least within the business context of our current users, requests are the more important of the two phenomena.

Our aim is to create tools that assist email users to identify and manage requests contained in incoming and outgoing email. We define a request as an utterance that places an obligation on an email recipient to schedule an action; perform (or not perform) an action; or to respond with some speech act. A simple example might be *Please call when you have a chance*. A more complicated request is *David will send you the latest version if there have been any updates*. If David (perhaps cc'ed) is a recipient of an email containing this second utterance, the utterance functions as a (conditional) request for him, even though it is addressed as a commitment to a third-party. In real-world email, requests are frequently expressed in such subtle ways, as we discuss in Section 4.

A distinction can be drawn between **message-level identification**—i.e., the task of determining *whether* an email message contains a request — and **utterance-level identification**—i.e., determining precisely where and how the request is expressed. In this paper, we focus on the task of message-level identification, since utterance-level identification is a significantly more problematic task: it is often the case that, while we might agree that a message contains a request or commitment, it is much harder to determine the precise extent of the text that conveys this request (see (Lampert et al., 2008b) for a detailed discussion of some of the issues here).

3 Related Work

Our request classification work builds on influential ideas proposed by Winograd and Flores (1986) in taking a language/action perspective and identifying speech acts in email. While this differs from the approach of most currently-used email systems, which routinely treat the content of email messages as homogeneous bags-of-words, there is a growing body of research applying ideas from Speech Act Theory (Austin, 1962; Searle, 1969) to analyse and enhance email communication.

Khosravi and Wilks (1999) were among the first to automate message-level request classification in email. They used cue-phrase based rules to classify three classes of requests: Request-Action, Request-Information and Request-Permission. Unfortunately, their approach was quite brittle, with the rules being very specific to the computer support domain from which their email data was drawn.

Cohen, Carvalho and Mitchell (2004) developed machine learning-based classifiers for a number of email speech acts. They performed manual email zoning, but didn't explore the contribution this made to the performance of their various speech act classifiers. For requests, they report peak F-measure of 0.69 against a majority class baseline accuracy of approximately 66%. Cohen, Carvalho and Mitchell found that unweighted bigrams were particularly useful features in their experiments, out-performing other features applied. They later applied a series of text normalisations and n-gram feature selection algorithms to improve performance (Carvalho and Cohen, 2006). We apply similar normalisations in our work. While difficult to compare due to the use of a different email corpus that may or may not exclude annotation disagreements, our request classifier performance exceeds that of the enhanced classifier reported in (Carvalho and Cohen, 2006).

Goldstein and Sabin (2006) have also worked on related email classification tasks. They use verb classes, along with a series of hand-crafted form- and phrase-based features, for classifying what they term **email genre**, a task which overlaps significantly with email speech act classification. Their results are difficult to compare since they include a mix of form-based classifications like **response** with more intent-based classes such as **request**. For requests, the results are rather poor, with precision of only 0.43 on a small set of personal mail.

The SmartMail system (Corston-Oliver et al., 2004) is probably the most mature previous work on utterance-level request classification. SmartMail attempted to automatically extract and reformulate action items from email messages for the purpose of adding them to a user's to-do list. The system employed a series of deep linguistic features, including phrase structure and semantic features, along with word and part-of-speech n-gram features. The authors found that word n-grams were highly predictive for their classification task, and that there was little difference in performance when the more expensive deep linguistic features were added. Based on this insight, our own system does not employ deeper linguistic features. Unfortunately, the results reported reveal only the aggregate performance across all classes, which involves a mix of both form-based classes (such as signature content address lines and URL lines), and intent-based classes (such as requests and promises). It is thus very difficult to directly compare the results with our system. Additionally, the experiments were performed over a large corpus of messages that are not available for use by other researchers. In contrast, we use messages from the widely-available Enron email corpus (Klimt and Yang, 2004) for our own experiments.

While several of the above systems involve manual processes for removing particular parts of message bodies, none employ a comprehensive, automated approach to email zoning.

We focus on the combination of email zoning and request classification tasks and provide details of how email zoning improves request classification — a task not previously explored. To do so, we require an automated email zone classifier. We experimented with using the Jangada system (Carvalho and Cohen, 2004), but found similar shortcomings to those noted by Estival et al. (2007). In particular, Jangada did not accurately identify forwarded or reply content in email messages from the email Enron corpus that we use. We achieved much better performance with our own Zebra zone classifier (Lampert et al., 2009); it is this system that we use for email zoning throughout this paper.

