Automatic Text Summarization of Emails on Mobile Platforms

Charles Revett

School of Computing and Communications

Lancaster University

charlierevett@gmail.com

http://lancs.ac.uk/ug/revett

Abstract

This paper presents a novel approach for email management on a mobile device, using automatic text summarization (ATS) techniques tailored to the unique set of requirements presented by summarising emails and the limitations of working on mobile platforms. It sets out to prove that the addition of email summaries to the conventional model of a mobile email client (app) can improve user productivity.

1 Introduction

Email has continued to be one of the world's most ubiquitous applications, used within all aspects of life. The recent rise of smart-phones has seen the tool increasingly be accessed via mobile (Zheng & Ni, 2006). As people demand the ability to access information anytime, anywhere and many users suffering from *email overload*, the need for summarization techniques to be integrated into mobile applications is crucial in reducing the time users spend reading, replying and organizing their emails (Whittaker & Sidner, 1996).

The approach proposed in this paper focuses on crafting a mobile application that adaptively creates high quality, succinct summaries at speed that cater for the tight constraints of a mobile phone. Indicative extracts will quickly highlight to the user what action needs to be taken for a given email. It follows previous work from other genres of summarization, as well as implementing email specific features.

1.1 Email Characteristics

Previous text summarization techniques targeting general written text such as news articles share many features with email summarization. However email is a unique linguistic genre with its own distinct characteristics.

Email is an asynchronous method of communication with multiple collaborators interacting at once. Similar to that of a face-to-face discussion, however unlike spoken dialogue, text is the single channel of communication available for the user to convey their intent. Unlike standard text summarization, the average body of text is much smaller and the language used is frequently written in an informal, grammatically incorrect chat format.

These characteristics pose a set of requirements that are not well suited to general text summarization techniques, thus this paper pulls the basic methodology from previous work and then adds novel features that are specific to handling email.

1.2 Mobile Limitations

Implementing a stand-alone text summarization system on a mobile phone both limits what functionality can be included and determines features that are necessary to overcome limitations. The three largest human-computer interaction (HCI) challenges faced when developing on a mobile device are; battery life, screen size and computing power (Goldberg, 2013). These key components heavily affect the nature of the application and the functionality that can be implemented. To keep the requirements of the application to a minimum, only proven text summarization methodologies have been included in the system.

The reduced screen resolution of a mobile phone compared to more conventional devices such as laptops or monitors means that not only the quality of the summary should be considered but also the length. Sentence simplification functionality is therefore considered an important part of the app.

The remainder of this paper is structured as follows. Section 2 provides an understanding of the background to automatic text summarization, and an overview of past email summarization research. Section 3 presents how the proof-of-concept application was implemented, including the natural language processing (NLP) techniques featured. Section 4 offers both qualitative and quantitative based evaluations of the application, taking into consideration a user study. Section 5 discusses the overall success of the paper in relation to the results gathered from the two evaluations. Section 6 concludes the paper by focusing on how well the primary objectives have been addressed, the limitations of the application and implications of the results gathered.

2 Background

This section gives an overview of the previous research that this paper uses as its foundation, around proven automatic extraction methods and more recently work in the area of email summarization.

Generating extractive summaries of emails can be seen as an extension of automatic extraction (Edmundson, 1969). Edmundson described the '*Four Basic Methods'* for an automatic extraction system as; cue words, key words, title words and sentence location. Each method assigned a numerical weight to every sentence in the document based on certain machine recognizable characteristics or clues. Experimental cycles of his work showed that the combined score of three of the methods (Cue-Title-Location) resulted in the highest mean coselection score. The *key words* method in isolation resulted in the lowest mean coselection score. This data analysis was taken into account when determining which features to include in the application.

Edmundson's *cue word* methodology attempts to house within an application a pre-generated dictionary which conveys a specific topic (e.g.

importance). It described the foundation for the majority of extractive systems, however it is worth expanding on his work especially when working in a specialised domain such as email. Pollock and Zamora developed a system for the Chemical Abstracts Service that relied on a cue word dictionary specific to the field of chemistry to formulate the numeric weighting of sentences (Pollock & Zamora, 1975). As emails are unlike formal written text and have a more informal, chat format, the approach of Pollock and Zamora can be applied to create an email specific cue word dictionary. An application can include multiple subdictionaries, with the typical approach being to include three: positively relevant (*bonus*) words; negatively relevant (*stigma*) words; and irrelevant (*null*) words (Inderjeet & Maybury, 1999).

Within Edmundson's initial research, the *location* of sentences within the document held the best individual results for indicating the most important sentences. Edmundson built upon earlier work (Baxendale, 1958) which found that important sentences were most likely to be located at the beginning and end of a paragraph. A more recent study of over 10,000 news articles showed that the title contained the highest density of keywords, followed by the first sentence of the second paragraph, third paragraph, and so on (Lin & Hovy, 1997). The overall theme of the results did not hold with every newspaper. This is an important point to keep in mind when considering research to build upon within this paper, as the language used in emails is dissimilar to the majority of previous corpora used for research purposes.

