

Text Summarization Tool Evaluation: A Feasibility Study for Generating Meeting Summaries

CPSC503 Final Report

Michael Ji
Department of Computer Science
University of Calgary

Abstract

Text summarization (or automatic summarization) is the creation of a shortened version of a text by a computer program and work on it dates back as far as 40 years. Text summarization has been applied with high rates of success to a wide range of areas in the academic and commercial sectors; however, not as much work has been focused in the meeting domains. A feasibility study of incorporating a text summarization tool to generate meeting summaries is examined by the means of a product evaluation. The evaluation process is carried out in 4 distinct phases: (1) preparation, (2) criteria establishment, (3) characterization, and (4) testing. The experiment conducted in the testing phase compare the three identified and characterized text summarization products, with respect to performance and acceptability. The test data (meeting transcript) is derived from a specific software engineering meeting, known as SCRUM. As a performance measurement, each product is examined by the degree to which the important sentences generated by the software matches the manually identified important sentences. Acceptability is measured by a survey where meeting participants grade the summaries by the means of answering questionnaires. The three products to be evaluated are: (1) Open Text Summarizer, (2) Pertinent Summarizer, and (3) TextAnalyst.

1 Introduction

Text summarization (or automatic summarization) is the creation of a shortened version of a text by a computer program. The product of this procedure still contains the most important points of the original text and is generally referred to as an abstract or a summary. Broadly, one distinguishes two approaches to text summarization: extraction and abstraction. Extraction techniques merely copy information deemed to be most important by the system to the summary, while abstraction involves paraphrasing sections of the source document. In general, abstraction can produce summaries that are more condensed than extraction, but these programs are considered much harder to develop. Both techniques exploit the use of natural language processing and/or statistical methods for generating summaries. And, the classical approaches to text summarization proposed by Luhn et al.[1,3,5] have established the basis for the discipline of text summarization techniques.

The applicability of text summarization is increasingly being exploited in the commercial sector, in areas of telecommunications, data mining, information retrieval, and in word processing with high probability rates of success [2]. In addition to its wide range of applicability in the commercial sector, emerging areas of text summarization include, multimedia and multi-document summarization; however, there has been less work performed in meeting summarization [4]. Therefore, as for our initial basis for the Alan project – robotic partner for agile software engineering team - our goal is to extend this applicability to the meeting domains to produce high-quality meeting summaries. To accomplish our task in hand requires a text summarization tool. But, rather than developing our own tool, a feasibility study was instigated to determine the success of making use of third party software. This in turn required a product evaluation to be carried out.

The goal of this report is to capture the product evaluation process in 4 distinct phases: (1) preparation, (2) criteria establishment, (3) characterization, and (4) testing. First and foremost, the preparation phase consists of requirement analysis and product research that identify three feasible products (text summarization tools). In the criteria establishment phase, evaluation criteria are established for the two sub-criteria (characteristic and testing). While the characterization phase comprises of the data collection for the criteria defined. Followed by the evaluation experiment (or testing) performed on the established testing criteria, as the final phase of the evaluation process. Furthermore, the discussion section discloses the results of the experiment and any follow-up work to be carried out. Finally, the important points of the report are re-emphasized in the concluding section.

2 Summarization Approaches

Classical approaches to text summarization date back nearly 40 years. These approaches all exploited the surface-level technique (such as identification of keywords, word frequency, etc) and are classic in the sense that it serves as a fundamental basis for both modern practical applications as well as motivating subsequent research[2]; hence, the following subsections briefly examines some of these approaches.

2.1 Term Frequency Method

Luhn's paper "The Automatic Creation of Literature Abstracts"[1] from 1958 describes a statistical approach to text summarization. To determine which sentences of an article may best serve in making up an abstract summary, a measurement known as the "significance factor" is used to compare and grade sentences. Luhn proposed that the frequency of word occurrence in an article provides a useful measurement of word significance. While as the relative position of those significant words within a sentence provides a useful measurement for determining the significance of the sentence as a whole. Therefore, the significance factor of a sentence was assigned based on these two measurements. With each sentence being associated to a significant factor allowed the sentences to be ranked, and one or several of the highest-ranking sentences were then selected to makeup the abstract summary.

2.2 Cue-Title-Location Method

Following Luhn's publication, Edmundson [3], compares the term frequency approach with other features, including cue phrases, title and heading words, and sentence location. Edmundson developed programs that allowed sentences to be assigned a weight value based on each of the four methods. Evaluation of the four methods against manually created extracts was performed using a corpus-based methodology. After his evaluation, Edmundson came to the conclusion that the three additional methods (cue phrases, title, and location) dominated term frequency measurements in producing abstract text summaries. In addition, he was able to infer that the combination of features (cue-title-location) produced significantly better results, with location being the best individual feature.

2.3 Tailoring Method

Based on previous methods, Pollock and Zamora [5] provides further insight into text summarization and extends the previous approaches to apply it to a particular subject area to acquire better results. Their system is focused on producing abstract summaries of chemistry articles, with the original text being compressed to 20%. The system relies heavily on the use of cue phrases specific to chemistry subdomains to differentiate between positive and negative terms, which are tested against the sentences to be included in the summary. Also, the use of term frequencies is exploited to modulate the effects of cue phrases. However, the most distinguishing feature that differentiates the Pollock and Zamora system from the previous methods is the utilization of elimination operations aimed to compact and normalize individual sentences.

3 Evaluation Process: Methodology

The product evaluation process consists of 4 distinct phases: (1) preparation, (2) criteria establishment, (3) characterization, and (4) testing. Each of the phases is carried out separately in the specified order to identify, characterize, and assess the state-of-the-art text summarization tools currently available. From here on end, the words "product", "software product", and "text summarization tool" will be used interchangeably, depending on the context to which the word applies.

3.1 Preparation

Initial research was carried out to identify state-of-the-art text summarization tools between two major classes of available software: open-source and commercial. General-purpose search engines (like Google, Alta Vista, etc), product reviews and polls were the main sources for identifying products satisfying our high-level requirements. In addition to the software classes, another key factor taken into consideration was the summarization approach. Therefore, the identified products display a range of software classes, in addition to the complexity of the summarization approach being used (i.e. simple – complex). Complexity is taken into consideration to determine whether the intricacy of the employed technique relates to the quality of the resultant summary.

Three text summarization tools were identified after conducting the initial research. First and foremost, “Open Text Summarizer” (OTS) as the name suggests is open-source and is both a library and a command line tool. Word processors such as AbiWord and Kword make use of this tool for summarizing text. Both second and third tools are commercial based software that varies considerably in regard to their summarization approaches and product capabilities. The “Pertinence Summarizer” (PS) is considered rather straightforward in comparison to the “TextAnalyst” (TA) when considering the summarization approaches and product capabilities. Detail information regarding each of the products aforementioned is examined in the Characterization phase (section 3.3) and experimental results are presented in the Testing phase (section 3.4) of the evaluation process.

3.2 Criteria Establishment

Criteria for product evaluation were established on the basis of characteristic and testing criteria. Characteristic criteria consist of attributes that characterize the product with respect to the features and functionalities. On the other hand, testing criteria defines the experimental measurements used for the assessment of the product.

3.2.1 Characteristic Criteria

This subsection defines the attributes that are used for characterizing the products. By characterizing products by their key features and functionalities one can easily tabulate and compare one product against another by simple means of assessment (visually or quantitatively).

Current Version

This indicates the current version number of the software in production (i.e. v1.3) or signifies the different types of software version that are available (i.e. sever edition vs. personal edition).

Company / Educational Institution

This is the name of the company or educational institution that produced the product.

Supported Document Format

What document formats (HTML, PDF, DOC, etc) are supported by the product.

Language Support

What languages (English, French, Chinese, etc) are supported by the product.

Summarization Approach

Types of technique (word frequency, lexicon, dictionary, etc) being employed by the text summarization engine to generate its summaries.

Summary Features

Supported features of the product that allows the customization of summaries being generated.

Platform

Types of platforms (Window, Linux, Unix) supported by the software product.

Hardware Requirements

List of hardware characteristics required to use the software product.

Software Requirements

List of required software to run the software product.

Development Support

Does the product support software development through API and/or SDK.

Documentation

What kind (web, soft / hard copy, code, etc) of documentation is provided together with the product.

Product Cost

Cost associated with purchasing the software product.

Licence Cost

This indicates the license cost associated with the installation of the software product.

First Release

Number of years on the market, obtained by current year minus the first release year of the product.