4 Email Request Classification

Identifying requests requires interpretation of the intent that lies behind the language used. Given this, it is natural to approach the problem as one of **speech act identification**. In Speech Act Theory, speech acts are categories like **assertion** and **request** that capture the intentions underlying surface utterances, providing abstractions across the wide variety of different ways in which instances of those categories might be realised in linguistic form. In this paper we focus on the speech acts that represent requests, where people are placing obligations upon others via actionable content within email messages.

The task of building automated classifiers is difficult since the function of conveying a request does not neatly map to a particular set of language forms; requests often involve what are referred to as **indirect speech acts**. While investigating particular surface forms of language is relatively unproblematic, it is widely recognised that “investigating a collection of forms that represent, for example, a particular speech act leads to the problem of establishing which forms constitute that collection” (Archer et al., 2008). Email offers particular challenges as it has been shown to exhibit a higher frequency of indirect speech acts than other media (Hassell and Christensen, 1996). We approach the problem by gathering judgments from human annotators and using this data to train supervised machine learning algorithms.

Our request classifier works at the message-level, marking emails as requests if they contain one or more request utterances. As noted earlier, we define a request as an utterance from the email sender that places an obligation on a recipient to schedule an action (e.g., add to a calendar or task list), perform an action, or respond. Requests may be conditional or unconditional in terms of the obligation they impose on the recipient. Conditional requests require action only if a stated condition is satisfied. Previous annotation experiments have shown that conditional requests are an important phenomena and occur frequently in email (Scerri et al., 2008; Lampert et al., 2008a). Requests may also be phrased as either a direct or indirect speech act.

Although some linguists distinguish between speech acts that require a physical response and those that require a verbal or information response, e.g., (Sinclair and Coulthard, 1975), we follow Searle’s approach and make no such distinction. We thus consider questions requiring an informational response to be requests, since they place an obligation on the recipient to answer.[1]

Additionally, there are some classes of request which have been the source of systematic human disagreement in our previous annotation experiments. One such class consists of **requests for inaction**. Requests for inaction, sometimes called **prohibitives** (Sadock and Zwicky, 1985), prohibit action or request negated action. An example is: *Please don’t let anyone else use the computer in the office*. As they impose an obligation on the sender, we consider requests for inaction to be requests. Similarly, we consider that meeting announcements (e.g., *Today’s Prebid Meeting will take place in EB32c2 at 3pm*) and requests to read, open or otherwise act on documents attached to email messages (e.g., *See attached*) are also requests.

Several complex classes of requests are particularly sensitive to the context for their interpretation. Reported requests are one such class. Some reported requests, such as *Paul asked if you could put together a summary of your accomplishments in an email*, clearly function as a request. Others do not impose an obligation on the recipient, e.g., *Sorry for the delay; Paul requested your prize to be sent out late December*. The surrounding context must be used to determine the intent of utterances like reported requests. Such distinctions are often difficult to automate.

Other complex requests include instructions. Sometimes instructions are of the kind that one might ‘file for later use’. These tend to not be marked as requests. Other instructions, such as *Your user id and password have been set up. Please follow the steps below to access the new environment*, are intended to be executed more promptly. Temporal distance between receipt of the instruction and expected action is an important factor to distinguish between requests and non-requests. Another influencing property is the likelihood of the trigger event that would lead to execution of the described action. While the example instructions above are likely to be executed, instructions for how to handle suspected anthrax-infected mail are (for most people) unlikely to be actioned.

Further detail and discussion of these and other

[1]Note, however, that not all questions are requests. Rhetorical questions are perhaps the most obvious class of non-request questions.

challenges in defining and interpreting requests in email can be found in (Lampert et al., 2008b). In particular, that paper includes analysis of a series of complex edge cases that make even human agreement in identifying requests difficult to achieve.

4.1 An Email Request Classifier

Our request classifier is based around an SVM classifier, implemented using Weka (Witten and Frank, 2005). Given an email message as input, complete with header information, our binary request classifier predicts the presence or absence of request utterances within the message.

For training our request classifier, we use email from the database dump of the Enron email corpus released by Andrew Fiore and Jeff Heer.[2] This version of the corpus has been processed to remove duplicate messages and to normalise sender and recipient names, resulting in just over 250,000 email messages. No attachments are included.

