Segmenting Email Message Text into Zones

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

Robert Dale ‡
rdale@ics.mq.edu.au

Cécile Paris †
‡Centre for Language Technology
Macquarie University
North Ryde 2109, Australia
cecile.paris@csiro.au

Abstract

In the early days of email, widely-used conventions for indicating quoted reply content and email signatures made it easy to segment email messages into their functional parts. Today, the explosion of different email formats and styles, coupled with the ad hoc ways in which people vary the structure and layout of their messages, means that simple techniques for identifying quoted replies that used to yield 95% accuracy now find less than 10% of such content. In this paper, we describe Zebra, an SVM-based system for segmenting the body text of email messages into nine zone types based on graphic, orthographic and lexical cues. Zebra performs this task with an accuracy of 87.01%; when the number of zones is abstracted to two or three zone classes, this increases to 93.60% and 91.53% respectively.

1 Introduction

Email message bodies consist of different functional parts such as email signatures, quoted reply content and advertising content. We refer to these as **email zones**. Many language processing tools stand to benefit from better knowledge of this message structure, facilitating focus on relevant content in specific parts of a message. In particular, access to zone information would allow email classification, summarisation and analysis tools to separate or filter out 'noise' and focus on the content in specific zones of a message that are relevant to the application at hand. Email contact mining tools such as that developed by Culotta et al. (2004), for example, might access the email signature zone, while tools that attempt to identify tasks or action items in email (e.g., (Bellotti et al., 2003; Corston-Oliver et al., 2004; Bennett and Carbonell, 2007; Lampert et al., 2007)) might restrict themselves to the sender-authored and forwarded content. Despite previous work on this problem, there are no available tools that can reliably extract or identify the different functional zones of an email message.

While there is no agreed standard set of email zones, there are clearly different functional parts within the body text of email messages. For example, the content of an email disclaimer is functionally different from the sender-authored content and from the quoted reply content automatically included from previous messages in the thread of conversation. Of course, there are different distinctions that can be drawn between zones; in this paper we explore several different categorisations based on our proposed set of nine underlying email zones.

Although we focus on content in the body of email messages, we recognise the presence of useful information in the semi-structured headers, and indeed make use of header information such as sender and recipient names in segmenting the unstructured body text.

Segmenting email messages into zones is a challenging task. Accurate segmentation 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 their messages. When replying to a message, for example, it is often useful to include all or part of the original message that is being replied to. Different email clients indicate quoted material in different ways. By default, 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.

To address these challenges, in this paper we describe Zebra, our email zone classification system. First we describe how Zebra builds and improves on previous work in Section 2. Section 3 then presents our set of email zones, along with details of the email data we use for system training and experiments. In Section 4 we describe two approaches to zone classification, one that is line-based and one that is fragment-based. The performance of Zebra across two, three and nine email zone classification tasks is presented and analysed in Section 5.

2 Related Work

Segmenting email messages into zones requires both text segmentation and text classification. The main focus of most work on text segmentation is topic-based segmentation of news text (e.g., (Hearst, 1997; Beeferman et al., 1997)), but there have been some previous attempts at identifying functional zones in email messages.

Chen et al. (1999) looked at both linguistic and two-dimensional layout cues for extracting structured content from email signature zones in email messages. The focus of their work was on extracting information from already identified signature blocks using a combination of two-dimensional structural analysis and one-dimensional grammatical constraints; the intended application domain was as a component in a system for email text-to-speech rendering. The authors claim that their system can be modified to also identify signature blocks within email messages, but their system performs this task with a recall of only 53%. No attempt is made to identify functional zones other than email signatures.

Carvalho and Cohen's (2004) Jangada system attempted to identify email signatures within plain text email messages and to extract email signatures and reply lines. Unfortunately, the 20 Newsgroups corpus[1] they worked with contains 15-year-old Usenet messages which are much more homogeneous in their syntax than contemporary email, particularly in terms of how quoted text from previous messages is indicated. As a result, using a very simple metric (a line-initial '>' character) to identify reply lines achieves more than 95% accuracy. In contrast, this same simple metric applied to the Enron email data we annotated detects less than 10% of actual reply or forward lines.