The *title words* feature described by Edmundson utilises terms located in titles and headings within the document. It allowed for a theme to quickly be built about the document. For this paper, the *subject* of an email will be taken as the source of *title* keywords. The weighting that this feature has on the overall sentence score has to be closely analysed, as the overall theme of a document may only be partially covered within its title.

The *key words* feature assumed that the more frequent terms within a document were the most salient. Edmundson's experimental results however showed that in isolation this methodology resulted in the lowest coselection score of the four features. More recently, a more complex approach was considered where the frequency of terms within two corpora are compared. This formed a quick and accurate method for key word discovery through highlighting the differentiating key words between two corpora (Rayson & Garside, 2000).

Research expanding on Edmundson's findings has formed four fully featured methodologies that lend themselves well to a wide variety of experiments. This paper will also see the addition of other features to cover the shortcomings of these four features and the difficulties of developing on a mobile device.

As discussed previously, a major drawback of developing an application for a mobile device is the limited screen size. Grefenstette developed a program for the blind which compacts a page of text and reads it back to the user (Grefenstette, 1998). This was to try and mimic the action of a sighted reader skim reading a page. The system eliminated words not in a stated criteria, thus reducing the word count. This method of *text reduction* was used as a foundation to solve the issue of reduced screen space within the application. Further work added new sources of knowledge to decide which phrases to remove from a sentence; syntactic knowledge and statistical knowledge pre-generated from a training corpus (Jing, 2000). There is a large scope for sentence reduction, however this needs to be balanced with the processing limitations of using a mobile device.

Functionality that reduces the amount of text that the summarization engine is required to compute, will in turn lower the processing power required. Thus a *sentence length cut-off* feature could be considered. It was found that short sentences tend to not be included in summaries (Kupiec, et al., 1995). Hence, a threshold (e.g. 5 words) could be implemented to act as a guard, thus reducing the load on the engine.

2.1 Email Summarization

Previous research conducted into the field of email summarization focuses on extracting the best techniques from document summarization systems due to low number of email-specific systems.

The FASil email summariser (Dalli, et al., 2004) concentrated on *named entity recognition* (NER). The summariser put forward an innovative Internet-based approach which allowed it to recognise proper names, locations, dates, titles and anaphors. This approach could not be implemented on a mobile device due to the restrictions and implications of 2G/3G network usage (Perrucci, et al., 2009).

A multi-document approach could be considered where a *fragment quotation graph* is constructed (Carenini, et al., 2007). Each node within the directed graph would represent an email and an edge between two nodes represents that they are within the same conversation. This method offers a more detailed representation of the structure of an email conversation. Lam et al. also exploited the thread structure of an email conversation to increase the knowledge pool available to highlight the important sentences within an email (Lam, 2002).

Similar to this paper, Muresan et al. describe work using machine learning approaches to identify rules for salient noun phrase extraction within individual emails (Muresan, et al., 2001).

2.2 Mobile Application using ATS

Few operational mobile applications have implemented automatic text summarization techniques. This is mainly due to the limited processing resources mobile phones could provide until very recently. This section will highlight two applications in particular which have utilised ATS techniques to solve the problem of information overload – *Summly* and *Textal.*

Summly was released for the iPhone in December 2011, allowing users to read a summarized version of a given news article. The application saw immediate success within the Apple App Store and quickly received venture capital funding (Summly, 2013).

Textal is an iPhone app which generates a wordcloud for the user based on a given book, document, website or twitter stream. Thus allowing the user to explore the statistics and relationships between words in the given text. The project differs from *Summly* as it is research orientated, however it also secured funding from two research councils (EPSRC and NCRM) (Textal, 2013).

The success of both of these applications shows that users respond positively to ATS techniques being included within mobile applications, thus highlighting ATS as a viable solution for tackling information overload.

An important difference to stress about the summaries produced by *Summly* is that their aim is to replace the news article. This paper instead focuses on creating indicative summaries which provide the user with the information necessary to understand the email and perform an action.

2.3 Problem Statement

The primary objective of this paper is to investigate if the addition of text summarization to a mobile email client increases user productivity. This paper will also act as a proof-of-concept, for whether the introduction of NLP techniques to the mobile platform is beneficial.

Quantifying if a user is more *productive* is difficult to pinpoint and would require a longitudinal study. Thus this paper presents both a qualitative and a quantitative based evaluation of results gathered.

The qualitative evaluation will give an insight into the unstructured data collected from a user study; whilst the quantitative evaluation will demonstrate a statistical investigation of the summaries formed by the application. Both will address the following hypothesis:

> *The addition of email summaries to the conventional model of a mobile email client improves user productivity.*

3 Proposed Solution

This section presents how the proof-of-concept application was designed, with an explanation of the summarization techniques at its core, as well as the reasoning for why they were included.

Figure 1 depicts how the original text received from the mail transfer agent flows through the system to result in a three sentence summary being formed. The first seven steps of Figure 1 remove any part of the original text which the sentence scoring algorithm does not require. These parts fall into three main categories:

1. Words present in the *stopList* dictionary.
2. Punctuation.
3. Sentences less than 5 words in length (*Short-Sentence Guard* feature).

This is a crucial part of the system as it drastically reduces the iterations of the sentence scoring algorithm, thus helping to limit the amount of resources the application uses.

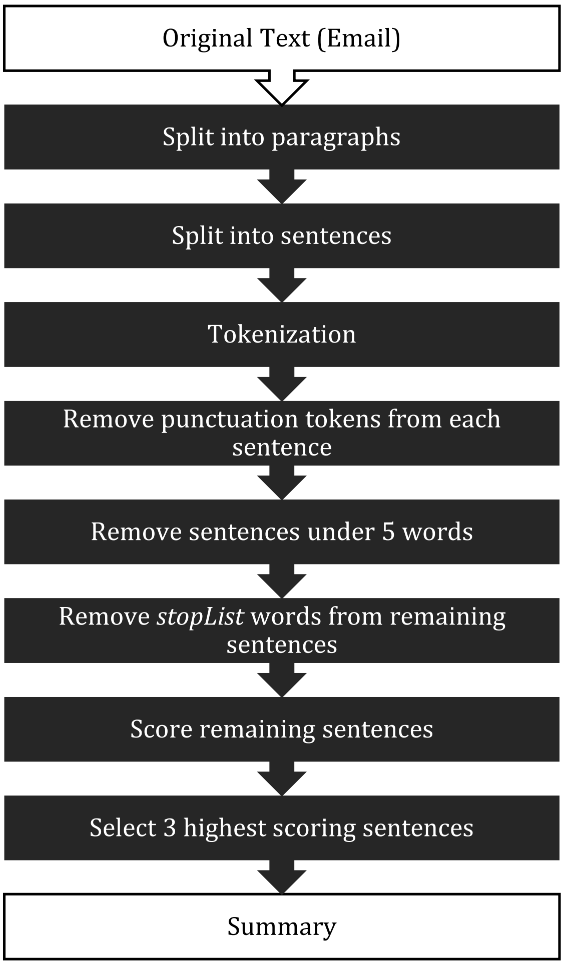

Original Text (Email)
Split into paragraphs
Split into sentences
Tokenization
Remove punctuation tokens from each sentence
Remove sentences under 5 words
Remove *stopList* words from remaining sentences
Score remaining sentences
Select 3 highest scoring sentences
Summary

Figure 1: Flow Chart depicting the sequence of events leading to a summary being generated.

The *stopList* dictionary allows for 'stop words' to be filtered out of the text prior to the system running the sentence scoring algorithm. This is a major performance optimisation, as by removing the insignificant words of the sentence, the system is not wasting resources computing inconsequential phrases. The dictionary (101 words) was initially formed from a list of the most common English words (compiled by *Textfixer)*. Throughout development words were added and removed to calibrate the dictionary to the precise specifications of this paper.

LingPipe was included within the application as an external library. It is a java toolkit used to process text is number of ways using computational linguistics. Two pieces of functionality were included from the toolkit; *sentence splitter* and *tokenizer*. The *sentence splitter* took each paragraph in turn and added each of its sentences to an array. This was also the point where the location of each sentence was stored, through the use of a wrapper class. The *tokenizer* took each sentence in turn and added each token to an array.

The original text from the body of the email was parsed using a bespoke HTML parser written especially for the application. It splits the email into paragraphs (or *blocks*) depending on the HTML tags. It then removes any HTML tags from the *blocks* to ensure no tags are displayed to the user.

3.1 Summarization Techniques

There are few functioning email summarization systems available for comparison, thus features chosen for the sentence scoring algorithm were document summarization features which had yielded successful results and could function within the constraints of the application.

Four features were chosen to be included within the application. Each section below describes how they were implemented and the reasoning for their selection.

Three of the features were selected from Edmundson's original research as he found that the combination of these features (Cue-Title-Location) resulted in the highest mean coselection score. The *keyword* feature was excluded from the application due to its disappointing experimental results and the major impact it would have on the resources used. These findings and their suitability for a mobile device were the driving force for selecting them.

Title Keywords

This feature is very similar to that of Edmundson's, where words located in titles or subheadings are used to build an overall theme of what the document is about.

The application takes the *subject* of the email as the sole title of the entire text. The *subject* is split into tokens, where each token roughly corresponds to a *word*. Any tokens which are 'stop words' or punctuation are removed. This leaves the system with an array of words which it deems to be important. The more title words that a sentence contains, the more important it is deemed to be.

The strong correlation between the subject of an email and its content meant that this was a necessary feature to include. It is also simple to implement and efficient, both advantageous characteristics when developing on a mobile device.

Sentence Location

The principles of this feature were based on the work of Lin & Hovy, who proposed that the most important emails were located in the first sentence of the second paragraph, third paragraph, and so on.

The application records the paragraph number and sentence number within the paragraph for each sentence when parsing the original text. This allows the system to later validate the location of a sentence, deeming it more important if it is located in the first sentence of a paragraph.

This feature was included in the system because Edmundson recorded it as having the best individual results for indicating the most important sentences and it again was an efficient elegant feature which suited the constraints of a mobile device.

Cue Keywords

This feature is heavily based on the work of Pollock & Zamora, as the system utilises pre-generated email specific *bonus* and *stigma* dictionaries to highlight important sentences.

The application implements this feature in much the same way as the *title keywords* feature. The system splits each sentence into tokens and discards any null tokens (‘stop words’ and punctuation). The more *bonus* words that a sentence contains, the more important it is deemed to be and the more *stigma* words that a sentence contains, the less important it is deemed to be.

This is by far the most resource intensive feature, due to the amount of iterations. However it allows for pre-generated dictionaries to be controlled, permitting the application to be fine-tuned to each user’s individual style of communication or technical field.

The *bonus* and *stigma* dictionaries were primarily generated through the use of *Wmatrix*, a web-based corpus processing environment (Rayson, 2008). The system allowed for a list to be generated of words relevant to particular semantic. The *bonus* dictionary contains 53 words, composed from the ‘importance’ semantic. The *stigma* dictionary contains 11 words, composed from the ‘unimportance’ semantic. Both of the dictionaries had words manually added and removed throughout development.

Short-Sentence Guard

This feature utilises the work done by Kupiec, who determined that short sentences should not be considered for a summary. Thus sentences have to be over 4 words long to be scored. This has a large impact on the amount of iterations the other features are required to perform, and therefore significantly reduces the amount of resources used.

3.2 Sentence Scoring Algorithm

The algorithm below assigns a numeric value to the given sentence *i* based on the three Edmundson concepts (Cue-Title-Location) detailed in Section 3.1.

$$\boldsymbol{S_i = w_1(B_i) - w_2(S_i) + w_3(T_i) + w_4(L_i)}$$

Where:

$\boldsymbol{S_i}$ Score of sentence *i*.

B_i Score of sentence *i* based on the number of *bonus* words it contains.

S_i Score of sentence *i* based on the number of *stigma* words it contains.

T_i Score of sentence *i* based on the number of title keywords it contains.

L_i Score of sentence *i* based on its location.

The *cue keywords* feature was split into two separate parts. This allowed for separate weightings for the *bonus* and *stigma* dictionaries.

The weighting for each feature (w_1 – w_4) were formed using trial & error throughout development. Due to the time scale available for this paper, it was not possible for multiple user studies to test a variety of different weightings.

The following weights were used:

$w_1 = 2$ Cue keywords (*Bonus*) weighting.

$w_2 = 1$ Cue Keywords (*Stigma*) weighting.

$w_3 = 3$ Title Keywords weighting.

$w_4 = 1$ Sentence location weighting.

3.3 Worked Sentence Scoring Example

This section will highlight each step of the sentence scoring algorithm individually.

Figure 3 shows the list of remaining tokens from the original sentence (displayed in Figure 2) after all 'stop words' and punctuation has been removed. The sentence scoring algorithm will only consider these tokens. Figure 4 is the list of tokens extracted from the subject of the email to form the *title keywords*.

```
We will schedule an opening meeting
with you the week June 4, 2001 to
discuss the Audit scope.
```

Figure 2: Original sentence

```
[will][schedule][opening][meeting]
[week][june][4][2001][discuss]
[audit][scope]
```

Figure 3: Remaining tokens from original sentence

```
[audit][notification]
```

Figure 4: Title Keywords

The algorithm will compare each token against the *bonus* dictionary, *stigma* dictionary and any *title keywords*.

The `[meeting]` token would be flagged as a *bonus* keyword. The `[audit]` token would be flagged as a *title* keyword. The sentence was located in the first sentence of the second paragraph, thus the algorithm would validate the sentence to be deemed as important based on its location.

The final score would be as follows:

$$\boldsymbol{S_i = w_1(B_i) - w_2(S_i) + w_3(T_i) + w_4(L_i)}$$

$$\boldsymbol{= 2(1) - 1(0) + 3(1) + 1(1)}$$

$$\boldsymbol{= 6}$$

4 Evaluation

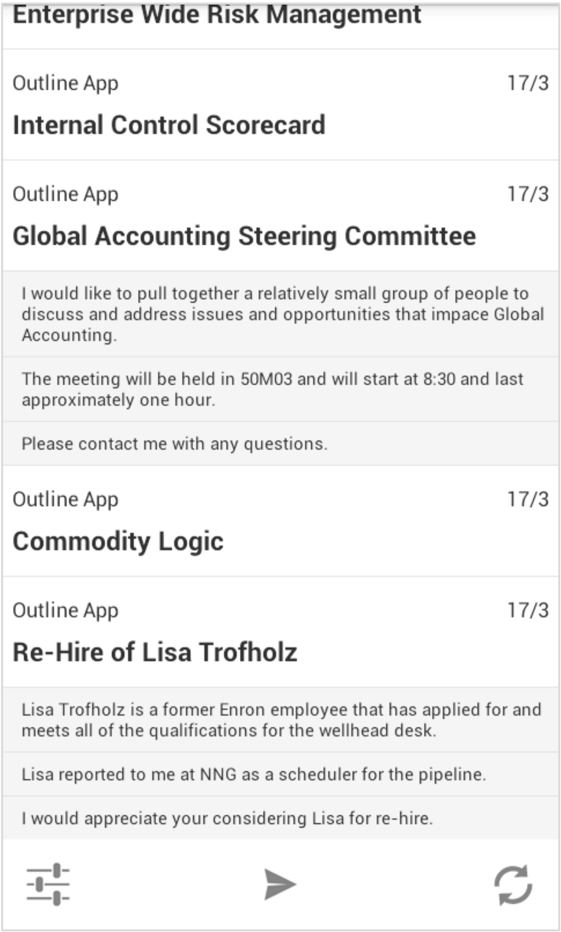
This section provides a qualitative discussion of the results gathered from a user study and a quantitative analysis of the sentence scoring algorithm against a baseline. Two evaluations were conducted to provide a more in-depth analysis to explore to what extent the application matched the *hypothesis*. The extended evaluation also helped to address the difficulty in measuring productivity.

4.1 User Study

The primary issue which the study sets out to explore was whether the summaries generated by the application highlighted enough information about the email for the user to feel confident in carrying out an action, thus improving efficiency. An action was described to the user as a common email interaction such as;

- Replying
- Forwarding
- Deleting
- Archiving
- Highlighting as important (starring)

The user study was conducted as a stand-alone user feedback survey. It consisted of a total of 10 participants. They were each contacted electronically or verbally, and screened to ensure a varied group of participants were chosen (age, job etc.). Each participant was given an outline of the application's aim and was allowed to freely explore the application, as shown in Figure 5. Each email used within the study was selected from the *Enron* email corpus, a recognised source of emails in Linguistics.


Enterprise Wide Risk Management

Outline App 17/3
Internal Control Scorecard

Outline App 17/3
Global Accounting Steering Committee

I would like to pull together a relatively small group of people to discuss and address issues and opportunities that impace Global Accounting.

The meeting will be held in 50M03 and will start at 8:30 and last approximately one hour.

Please contact me with any questions.

Outline App 17/3
Commodity Logic

Outline App 17/3
Re-Hire of Lisa Trofholz

Lisa Trofholz is a former Enron employee that has applied for and meets all of the qualifications for the wellhead desk.

Lisa reported to me at NNG as a scheduler for the pipeline.

I would appreciate your considering Lisa for re-hire.

Figure 5: Application User Interface

Participant observation (Ethnography) was employed so as to collect any further data. The researcher imposed a minimal presence when observing the user throughout the study, as to not impose their own bias on the data. Their aim was to take note of and answer any questions the user may have during the study. Observing the participants the researcher saw that many struggled to grasp the concept when reading the user study blurb, with multiple participants asking for clarification. This may be due to the new functionality included in the application being new to them or the blurb not being clear enough.

The survey consisted of 2 questions aimed at calculating the amount of important emails the user receives per day and 6 questions targeting the primary issue of the study. The 6 questions aimed at analysing the quality of each summary and the overall aim of the study required the original emails as reference, therefore these were also provided to the participants.

4.2 User Study Results

The first 2 questions contained in the survey set out to calculate the amount of important emails that the user receives on a daily basis. As shown in Figure 6, the proportion of important emails a user receives each day is heavily outweighed by the amount of emails they deem to be insignificant. The study showed that on average the users considered 18% of the emails received daily to be important. This analysis clearly shows that there is a need for a more efficient manner of email management, in order to reduce time reading irrelevant emails.

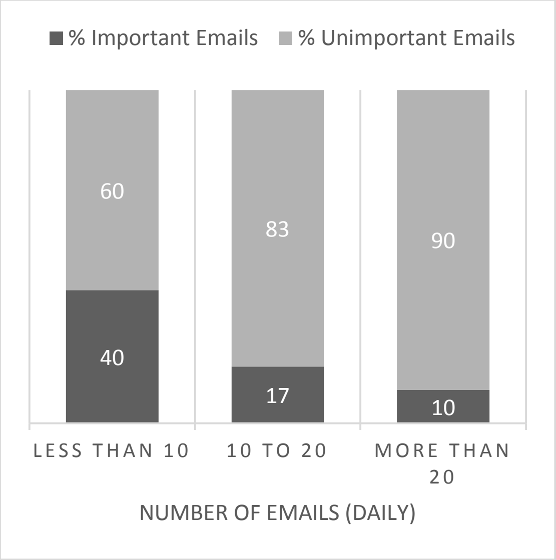


Figure 6: Stacked bar chart displaying the proportion of important emails a user receives each day.

Further analysis of Figure 6, shows that the less emails that a user receives, the larger the proportion of emails that they deem to be important. This highlights that a mobile email client utilizing automatic text summarization techniques would be best suited to users receiving over 20 emails per day.

The 6 questions aimed at analysing the application itself comprised 5 statements which the user was asked to state their opinion of ('Strongly Agree' to 'Strongly Disagree') and a comment box. The users were asked to highlight any particularly good or bad summaries in the comment box, as well as any observations about the application. Each section below highlights the responses gathered to each of the 5 statements, as well as a possible explanation behind negative responses.

Statement 1: *"The summaries provided an informative reflection of what the emails contained."*

All ten of the participants responded positively to this statement, with 40% of users answering that they 'Strongly Agree' with the statement. This reinforces that document summarization techniques function well when dealing with the informal style of communication found in emails.

Statement 2: *"After reading the summaries, I did not feel the need to open the emails for further reading."*

All ten of the users responded positively to this statement, with 50% of users answering that they 'Strongly Agree' with the statement. The distribution of answers demonstrates that a summary is an adequate replacement for the original email. Thus highlighting that automatic text summarization techniques fit well within both the constraints of a mobile device and the characteristics of email.

Statement 3: *"Each summary provided enough information for me to carry out an action (e.g. delete, archive etc)."*

80% of users responded positively to this statement, with 40% of participants answering that they 'Strongly Agree' with the statement. 20% of users highlighted that they neither agreed nor disagreed with the statement. A possible reason for this lack of clarity may be that short *question* sentences often received very low scores. Therefore resulting in these sentences being left out of the summary. A question will often result in an action, thus the system could easily exclude the overall action of the email from the summary. This could have been solved by creating a feature within the sentence scoring algorithm which deems a sentence more important if it includes a question mark or particular *action* phrases such as 'Can you please'.

Statement 4: *"Each summary pin-pointed the 3 most important sentences from within each email."*

This statement received a positive response from 60% of the participants, with 20% stating that they 'Strongly Agree' with it. 40% of the users responded that they neither agreed nor disagreed with the statement. Similar to *statement 3*, this could be a result of sentences containing the overall action of the email being excluded from the summary.

Statement 5: *"Summarising the emails improved my efficiency when managing the inbox."*

90% of participants responded positively to this statement, with 50% declaring that they 'Strongly Agree' with the statement. This was reiterated within the responses gathered from the comment box, where participants took time to highlight their preference for this method of email management. These responses further reinforce the motivation behind this project and the need for ATS techniques to be included within mobile email clients.

Responses gathered from the final question asking for the participant's overall opinion of the application and its techniques highlighted that 50% found some sentences to be too long. The sentence scoring algorithm does not take into account the overall length of a sentence and will extract a copy of the entire text to include in the summary. The algorithm also does not normalise a sentence's score based on its length. As a longer sentence has a higher probability to include *cue* and *title* keywords and thus receive a higher score, it is expected for summaries to favour longer sentences.

To address this issue, a feature similar to the work conducted by Grefenstette could be implemented which removes a certain criteria (e.g. pronouns and adjectives) of words to reduce the overall length of the sentence. Alternatively, the score a sentence receives could be proportional to its length, thus forcing the system to no longer favour longer sentences. This feature would also help the system to better recognise shorter *question* sentences (highlighted in the analysis of Statement 3).

Both these features would address the issue of summary length, however they could result in a level of clarity being lost.

One particcpant highlighted that they would find it hard to completely trust the system's judgement of what it deems to be an important sentence. A different participant also highlighted that deeming a sentence more important based on who sent it would be a crucial feature. This functionality (known as *VIP*) has been successfully implemented within the standard iOS email client shipped with all Apple mobile devices.

Both users demonstrated that they would require some a specific sort of user manipulation of the sentence scoring algorithm. Multiple users all found that a way to customize how the system scores a sentence would have made the application better. The time-scale of this paper did not allow for an extended user study, where participants explore the application over a number of weeks during their daily routine. This kind of user study would allow the types of customization necessary to be highlighted. An elongated user evaluation of the study would also have allowed for a greater insight into whether the user's productivity was being increased as they would be using the application within their daily routine.

Many of the points brought up by the participants of the user study would raise a deeper concern if this application were to be released as a completed product. However the application itself was produced purely as a proof-of-concept, and the overall trend of the user study suggests that the application is increasing the user's level of productivity and therefore to some extent matching the *hypothesis*.

4.3 Quantitative Evaluation

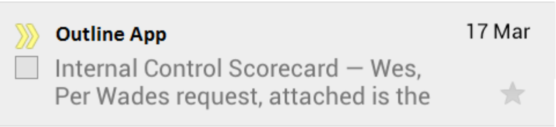
This section of the paper will compare and contrast the statistical accuracy of the summaries generated by the application against that of a baseline. This part of the evaluation was conducted as a classification task, where the *precision* and *recall* were calculated for both the proof-of-concept application and the baseline.

10 emails were manually classified, showing what the researcher deemed to be the most important sentences within the email. These sentences were set as the *evaluation metric*, and were used to calculate the *precision* and *recall*. This methodology was used because the number of unimportant sentences heavily outweigh the amount of important sentences.

The baseline being used within this part of the evaluation is the Gmail application, considered to be the conventional model for a mobile email client. Within its user interface, Gmail displays the following details for each email:

- Sender
- Date received
- Subject
- First *n* characters of email

The amount of characters displayed of the email (n) is dependant on the length of subject of the email. The user interface allows for up to 60 characters to be displayed to the user. Therefore if the character length of the subject is less than 60, then the remaining characters will be filled with a segment of the text from the body of the email (as shown in Figure 7).


Outline App 17 Mar

□ Internal Control Scorecard — Wes,
Per Wades request, attached is the

Figure 7: Screenshot showing how the details of an email is displayed to the user within the Gmail application.

In this classification task, we will be using the following 2-by-2 contingency table:

	Manually Classified by Researcher	
Automatically Classified by System	*tp*	*fp*
	fn	*tn*

Where:

tp The sentence was classified as important by both the researcher and the system [*true positive*].

fp The system classified the sentence as important however the researcher classified it as unimportant [*false positive*].

tn The sentence was classified as unimportant by both the researcher and the system [*true negative*].

fn The system classified the sentence as unimportant however the researcher classified it as important [*false negative*].

These 4 terms were used to calculate the *precision* and *recall* of the two systems. Below details the definition of both, as well as how to calculate them.

Precision

Of the sentences deemed as important by the system, what percentage of them were also classified as important by the researcher.

$$\boldsymbol{P = \frac{tp}{tp + fp}}$$

Recall

Of the sentences classified as important by the researcher, what percentage of them the system also deemed important.

$$\boldsymbol{R = \frac{tp}{tp + fn}}$$

4.3 Quantitative Results

This section of the evaluation analyses the *precision* and *recall* results gathered for both the application developed as part of this paper and the baseline.

The baseline (Gmail application) had an average *recall* value of 25.0% and an average *precision* value of 70.0%. As the first sentence is only able to be displayed to the user, the precision of the baseline application is very hit-or-miss. The experimental results from Lin & Hovy study of sentence location showed that the first sentence of a paragraph is more likely to important. Thus by only displaying the first sentence of the email, the baseline results in it having a high precision value.

Comparing this to the proof-of-concept application, it received an average recall value of 76.6% and an average precision value of 83.3%.

These results clearly highlight that, on average, the proof-of-concept application is both out-performing the baseline, as well as highlighting that it is deeming the same sentences important as the researcher, manually classifying each sentence, the majority of the time.

A possible reason for the *precision* and *recall* values not being higher is that the summary is constrained to a fixed sentence count. The researcher, manually classifying the sentences of each of the 10 emails, found that 5 of the emails required a different sized summary. This was also highlighted by two participants from the user study. They added that the number of sentences forming the summary should be proportional to the length of the original email.

Normally when constructing an NLP system, there needs to be a trade off between *precision* and *recall*, where the system is aiming for good results in one category. It is therefore very rare that in this novel approuch both of the values are high. Further development of the application could allow the user to scale their preference for each category, thus addressing the problem of customization discussed in Section 4.2.

The quantitative results gathered for the application both reinforce the fact that it improves user productivity and matches the matches the *hypothesis* to some extent, but also performs better than the leading mobile email client (Gmail).

5 Discussion

This paper, although limited by the amount of data gathered due to the time constraints available, still manages to display sufficient data to reinforce the motivation behind this project, showing that previous automatic text summarization techniques can be tweaked to successfully deal with the summarization of emails on a mobile device. The results gathered from both the user study and quantitative evaluation clearly show that production of high quality succinct email summaries is possible on a mobile device, and provides a successful alternative to the conventional mobile email client.

The summarization features included within the application are very similar to that of a subset of the features Edmundson proposed in his original research. The paper has manipulated his work through the use of more recent studies conducted and testing throughout development to create a system that would be familiar to anyone within the field of automatic text summarization, however unique enough for it to succeed in this unexplored area.

The limited number of participants of the user study still allow for the results to be interpreted as an analysis of anyone dealing with email management, as information overload is a universal problem. This is highlighted clearly within the user study, with participants responding that on average they deem only 18% of emails received daily to be important. The quantitative evaluation clearly highlighted the accuracy of the application and that even with the small amount of features implemented, it was generating summaries very close to those a human would.

Both evaluations clearly highlighted how the application could be extended in the future, as well as demonstrating that this is a viable model for a mobile email client that users would like to use during their daily routine. The number of possible extensions requested by participants of the user study shows that there is a large scope for an application of this kind, and also the enthusiasm from users for a product similar to the proof-of-concept application to be developed in the future. Although the user study could not be conducted over a number of weeks, it still perfectly highlighted the problems with the system and how it would be tweaked if development were to continue.

Three main issues were found with the system; sentence length, exclusion of questions and lack of user customization. Each of these problems had little effect on the quality of the summaries produced, as demonstrated in the quantitative evaluation, however they did effect the overall experience of using the application. This would have been verified much more clearly if the users had to use the application on a daily basis as part of an extended study.

6 Conclusion

This paper set out to provide evidence for the need of automatic text summarization techniques within mobile email clients. It did this through the development of a proof-of-concept application. This application set out to increase a user's level of productivity when managing their emails, thus reducing the time spent on their mobile device and potentially increasing their work efficiency in other parts of life.

In section 2 of this report, the background to automatic text summarization was discussed, as well an overview of related non-mobile email summarization systems. Section 3 described how the proof-of-concept application functions and the core summarization techniques that were included in the application, as well as the reasoning for why each of them was selected. Section 4 presented both a qualitative and a quantitative based evaluation of the application. This highlighted how a group of 10 participants responded to the application and how it compared to the conventional mobile email client model (Gmail). Section 5 then went on to discuss to what extent the *hypothesis* was matched, using the data gathered from the two evaluations.

This paper suggests that the universal problem of information overload seen on a daily basis by email users can be reduced by the implementation of automatic text summarization techniques to mobile email clients. Although the functionality included within the proof-of-concept application was primitive, the data gathered proved that the *hypothesis* was met. Possible future development to the application could pose it as a viable tool for users. The data collected in this paper depicts how further research or an attempt at developing a public end product could see this becoming a reality.

References

Baxendale, P. B., 1958. Machine-made index for technical literature—an experiment. *IBM Journal of Research and Development ,* 2(4), pp. 354-361.

Carenini, G., Ng, R. T. & Zhou, X., 2007. Summarizing email conversations with clue words. *Proceedings of the 16th international conference on World Wide Web. ACM.*

Dalli, A., Xia, Y. & Wilks, Y., 2004. Fasil email summarisation system. *Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics.*

Edmundson, H. P., 1969. New methods in automatic extracting. *Journal of the ACM (JACM),* 16(2), pp. 264-285.

Goldberg, K., 2013. *10 Biggest Challenges of Human Computer Interaction for Mobile.* [Online] Available at: http://ux.walkme.com/mobile-human-computer-interaction/

Grefenstette, G., 1998. Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind.. *Working notes of the AAAI Spring Symposium on Intelligent Text summarization,* pp. 111-118.

Inderjeet, M. & Maybury, M. T., 1999. Advances in automatic text summarization. In: s.l.:MIT press, p. 30.

Jing, H., 2000. Sentence reduction for automatic text summarization. *Proceedings of the sixth conference on Applied natural language processing. Association for Computational Linguistics.*

Kupiec, J., Pedersen, J. & Chen, F., 1995. A trainable document summarizer. *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval.*

Lam, D., 2002. *Exploiting e-mail structure to improve summarization,* s.l.: IBM Research.

Lin, C.-Y. & Hovy, E., 1997. Identifying topics by position. *Proceedings of the fifth conference on Applied natural language processing (Association for Computational Linguistics).*

Muresan, S., Tzoukermann, E. & L. Klavans, J., 2001. Combining linguistic and machine learning techniques for email summarization. *Proceedings of the 2001 workshop on Computational Natural Language Learning-Volume 7. Association for Computational Linguistics.*

Perrucci, G. P., Fitzek, F. H. & Sasso, G., 2009. On the impact of 2G and 3G network usage for mobile phones' battery life.. *Wireless Conference, 2009. EW 2009. European. IEEE.*

Pollock, J. J. & Zamora, A., 1975. Automatic abstracting research at chemical abstracts service. *Journal of Chemical Information and Computer Sciences,* 15(4), pp. 226-232.

Rayson, P., 2008. Wmatrix: a web-based corpus processing environment.

Rayson, P. & Garside, R., 2000. Comparing corpora using frequency profiling. *Proceedings of the workshop on Comparing Corpora. Association for Computational Linguistics.*

Summly, 2013. *Summly - About.* [Online]
Available at: http://summly.com/about.html
[Accessed 04 03 2013].

Textal, 2013. *Textal - About.* [Online]
Available at: http://www.textal.org/about
[Accessed 16 03 2013].

Whittaker, S. & Sidner, C., 1996. Email overload: exploring personal information management of email. *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground,* pp. 276-283.

Zheng, P. & Ni, L. M., 2006. Spotlight: the rise of the smart phone. *Distributed Systems Online, IEEE,* 7(3).