Our request classifier training data is drawn from a collection of 664 messages that were selected at random from the Enron corpus. Each message was annotated by three annotators, with overall kappa agreement of 0.681. From the full dataset of 664 messages, we remove all messages where annotators disagreed for training and evaluating our request classifier, in order to mitigate the effects of annotation noise, as discussed in (Beigman and Klebanov, 2009). The unanimously agreed data set used for training consists of 505 email messages.

4.2 Request Classification Features

The features we use in our request classifier are:

- message length in characters and words;
- number and percentage of capitalised words;
- number of non alpha-numeric characters;
- whether the subject line contains markers of email replies or forwards (e.g. `Re:`, `Fw:`);
- the presence of sender or recipient names;
- the presence of sentences that begin with a modal verb (e.g., *might*, *may*, *should*, *would*);
- the presence of sentences that begin with a question word (e.g, *who*, *what*, *where*, *when*, *why*, *which*, *how*);
- whether the message contains any sentences that end with a question mark; and
- binary word unigram and word bigram features for n-grams that occur at least three times across the training set.

[2] http://bailando.sims.berkeley.edu/enron/enron.sql.gz

Before generating n-gram features, we normalise the message text as shown in Table 1, in a manner similar to Carvalho and Cohen (2006). We also add tokens marking the start and end of sentences, detected using a modified version of Scott Piao's sentence splitter (Piao et al., 2002), and tokens marking the start and end of the message.

Symbol Used	Pattern
numbers	Any sequence of digits
day	Day names or abbreviations
pronoun-object	Objective pronouns: *me*, *her*, *him*, *us*, *them*
pronoun-subject	Subjective pronouns: *I*, *we*, *you*, *he*, *she*, *they*
filetype	.doc, .pdf, .ppt, .txt, .xls, .rtf
multi-dash	3 or more sequential '–' characters
multi-underscore	3 or more sequential '_' characters

Table 1: Normalisation applied to n-gram features

Our initial request classifier achieves an accuracy of 72.28%. Table 2 shows accuracy, precision, recall and F-measure results, calculated using stratified 10-fold cross-validation, compared against a majority class baseline. Given the well-balanced nature of our training data (52.08% of messages contain a request), this is a reasonable basis for comparison.

	Majority Baseline		No Zoning Classifier	
	Request	Non-Request	Request	Non-Request
Accuracy	52.08%		72.28%	
Precision	0.521	0.000	0.729	0.716
Recall	1.000	0.000	0.745	0.698
F-Measure	0.685	0.000	0.737	0.707

Table 2: Request classifier results without email zoning

An error analysis of the predictions from our initial request classifier uncovered a series of classification errors that appeared to be due to request-like signals being picked up from parts of messages such as email signatures and quoted reply content. It seemed likely that our request classifier would benefit from an email zone classifier that could identify and ignore such message parts.

5 Improving Request Classification with Email Zoning

Requests in email do not occur uniformly across the zones that make up the email message. There are specific zones of a message in which requests are likely to occur.

Unfortunately, accurate classification of email zones is difficult, hampered by the lack of standard syntax used by different email clients to indicate different message parts, and by the *ad hoc* ways in which people vary the structure and layout of their messages. For example, different email clients indicate quoted material in a variety of ways. Some prefix every line of the quoted message with a character such as '>' or '|', while others indent the quoted content or insert the quoted message unmodified, prefixed by a message header. Sometimes the new content is above the quoted content (a style known as **top-posting**); in other cases, the new content may appear after the quoted content (**bottom-posting**) or interleaved with the quoted content (**inline replying**). Confounding the issue further is that users are able to configure their email client to suit their individual tastes, and can change both the syntax of quoting and their quoting style (top, bottom or inline replying) on a per message basis.

Despite the likelihood of some noise being introduced through mis-classification of email zones, our hypothesis was that even imperfect information about the functional parts of a message should improve the performance of our request classifier.

Based on this hypothesis, we integrated Zebra (Lampert et al., 2009), our SVM-based email zone classifier, to identify the different functional parts of email messages. Using features that capture graphic, orthographic and lexical information, Zebra classifies and segments the body text into nine different email zones: author content (written by the current sender), greetings, signoffs, quoted reply content, forwarded content, email signatures, advertising, disclaimers, and automated attachment references. Zebra has two modes of operation, classifying either message fragments — whitespace separated sets of contiguous lines — or individual lines. We configure Zebra for line-based zone classification, and use it to extract only lines classified as author, greeting and signoff text. We remove the content of all other zones before we evaluate features for request classification.