Usenet messages are also markedly different from contemporary email when it comes to email signatures. Most Usenet clients produced messages which conformed to RFC3676 (Gellens, 2004), a standard that formalised a "long-standing convention in Usenet news ... of using two hyphens `--` as the separator line between the body and the signature of a message." Unfortunately, this convention has long since ceased to be observed in email messages. Carvalho and Cohen's email signature detection approach also benefits greatly from a simplifying assumption that signatures are found in the last 10 lines of an email message. While this holds true for their Usenet message data, it is no longer the case for contemporary email.

In attempting to use Carvalho and Cohen's system to identify signature blocks and reply lines in our own work, we identified similar shortcomings to those noted by Estival et al. (2007). In particular, Jangada did not accurately identify forwarded or reply content in email data from the Enron email corpus. We believe that the use of older Usenet-style messages to train Jangada is a significant factor in the systematic errors the system makes in failing to identify quoted reply, forwarded and signature content in messages formatted in the range of message formats and styles popularised by Microsoft Outlook. These errors are a fundamental problem with Jangada, especially since Outlook is the most common client used to compose messages in our annotated email collection drawn from the Enron corpus. More generally, we note that Outlook is the most popular email client in current use, with an estimated 350–400 million users worldwide,[2] representing anywhere up to 40% of all email users.[3]

More recently, as part of their work on profiling

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

[2] Xobni Co-founder Adam Smith and former Engineering VP Gabor Cselle have both published Outlook user statistics. See http://www.xobni.com/asmith/archives/66 and http://gaborcselle.com/blog/2008/05/xobnis-journey-to-right-product.html.

[3] http://www.campaignmonitor.com/stats/email-clients/

authors of email messages, Estival et al. (2007) classified email bodies into five email zones. Their paper does not provide results for five-zone classification, but they report accuracy of 88.16% using a CRF classifier to distinguish three zones: reply, author and signature. We use their classification scheme as the starting point for our own set of email zones.

3 Email Zones

As noted earlier, we refer to the different functional components of email messages as **email zones**. The zones we propose refine and extend the five categories — *Author Text*, *Signature*, *Advertisement* (automatically appended advertising), *Quoted Text* (extended quotations such as song lyrics or poems), and *Reply Lines* (including forwarded and reply text) — identified by Estival et al. (2007).

We consider that each line of text in the body of an email message belongs to one of nine more fine-grained email zones. We intend our nine email zones to be abstracted and adapted to suit different tasks. To illustrate, we present the zones below abstracted into three classes: sender-authored content, boilerplate content, and content quoted from other conversations. This is the zone partition we use to generate the three-zone results reported in Section 5. This categorisation is useful for problems such as finding action items in email messages: such detection tools would look in text from the sender-authored message zones for new action item information, and could also look in quoted conversation content to link new action item information (such as reported completions) to previous action item content.

Our nine email zones can also be reduced to a binary scheme to distinguish text authored by the sender from text authored by others. This distinction is useful for problems such as author attribution or profiling tasks. In this two-class case, the sender-authored zones would be *Author*, *Greeting*, *Signoff* and *Signature*, while the other-authored zones would be *Reply*, *Forward*, *Disclaimer*, *Advertising* and *Attachment*. This is the partition of zones we use in our two-zone experiments reported in Section 5.

3.1 Sender Zones

Sender zones contain text written by the current email sender. The *Greeting* and *Signoff* zones are sub-zones of the *Author* zone, usually appearing as the first and last items respectively in the *Author* zone. Thus, our proposed sender zones are:

1. **Author:** New content from the current email sender. This specifically excludes any text authored by the sender that is included from previous messages.

2. **Greeting:** Terms of address and recipient names at the beginning of a message (e.g., *Dear/Hi/Hey Noam*).

3. **Signoff:** The message closing (e.g., *Thanks/Cheers/Regards, John*).

3.2 Quoted Conversation Zones

Quoted conversation zones include both content quoted in reply to previous messages in the same conversation thread and forwarded content from other conversations.[4] Our quoted conversation zones are:

4. **Reply:** Content quoted from a previous message in the same conversation thread, including any embedded signatures, attachments, advertising, disclaimers, author content and forwarded content. Content in a reply content zone may include previously sent content authored by the current sender.