3.2.2 Testing Criteria

This subsection defines the testing criteria that are used for the experimental evaluation to provide an assessment between the three software products. Specific measurements for each of the criteria are defined below.

Performance

As a performance measurement, each product is examined by the degree to which the important sentences extracted (or identified) by the software matches the manually identified important sentences. It is a pure extraction in the sense that the text summarization tools do not perform additional *compaction techniques*¹ on top of the extracted sentences. For example, the important information contained within a sentence (or a group of words) might be represented by the identification of concept words, rather than the whole sentence itself that may contain uninformative words (such as *determiners*² - “the”, “a”, “this”, “these”). Therefore without compaction, sentence extractions performed by the text summarization tools correspond to the original sentences from the document.

By examining the document as a set of sentences, sentence extraction can be viewed as a classification problem with the task of classifying each sentence into one of two categories: either salient or non-salient. Therefore, evaluation of the automatic sentence extraction tool (text summarization tool) can be achieved by the degree to which its classification corresponds to the human-generated classification. The summary of the classification of salient and non-salient sentences is depicted in a Classification Matrix, as shown in Table 1. The variable A represents the number of times that the human-identified salient sentence matches a machine-identified salient sentence. The variable D represents the number of times that the human and the machine agree that a sentence is not salient (or non-salient). The variable B and C represent the number of times that the human and the machine disagree on the classification of sentence saliency.

¹ Compaction techniques – are methods to encapsulate salient information in its minimal form.

² Determiner – are words which quantify or identify nouns.

	Classified as salient sentence by the human	Classified as non-salient sentence by the human
Classified as salient sentence by the machine	A	B
Classified as non-salient by the machine	C	D

Table 1: Sentence Classification Matrix

The variables (*A*, *B*, *C*, and *D*) can be further mapped into mathematical formulas to generate specific measurements to evaluate the performance of the automatic sentence extraction tools. The two measurements in consideration are *Precision* and *Recall* that can be calculated as follows:

$$\text{Precision} = A / (A + B)$$

$$\text{Recall} = A / (A+C)$$

Precision is the proportion of extracted sentences that are actually relevant (salient), i.e. text summarization tool extracts 20 salient sentences (classified by machine), but only 7 out of the 20 sentences are actually salient when matched with the human classification. *Recall* is the proportion of relevant (salient) sentences extracted, i.e. text summarization tool correctly identifies 7 out of 14 total salient sentences (classified by human). Both, measurements (precision and recall) provide a quantitative means to compare the software products in the evaluation experiment.

Acceptability

Acceptability is measured by a survey to which meeting participants grade the summaries by the means of answering questionnaires. The questionnaire is answered by selecting one of the appropriate weighted choices: Unsatisfactory = 1 point, Somewhat Satisfactory = 2 point, Satisfactory = 3 point, Above Satisfaction = 4 point, and Exceeds Satisfaction = 5 point. The overall grade (total sum of weighted choices attained from the questionnaires) will in turn imply the quality of the summary generated by the software product. Conducting a survey is informative as it removes the subjectivity of the experimenter and allows multiple judgments to be taken into consideration. Respectively, this provides a more accurate and realistic assessment on the experimental evaluation. An example use of the survey questionnaires can be seen in Table 2.

Summary A					
Questions:	Weight Choices (select 1 appropriate choice)				
	Unsatisfactory = 1	Somewhat Satisfactory = 2	Satisfactory = 3	Above Satisfaction = 4	Exceeds Satisfaction = 5
<i>Specific</i>					
To what degree of satisfaction does the resultant summary capture the important information that you conveyed during the meeting?		*			
To what degree of satisfaction is your important information that was captured accurately represented?		*			
To what degree of satisfaction can the resultant summary be considered as an effective meeting minute?		*			
<i>General</i>					
Readability (i.e. dangling references, premature sentence termination, redundancy)				*	
Comprehension (i.e. ability to grasp / interpret the meaning)		*			
Content (i.e. inadequate, misleading, unnecessary information)			*		
Utility / Effectiveness (i.e. how will it affects the completion of other tasks or how helpful is the summary)		*			
Overall Impression (i.e. overall quality of the summary)			*		
Total for each choice	0	10	6	4	0
OVERALL GRADE (Total Score)	20				
Additional Comments (if needed)					
Summary produced does not represent the important information from the meeting in an effective manner. The information within the summary needs to be formatted in such away to represent the information distinctively and clearly.					

Table 2: Survey Questionnaire

3.3 Product Characterization

A detailed examination was conducted on each of the products according to the characterization criteria established in the criteria establishment phase (section 3.2). The necessary information was obtained either by direct contact (sales representative), reading white papers, product specifications, and/or by product use. The findings resulted in the Software Criteria Matrix presented in Table 3. Due to limited space only part of the matrix has been included, but the full matrix can be found within the accompanying *Appendix: Figure 1* attached to this report. This matrix compares the three software products.

	Commercial	Commercial	Open Source
	Pertinence Summarizer	TextAnalyst	Open Text Summarizer (Libots)
Version	Personal Server Edition	TextAnalyst 2.3: software for automated semantic analysis TextAnalyst COM (SDK): toolkit of text analysis algorithms	0.4.2
Company / Educational Institution	Pertinence Mining http://www.pertinence.net	Megaputer http://www.megaputer.com	Open Source http://libots.sourceforge.net/
Supported Document Format	HTML, PDF, DOC, RTF, TXT and PPT, XLS according to the platforms used	TXT	TXT, HTML
Language Support	Supports ten languages (West European and Asian languages, Arabic and Russian)	English	English, German, Spanish, Russian, Hebrew, Esperanto and other languages
Summarization Approach	Lexicon, Thesaurus	Semantic network analysis - set of most imp. concepts (word and word comb.) and relationship b/w them - method that uses Artificial Neural Networks to extract semantic information from the text Statistical analysis Dictionary	Statistical analysis based of word frequency

Table 3: Software Criteria Matrix

3.4 Product Testing

The following experiment is devised to evaluate the software products on the basis of performance and acceptability. Therefore, an appropriate test data, experimental setup, and Test results are detailed below.

3.4.1 Test Data

The test data is derived from a specific Software Engineering (SE) meeting at the University of Calgary (UofC), known as SCRUM. The purpose of the SCRUM meeting is to gather the SE team members to share status. At each meeting, each participant answers the following 3 questions:

- 1) Tasks they have worked on / completed since the last meeting.
- 2) What blocks if any they found in performing their tasks
- 3) What they will be working on till the next meeting

The SCRUM master, usually the supervisor, leads the meeting, which runs for a short period of time (~15 min). The reasoning behind the selected meeting was due to its short duration and simple structure, with the mindset of increasing the likelihood for the text summarization tools to generate effective meeting summaries (i.e. extract salient information from the meeting transcript).

The test data was manually transcribed by hand using the pre-recorded audio data from two SCRUM meetings (October 25, 2005 and November 1, 2005). To easily annotate the speakers associated with the text, speaker identification tags were manually inserted. However, for the purpose of the experiment the speaker tags need to be removed as to acquire accurate results – inclusion of speaker tags preceding the dialog skews the results of the keyword frequency even though they are not actually keywords in a spoken dialog. Part of the resultant transcript for the October 25, 2005 SCRUM meeting is shown in Figure 1 (refer to *Appendix A: Figure 2 and Figure 3* for the full transcripts).

```
Frank: Then let's get started. What did you do since the last time?
Rob: Um. When was the last meeting? Wednesday? So I got I got some of
the initial GDF structure setup. Um. I'm still working on that. So you
know try refreshing. Going through making sure all these stuffs are
setup properly. And a...I imported some of the video. Or actually I
brought all the video imported but I haven't started analyzing yet.
Frank: Um.
Rob: That's about all that's happened.
Frank: And that's to the next to Thursday?
Rob: Um. Thursday I hope the some of the GEF some more of the GEF stuff
done. Um. Specifically I want to get the other part we did and one on
carbon screen.
Frank: Ok. Anything important?
Charles: I have changed the e-mail address and other things into
numbers.
Beth: Also the declarations.
Frank: Ah! yeah. Ok. Enum. So that's done. So what's the plan to the
next day?
Charles: This time I give you constraint or sequel connection.
Beth: That's the fixture to EJB. Like remote connection. The task of
the task.
```

**Figure 1: October 25, 2005
Meeting Transcript**

In addition to the test data, manual identification of salient sentences was achieved for the test data inconsideration (October 25, 2005 and November 1, 2005) by highlighting salient sentences on the transcript. The resultant part of the transcript with highlighted salient sentences for the October 25, 2005 SCRUM meeting is shown in Figure 2 (refer to *Appendix A: Figure 4 and Figure 5* for the full transcripts with salient sentence identification). This manual identification of salient sentences by a human is compared against the machine classification to obtain the appropriate performance measurements (precision and recall). It is important to note that the identification of salient sentences is subjective and the performance measurements will vary depending on the person assigned to complete the task. Nevertheless, for the experiment in consideration this is less

problematic because of the small amount of test data in consideration that include clearly identifiable salient sentences.

Frank: Then let's get started. What did you do since the last time?
Rob: Um. When was the last meeting? Wednesday? So I got I got some of the initial GDF structure setup. Um. I'm still working on that. So you know try refreshing. Going through making sure all these stuffs are setup properly. And a...I imported some of the video. Or actually I brought all the video imported but I haven't started analyzing yet.
Frank: Um.
Rob: That's about all that's happened.
Frank: And that's to the next to Thursday?
Rob: Um. Thursday I hope the some of the GEF some more of the GEF stuff done. Um. Specifically I want to get the other part we did and one on carbon screen.
Frank: Ok. Anything important?
Charles: I have changed the e-mail address and other things into numbers.
Beth: Also the declarations.
Frank: Ah! yeah. Ok. Enum. So that's done. So what's the plan to the next day?
Charles: This time I give you constraint or sequel connection.
Beth: That's the fixture to EJB. Like remote connection. The task of the task.

**Figure 2: October 25, 2005
Meeting Transcript with Salient Sentence Identification**

3.4.2 Test Setup

Measuring Performance

The evaluation experiment for measuring performance is conducted in the following manner.

Baseline

First of all, baseline performance measurements are obtained for each of the text summarization tool by generating summaries at 50% (summary length) using the two test data. Baseline performance is obtained by utilizing the tool in its *base-configuration*³ and only modifying the summary length feature. This is an important measurement as it provides a starting point to which the products can individually be assessed with subsequent experiments.

Customization – Excessive Keywords

Upon completion of the baseline, performance measurements are then obtained for the summarization tool at 50% (summary length) with *customization features*⁴ using excessive keywords that specifically aid the text summarization tool to correctly identify salient sentences. These keywords contain domain relevant words, indicator

³ Base-configuration – utilizing the tool without exploiting any of its extendible features to improve the overall summary being generated.

⁴ Customization features – are extendible features (i.e. preferences and settings, parameters input, etc) provided by the software tool to alter system performance.

words, and other words that signify the importance of sentences. Domain relevant words include words that are specific to the domain in consideration (i.e. software engineering and/or the specific projects). Indicator words include words that aid in the identification of important information. For example, salient information in the SCRUM meeting correspond to the answers to the 3 questions being asked; hence, indicator words such as started, completed, working, plans, etc are good identifiers which signal the trailing of important information. Other words which signify saliency are degree words such as specifically, furthermore, importantly, significantly, etc. The set of excessive keywords are derived for each test data by manually analyzing the transcript and used to obtain the performance measurements. This measurement is important as to deduce the performance of the text summarization tool when extensive aiding is supplied by human intervention.

Customization - Generalized Keywords

Finally, performance measurements are then obtained for the summarization tool at 50% (summary length) with *customization features*⁴ using generalized keywords. Generalized keywords only constitute the set of domain relevant words that are identified from the test data in consideration. It is generalized in the sense that domain relevant words are not specific to the test data in consideration, whereas indicator and other words that signify saliency may only apply to the particular test data to which it is derived from. Furthermore, it is unacceptable to derive a set of excessive keywords for each SCRUM meeting that needs to be summarized; hence, having a single set of generalized keywords that can evolve (change) and become applicable to all SCRUM meetings would be the ultimate approach for tackling the text summarization problem. Therefore, this measurement provides the initial bases to infer the performance of the text summarization tools when taking a generalized approach.

Measuring Acceptability

The evaluation experiment for measuring acceptability is conducted in the following manner. A survey is performed using generated summaries for the latest test data (November 24, 2005) in the following two cases: 1) using the baseline configuration and 2) using generalized keywords. The resultant grades from the survey are used to infer the overall qualities of the summaries generated by the text summarization tool. The Open Text Summarizer will not be included as part of the survey, as it cannot support customization capabilities; hence, only the Pertinence Summarizer and the TextAnalyst will be considered. In addition, text summarization tools using specific keywords will also not be considered in the survey, as this kind of explicit approach to text summarization is restricted to the test data in consideration.

3.4.3 Test Results

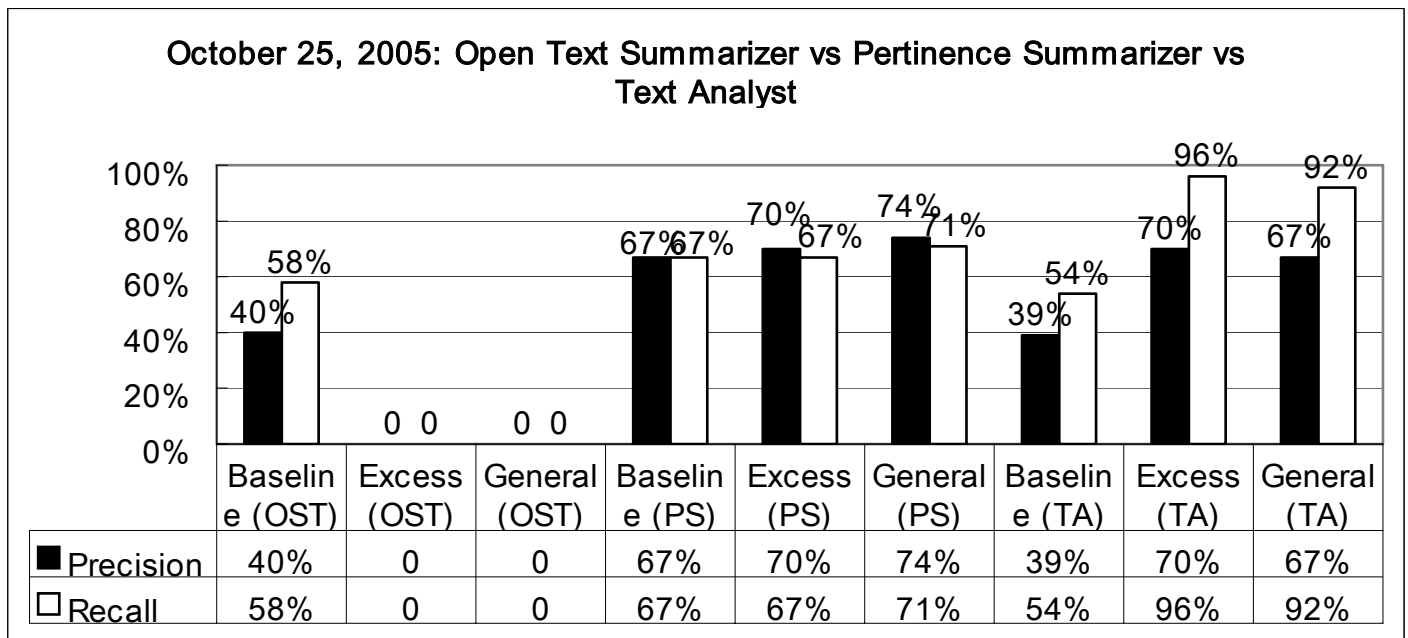
Performance

Table 4 presents the text summarization tool performance results (PR = precision and RC = recall) for the baseline, excess keywords, and generalized keywords summarization techniques using the test data in consideration (October 25, 2005 and November 1, 2005). As well, Figure 3 and 4 presents the performance results in a graphical manner for each test data as to illustrate the comparison between the text summarization tools.

Open Text Summarizer (OTS)			Pertinence Summarizer (PS)			Text Analyst (TA)		
Baseline	Excess	General	Baseline	Excess	General	Baseline	Excess	General
PR=40% RC=58%	N/A	N/A	PR=67% RC=67%	PR=70% RC=67%	PR=74% RC=71%	PR=39% RC=54%	PR=70% RC=96%	PR=67% RC=92%
October 25, 2005			October 25, 2005			October 25, 2005		
PR=38% RC=55%	N/A	N/A	PR=59% RC=69%	PR=62% RC=69%	PR=60% RC=69%	PR=42% RC=60%	PR=49% RC=69%	PR=44% RC=62%
November 1, 2005			November 1, 2005			November 1, 2005		

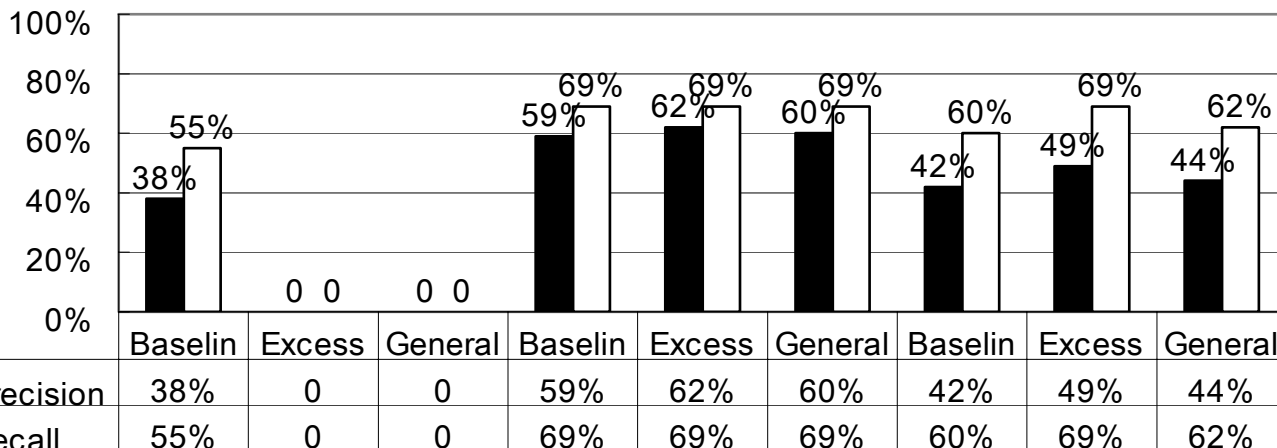
*PR = Precision, RC = Recall

Table 4: Text Summarization Tool Performance Results



**Figure 3: October 25, 2005
Text Summarization Tool Performance Comparison**

November 01, 2005: Open Text Summarizer vs Pertinence Summarizer vs TextAnalyst



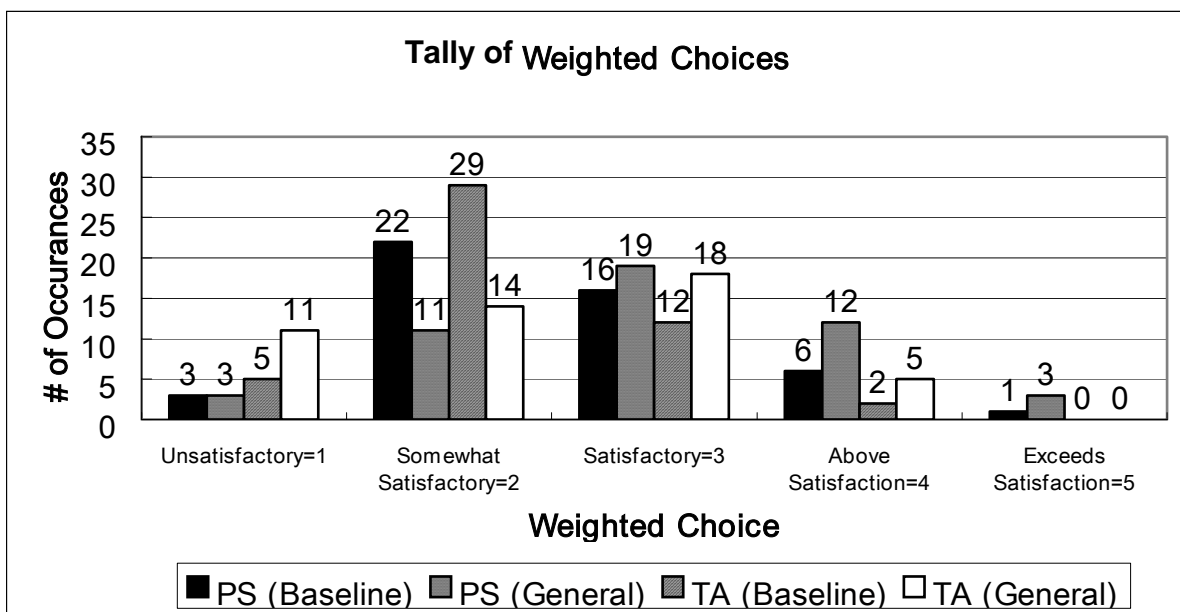
**Figure 4: November 01, 2005
Text Summarization Tool Performance Comparison**

As can be seen from the data, the performance measurements (precision and recall) using excess and generalized keywords are not available (N/A) for the OTS since the summarization tool does not support customization-using keywords. A general trend can be observed for the text summarization tools that support customization-using keywords (PA and TA), where the performance measurements obtained are the lowest for the baseline configuration, the highest when using excess keywords, and in-between when making use of generalized keywords. However, as the data show, there are exceptional cases such as the performance measurements for the PS for October 25, 2005 and November 1, 2005 data. In the former case (PS: October 25, 2005), the measurements for precision and recall were both higher when using generalized keywords. This observation simply goes against the hypothetical expectation that one might anticipate with regard to the optimal performance of the text summarization tool being directly correlated to the amount of aiding supplied. Therefore, it provides an example where too much aiding might have misled the text summarization tool in regard to identifying the appropriate salient sentences. In the later case (PS: November 1, 2005), the measurements for recall are all equivalent (at RC=69%) as the PS tool (machine) was able to successfully extract the same number of salient sentences that matched with the human identification using the 3 different summarization techniques; hence, resulting in the same ratio to be calculated for recall.

Overall, the PS performed the best, followed by the TA and then the OTS (due to its non-support for customization). Furthermore, the TA benefits the most by making use of the customization features, whereas the PS benefits but to a lesser degree. With respect to test data size in sentences, November 1, 2005 test data (151 sentences) is approximately twice the size of the October 25, 2005 test data (64 sentences). The implications of test data length observed in this experiment suggest an overall decrease in performance for the text summarization tools.

Acceptability

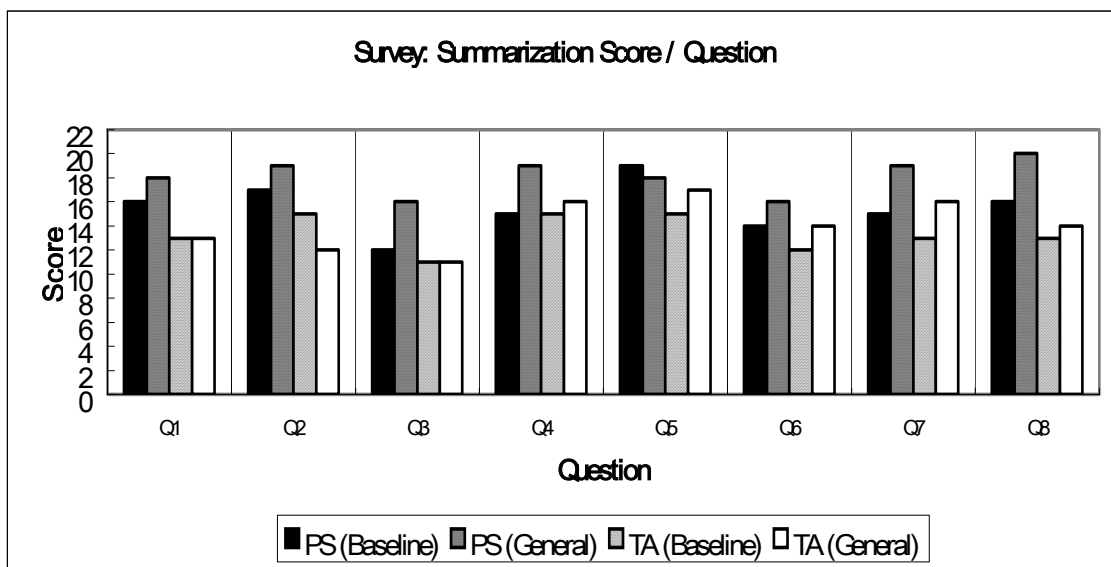
The results of the survey are as follows, 6 responses were successfully received from the 7 meeting participants. Figure 5 presents the overall tally of the weighted choices received from the survey questionnaire for each of the summaries generated by the text summarization tool with their respective techniques (baseline / generalized keywords). Since the questionnaire consisted of 8 questions (Q1 – Q8) and there were 6 participants (= 6 occurrence / question), the sum of all occurrences of weighted choices for each summary should equal 48 (8 question x 6 occurrence / question). For example, the PS using generalized keywords (PS (General)) received the following occurrences of weighted choices from the 6 meeting participants who graded the summaries based on the 8 questions: 3 Unsatisfactory, 11 Somewhat Satisfactory, 19 Satisfactory, 12 Above Satisfaction, and 3 Exceeds Satisfaction. The number of occurrences of weighted choices received totals 48 (3+11+19+12). A tally of the survey results provides quick means to assess and incorporate how the meeting participants felt about the resultant summaries generated by the text summarization tools when employing different techniques.



**Figure 5: November 24, 2005
Survey – Tally of Weighted Choices**

As can be seen from the figure, the meeting participants were either Somewhat Satisfied or Satisfied with the resultant summaries, the highest # of occurrences results for each summary in the two weighted choices aforementioned. Also, the meeting participants felt that the resultant summaries generated using generalized keywords produced better summaries compared to the baseline configuration for both the text summarization tools. This is illustrated by the highest # of occurrences corresponding to the Satisfactory weighted choice when making use of keywords (PS (General) = 19, TA (General) = 18), compared to the highest # of occurrences in the Somewhat Satisfactory weighted choice when using the baseline configuration (PS (Baseline)=22, TA (Baseline)=29).

Even though the quick analysis of the survey tally suggests that the generated summaries were either considered Somewhat Satisfactory or Satisfactory. A more detail analysis must be taken into consideration to determine how well these Somewhat Satisfactory or Satisfactory summaries compared against each other; hence, necessitating the need of a summarization score. Figure 6 presents the obtained scores for each of the summaries generated by the text summarization tool with their respective techniques from the survey questionnaire, which comprises of the 8 questions (Q1 – Q8). The summarization scores for each question was obtained by examining all the survey responses and counting the # of occurrences for each weighted choice (Unsatisfactory = 1 point, Somewhat Satisfactory = 1 point, etc) and multiplying this number with the corresponding weights (1 – 5). For example, Q1 score = (# of occurrences for Unsatisfactory * 1) + (# of occurrences for Somewhat Satisfactory * 2) + ... + (# of occurrences for Exceeds Satisfaction * 5). The overall summarization grade (total score) was obtained by taking the sum of all the individual scores received for each question (Q1 – Q8); this is depicted in Figure 7.



Q1	Capturing Saliency
Q2	Accurate Represented
Q3	Effective Meeting Minute
Q4	Readability
Q5	Comprehension
Q6	Content
Q7	Utility / Effectiveness
Q8	Overall Impression

Figure 6: November 24, 2005
Survey – Summarization Score per Question

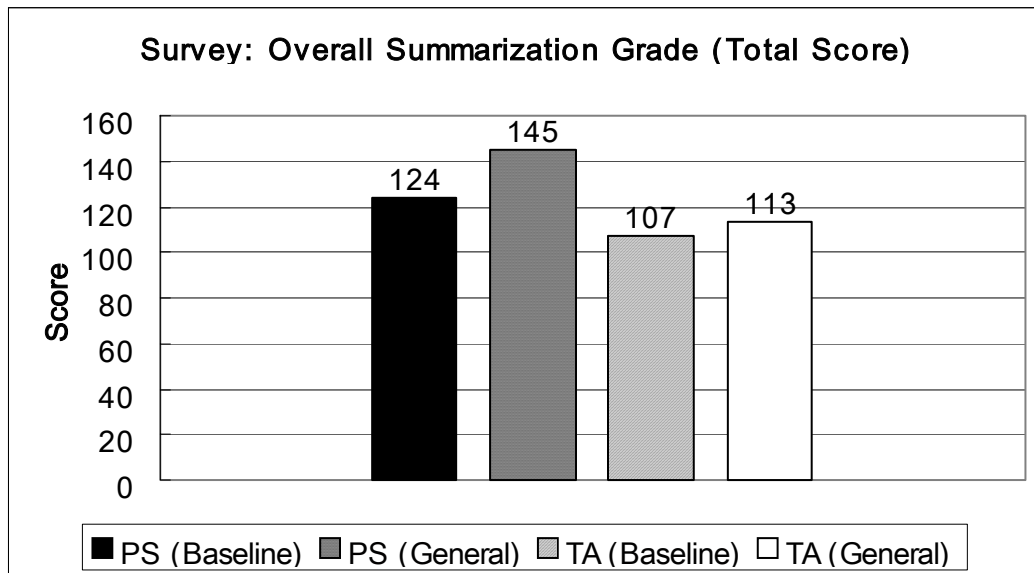


Figure 7: November 24, 2005
Survey – Overall Summarization Grade (Total Score)

As can be seen from the data in Figure 6, PS using generalized keywords (PS (General)) obtained the highest score for nearly all of the questions in consideration which include decisive factor such as capturing and accurately representing salient information (Q1 and Q2), Readability (Q4), Content (Q6), Utility / Effectiveness (Q7), etc. Examination of the scores received per question also allows one to determine the areas where the generated summaries need improvements. For example, although the PS using the baseline configuration (PS (Baseline)) obtained relatively good scores for most of the questions, the score obtained for Q3 is quite low; hence, one might consider focusing on the enhancements to the summary as to produce a more effective meeting minute. Nevertheless, the overall summarization grade (total score) as shown in Figure 7, can be used as the basis to compare against the different summaries generated by the text summarization tools with their respective techniques. The results are as follows; the PS using generalized keywords obtained the best overall summarization grade (total score=145), PS at baseline configuration came second (total score=124), TA using

generalized keywords came third (total score=113), and worst score was acquired by the TA at baseline configuration (total score=107). Furthermore, the resultant overall grade can be used to deduce the quality of the summary; hence, the PS using generalized keywords (PS (General)) generated the best quality summaries.

4 Discussion

The experimental results suggest that making use of summarization techniques such as the incorporation of specific / generalized keywords can improve the performance of the text summarization tool. However, in some cases extensive aiding of the text summarization tool may also lead to degradation in performance. With the small test data inconsideration, PS attained the best results with respect to performance and acceptability, while utilizing all 3 different summarization techniques (baseline, specific keywords, generalized keywords). The results of the survey imply that the overall impressions of the summaries generated by the text summarization tools are either considered Somewhat Satisfactory or Satisfactory. Among the satisfactory summaries, PS using generalized keywords acquired the best overall summarization grade, which appears to correspond to the obtained results from the performance measurements. This in turn implies that PS using generalized keywords produced the best quality summaries over the summarization tools examined in the survey with respect to their employed techniques (baseline and generalized keywords).

Other general factors that might affect the overall performance of the summarization tools are summary length, test data size, summarization approaches, and etc. Although these factors were not directly measured due to the limited time constraints, indirect observations of these effects were evident during experimental conduct and therefore will be noted. Performance seemed to be directly related to the summary length as increasing the length of the summary enhances the likelihood of capturing salient information within a fixed length text, which increased recall but in the cost of degrading precision. Hence, balancing these two measurements (precision and recall) might be important as to obtain an overall good quality performance. Also, performance appeared to decrease as data size increases, this might be attributed to the increase in the number of sentences that need to be examined in retrospect to the smaller percentage of salient sentences that exist within the longer text. Summarization approaches utilized by the text summarization tools were an interesting factor to note, as it appeared that the complexity of the employed technique did not have a positive correlation with the summarization performance. This was somewhat apparent from the experimental results where the PS (less complex approach) outperformed the TA (more sophisticated approach). By no means are these the only factors that may contribute to the overall summarization performance, but nevertheless they are important enough that their consequences must be taken into consideration when trying to produce effective summaries.

Overall, PS seems to be the candidate product of choice among the evaluated products. However, one must not forget to consider the advantages and disadvantages of each product with respect to their features and capabilities, while taking into consideration the limited experimental results before making any decisive decision. Clearly, the limitations of the experimental setup include, the small number of test data examined and its relatively small data size, human subjectivity of salient sentence identification and keyword determinacy to which affects the resulting performance measurements, and small amount of allocated time to carryout the experiment.

Finally, although the current state-of-the-art summarization tools evaluated were unable to generate effective meeting summaries on its own – as indicated by the survey where the summaries merely received a satisfactory grade. Nevertheless, these tools with their corresponding API's provide fundamental basis to which the existing summarization approaches can be further tailored to enhance the overall quality of the resultant summaries. One of the promising techniques that were introduced in this experiment was the use of generalized keywords as to help aid the summarization tool in identifying salient sentences. Other techniques might include compaction of salient sentences to remove redundancy / uninformative information, or even the use of exclusion keywords to assign lesser weights to sentences that contain these words. Thus, appropriate summarization techniques should be investigated, in addition to experimental conduct as to determine the degree of quality improvement. Furthermore, formatting of the resultant summary in an effective and concise manner would enhance its readability and comprehension; hence, improving the overall quality of the summary. As a result, rather than developing a tool from scratch, making use of third party software to support the development of automating meeting summaries seems to be the most practical solution.

5 Conclusion

Feasibility analysis was examined by the way of a production evolution to determine the likelihood of incorporating a third party text summarization tool. The product evaluation process consisted of 4 distinct phases: (1) preparation, (2) criteria establishment, (3) characterization, and (4) testing. Three software products identified and characterized are Open Text Summarizer (open-source), Pertinence Summarizer (commercial), and TextAnalyst (commercial), respectively. Furthermore, testing criteria (performance and acceptability) was determined to which the evaluation experiment was applied to gather the specific measurements (precision, recall, summary grade). The results of the experiment suggest that the current state-of-the-art text summarization tools evaluated cannot produce effective summaries on its own. Rather, it provides the initial basis to which the summarization tools can be tailored to enhance the overall quality of the resultant summaries. As a result, follow-up work should consist of the investigation of appropriate techniques to improve the overall quality of the summaries and experimental conduct as to measure its degree of improvement. Nonetheless, Pertinence Summarizer attained the best results compared to the other two products.

6 Bibliography and Citations

References

- [1] H. P. Luhn, "*The automatic creation of literature abstracts*," IBM Journal of Research and Development, pp. 155--164, April 1958.
- [2] Mark T. Maybury, Inderjeet Mani, *Advances in Automatic Text Summarization*, MIT Press, Cambridge, MA, 1999.
- [3] P. Edmundson. "*New methods in automatic extracting*," Journal of the ACM, vol. 16, no. 2, pp. 264--285, 1969.
- [4] G. Murray, S. Renals, and J. Carletta. "*Extractive summarization of meeting recordings*," In Proc. Interspeech, 2005.
- [5] J. J. Pollock and A. Zamor. "*Automatic Abstracting Research at Chemical Abstracts Service*," Journal of Chemical Information and Computer Sciences, 15(4), 226-232(1975).

Appendix

	Commercial	Commercial	Research	Open Source
	Pertinence Summarizer	TextAnalyst	Extractor4Speech / Extractor	Open Text Summarizer (Libots)
Version	Personal Server Edition	TextAnalyst 2.3: software for automated semantic analysis TextAnalyst COM (SDK): toolkit of text analysis algorithms	Extractor v7.2: text summarization	0.4.2
Company / Educational Institution	Pertinence Mining http://www.pertinence.net	Megaputer http://www.megaputer.com	Extractor: DBI Technologies Inc. http://www.dbi-tech.com/products.asp	Open Source http://libots.sourceforge.net/
Supported Document Format	HTML, PDF, DOC, RTF, TXT and PPT, XLS according to the platforms used	TXT	Extractor v7.2: TXT, HTML, DOC, PDF, and Email	TXT, HTML
Language Support	Supports ten languages (West European and Asian languages, Arabic and Russian)	English	English, French, German, Japanese, Korean, Spanish	English, German, Spanish, Russian, Hebrew, Esperanto and other languages
Summarization Approach	Lexicon, Thesaurus	Semantic network analysis - set of most imp. concepts (word and word comb.) and relationship b/w them - method that uses Artificial Neural Networks to extract semantic information from the text Statistical analysis Dictionary	Genetic Approach (Genetic extraction algorithm)– adaptability to different domains (w/o training); statistical analysis based on approx. 20 different measurements (frequency, key phrases, sentence location, # of words in a phrase, proximity of words, etc)	Statistical analysis based of word frequency
Summarization Features	Key word inclusion / exclusion; variable summary length	Concept and relationship extraction; inclusion / exclusion keyword list; natural language queries for sentence retrieval; variable summary length	Keyword / key phrases extraction; highlighting key sentences and sentence marking; keyword inclusion / exclusion; frequency ranking; multi- document processing; variable summary length	Does not support targeted / customized summaries; variable summary length
Platform	Window, Linux, Unix	Windows	Linux, Solaris, Windows	Linux, Unix, Windows

Hardware Requirements	Any Workstation	Minimum: <ul style="list-style-type: none"> ▪ Intel 486 processor or better ▪ 16 / 24 Mbytes RAM ▪ 10 MB hard disk space 	Any Workstation	Any Workstation
Software Requirements	Java Runtime Environment	Windows 95/98/2000 Windows NT 4.0/ 2000 Profession	None	For windows platform: GladeWin32 (runtime) http://gladewin32.sourceforge.net/ GNUwin32(popt) http://gnuwin32.sourceforge.net/
Development Support	Java API and Web API services are available	TextAnalyst COM (SDK):	Extractor SDK: API support: C (C, C++, VC++), Java, Visual Basic, Python, Perl	Source Code
Documentation	N/A	Corresponding Documentation	Corresponding Documentation	Source Code Documentation
Product Cost (Canadian \$)	Additional cost: \$141.47 / language Personal Edition: \$169.20 Server Edition (2 users): \$1414.75	TextAnalyst 2.3: \$1,290 TextAnalyst COM (SDK): \$9,990	Extractor SDK: \$8,907 Educational Discount (50%): \$4,454	Free
Licence Cost (Canadian \$)	Server edition: \$141.47 / user	N/A	N/A	None
First Release	N/A	1999 (6 years)	2004 (1 year)	N/A

Figure 1: Software Criteria Matrix

Frank: Then let's get started. What did you do since the last time?

Rob: Um. When was the last meeting? Wednesday? So I got I got some of the initial GDF structure setup. Um. I'm still working on that. So you know try refreshing. Going through making sure all these stuffs are setup properly. And a...I imported some of the video. Or actually I brought all the video imported but I haven't started analyzing yet.

Frank: Um.

Rob: That's about all that's happened.

Frank: And that's to the next to Thursday?

Rob: Um. Thursday I hope the some of the GEF some more of the GEF stuff done. Um. Specifically I want to get the other part we did and one on carbon screen.

Frank: Ok. Anything important?

Charles: I have changed the e-mail address and other things into numbers.

Beth: Also the declarations.

Frank: Ah! yeah. Ok. Enum. So that's done. So what's the plan to the next day?

Charles: This time I give you constraint or sequel connection.

Beth: That's the fixture to EJB. Like remote connection. The task of the task.

Charles: And I'll try to try to update the output out stream to the input stream system.

Frank: Ok. So next time is on Thursday. So you can get all that done by Thursday?

Charles: No.

Frank: So what's the plan to Thursday?

Charles: Thursday?

Frank: Yeah.

Charles: Work on Fixture.

Frank: Fixture. ok.

Connie: I'm working with Charles. Working on EJB.

Beth: And for me, I..I'm still working on that new project. I'm trying to opt the project. And since I'm going to hold like a presentation for the people in the university lab, so let them be familiar with Junit something and extension of JUnit like XML unit, jimmy, JFCunit, which is connected to ...related to swing face.

Frank: When is it? When is this presentations?

Beth: I think in less than a week. Because if I just wrote like a letter to the entry and then they have to decide.

Frank: Let me know when it is. Then I'll like to...

Shelly: Um. I setup that Open CV library that Dr. Boyd recommended. I'm trying it out. But it was all the tutorials are done for Visual Studio 6. Not Visual Studio .net.

Rob: Oh. Good luck.

Shelly: So I got Visual Studio 6 from someone else ...um...I'm going to try that one out. And I'm also researching about remote framework for programming Aibo.

Frank: Yep. You are looking. You are simply sitting in. So...

Ruth: Ok. Um... I got this wonderful recording script going. Hopefully it's still going. Um... and..Shelly and I have been looking at this Remote Framework for programming. It's sort of higher level than tekkotsu.

Frank: What's the plan for Thursday?

Ruth: The plan till thursday? Actually the next task over there was..ah...either get the dog to not fall off the table, or ...

Frank: Follow the, follow where the sound is.

Ruth: Yes. That's the other one. I think actually follow the sound would be better. I'll do that.

Frank: Ok.

Ruth: Well, No sleep for me.

Frank: Anybody running into trouble that they can't solve themselves. No? Then we are done.

Ruth: Yei.

Frank: 10 minutes.

**Figure 2: October 25, 2005
Meeting Transcript**

Ruth: I've got some recordings of ah Dave um and I reading out that UML sheet with three different Aibos to see if either there are any difference in sound quality.
Frank: Ah. Ok
Ruth: Um. It's terrible in all three. Because we think that his voice and the noise of the lab were kind of the same level and we think that it might um be very difficult to pick anything up. My voice seems alright.
Frank: Because of the higher pitch?
Ruth: I think so. I think so. But I think his voice is very low. And the noise of the HCI lab kind of blended in. So.
Frank: What's the next step?
Ruth: Next step for me would be to have the Aibo look at whoever's talking.
Frank: Ah. Ok.
Ruth: That's my goal for next
Frank: Until Thursday?
Ruth: Thursday? Maybe next Tuesday. I won't start anything by Thursday.
Frank: Ok. Any problems?
Ruth: I'm not getting any sleep again.
Frank: Ok. It'll be a problem
Rob: Oh well.
Ruth: Oh well.
Frank: What did you do?
Beth: So I'm before taking the presentation and also applied for the examples for unit extensions like ... like open tools.
Frank: Yep
Beth: And how to present some. I sat with Charles and sorted some problems about DB3.
Frank: Next plan?
Beth: Next plan finish the presentation preparation and I will like a . I have a presentation next week.
Frank: Ok. Practice.
Rob: Um. Yeah, I haven't done a whole lot. I've been busy with course work.
Frank: Really?
Rob: Um, Yeah. It's just a everything's due on Wednesday for Saul's course. So I'm trying to get all that stuffs done first.
Frank: But then you have.
Rob: And then I have basically from Thursday afternoon until following wednesday afternoon to get all the rally stuff done.
Frank: Nothing else.
Rob: I started looking at some of the video from the first couple of the iteration planning meetings.
Frank: Ah Ok.
Rob: But that's. yeah. Pretty much everything's on hold right now until I get this paper out the door.
Frank: So the comment that I gave you were minor on your presentation.
Ruth: No, no, no.
Rob: That's a totally different presentation, but yeah.
Frank: Yeah. Too much work.
Rob: Not yet.
Ruth: Don't start that. Ok
Frank: What did you since the last meeting?
Shelly: Um. I tried out few face recognition open source programs. None of them are working very nicely. Can't even recognize my own face. So I gave them about fifty training data, but it still can't find my face in there, so.
Frank: Then think about cheating. How can we help the software to actually recognize people. I don't care about faces. In the end, the goal is actually to attribute bound to specific person. You don't have to do face recognition. For example, my bright orange is very easy to recognize probably. Or the pink stripes.
Shelly: Right. At this point I think I'm just going to move onto voice recognition part. Then
Frank: He likes it.
Ruth: I think it's the voice recognition to play the song.
Shelly: Oh.

Ruth: Ok. Go on.
Shelly: Not my plan.
Frank: What's the concrete next step this week?
Shelly: Um. Find out if there is a. Actually No. I'm just going to get that movie and dessemble them into different frequencies and see if.
Frank: Frequencies?
Shelly: Of field. Sound. Voice. And if find out registers of each voices.
Frank: Ok.
Shelly: And if see if I can do some kind of identity recognition through voices.
Frank: Ok. Good.
Connie: I've been working with Charles on EJB stuff. And getting the test to work, but I've been in the middle of midterms and assignments. So he's been doing all the work. And then.
Charles: Now I can. I can do something from my computer to the server. Because I think can control the second bit.
Frank: And you are testing against the new EJB3 immitation of MASE. Charles: Yes. EJB3. Not MASE, but EJB3 from test. However test, some simple test. But when I find the time to fix the fixture, it doesn't work. There's something wrong with that.
Frank: Typically that that is. What's the problem that you are getting.
Charles: It says that it can't find a class.
Frank: It's a class path problem. It's a class path problem.
Charles: Classpath?
Frank: Yeah. So basically, if it can't find the path, it basically means that your java file or whatever with that fixture is not on the classpath that you are currently using.
Beth: But I know that the fixture should be on the server.
Frank: Fixture doesn't have to be on the server. But your server. Client job from the server side has to be on the fixture. classpath of the client.
Charles: Yes
Frank: to be on that classpath.
Beth: The recovers is right now I think it's because when we do package of EJB3 we test all the things under being directory. So all the test files and fixture files are exact and will be include.
Frank: On the server?
Beth: Yeah. Will be packaged. Because we haven't changed the package or something. If we are handling works on server, because the tables itself JM dot jar. But I think we have to separate. Now server, we can only include session beans and answer beans.
Frank: Yes. But you said that client is creating the errors.
Charles: No. Just I want to try out on the same machine. the error comes up and then.
Beth: In Jboss
Charles: Now I'm trying separate EJB with my time.
Frank: But I thought it's already working. You have a server and you have a client on another machine. Is this working?
Charles: Not yet. I'm trying to do that.
Frank: so on the same machine it works. But separate machine it doesn't.
Charles: No. On the same machine it doesn't work. But I think on separate machine it seems to work.
Beth: He means why. Because in JBoss environment, because we included all test, in the EJB3 that fire. So we don't we haven't included fixture in fitness dot jar onto the JBoss server. Now JBoss server reported an error like . It cannot find the column fixture. But it is not right.
Frank: JBoss server shoudln't execute column fixtures.
Beth: Yes. So we have to separate the things. Now when we package EJB3 and we package all the things under beans. It includes everything from beans. Now we have to separate like server should only include session beans and client.
Frank: And there is a client java file that you have to include on the machine where the fitness runs. So you can actually grade a connection to the J and the I server. Look and download client java file and you can get copies.
Beth: Actually we got the title working just like yesterday and in that title, we only created like a small java client files. So in that project, we included time dot jar with some other kind of requirements.
Frank: So do you think it can be done by Thursday?
Charles: I think so.

Frank: Good
Charles: This Thursday. Simple test. What do you mean?
Frank: JBoss, Client Fitness connection.
Charles: Client Fitness JBoss. JBoss connection? This is server.
Frnak: This is JBoss Server. This is another machine with your tests. And then run the test again on the server.Charles: Ok. I think I can do that.
Frank: That's why you are there.
Charles: There will be some difficulty there, but I'll try.
Frank:Good. Then we are done for today?
Connie: Is there an iteration meeting tomorrow?
Frank: Is it tomorrow or next week?
Rob: Following two weeks is tomorrow.
Frank: I forget everything until something pops up on my computer. In my schedule so. so how are we doing?
Rob: Mine are all in half way stages. Which is not good.
Frank: Not how we do in agile environment. You start a task. You finish it. You shouldn't have multiple tasks un half stages.
Rob: I know. It's only you can only put in certain hours before you have to switch.
Frank: Yep. It can be done. So we are not meeting tomorrow. Let's have a look and decide whether we want to have the iteration meeting planned back.

**Figure 3: November 1, 2005
Meeting Transcript**

Frank: Then let's get started. What did you do since the last time?

Rob: Um. When was the last meeting? Wednesday? So I got I got some of the initial GDF structure setup. Um. I'm still working on that. So you know try refreshing. Going through making sure all these stuffs are setup properly. And a...I imported some of the video. Or actually I brought all the video imported but I haven't started analyzing yet.

Frank: Um.

Rob: That's about all that's happened.

Frank: And that's to the next to Thursday?

Rob: Um. Thursday I hope the some of the GEF some more of the GEF stuff done. Um. Specifically I want to get the other part we did and one on carbon screen.

Frank: Ok. Anything important?

Charles: I have changed the e-mail address and other things into numbers.

Beth: Also the declarations.

Frank: Ah! yeah. Ok. Enum. So that's done. So what's the plan to the next day?

Charles: This time I give you constraint or sequel connection.

Beth: That's the fixture to EJB. Like remote connection. The task of the task.

Charles: And I'll try to try to update the output out stream to the input stream system.

Frank: Ok. So next time is on Thursday. So you can get all that done by Thursday?

Charles: No.

Frank: So what's the plan to Thursday?

Charles: Thursday?

Frank: Yeah.

Charles: Work on Fixture.

Frank: Fixture. ok.

Connie: I'm working with Charles. Working on EJB.

Beth: And for me, I..I'm still working on that new project. I'm trying to opt the project. And since I'm going to hold like a presentation for the people in the university lab, so let them be familiar with Junit something and extension of JUnit like XML unit, jimmy, JFCunit, which is connected to ...related to swing face.

Frank: When is it? When is this presentations?

Beth: I think in less than a week. Because if I just wrote like a letter to the entry and then they have to decide.

Frank: Let me know when it is. Then I'll like to...

Shelly: Um. I setup that Open CV library that Dr. Boyd recommended. I'm trying it out. But it was all the tutorials are done for Visual Studio 6. Not Visual Studio .net.

Rob: Oh. Good luck.

Shelly: So I got Visual Studio 6 from someone else ...um...I'm going to try that one out. And I'm also researching about remote framework for programming Aibo.

Frank: Yep. You are looking. You are simply sitting in. So...

Ruth: Ok. Um... I got this wonderful recording script going. Hopefully it's still going. Um... and..Shelly and I have been looking at this Remote Framework for programming. It's sort of higher level than tekkotsu.

Frank: What's the plan for Thursday?

Ruth: The plan till thursday? Actually the next task over there was..ah...either get the dog to not fall off the table, or ...

Frank: Follow the, follow where the sound is.

Ruth: Yes. That's the other one. I think actually follow the sound would be better. I'll do that.

Frank: Ok.

Ruth: Well, No sleep for me.

Frank: Anybody running into trouble that they can't solve themselves. No? Then we are done.

Ruth: Yei.

Frank: 10 minutes.

Figure 4: October 25, 2005
Meeting Transcript with Salient Sentence Identification

Ruth: I've got some recordings of ah Dave um and I reading out that UML sheet with three different Aibos to see if either there are any difference in sound quality.

Frank: Ah. Ok

Ruth: Um. It's terrible in all three. Because we think that his voice and the noise of the lab were kind of the same level and we think that it might um be very difficult to pick anything up. My voice seems alright.

Frank: Because of the higher pitch?

Ruth: I think so. I think so. But I think his voice is very low. And the noise of the HCI lab kind of blended in. So.

Frank: What's the next step?

Ruth: Next step for me would be to have the Aibo look at whoever's talking.

Frank: Ah. Ok.

Ruth: That's my goal for next

Frank: Until Thursday?

Ruth: Thursday? Maybe next Tuesday. I won't start anything by Thursday.

Frank: Ok. Any problems?

Ruth: I'm not getting any sleep again.

Frank: Ok. It'll be a problem

Rob: Oh well.

Ruth: Oh well.

Frank: What did you do?

Beth: So I'm before taking the presentation and also applied for the examples for unit extensions like ... like open tools.

Frank: Yep

Beth: And how to present some. I sat with Charles and sorted some problems about DB3.

Frank: Next plan?

Beth: Next plan finish the presentation preparation and I will like a . I have a presentation next week.

Frank: Ok. Practice.

Rob: Um. Yeah, I haven't done a whole lot. I've been busy with course work.

Frank: Really?

Rob: Um, Yeah. It's just a everything's due on Wednesday for Saul's course. So I'm trying to get all that stuffs done first.

Frank: But then you have.

Rob: And then I have basically from Thursday afternoon until following wednesday afternoon to get all the rally stuff done.

Frank: Nothing else.

Rob: I started looking at some of the video from the first couple of the iteration planning meetings.

Frank: Ah Ok.

Rob: But that's. yeah. Pretty much everything's on hold right now until I get this paper out the door.

Frank: So the comment that I gave you were minor on your presentation.

Ruth: No,no,no.

Rob: That's a totally different presentation, but yeah.

Frank: Yeah. Too much work.

Rob: Not yet.

Ruth: Don't start that. Ok

Frank: What did you since the last meeting?

Shelly: Um. I tried out few face recognition open source programs. None of them are working very nicely. Can't even recognize my own face. So I gave them about fifty training data, but it still can't find my face in there, so.

Frank: Then think about cheating. How can we help the software to actually recognize people. I don't care about faces. In the end, the goal is actually to attribute bound to specific person. You don't have to do face recognition. For example, my bright orange is very easy to recognize probably. Or the pink stripes.

Shelly: Right. At this point I think I'm just going to move onto voice recognition part. Then

Frank: He likes it.

Ruth: I think it's the voice recognition to play the song.

Shelly: Oh.

Ruth: Ok. Go on.
Shelly: Not my plan.
Frank: What's the concrete next step this week?
Shelly: Um. Find out if there is a. Actually No. I'm just going to get that movie and dessemble them into different frequencies and see if.
Frank: Frequencies?
Shelly: Of field. Sound. Voice. And if find out registers of each voices.
Frank: Ok.
Shelly: And if see if I can do some kind of identity recognition through voices.
Frank: Ok. Good.
Connie: I've been working with Charles on EJB stuff. And getting the test to work, but I've been in the middle of midterms and assignments. So he's been doing all the work. And then.
Charles: Now I can. I can do something from my computer to the server. Because I think can control the second bit.
Frank: And you are testing against the new EJB3 immitation of MASE. Charles: Yes. EJB3. Not MASE, but EJB3 from test. However test, some simple test. But when I find the time to fix the fixture, it doesn't work. There's something wrong with that.
Frank: Typically that that is. What's the problem that you are getting.
Charles: It says that it can't find a class.
Frank: It's a class path problem. It's a class path problem.
Charles: Classpath?
Frank: Yeah. So basically, if it can't find the path, it basically means that your java file or whatever with that fixture is not on the classpath that you are currently using.
Beth: But I know that the fixture should be on the server.
Frank: Fixture doesn't have to be on the server. But your server. Client job from the server side has to be on the fixture. classpath of the client.
Charles: Yes
Frank: to be on that classpath.
Beth: The recovers is right now I think it's because when we do package of EJB3 we test all the things under being directory. So all the test files and fixture files are exact and will be include.
Frank: On the server?
Beth: Yeah. Will be packaged. Because we haven't changed the package or something. If we are handling works on server, because the tables itself JM dot jar. But I think we have to separate. Now server, we can only include session beans and answer beans.
Frank: Yes. But you said that client is creating the errors.
Charles: No. Just I want to try out on the same machine. the error comes up and then.
Beth: In Jboss
Charles: Now I'm trying separate EJB with my time.
Frank: But I thought it's already working. You have a server and you have a client on another machine. Is this working?
Charles: Not yet. I'm trying to do that.
Frank: so on the same machine it works. But separate machine it doesn't.
Charles: No. On the same machine it doesn't work. But I think on separate machine it seems to work.
Beth: He means why. Because in JBoss environment, because we included all test, in the EJB3 that fire. So we don't we haven't included fixture in fitness dot jar onto the JBoss server. Now JBoss server reported an error like . It cannot find the column fixture. But it is not right.
Frank: JBoss server shoudln't execute column fixtures.
Beth: Yes. So we have to separate the things. Now when we package EJB3 and we package all the things under beans. It includes everything from beans. Now we have to separate like server should only include session beans and client.
Frank: And there is a client java file that you have to include on the machine where the fitness runs. So you can actually grade a connection to the J and the I server. Look and download client java file and you can get copies.
Beth: Actually we got the title working just like yesterday and in that title, we only created like a small java client files. So in that project, we included time dot jar with some other kind of requirements.
Frank: So do you think it can be done by Thursday?
Charles: I think so.

Frank: Good
Charles: This Thursday. Simple test. What do you mean?
Frank: JBoss, Client Fitness connection.
Charles: Client Fitness JBoss. JBoss connection? This is server.
Frnak: This is JBoss Server. This is another machine with your tests. And then run the test again on the server.
Charles: Ok. I think I can do that.
Frank: That's why you are there.
Charles: There will be some difficulty there, but I'll try.
Frank: Good. Then we are done for today?
Connie: Is there an iteration meeting tomorrow?
Frank: Is it tomorrow or next week?
Rob: Following two weeks is tomorrow.
Frank: I forget everything until something pops up on my computer. In my schedule so. so how are we doing?
Rob: Mine are all in half way stages. Which is not good.
Frank: Not how we do in agile environment. You start a task. You finish it. You shouldn't have multiple tasks un half stages.
Rob: I know. It's only you can only put in certain hours before you have to switch.
Frank: Yep. It can be done. So we are not meeting tomorrow. Let's have a look and decide whether we want to have the iteration meeting planned back.

Figure 5: November 1, 2005
Meeting Transcript with Salient Sentence Identificatio