6 Results and Discussion

Classifying the zones in email messages and applying our request classifier to only relevant message parts significantly increases the performance of the request classifier. As noted above, without zoning, our request classifier achieves accuracy of 72.28% and a weighted F-measure (weighted between the F-measure for requests and non-requests based on the relative frequency of each class) of 0.723. Adding the zone classifier, we increase the accuracy to 83.76% and the weighted F-measure to 0.838. This corresponds to a relative increase in both accuracy and weighted F-measure of 15.9%, which in turn corresponds to an error reduction of more than 41%. Table 3 shows a comparison of the results of the non-zoning and zoning request classifiers, generated using stratified 10-fold cross-validation. In a two-tailed paired t-test, run over ten iterations of stratified 10-fold cross-validation, the increase in accuracy, precision, recall and f-measure were all significant at p=0.01.

	No Zoning		With Zoning	
	Request	Non-Request	Request	Non-Request
Accuracy	72.28%		83.76%*	
Precision	0.729	0.716	0.849*	0.825*
Recall	0.745	0.698	0.837*	0.839*
F-Measure	0.737	0.707	0.843*	0.832*

Table 3: Request classifier results with and without email zoning (* indicates a statistically significant difference at p=0.01)

6.1 Lexical Feature Contribution

As expected, lexical information is crucial to request classification. When we experimented with removing all lexical (n-gram) features, the non-zoning request classifier accuracy dropped to 57.62% and the zoning request classifier accuracy dropped to 61.78%. In contrast, when we apply *only* n-gram features, we achieve accuracy of 71.49% for the non-zoning classifier and 83.36% for the zoning classifier. Clearly, lexical information is critical for accurate request classification, regardless of whether email messages are zoned.

Using Information Gain, we ranked the n-gram features in terms of their usefulness. Table 4 shows the top-10 unigrams and bigrams for our non-zoning request classifier. Using these top-10 n-grams (plus our non-n-gram features), we achieve only 66.34% accuracy. These top-10 n-grams do not seem to align well with linguistic intuitions, illustrating how the noise from irrelevant message parts hampers performance. In particular, there were several similar, apparently automated messages that were annotated (as non-requests) which appear to be the source of several of the top-10 n-grams. This strongly suggests that without zoning, the classifier is not learning features from the training set at a useful level of generality.

Word Unigrams	Word Bigrams	
	Word 1	Word 2
pronoun-object	let	*pronoun-object*
please	*pronoun-object*	know
iso	*start-sentence*	no
pronoun-subject	start	date
hourahead	hour	:
attached	;	hourahead
let	hourahead	hour
westdesk	*start-sentence*	start
parsing	westdesk	/
if	iso	final

Table 4: Top 10 useful n-grams for our request classifier without zoning, ranked by Information Gain

In contrast, once we add the zoning classifier, the top-10 unigrams and bigrams appear to correspond much better with linguistic intuitions about the language of requests. These are shown in Table 5. Using these top-10 n-grams (plus our non-n-gram features), we achieve 80% accuracy. This suggests that, even with our relatively small amount of training data, the zone classifier helps the request classifier to extract fairly general n-gram features.

Interestingly, although lexical features are very important, the top three features ranked by Information Gain are non-lexical: message length in words, the number of non-alpha-numeric characters in the message and the number of capitalised words in the message.

Word Unigrams	Word Bigrams	
	Word 1	Word 2
please	?	*end-sentence*
?	*pronoun-object*	know
pronoun-object	let	*pronoun-object*
if	*start-sentence*	please
pronoun-subject	if	*pronoun-subject*
let	*start-sentence*	thanks
to	please	let
know	*pronoun-subject*	have
thanks	thanks	*comma*
do	start	date

Table 5: Top 10 useful n-grams for our request classifier with zoning, ranked by Information Gain

6.2 Learning Curves

Figure 1 shows a plot of accuracy, precision and recall versus the number of training instances used to build the request classifier. These results are calculated over zoned email bodies, using the average across ten iterations of stratified 10-fold cross-validation for each different sized set of training instances, implemented via the `FilteredClassifier` with the `Resample` filter in Weka. Given our pool of 505 agreed message annotations, we plot the recall and precision for training instance sets of size 50 to 505 messages.

There is a clear trend of increasing performance as the training set size grows. It seems reasonable to assume that more data should continue to facilitate better request classifier performance. To this end, we are annotating more data as part of our current and future work.