5. **Forward:** Content from an email message outside the current conversation thread that has been forwarded by the current email sender, including any embedded signatures, attachments, advertising, disclaimers, author content and reply content.

3.3 Boilerplate Zones

Boilerplate zones contain content that is reused without modification across multiple email messages. Our proposed boilerplate zones are:

6. **Signature:** Content containing contact or other information that is automatically inserted in a message. In contrast to disclaimer or advertising content, signature content is usually templated content written once by the email author, and automatically or semi-automatically included in email messages. A

[4] Although we recognise the need for the *Quoted Text* zone proposed by Estival et al. (2007), no such data occurs in our collection of annotated email messages. We therefore omit this zone from our current set.

user may also use a *Signature* in place of a *Signoff*; in such cases, we still mark the text as a *Signature*.

7. **Advertising:** Advertising material in an email message. Such material often appears at the end of a message (e.g., *Do you Yahoo!?*), but may also appear prefixed or inline with the content of the message, (e.g., in sponsored mailing lists).

8. **Disclaimer:** Legal disclaimers and privacy statements, often automatically appended.

9. **Attachment:** Automated text indicating or referring to attached documents, such as that shown in line 16 of Figure 1. Note that this zone does not apply to manually authored reference to attachments, nor to the actual content of attachments (which we do not classify).

3.4 Email Data and Annotation

The training data for our zone classifier consists of 11881 annotated lines from almost 400 email messages drawn at random from the Enron email corpus (Klimt and Yang, 2004).[5] We use the database dump of the corpus released by Andrew Fiore and Jeff Heer.[6] 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. Following Estival et al. (2007), we used only a single annotator since the task revealed itself to be relatively uncontroversial. Each line in the body text of selected messages was marked by the annotator (one of the authors) as belonging to one of the nine zones. After removing blank lines, which we do not attempt to classify, we are left with 7922 annotated lines as training data for Zebra. The frequency of each zone within this annotated dataset is shown in Table 3.

Figure 1 shows an example of an email message with each line annotated with the appropriate email zone. Two zone annotations are shown for each line (in separate columns), one using the nine fine-grained zones and the second using the abstracted three-zone scheme described in Section 3. Note, however, that not all of the nine fine-grained zones, nor all of the three abstracted zones, are actually present in this particular message.

[5]This annotated dataset is available from http://zebra.thoughtlets.org/.

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

4 Zone Segmentation and Classification

Our email zone classification system is based around an SVM classifier using features that capture graphic, orthographic and lexical information about the content of an email message.

To classify the zones in an email message, we experimented with two approaches. The first employs a two-stage approach that segments a message into zone fragments and then classifies those fragments. Our second method simply classifies lines independently, returning a classification for each non-blank line in an email message. Our hypothesis was that classifying larger text fragments would lead to better performance due to the text fragments containing more cues about the zone type.

4.1 Zone Fragment Classification

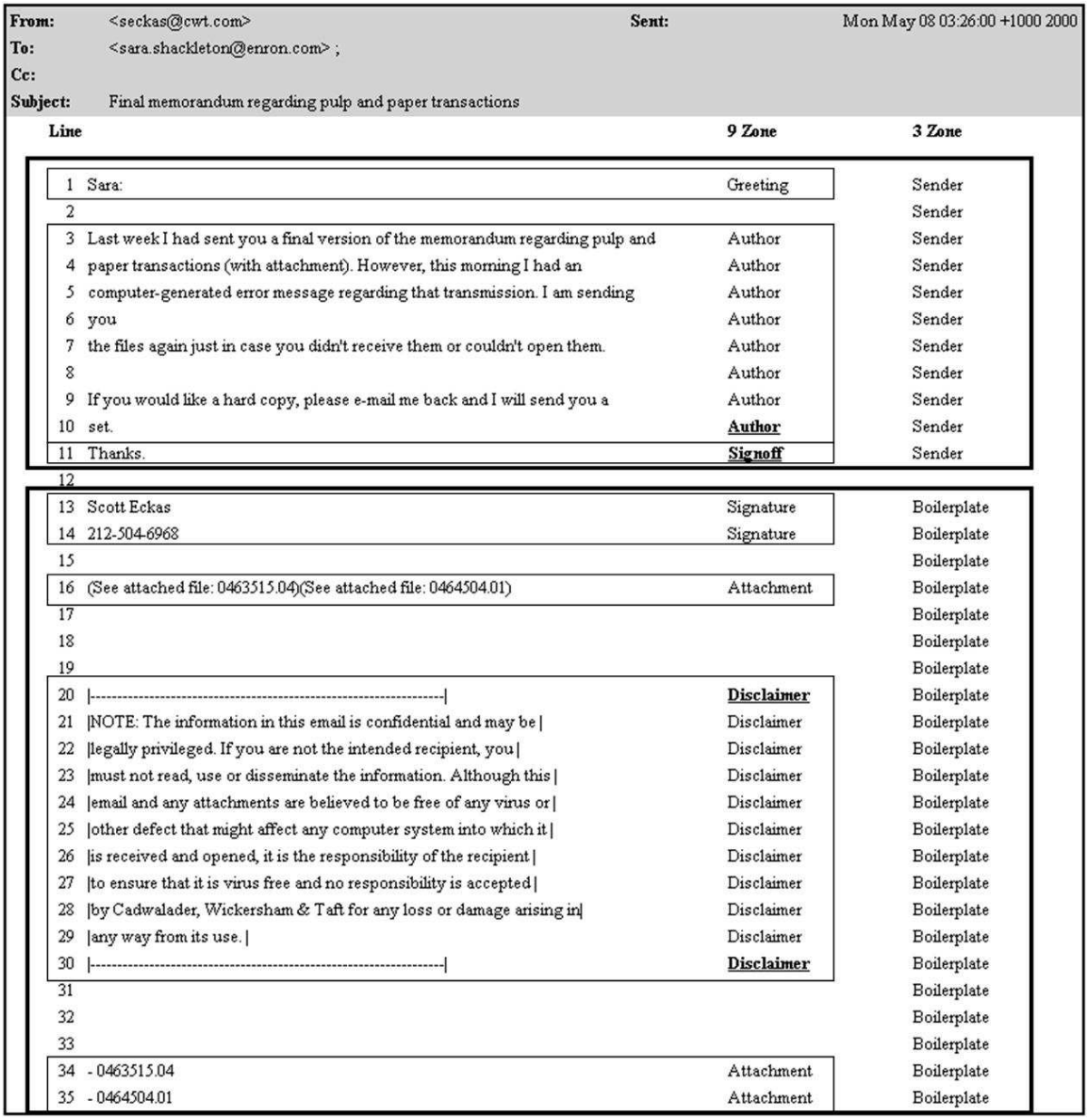
Zone fragment classification is a two-step process. First it predicts the zone boundaries using a simple heuristic, then it classifies the resulting *zone fragments*, the sets of content lines that lie between these hypothesised boundaries.

In order to determine how well we can detect zone boundaries, we first need to establish the correct zone boundaries in our collection of zone-annotated email messages.

4.1.1 Zone Boundaries

A zone boundary is defined as a continuous collection of one or more lines that separate two different email zones. Lines that separate two zones and are blank, contain only whitespace or contain only punctuation characters are called **buffer lines**.

Since classification of blank lines between zones is often ambiguous, empty or whitespace-only buffer lines are not included as content in any zone, and thus are not classified. Instead, they are treated as strictly part of the zone boundary. In Figure 1, these lines are shown without any zone annotation. Zone boundary lines that are included as content in a zone have their zone annotation styled in bold and underlined. The important point here is that zone boundaries are specific to a zone classification scheme. For nine-zone classification of the message in Figure 1, there are six zone boundaries: line 2, lines 10–11, line 12, line 15, lines 17–20, and lines 30–33. For three-zone clas-



From: <seckas@cwt.com> **Sent:** Mon May 08 03:26:00 +1000 2000
To: <sara.shackleton@enron.com> ;
Cc:
Subject: Final memorandum regarding pulp and paper transactions

Line		9 Zone	3 Zone
1	Sara:	Greeting	Sender
2			Sender
3	Last week I had sent you a final version of the memorandum regarding pulp and	Author	Sender
4	paper transactions (with attachment). However, this morning I had an	Author	Sender
5	computer-generated error message regarding that transmission. I am sending	Author	Sender
6	you	Author	Sender
7	the files again just in case you didn't receive them or couldn't open them.	Author	Sender
8		Author	Sender
9	If you would like a hard copy, please e-mail me back and I will send you a	Author	Sender
10	set.	**Author**	Sender
11	Thanks.	**Signoff**	Sender
12			
13	Scott Eckas	Signature	Boilerplate
14	212-504-6968	Signature	Boilerplate
15			Boilerplate
16	(See attached file: 0463515.04)(See attached file: 0464504.01)	Attachment	Boilerplate
17			Boilerplate
18			Boilerplate
19			Boilerplate
20	|----------|	**Disclaimer**	Boilerplate
21	|NOTE: The information in this email is confidential and may be|	Disclaimer	Boilerplate
22	|legally privileged. If you are not the intended recipient, you|	Disclaimer	Boilerplate
23	|must not read, use or disseminate the information. Although this|	Disclaimer	Boilerplate
24	|email and any attachments are believed to be free of any virus or|	Disclaimer	Boilerplate
25	|other defect that might affect any computer system into which it|	Disclaimer	Boilerplate
26	|is received and opened, it is the responsibility of the recipient|	Disclaimer	Boilerplate
27	|to ensure that it is virus free and no responsibility is accepted|	Disclaimer	Boilerplate
28	|by Cadwalader, Wickersham & Taft for any loss or damage arising in|	Disclaimer	Boilerplate
29	|any way from its use.|	Disclaimer	Boilerplate
30	|----------|	**Disclaimer**	Boilerplate
31			Boilerplate
32			Boilerplate
33			Boilerplate
34	- 0463515.04	Attachment	Boilerplate
35	- 0464504.01	Attachment	Boilerplate

Figure 1: An example email message marked with both nine- and three-zone annotations.

sification, the only zone boundary consists of line 12, separating the sender and boilerplate zones.

Based on these definitions, there are three different types of zone boundaries:

1. **Blank boundaries** contain only empty or whitespace-only buffer lines. Lines in these zone boundaries are strictly separate from the zone content. An example is Line 12 in Figure 1, for both the three- and nine-zone classification.
2. **Separator boundaries** contain only buffer lines, but must contain at least one punctuation-character buffer line that is retained as content in one or both zones. In Figure 1, an example is the zone boundary containing lines 17–20 that separates the *Attachment* and *Disclaimer* zones for nine-zone classification, since line 20 is retained as part of the *Disclaimer* zone content.
3. **Adjoining boundaries** consist of the last content line of the earlier zone and the first content line of the following zone. These boundaries occur where no buffer lines exist between the two zones. An example is the zone boundary containing lines 10 and 11 that separates the *Author* and *Signoff* zones in Figure 1 for nine-zone classification.

4.1.2 Hypothesising Zone Boundaries

To identify zone boundaries in unannotated email data, we employ a very simple heuristic approach. Specifically, we consider every line in the body of an email message that matches any of the following criteria to be a zone boundary:

1. A blank line;
2. A line containing only whitespace; or
3. A line beginning with four or more repeated punctuation characters, optionally prefixed by whitespace.

Our efforts to apply more sophisticated machine-learning techniques to identifying zone boundaries could not match the 90.15% recall achieved by this simple heuristic. The boundaries missed by the simple heuristic are all **adjoining boundaries**, where two zones are not separated by any buffer lines. An example of a boundary that is not detected by our heuristic is the zone boundary between the *Author* and *Signoff* zones in Figure 1 formed by lines 10 and 11.

Obviously, our simple boundary heuristic detects **actual boundaries** as well as **spurious boundaries** that do not actually separate different email zones. Unsurprisingly, the number of spurious boundaries is large. The precision of our simple heuristic across our annotated set of email messages is 22.5%, meaning that less than 1 in 4 hypothesised zone boundaries is an actual boundary. The underlying email zones average more than 12 lines in length, including just over 8 lines of non-blank content. Due to the number of spurious boundaries, fragments contain less than half this amount — approximately 3 lines of non-blank content on average. One of the most common types of spurious boundaries detected are the blank lines that frequently separate paragraphs within a single zone.

For three-zone classification, the set of predicted boundaries remains the same, but there are less actual boundaries to find, so recall increases to 96.3%. However, because many boundaries from the nine-zone classification are not boundaries for the three-zone classification, precision decreases to 14.7%.

4.1.3 Classifying Zone Fragments

Having segmented the email message into candidate zone fragments, we classify these fragments using the SMO implementation provided by Weka (Witten and Frank, 2005) with the features described in Section 4.3.

Although our boundary detection heuristic has better than 90% recall, the small number of actual boundaries that are not detected result in some zone fragments containing lines from more than one underlying email zone. In these cases, we consider the mode of all annotation values for lines in the fragment (i.e., the most frequent zone annotation) to be the gold-standard zone type for the fragment. This, of course, may mean that we somewhat unfairly penalise the accuracy of our automated classification when Zebra detects a zone that is indeed present in the fragment, but is not the most frequent zone.

4.2 Line Classification

Our line-based classification approach simply extracts all non-blank lines from an email message and classifies lines one-by-one, using the same features as for fragment-based classification. This approach is the same as the signature and reply line classification approach used by Carvalho and Cohen (2004).

4.3 Classification Features

We use a variety of graphic, orthographic and lexical features for classification in Zebra. The same features are applied in both the line-based and the fragment-based zone classification (to either individual lines or zone fragments). In the description of our features, we refer to both single lines and zone fragments (collections of contiguous lines) as **text fragments**.

4.3.1 Graphic Features

Our graphic features capture information about the presentation and layout of text in an email message, independent of the actual words used. This information is a crucial source of information for identifying zones. Such information includes how the text is organised and ordered, as well as the 'shape' of the text. The specific features we employ are:

- the number of words in the text fragment;
- the number of Unicode code points (i.e., characters) in the text fragment;
- the start position of the text fragment (equal to one for the first line in the message, two for the second line and increasing monotonically

through the message; we also normalise the result for message length);
- the end position of the text fragment (calculated as above and again normalised for message length);
- the average line length (in characters) within the text fragment (equal to the line length for line-based text fragments);
- the length of the text fragment (in characters) relative to the previous fragment;
- the length of the text fragment (in characters) relative to the following fragment;
- the number of blank lines preceding the text fragment; and
- the number of blank lines following the text fragment.

4.3.2 Orthographic Features

Our orthographic features capture information about the use of distinctive characters or character sequences including punctuation, capital letters and numbers. Like our graphic features, orthographic features tend to be independent of the words used in an email message. The specific orthographic features we employ include:

- whether all lines start with the same character (e.g., '>');
- whether a prior text fragment in the message contains a quoted header;
- whether a prior text fragment in the message contains repeated punctuation characters;
- whether the text fragment contains a URL;
- whether the text fragment contains an email address;
- whether the text fragment contains a sequence of four or more digits;
- the number of capitalised words in the text fragment;
- the percentage of capitalised words in the text fragment;
- the number of non-alpha-numeric characters in the text fragment;
- the percentage of non-alpha-numeric characters in the text fragment;
- the number of numeric characters in the text fragment;
- the percentage of numeric characters in the text fragment;
- whether the message subject line contains a reply syntax marker such as *Re:* ; and
- whether the message subject line contains a forward syntax marker such as *Fw:*.

4.3.3 Lexical Features

Finally, our lexical features capture information about the words used in the email text. We use unigrams to capture information about the vocabulary and word bigram features to capture short range word order information. More specifically, the lexical features we apply to each text fragment include:

- each word unigram, calculated with a minimum frequency threshold cutoff of three, represented as a separate binary feature;
- each word bigram, calculated with a minimum frequency threshold cutoff of three, represented as a separate binary feature;
- whether the text fragment contains the sender's name;
- whether a prior text fragment in the message contains the sender's name;
- whether the text fragment contains the sender's initials; and
- whether the text fragment contains a recipient's name.

Features that look for instances of sender or recipient names are less likely to be specific to a particular business or email domain. These features use regular expressions to find name occurrences, based on semi-structured information in the email message headers. First, we extract and normalise the names from the email headers to identify the relevant person's given name and surname. Our features then capture whether one or both of the given name or surname are present in the current text fragment. Features which detect user initials make use of the same name normalisation code to retrieve a canonical form of the user's name, from which their initials are derived.

5 Results and Discussion

Table 1 shows Zebra's accuracy in classifying email zones. The results are calculated using 10-fold cross-validation. Accuracy is shown for three tasks — nine-, three- and two-zone classification — using both line and zone-fragment classification. Performance is compared against a majority class baseline in each case.

Zebra's performance compares favourably with previously published results. While it is difficult to

	2 Zones		3 Zones		9 Zones	
	Zebra	Baseline	Zebra	Baseline	Zebra	Baseline
Lines	93.60%	61.14%	91.53%	58.55%	87.01%	30.94%
Fragments	92.09%	62.18%	91.37%	59.44%	86.45%	30.36%

Table 1: Classification accuracy compared against a majority baseline

	2 Zones		3 Zones		9 Zones	
	Zebra	Baseline	Zebra	Baseline	Zebra	Baseline
Lines	90.62%	61.14%	86.56%	58.55%	81.05%	30.94%
Fragments	91.14%	62.18%	89.44%	59.44%	82.55%	30.36%

Table 2: Classification accuracy, without word n-gram features, compared against a majority baseline

directly compare, since not all systems are freely available and they are not trained or tested over the same data, our three-zone classification (identifying sender, boilerplate and quoted reply content) is very similar to the three-zone task for which (Estival et al., 2007) report 88.16% accuracy for their system and 64.22% accuracy using Carvalho and Cohen's Jangada system. Zebra outperforms both, achieving 91.53% accuracy using a line-based approach. In the two-zone task, where we attempt to identify sender-authored lines, Zebra achieves 93.60% accuracy and an F-measure of 0.918, exceeding the 0.907 F-measure reported for Estival et al.'s system tuned for exactly this task.

Interestingly, the line-based approach provides slightly better performance than the fragment-based approach for each of the two-zone, three-zone and nine-zone classification tasks. As noted earlier, our original hypothesis was that zone fragments would contain more information about the sequence and text shape of the original message, and that this would lead to better performance for fragment-based classification.

When we restrict our feature set to those that look only at the text of the line or zone fragment, the fragment-based approach does perform better than the line-based one. Using only word unigram features, for example, our fragment classifier achieves 78.7% accuracy. Using the same features, the line-based classifier achieves only 57.5% accuracy. When we add further features that capture sequence and shape information from outside the text fragment being classified (e.g., the length of a text segment compared to the text segment before and after, and whether a segment occurs after another segment containing repeated punctuation or the sender's name), the line-based approach achieves a greater increase in accuracy than the fragment-based approach. This presumably is because individual lines intrinsically have less information about the message context, and so benefit more from the information added by the new features.

We also experimented with removing all word unigram and bigram features to explore the classifier's portability across different domains. This removed all vocabulary and word order information from our feature set. In doing so, our feature set was reduced to less than thirty features, consisting of mostly graphic and orthographic information. The few remaining lexical features captured only the presence of sender and recipient names, which are independent of any particular email domain. As expected, performance did drop, but not dramatically. Table 2 shows that average performance without n-grams (across two-, three- and nine-zone tasks) for line-based classification drops by 4.67%. In contrast, fragment-based classification accuracy drops by less than half this amount — an average of 2.26%. This suggests that, as we originally hypothesised, there are additional non-lexical cues in zone fragments that give information about the zone type. This makes the zone fragment approach potentially more portable for use across email data from different enterprise domains.

Of course, classification accuracy gives only a limited picture of Zebra's performance. Table 4 shows precision and recall results for each zone in the nine-zone line-based classification task. Per-

	Total	Author	Signature	Disclaim	Advert	Greet	Signoff	Reply	Fwd	Attach
Author	2415	2197	56	9	4	14	31	43	53	8
Signature	383	93	203	4	0	0	20	28	31	4
Disclaim	97	30	4	52	0	0	0	2	9	0
Advert	83	47	1	1	20	0	0	7	7	0
Greet	85	8	0	0	0	74	2	0	1	0
Signoff	195	30	5	0	0	0	147	11	2	0
Reply	2451	49	10	3	2	1	10	2222	154	0
Fwd	2187	72	13	7	8	1	3	125	1958	0
Attach	26	4	0	0	0	0	0	1	1	20

Table 3: Confusion Matrix for 9 Zone Line Classification

formance clearly varies significantly across the different zones. For *Author*, *Greeting*, *Reply* and *Forward* zones, performance is good, with F-measure > 0.8. This is encouraging, given that many email tools, such as action-item detection and email summarisation would benefit from an ability to separate author content from reply content and forwarded content. The *Advertising*, *Signature* and *Disclaimer* zones show the poorest performance, particularly in terms of Recall. The *Advertising* and *Disclaimer* zones are almost certainly hindered by a lack of training data; they are two of the smallest zones in terms of number of lines of training data. The relatively poor *Signature* class performance is more interesting. Given the potential confusion between *Signoff* content and *Signature*s that function as *Signoff*s, one might expect confusion between *Signoff* and *Signature* zones, but Table 3 shows this is not the case. Instead, there is significant confusion between *Signature* and *Author* content, with almost 25% of *Signature* lines misclassified as *Author* lines. When word n-grams are removed from the feature set, the number of these misclassifications increases to almost 50%. These results reinforce our observation that the task of email signature extraction is much more difficult that it was in the days of Usenet messages.

Zone	Precision	Recall	F-Measure
Author	0.868	0.910	0.889
Signature	0.695	0.530	0.601
Disclaimer	0.684	0.536	0.601
Advertising	0.588	0.241	0.342
Greeting	0.822	0.871	0.846
Signoff	0.690	0.754	0.721
Reply	0.911	0.907	0.909
Forward	0.884	0.895	0.889
Attachment	0.625	0.769	0.690

Table 4: Precision and recall for nine-zone line classification

6 Conclusion

Identifying functional zones in email messages is a challenging task, due in large part to the diversity in syntax used by different email software, and the dynamic manner in which people employ different styles in authoring email messages. Zebra, our system for segmenting and classifying email message text into functional zones, achieves performance that exceeds comparable systems, and that is at a level to be practically useful to email researchers and system builders. In addition to releasing our annotated email dataset, the Zebra system will also be available for others to use[7].

Because we employ a non-sequential learning algorithm, we encode sequence information into the feature set. In future work, we plan to determine the effectiveness of using a sequential learning algorithm like Conditional Random Fields (CRF). We note, however, that Carvalho and Cohen (2004) demonstrate that using a non-sequential learning algorithm with sequential features, as we do, has the potential to meet or exceed the performance of sequential learning algorithms.

Acknowledgments

The authors are grateful to the anonymous reviewers for their insightful comments and suggestions.

[7] See http://zebra.thoughtlets.org for access to the annotated data and Zebra system

References

Douglas Beeferman, Adam Berger, and John Lafferty. 1997. Text segmentation using exponential models. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, pages 35–46, Providence, RI.

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, USA, April 5-10.

Paul N Bennett and Jaime G Carbonell. 2007. Combining probability-based rankers for action-item detection. In *Proceedings of NAACL HLT 2007*, pages 324–331, Rochester, NY, April.

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.

Hao Chen, Jianying Hu, and Richard W Sproat. 1999. Integrating geometrical and linguistic analysis for email signature block parsing. *ACM Transactions on Information Systems*, 17(4):343–366, October. ISSN: 1046-8188.

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, July.

Aron Culotta, Ron Bekkerman, and Andrew McCallum. 2004. Extracting social networks and contact information from email and the web. In *Proceedings of the Conference on Email and Anti-Spam (CEAS)*.

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, Sept 19-21.

R. Gellens. 2004. RFC3676: The text/plain format and delsp parameters, February.

Marti A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

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

Andrew Lampert, Cécile Paris, and Robert Dale. 2007. Can requests-for-action and commitments-to-act be reliably identified in email messages? In *Proceedings of the 12th Australasian Document Computing Symposium*, pages 48–55, Melbourne, Australia, December 10.

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