6.3 Error Analysis

To explore the errors made by our request classifier, we examined the output of our zoning request classifier using our full feature set, including all word n-grams.

Approximately 20% of errors relate to requests that are implicit, and thus more difficult to detect from surface features. Another 10% of errors are due to attempts to classify requests in inappropriate genres of email messages. In particular, both marketing messages and spam frequently include request-like, directive utterances which our annotators all agreed would not be useful to mark as re-

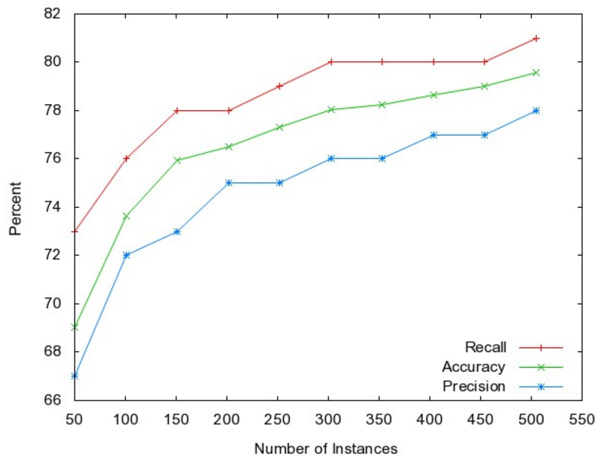

Figure 1: Learning curve showing recall, accuracy and precision versus the number of training instances

quests for an email user. Not unreasonably, our classifier is sometimes confused by the content of these messages, mistakenly marking requests where our annotators did not. We intend to resolve these classification errors by filtering out such messages before we apply the request classifier.

Another 5% of errors are due to request content occurring in zones that we ignore. The most common case is content in a forwarded zone. Sometimes email senders forward a message as a form of task delegation; because we ignore forwarded content, our request classifier misses such requests. We did experiment with including content from forwarded zones (in addition to the author, greeting and signoff zones), but found that this reduced the performance of our request classifier, presumably due to the additional noise from irrelevant content in other forwarded material. Forwarded messages are thus somewhat difficult to deal with. One possible approach would be to build sender-specific profiles that might allow us to deal with forwarded content (and potentially content from other zones) differently for different users, essentially learning to adapt to the different styles of different email users.

A further 5% of errors involve errors in the zone classifier, which leads to incorrect zone labels being applied to zone content that we would wish to include for our request classifier. Examples include author content being mistakenly identified as signature content. In such cases, we incorrectly remove relevant content from the body text that is passed to our request classifier. Improvements to the zone classifier would resolve these issues.

As part of our annotation task, we also asked coders to mark the presence of **pleasantries**. We define a pleasantry as an utterance that could be a request in some other context, but that does not function as a request in the context of use under consideration. Pleasantries are frequently formulaic, and do not place any significant obligation on the recipient to act or respond. Variations on the phrase *Let me know if you have any questions* are particularly common in email messages. The context of the entire email message needs to be considered to distinguish between when such an utterance functions as a request and when it should be marked as a pleasantry. Of the errors made by our request classifier, approximately 5% involve marking messages containing only pleasantries as containing a request.

The remaining errors are somewhat diverse. Close to 5% involve errors interpreting requests associated with attached files. The balance of almost 50% of errors involve a wide range of issues, from misspellings of key words such as *please* to a lack of punctuation cues such as question marks.

7 Conclusion

Request classification, like any form of automated speech act recognition, is a difficult task. Despite this inherent difficulty, the automatic request classifier we describe in this paper correctly labels requests at the message level in 83.76% of email messages from our annotated dataset. Unlike previous work that has attempted to automate the classification of requests in email, we zone the messages without manual intervention. This improves accuracy by 15.9% relative to the performance of the same request classifier without the assistance of an email zone classifier to focus on relevant message parts. Although some zone classification errors are made, error analysis reveals that only 5% of errors are due to zone misclassification of message parts. This suggests that, although zone classifier performance could be further improved, it is likely that focusing on improving the request classifier using the existing zone classifier performance will lead to greater performance gains.

References

Dawn Archer, Jonathan Culpeper, and Matthew Davies, 2008. *Corpus Linguistics: An International Handbook*, chapter Pragmatic Annotation, pages 613–642. Mouton de Gruyter.

John L Austin. 1962. *How to do things with words*. Harvard University Press.

Eyal Beigman and Beata Beigman Klebanov. 2009. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJCNLP*, pages 280–287, Singapore.

Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, and Ian Smith. 2003. Taking email to task: The design and evaluation of a task management centred email tool. In *Computer Human Interaction Conference*, CHI, pages 345–352, Ft Lauderdale, Florida.

Vitor R Carvalho and William W Cohen. 2004. Learning to extract signature reply lines from email. In *Proceedings of First Conference on Email and Anti-Spam (CEAS)*, Mountain View, CA, July 30-31.

Vitor R. Carvalho and William W. Cohen. 2006. Improving email speech act analysis via n-gram selection. In *Proceedings of HLT/NAACL 2006 - Workshop on Analyzing Conversations in Text and Speech*, pages 35–41, New York.

William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. 2004. Learning to classify email into "speech acts". In *Conference on Empirical Methods in Natural Language Processing*, pages 309–316, Barcelona, Spain.

Simon H. Corston-Oliver, Eric Ringger, Michael Gamon, and Richard Campbell. 2004. Task-focused summarization of email. In *ACL-04 Workshop: Text Summarization Branches Out*, pages 43–50.

Nicolas Ducheneaut and Victoria Bellotti. 2001. E-mail as habitat: an exploration of embedded personal information management. *Interactions*, 8(5):30–38.

Dominique Estival, Tanja Gaustad, Son Bao Pham, Will Radford, and Ben Hutchinson. 2007. Author profiling for English emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 263–272, Melbourne, Australia.

Jade Goldstein and Roberta Evans Sabin. 2006. Using speech acts to categorize email and identify email genres. In *Proceedings of the 39th Hawaii International Conference on System Sciences*, page 50b.

Lewis Hassell and Margaret Christensen. 1996. Indirect speech acts and their use in three channels of communication. In *Proceedings of the First International Workshop on Communication Modeling - The Language/Action Perspective*, Tilburg, The Netherlands.

Hamid Khosravi and Yorick Wilks. 1999. Routing email automatically by purpose not topic. *Journal of Natural Language Engineering*, 5:237–250.

Bryan Klimt and Yiming Yang. 2004. Introducing the Enron corpus. In *Proceedings of the Conference on Email and Anti-Spam (CEAS)*.

Andrew Lampert, Robert Dale, and Cécile Paris. 2008a. The nature of requests and commitments in email messages. In *Proceedings of EMAIL-08: the AAAI Workshop on Enhanced Messaging*, pages 42–47, Chicago.

Andrew Lampert, Robert Dal e, and Cécile Paris. 2008b. Requests and commitments in email are more complex than you think: Eight reasons to be cautious. In *Proceedings of Australasian Language Technology Workshop (ALTA2008)*, pages 55–63, Hobart, Australia.

Andrew Lampert, Robert Dale, and Cécile Paris. 2009. Segmenting email message text into zones. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 919–928, Singapore.

Wendy E. Mackay. 1988. More than just a communication system: Diversity in the use of electronic mail. In *ACM conference on Computer-supported cooperative work*, pages 344–353, Portland, Oregon, USA.

Denise E. Murray. 1991. *Conversation for Action: The Computer Terminal As Medium of Communication*. John Benjamins Publishing Co.

Scott S L Piao, Andrew Wilson, and Tony McEnery. 2002. A multilingual corpus toolkit. In *Proceedings of 4th North American Symposium on Corpus Linguistics*, Indianapolis.

Jerry M. Sadock and Arnold Zwicky, 1985. *Language Typology and Syntactic Description. Vol.I Clause Structure*, chapter Speech act distinctions in syntax, pages 155–96. Cambridge University Press.

Simon Scerri, Myriam Mencke, Brian David, and Siegfried Handschuh. 2008. Evaluating the ontology powering smail — a conceptual framework for semantic email. In *Proceedings of the 6th LREC Conference*, Marrakech, Morocco.

John R. Searle. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

John Sinclair and Richard Malcolm Coulthard. 1975. *Towards and Analysis of Discourse - The English used by Teachers and Pupils*. Oxford University Press.

Steve Whittaker and Candace Sidner. 1996. Email overload: exploring personal information management of email. In *ACM Computer Human Interaction conference*, pages 276–283. ACM Press.

Terry Winograd and Fernando Flores. 1986. *Understanding Computers and Cognition*. Ablex Publishing Corporation, Norwood, New Jersey, USA, 1st edition. ISBN: 0-89391-050-3.

Ian Witten and Eiba Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition.