Email search visualization: An efficient way for searching email

Tung Vuong

MSc Thesis

Helsinki 29.04.2014

UNIVERSITY OF HELSINKI

Department of Computer Science

HELSINGIN YLIOPISTO - HELSINGFORS UNIVERSITET - UNIVERSITY OF HELSINKI

Tiedekunta - Fakultet –Faculty	Laitos - Institution - Department
Faculty of Science	Department of Computer Science

Tekijä - Författare - Author

Vuong Thanh Tung

Työn nimi — Arbetets titel — Title

Email search visualization: An efficient way for searching email

Oppiaine _ Läroämne _ Subject

Computer Science

Työn laji - Arbetets art - Level	Aika-Datum-Month and year	Sivumäärä-Sidoantal-Number of pages
MSc Thesis	29.04.2014	66

Tiivistelmä _ Referat _ Abstract

Originally email was designed for messaging channel between individuals. However, it is now being used as personal file archiving. The growth of email quantity over time makes searching email hard and time-consuming. It even becomes extremely frustrating for people who have huge amount of emails. Firstly, when searching an email, people often look for a piece of information which is temporarily forgotten. If we can remind of them some related information, searching activity will become faster and easier. Secondly, since memory of information is temporarily lost, it is hard for people to define keyword to type in search box. If people misspell the word, search result will return no hits at all. In order to solve those issues, we provide an interface that support people seeking email more interactively. The interface is the combination of studies from email visualization and keyword extraction technique. Email visualization is the process of visually display email attributes such as sender, recipient, date, and content on the interface. Keyword extraction is a technique of extracting words directly from email body. Extracted keyword can describe related topical word which is meaningful and can be easily understood by users. Main objective of the interface is helping people to reinstate their memory of email information while visually interacting with different email attributes.

ACM Computing Classification System (CCS):
H.3.1 [Content Analysis and Indexing]
H.5.1 [Multimedia Information Systems]
H.5.2 [User Interfaces]

Avainsanat - Nyckelord - Keywords

Information visualization, natural language processing, email search.

Säilytyspaikka - Förvaringsställe - Where deposited

Muita tietoja - övriga uppgifter - Additional information

Contents

CHAPTER 1 - INTRODUCTION

Electronic mail (email) is one of the most successful technologies. Millions of email users spend a significant proportion of time on checking their mails. Research also shows that email greatly helps connecting people regardless of time and geographical location [Bikson92]. Original conception of email was exchanging digital mails between individuals. Over decades, it has been evolved beyond its definition of just an asynchronous communication channel [Gwizdka02]. People nowadays consider email as a tool for daily task management, file repository, or personal information archive. These additional functions are identified as a phenomenon called "email overload" [Whittaker96]. Researchers also address the most major problem with this phenomenon relates to personal archiving. The reason is the growth in mailbox size makes it hard to re-find past emails, related links, and attachments. Indeed, evidence shows that emails hold the worst result of the ease of finding information, but are the most searched information items at work settings [Elsweiler11][Robbins06][Cutrell03]. Besides, two additional problems also cause difficulty for finding email information. First, people only begin to look for an email when they are trying to figure out a piece of information that is temporarily forgotten. They scroll up and down a collection of emails hopefully to find their wanted information. Scrolling behavior consumes high amount of time because a long list of emails could not be fit into a single screen. Second, the moment when scrolling behavior failed, they start typing query into search box. Problem arises when they don't precisely know what keyword to use. If keyword is a common word, it returns too many hits; and if keyword is misspelled, it returns no hits at all. Thus, people will be involved into a sequence of typing different keywords hopefully to get an adequate amount of returned hits. In this thesis, I analyze the root cause of those problems and propose the mechanism which helps to reduce the complexity of dealing with such large email information data sets.

1.1 Objective

Traditional email clients support full-text search engine with textual information. However, the interface designed with textual information displayed on screen might decrease the perception of users and slow down user's responsiveness. Reasons are too much text confuses users and it's hard to skim through every email to find the right information. Therefore, first objective of this thesis is to create novel visual search interface that support interactive email seeking beyond query and response. Unlike other search engines, proposed visual interface allows people to

directly interact with associative email attributes on the interface. Purpose of this visual interaction is to present essential information that effectively reminds people of where wanted information is located in the mailbox.

Second objective is to investigate retrieval techniques that are needed to support the interactive visual interface. Techniques discussed in this thesis are how suggested keywords can be extracted from email body and what layout should be utilized to increase the quality of search function. Suggested keyword is key success for the email interface. Extracted word needs to be meaningful and is able to describe the topic of email conversation. Layout of the interface is designed in the way that it can eradicate scrolling behavior.

1.2 Method

Firstly, the main method represented in this thesis is about integrating a prominent collection of techniques called Information Visualization (InfoVis) into email search system. InfoVis is a study on computer graphics that reinforce human perception toward data retrieval [Ware12]. By integrating visualization technique into email search engine, apparently we can avoid scrolling behavior during search activity. Raw data retrieved from mailbox will be tailored and converted into visualized content to allow human cognition during search process. Visualized information must base on timeline layout within user's mailbox because people always perceive email as the information storage ordered by time. Visualized relationship between emails' attributes such as content, date, senders, and recipients should be shown in the way that it helps to reinstate user's memory of email information.

Secondly, a set of suggested keywords describing brief email information should be presented on the interface. Those keywords can help to reduce the amount of typing users must perform during search process. In order to do that, we need useful features from several related technologies such as Apache Lucene Core, Natural Language Processing, and Java Mail.

Apache Lucene Core is cross-platform and high performance search engine library that provides full-featured text search by utilizing Java-based indexing.

Java Mail API provides framework for processing emails from IMAP server. IMAP is a protocol that allows email clients to access remotely and retrieves information from mail server.

Natural Language Processing (NLP) is the computerized approach to analyzing and processing text. The purpose of using NLP in this thesis is producing a set of appropriate suggested keywords.

The thesis will be organized in the following order. First chapter will discuss about email information retrieval background and difficulties in re-finding email information. Second chapter will present technologies used in the thesis. Third chapter will focus on the development and design principles of application interface. Fourth chapter will present the implementation process. Final chapter will shows the statistics and evaluation result from the new interface of email information retrieval system.

CHAPTER 2 – BACKGROUND

Over last few years, people have witnessed the emergence and usefulness of many new communication technologies including instance messaging, mobile, voice, video conference, and social network. Nonetheless, email is still the most commonly used at work settings [Whittaker11]. Despite the fact of the people relies so much on email at workplace, its usage is still poorly understood. In reality, email critically affects company productivity because all business processes, contact information, and people archives are stored within their mailbox. Thus, mailbox now becomes even more important, and email information retrieval is a very crucial daily task.

2.1 Email Information Retrieval

Based on previous work [Whittaker11], email information retrieval is classified into two specific strategies regarding different behaviors in finding information. First strategy is called *preparatory* organization which is more about email management for later retrieval. In contrast to preparatory, opportunistic strategy refers to search and retrieve method which skips the preparing step.

Preparatory strategy implies *folder management*. Emails are deliberately organized into specific memorable folders for later retrieval. This strategy seems to be efficient for re-finding information because each folder has meaningful name and contains less amount of emails. Nowadays with advances in email application clients, people do not need to manually classify emails into folders. The application automatically does the job by preparing matching criteria for each folder beforehand. However, there are two drawbacks for this strategy.

- Mailbox grows quickly overtime leading to possibility of each folder comprises of many emails. Thus, each folder becomes large, and it is still frustrating to re-find specific information.
- Whittaker and Sidner argued that for management purpose, creating many folders is inevitable [Whittaker11]. Eventually, people must maintain the consistency between folder name and filtering criteria to avoid duplication. It will definitely take more time for folder preparation than finding information. Thus, the name *preparatory* implies this strategy.

Opportunistic strategies refer to pure access behavior which is *search and retrieve* procedure. Opportunistic behaviors such as searching, sorting, and scrolling do not require efforts to organize folders but it's time consuming. In contrast to preparatory strategy, it is not costly to enact and it reduces the complexity in folder tree. However, without folders, some important emails will be overlooked when unorganized huge amount of emails in inbox accumulate overtime.

Both strategies have pros and cons, but they have a very tight relationship. Preparatory strategy eventually needs to depend on access behavior when folders accumulate sufficient large number of emails. Based on actual statistics and analysis, research also indicates that foldering produce less efficiency and no more successful than access behavior [Whittaker11]. The actual fact is people reliance on foldering technique tend to have longer duration to re-finding information. The reason is selecting the right folder consumes time. Additional factor related to preparatory method is poorly organized folders dramatically reduce success rate of finding information [Whittaker96]. When foldering technique fails, people likely go back to their pure access behaviors. Research analysis also indicates that opportunistic strategy was both efficient and led to more successful retrieval [Whittaker11]. Moreover, prior work believes manual foldering method is onerous to enact. Therefore, opportunistic strategy is more efficient and better technique to re-find email information. Nonetheless, opportunistic behaviors are still perceived as a very challenging task, especially when time elapsed since last access is long [Elsweiler11].

2.2 Difficulties in re-finding email information

Re-finding email information can be assessed as a task which is relatively easy and straightforward, or it can be very challenging, time-consuming and frustrating [Elsweiler11]. Previous research field called Personal Information Management (PIM) studies on behaviors of how people manage their information and attempts to develop a system that helps people re-find their information [Elsweiler11]. In PIM research domain, re-finding behavior is described as an act of re-accessing and re-using information that people have accessed in the past [Dumais03]. The investigation also reveals three main classified tasks which are performed to re-find information [Elsweiler07]:

Lookup task involves searching for particular information such as password, personal ID, or telephone information. This piece of information is very small and specific that people just

want to know while ignoring other information within the same body of email. The difficulty of this task is memory related to email body is mostly forgettable. Thus users might or might not exactly know the source contains that piece of information.

Item task involves looking for an email with full body text, perhaps to forward to someone or just to review information. In contrast to lookup task, people know the full resource that they are looking for, but they don't know where to find it. They want to retrieve entire body of email resource to complete this task.

Multi-item task is the task that requests to retrieve numerous email messages or a conversation between senders. Often this task requires people to process and collate email body in order to complete the task.

There is also a situation where a task might involve both lookup task and item task simultaneously. For instance, an employee wants to forward full details information about the contract with contract number to sale department. This task seems to be straightforward because the keyword (or contract number) is known and email body is remembered.

Statistical analysis result from previous study indicates that there are majority of tasks are short-timing and straightforward [Elsweiler11]. However there are also some tasks might be problematic, time-consuming. The problem was underlined clearly by examining Message Uncertainty Ratio (MUR) [Elsweiler11]. MUR shows that most of the problems were caused by memory lost and time elapsed since last view. If email messages are viewed only once or twice and need to be re-accessed, then it will apparently causes the problem.

Furthermore, a previous analysis mentions several behaviors or approaches used to complete the task [Elsweiler11]. They are the use of folder, the use of sorting, submitted queries, and re-finding overtime. People might have one or multiple behaviors at a time in order to re-find their wanted information. Each of behaviors has its own difficulties and needs to be resolved.

- *The use of folder* [Whittaker11]: as mentioned in previous section. Use of folder refers to preparation strategy. This behavior is proven that there is no more efficient than opportunistic search behavior [Whittaker11]. The evidence shows people occasionally become disorientated when looking within folders. If the folder does not contain desired

piece of information, people must re-select the right folder to continue the task and it leads to more disoriented behavior.

- *The use of sorting in re-finding* [Whittaker11]: there are several different types of sort such as sort by rank or relevance, date, author, subject, to, and other. Analyzed data from previous work displays search results are most likely to be sorted based on date [Dumais03][Robbins06]. Researchers believe that the reason people want to sort search results by date because of human memory [Robbins06]. Memorable information usually is organized based on time. It becomes easier to recall memory when people refer to each time period. However, after queries return many search results sorted by date, people still encounter the difficulty of scrolling and manually searching for correct piece of email information. When such a case happens, the next attribute people often refer to is author. The reason is people might partially remember email information that they are looking for, and partial information most likely consists of author's name. In summary, two crucial attributes people often use to sort search results are date, and author. Thus, these criteria are always taken into consideration when it comes to the design of email application client.
- *Submitted queries or keywords* [Whittaker11]*:* Searching seems to be likely a very popular tactics. Different queries are submitted in order to complete the task. Researchers indicate that most submitted queries tend to be a short term query. A single keyword is popularly used for several first attempts, and later it increases in length and number of words. Perhaps, it indicates that after failing with shorter queries, people tend to reformulate the query with longer keywords. Interestingly author's name is the most prevalent in almost all submitted queries. In term of query performance, most of queries return empty or too many results, only over one third returns adequate number of emails. Additionally, the number of attempts for each task is often more than 1. It simply means keywords are difficult to be defined, and people most likely remember only partial words in email information.
- *Re-finding overtime* [Whittaker11]: this behavior refers to the fact that all re-found messages from search result are likely be re-found again within 4 months. Intuitively, the more people try to re-find the same piece of information, the less time-consuming it takes for each re-find task.

In summary for this chapter, email re-finding is a very challenging in either lookup task or item task. Most of tasks are very easy and straight-forward, but some evidence shows that some cases are extremely difficult. This is due to disorientation of memory and folder (MUR and FUR). Most of search queries consist of a single partial keyword and people try at least more than 1 attempt in order to complete re-finding task. The reason is keywords are hard to be defined and people can only remember partial words from email content. Last but not least, date and time is most favorable attribute when it comes to sorting search result information.

CHAPTER 3 –EMAIL SEARCH VISUALIZATION AS PROMISING TECHNOLOGY

In this thesis, email search visualization system comprises of many visual objects which depict relationship between email attributes such as senders, recipients, email topic, and time. Purpose of email visualization is to reinstate user memory of email content in order to allow users to

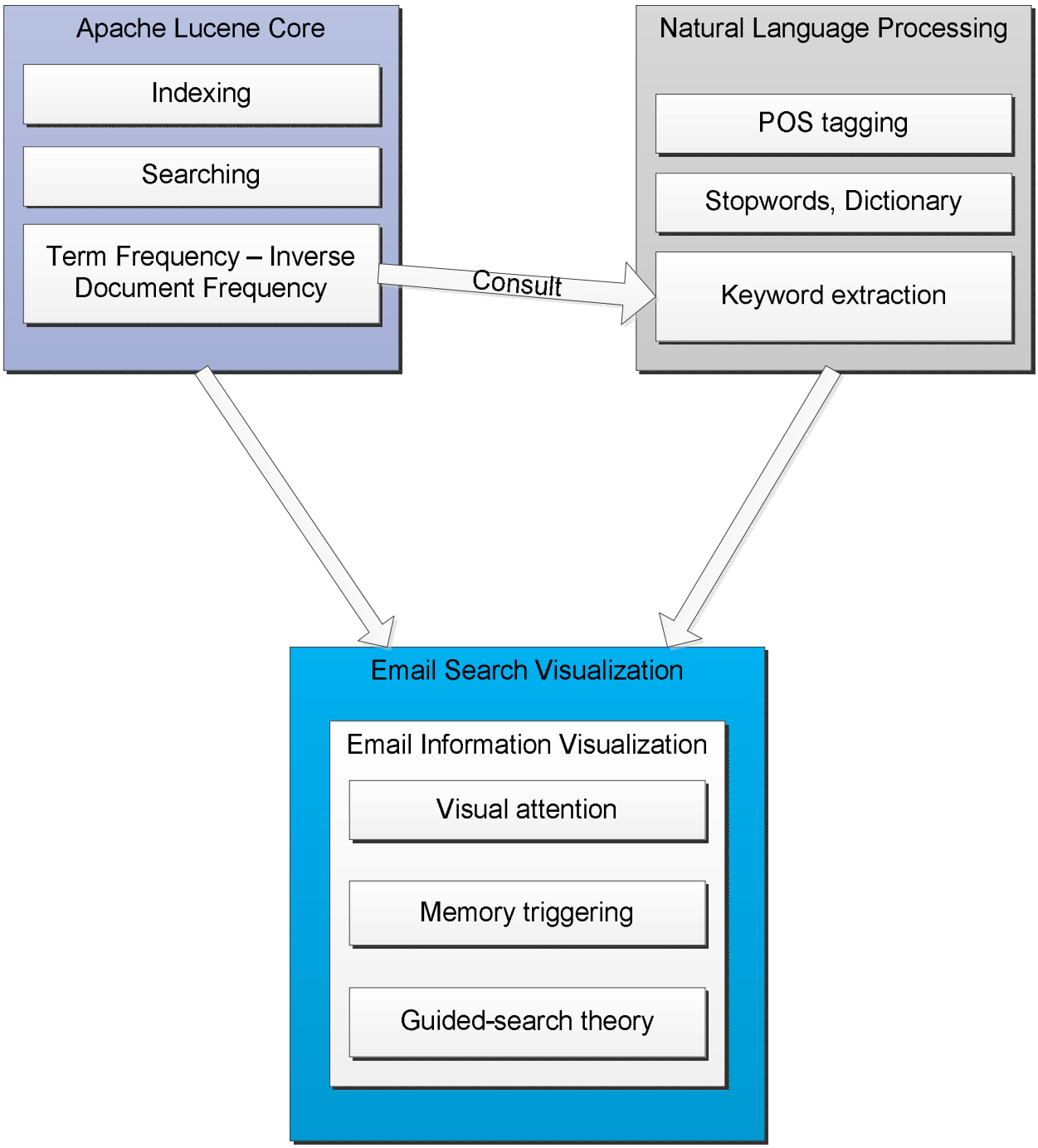

Figure 1. Technology overview in Email Search Graph Application

efficiently retrieve their desired email information. In PIM domain, Dumais et al and Cutrell et al stress that memory is crucial for re-finding emails, and they aim to provide associative and contextual cues to trigger memory revival during search activity [Dumais03][Robbins06]. Besides, submitted right keyword might efficiently retrieve right email information within one or two attempts. These two criteria are deliberately taken into accounts when designing an application interface.

3.1 Technology Overview

The thesis focuses on several technologies which fall into three general categories such as Visualization, search engine functionalities with Apache Lucence Core, and keyword extraction with Stanford Natural Language Processing. Figure 1 shows the overview of three technologies with their specific components involved. This chapter is a brief overview of the previous work on visualization, keyword extraction, and search engine functionalities.

Email messages are unlike journal articles, newspaper, scientific papers, books or web pages. Writing email syntax is very arbitrary because it's a private text, document, memo, and conversation. There is no grammar check and peer review. Therefore, in email context, extracting key term becomes more challenging task. We need to apply Natural Language Processing into our system. In order to retrieve properly important terms in email messages, it must involve several technologies such as Part-of-speech (POS) tagging, Stopword filtering, Dictionary check, and identifying collocation of words.

POS tagging: is described as a method to classify word into categories such as noun, adjective, verb, etc.

Stopword filtering: is used to filter all unnecessary words which are not considered as a keyword.

Dictionary check: since people often misspell the word in email message. We need to check the spelling and filter all meaningless words.

Indentifying collocation of words: After all words are POS tagged and checked, we need to identify whether a word is single word or compound words. Reason is key term extracted from message needs to be understandable for users so that they can remember email content.

Similar to other traditional search engine, email visualization system also consists of search engine functionalities for searching email information. However, implementing the entire search engine is beyond the scope of the application. We only apply some fundamental features of the open source search engine libraries provided Apache Lucene Core.

3.2 Email Visualization

Earlier researches [Kerr03][Frau05][Healey12][Sudarsky02] aim at identifying current problem of email usage to provide solutions with visualized interface. Most of the works fall into three categories:

- Thread-based visualization [Kerr03]
- Context-based visualization [Frau05][Healey12][Viégas06]
- Contact-based visualization [Sudarsky02]

Thread Arcs is typical thread-based visualization [Kerr03]. In Thread Arcs, email threads are transformed into graphics in order to describe context of conversation between people. Thread Arcs visualize email threads into chronological layout which helps to find emails within a particular conversation. It is believed that this chorological layout can improve quality of search results. Reason is mailbox act as a personal historical archive, thus people prefer looking up their history based on timeline.

In context-based visualization, key term or keyword describes the topic of email conversation. In contact-based visualization, contacts are people involved in that conversation. People are either the sender or recipient. Previous studies indicate the usefulness of visualizing keywords and contacts in email system [Viégas06] [Sudarsky02]. Both of them are very crucial attributes which help to remind users of their email information. These attributes are taken into consideration when designing the email search system.

Three visualization categories have their own pros and cons. However, they all have common discussions about visualization technique. Firstly, 2 crucial criteria including time ordering and memory triggering are all discussed in 3 cases. Secondly, a set of predefined keywords and contacts are very important attributes when user perform search query. In this thesis, the objective is to reducing the effort of typing query into search box. Reason is users often misspell keywords since they don’t exactly remember what they are looking. In order to do that, we

provide a set of visual predefined keywords and person names for users to select as search query. Another objective is providing users the ability to relate their email messages by visually presenting topics and email contacts.

3.2.1 Information Visualization

Dumais et al. designs an interface called *Phlat* effectively provides contextual cues to triggering human memory during search process [Dumais03]. However, the interface designed with textual information displayed on screen might decrease the perception of users and slow down user's responsiveness. By utilizing visual information, users can quickly perceive information and trigger memory more easily.

Information Visualization is the method of visually representing abstract data to reinforcing human perception. Sandra and Rune suggests visualization greatly supports re-finding information [Sudarsky02]. Firstly, a search query might return none or too many hits which impossible to trace back memory. Secondly, queries are often based on incomplete information which requires prior knowledge of sender, content, and date. Thirdly, email content often could not fit into a single screen. Thus, people really need to have some guideline or tooltip to support during search activities. These issues can be addressed by visualizing email information. Discussions of visual attention, memory triggering, and guided-search theory in visualized information show some mechanisms stimulating search process [Healey12]. Figure 2 presents

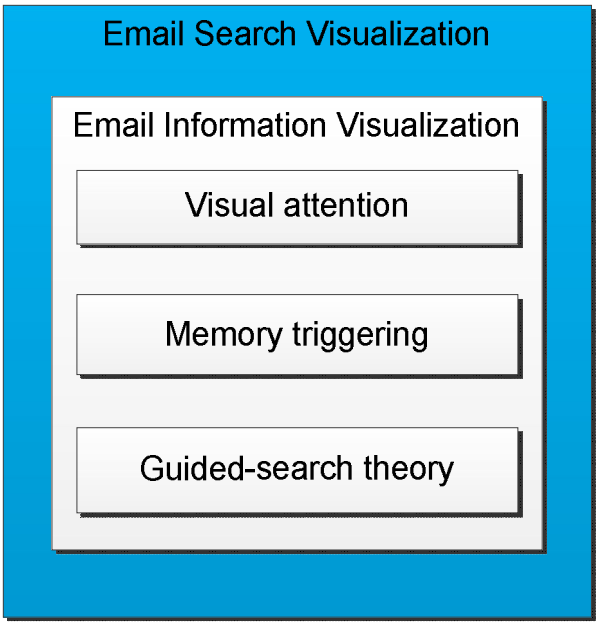

Figure 2. Fundamental features of Email Search Graph

several fundamental features are used in email search visualization system.

Visual attention [Healey12]: is the term used to denote mechanisms that determine which region of images are selected for more data processing. This is the emphasis on theories of attention by utilizing a variety of metaphors to triggering human natural perception. Proper choice of visual features and graphics apparently draw the focus of user attention to areas containing important information. Sender, recipient name, and date are most importation information which is worth attracting user attention. Instead of going through huge size of content of email, user attention are drawn to those separate attributes first allowing user to perceive email information. Through perception, user not only can collect a set of hints for further search queries, but also remember previous accessed information.

Memory triggering [Jovicic00]: an email always contains an event which is either personal or informational. In case of personal event, people, activity, and place are typically well remembered. While exact time is not described properly and needs to be searched for. In contrast to personal event, informational event is news information which has different characteristics and often lack of all contextual information. The only way to reminding people of the event is to perform keyword search. Keywords described here is topic related to sender, recipient, subject, date, and content of email body. Thus, memory revival greatly depends on the presence of the context surrounding the event. Visualized objects emphasized on user interface apparently allow people to remember the event. A visualized object is mapped to sender, time, and topic attribute of an email.

Guided-search theory [Wolfe89]: As more attention is drawn, the visualized object effectively guides users throughout steps to help completing the search task. The guiding is based on contextual cues regarding events. Contextual cueing is driven by beneficial interaction between memory and spatial attention. For instance, the application interface provides keywords describing email content. Then keywords are visualized into graphics in relationship with sender and timeline. Visualized keywords attract user attention, remind them of email information and guide them through the entire search activity.

Effectively showing overview of large dataset can reveal unknown patterns on email data. In the study with information visualization, there is always partial piece of information is missing from

user memory. By drawing attention and guiding people throughout search activity, memory related to email events can be revived, thus allowing search task to be completed rapidly.

3.2.2 Role of memory in email search

Previous sections emphasize the important role of memory play in PIM [Elsweiler11]. People tend successfully retrieve their information based on what they remember. Some studies also shows people actually adapt to email management strategy in order to compensate their memory limitation [Sudarsky02][Bellotti05]. Problems were people could not precisely remember the whole email context despite of whether the email contains personal or informational events [Jovicic00]. In order to solve memory issue, our application needs to focus on the following perspectives:

- Time elapsed since last information access
- Type of task inside emails
- Users involved inside email conversation

Time: Nature of memory is transient and its quality degrades overtime. Prior literature discusses abundantly about episodic memory [Baillie08]. Episodic memory is described as a separate memory that store information in episodes. Episodes include information such as location of events, person involved, others events occur before and after. Purpose of episodic memory is to study how much information people actually remember when time goes by. Therefore, when designing email re-finding system, time is importantly taken into consideration.

Task: Users are unable to type in the keywords because they don't precisely remember what they are looking for. Keywords are seen as partial information involved which also belong to temporal memory. In order to reinforce temporal memory to be restored, informational objects regarding email information need to be discovered by users. Tasks or discussions involved are informational objects. Users often look for emails based on temporal memory of tasks. By visually presenting partial information of these tasks or topics, it not only assists users remember full information of tasks but also allow people to remember when and how tasks were performed.

User: In addition to task, remembering different group of users involved in the email conversation also improves quality of memory. Users might be a sender or people in recipient group. They also act as important information that helps triggering human memory.

Three attributes above have been discussed to conclude the design of our implementation. Email Search Visualization is the name of the implementation. It will emphasize on time as a main principle, then tasks and users involved as secondary principles. Logically, our design becomes organization of features comprising of timeline, people, and keywords.

Organization of timeline

Temporal cycles are considered as the most important aspects of our implementation. In particular, time period are distinguished by column and color to provide easy temporal reference. Messages are placed throughout each time period indicating their ordered by time. Since email context are gradually lost over time and dating errors consequently increase, month level might be fine for older messages.

Organization of users

It is conceivable that large mailbox might contain a huge list of contacts which is very difficult to manage. Allowing users to glimpse over a collection of contacts based on relevancy might alleviate the problem. Contacts are in relevancy order but not alphabet order because it would benefit users during searching. Users only expect to see some relevant people related to search query.

Relationship between users and messages also is taken into account when designing the application. Additionally, type of users should be indicated clearly to isolate senders and recipients.

Organization of keywords

Topical words describing messages are crucial to trigger user's memory. A set of predefined keywords displayed in relevancy order extremely play important role in information re-finding process. With that, people no longer become frustrating to try performing correct query keywords.

Relationship between keywords and messages should also be shown in order to allow people to remember each message's topic.

3.3 Keywords extraction with Stanford Natural Language Processing

In section 2.2, we had a very brief discussion about the importance of different attributes including topical terms and contact names. This section, we will discuss the technique of how terms can be extracted from the full text of emails. Key term is a pivotal factor that affects the whole searching task. Instead of forcing users manually typing query in the search box, recommending a set of predefined keyword is a very thoughtful strategy. It not only eliminates time of typing keywords but also reminds people of the topic of email body. When memory is revived from reading those topical words, people can easily find what they are looking for within very short period of time. In order to extracting meaningful terms from text, we need to carefully apply Natural Language Processing approach. Figure 3 shows components involved in Natural Language processing

3.3.1 Natural Language Processing Definition

Natural Language Processing is the research area of how computers can be used to manipulate natural language text or speech [Chowdhury03]. Goals of NLP is creating artificial intelligent for computers to do linguistics analysis, and achieve human-like language processing [Chowdhury03]. NLP helps computers to extract words from a given text that have specific meaning or connotation describing email context. These words are called key terms or keywords.

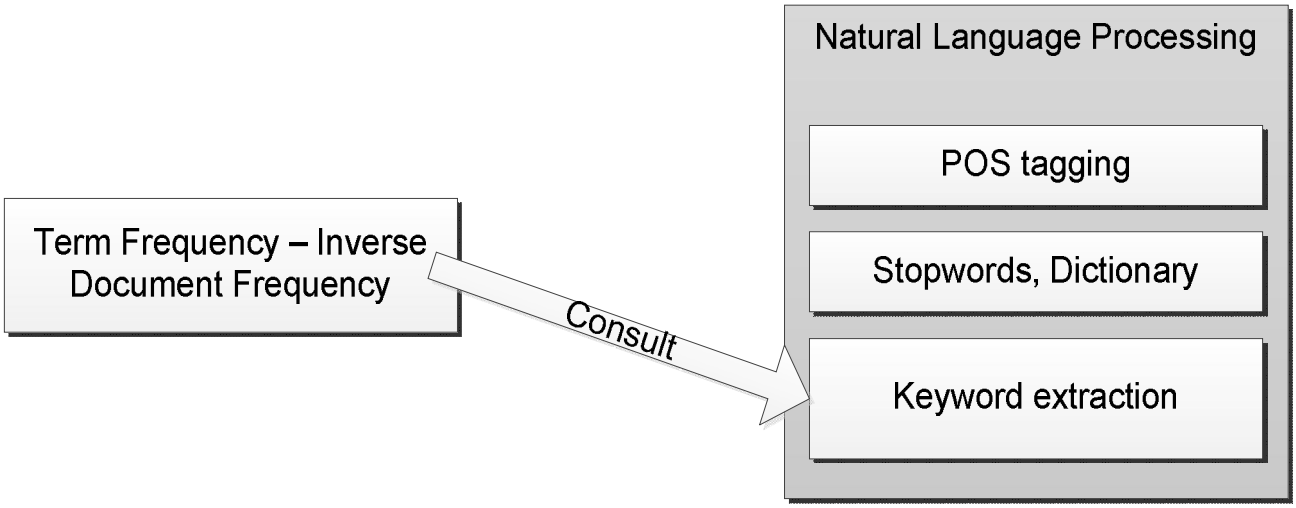

Figure 3. Components included in Stanford Natural Language processing

By looking at those key terms, memory about certain email context can be recovered, thus allowing people to retrieve email information faster. Liddy [Liddy98] distinguished seven different levels of linguistics analysis that people can use to extract meaning words:

- Phonological: this level interprets speech sounds across words. To be specific, it deals with the variations of pronunciation within a sentence.
- Morphological: deals with smallest components of words or morphemes. Morphemes include prefix, suffix, and the root itself. For instance, the word “preannounced” can be broken into 3 morphemes: “pre” is a prefix, “announce” is a root, and “ed” is a suffix.
- Lexical: interprets the meaning of individual words. This level process word-level understanding and assign property to a single word or parts of speech.
- Syntactic: analyzes words to discover grammatical structure of a sentence. This level requires both grammar and parser. The output of this processing is revealing the structural dependency between words within a sentence. For instance, adjective, noun, verb, adverb, pronoun, and so on are grammatical components used within a sentence.
- Semantic: produces possible meaning of the whole sentence by analyzing interactions between the meanings of words. For instance, a word alone has many meanings, but it has one specific meaning when it stays within a sentence. By analyzing a word within a sentence, semantic level can retrieve both correctly meanings of word and sentence.
- Discourse: uses document structures to deal with text longer than a sentence. Text can be understood as concatenated sentences having a single property or meaning.
- Pragmatic: deals with knowledge from outside content of a document. It requires world knowledge in order to explain extra meaning of a text.

Most current NLP systems only implement modules that satisfy lower level of language processing. The reason is higher levels are unnecessary to be implemented and dealing with smaller units within lower levels is easier and more effective.

For identifying and extracting terms from texts, previous work only focuses on 4 lower-levels of linguistics processing [Daille03]. Those levels are just sufficient for identifying specific key terms with reliable syntactic analysis. In our email search system, we also simply use those 4 levels.

3.3.2 Algorithm for extracting keywords

Key terms extraction or automatic term recognition is the important task in information retrieval research. Reasons are terms related to document content acts as short descriptions. Searching activities become very easy going tasks because people know what they are looking for. In email information retrieval case, emails are personal memory of users and short description acts like memory triggering unit. It does not only allow people to know what they are looking for but also reminds them of the whole email content. Recovering users' memory affects search process very much because people immediately know where to get that forgotten information.

For email information, to users, all key terms are important even with just a single noun. Thus, extracting key terms from email information requires retrieving all relevant words. Prior section indicates the importance of 4 lower levels of linguistics processing, thus we will apply them into our strategy. Our generic strategy includes POS tagging, stopwords removal, dictionary check, and syntactic check. POS tagging maps to phonological and morphological, stopwords removal and dictionary maps to lexical level, and syntactic check maps to syntactic level.

POS tagging:

Most of NLP systems demand POS tagging which is the initial stage prior to elaborate complex data syntax analysis. Generally, POS tagging is gateway to successful text analysis, thus it must be accurate and efficient. POS tagging or chunking is the series of processes for identifying proper trunks or words from a set of tokens [Kudo01]. Words retrieved from the identification process are classified into some grammatical classes such as noun, pronoun, adjective, verb, adverb, etc. In this thesis,

I will use Stanford POS tagging algorithm. Stanford POS tagger is the API adopted from the Supports Vendor Machine (SVM) tool [Diab04]. SVM is a prominent tool is intended to comply with all requirements of modern NLP technology [Giménez04]. It includes properties including simplicity, flexibility, robustness, portability, accuracy, and efficiency that NLP researchers recommend for [Giménez04]. Tagger is very easy to use, accepting standard pipelining, easy configuration file with very few parameters. The following steps are how Stanford POS tagging process works with email information.

- Strip all html tags from email body text

Number	Tag	Description	Number	Tag	Description
1.	CC	Coordinating conjunction	19.	PRP$	Possessive pronoun
2.	CD	Cardinal number	20.	RB	Adverb
3.	DT	Determiner	21.	RBR	Adverb, comparative
4.	EX	Existential *there*	22.	RBS	Adverb, superlative
5.	FW	Foreign word	23.	RP	Particle
6.	IN	Preposition or subordinating conjunction	24.	SYM	Symbol
7.	JJ	Adjective	25.	TO	*to*
8.	JJR	Adjective, comparative	26.	UH	Interjection
9.	JJS	Adjective, superlative	27.	VB	Verb, base form
10.	LS	List item marker	28.	VBD	Verb, past tense
11.	MD	Modal	29.	VBG	Verb, gerund or present participle
12.	NN	Noun, singular or mass	30.	VBN	Verb, past participle
13.	NNS	Noun, plural	31.	VBP	Verb, non-3rd person singular present
14.	NNP	Proper noun, singular	32.	VBZ	Verb, 3rd person singular present
15.	NNPS	Proper noun, plural	33.	WDT	Wh-determiner
16.	PDT	Predeterminer	34.	WP	Wh-pronoun
17.	POS	Possessive ending	35.	WP$	Possessive wh-pronoun
18.	PRP	Personal pronoun	36.	WRB	Wh-adverb

Figure 4. Alphabetical list of part-of-speech tags used in the Penn Treebank Project [Penn Treebank]

- Lowercase all words in the text
- Break text into a series of single words
- Tag each every word with grammatical classes

The result is we get a series of individual words with their own POS tag or grammatical classes. Figure 4 shows all possible tagging for words from standard Penn Treebank POS tags.

Stopwords removal:

Stopword list is described as a negative dictionary which is the device filters unnecessary poor meaningful terms for indexing [Fox89]. Stop words from the list also are never used for search query. First reason is stop words are most frequently occurring words in English. Thus, one search query with stop words return so many hits, people will avoid that. Secondly, stop words have no meanings, and users never use them in search query. Stopword list includes the following words [Fox89].

- Preposition: of, from, at, on, in, etc.
- Pronoun: the, he, this, those, them, etc.
- Question words: where, what, when, etc.
- Most frequently occurring words in English: go, have, like, would, keep, etc.
- Similar words with or different versions of most frequent occurring words: went, had, gone, liked, etc.
- All single letter words: from a to z.
- All above words with prefix and suffix.

In the case of email information, we need to add some words occurring frequently in emails to the stopword. They are very arbitrary and needs to be manually added. Some of them are “dear, regards, sincerely yours, etc”.

Dictionary check:

Dictionary check is in lexical level of linguistic processing. After stop words have been removed, the meaning of words needs to be interpreted. Since the resource of our research is English mailbox, therefore it is mandatory to check whether words extracting from text are English. By using Java Spell Checker, eventually those non-English words are removed from the list.

Syntactic check:

Until this stage, the list of words has been updated and waits for last processing. As the name implied for this step, syntactic check is responsible to check relationship and connection between words. If words are connected grammatically, they are combined into a compound terms or keywords. Figure 5 shows pseudo code of how the compound terms or keywords are formed.

```
Loop through list of terms
 If term is Noun or Adjective
   If next term and next next term are Noun or Adjective
     keyword = term + next term + next next term
   Else if next term is Noun or Adjective
     keyword = term + next term
   Esle
     keywrod = term
 Else term is not Noun or Adjective
   Does nothing
```

Figure 5. Pseudo code to extract keywords from a list of broken terms

Prior work suggests that most of search query fall into a single or double word query, only very few queries

performed with 3 words [Whittaker11]. Therefore, our algorithm focuses on extracting the keyword with less than 3 words. Figure 5 indicates that only the collation of adjective and noun can be called a compound word. In email context, a compound word might identify which topic email discuss about. According to prior work, when users remember the email topic, search activities will be easy and fast [Baillie08]. However, problems arise when so many topical words or keywords need to be displayed on screen. We need to choose which set of keywords are most relevant to users. Apache Lucene search engine is the potential solution to deal with such issue.

3.4 A discussion on search engine

Search engine is an application component comprising of functions that enable effective retrieval of text documents in database [Milosavljevic10]. Data structure was generated in order to facilitate the search process. That data structure is called an index, and the process of creating an index is called indexing. Document's content is parsed into single words or tokens. Tokens are indexed within search engine for purpose of searching by words. All information management system relies on one core component which is search engine, and our system is no exception. There are many search engines used for indexing structured documents such as Zebra [Zebra], Dspace [Tansley03], etc. However, for our information retrieval, we use Apache Lucene which is the open source search engine library. There are many reasons we choose Lucene. First, the system's platform is web application which requires server side script is Java, and Lucene is written entirely in Java. Secondly, Apache Lucene is open source, full-text, high performance, cross-platform search engine library that supports very well for an email retrieval system. Figure 6 describes three crucial components which we utilize in the thesis.

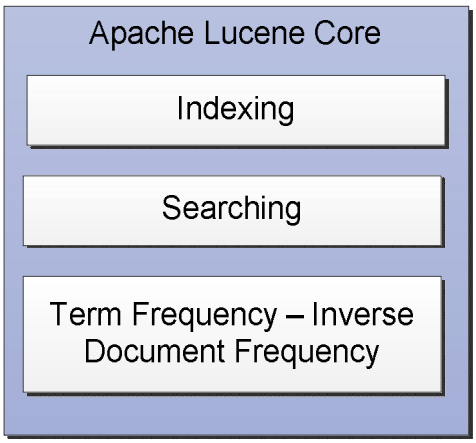

Figure 6. Components included in Apache Lucene Core

3.4.1 Apache Lucene Overview

Apache Lucene is a software package for full text-based information retrieval [Lucene]. It is simple and powerful because it does not require any knowledge of indexing and retrieval functions. Lucene provides an API that alleviates complexity of indexing and searching, thus enable programmers to focus on functionality of the application. An API is a set of routines, object classes, structures, and protocols that support building an application. Lucene relies on the following fundamental concepts.

- *Fields*: includes name and unstructured textual content. Each field corresponds to a piece of data that is retrieved from index during search. A field in our email retrieval system is an email attribute such as subject, senders, recipients, email body, and date. Table 1 shows an example of Index Fields of email retrieval system.
- *Documents*: contains a collection of fields. A document can be considered as an entire email.
- *Analyzer*: used to extract tokens or words out of textual content for purpose of indexing
- *Index*: A set of documents are stored by data structures to provide efficient information retrieval.

Field	Lucene Type	Description
Subject	Field.Text Field.Store	Full subject extracted from email String format
Sender	Field.Text Field.Store	Sender from email String format, only get name of email address
Recipients	Field.Text Field.Store	A collection of recipients from email String format, only get name of email address
Msgid	Field.Int Field.Store	Msgid from email Integer format
Date	Field.Text Field.Store	Date received email. Format as String YYYY MM DD

Content	Field.Text Field.Store Field.StoreTermVector Field.StoreTermVectorOffset	Full text of email content body String format, delete all html tags in email content

Table 1. An Example of Index Lucene Field

Indexing Process:

Indexing process includes 3 main steps:

- Text analysis
- Conversion of original email information into Lucene documents
- Index creation

Text analysis refers to the operation of parsing field content into a stream of tokens. Tokens are single words that already went through the process of lowercase transformation and stop word removal. Document conversion represents the process of converting email information into Lucence document. During index creation, appropriate index data structures are updated in database to facilitate searching process.

Searching Process:

Searching process comprises of the following order such as query formulation, query execution, and results retrieval. Queries are formulated programmatically by using several Lucene query types. Query execution uses previous built index. Retrieved results are ordered based on relevancy scoring mechanism. Figure 7 shows the fomula for the basic of calculating score of document. Lucene API supports several types of query as follows [Lucene]:

- *TermQuery:* a simplest type of query includes only one term in the query. The query requires 2 elements, a field and a term or keyword.
- *WildcardQuery:* Search for contents expressed with wildcard characters. Wildcard characters are either (*) - zero or many and (?) – one or many.
- *RangeQuery:* facilitate searching from starting term to ending term.

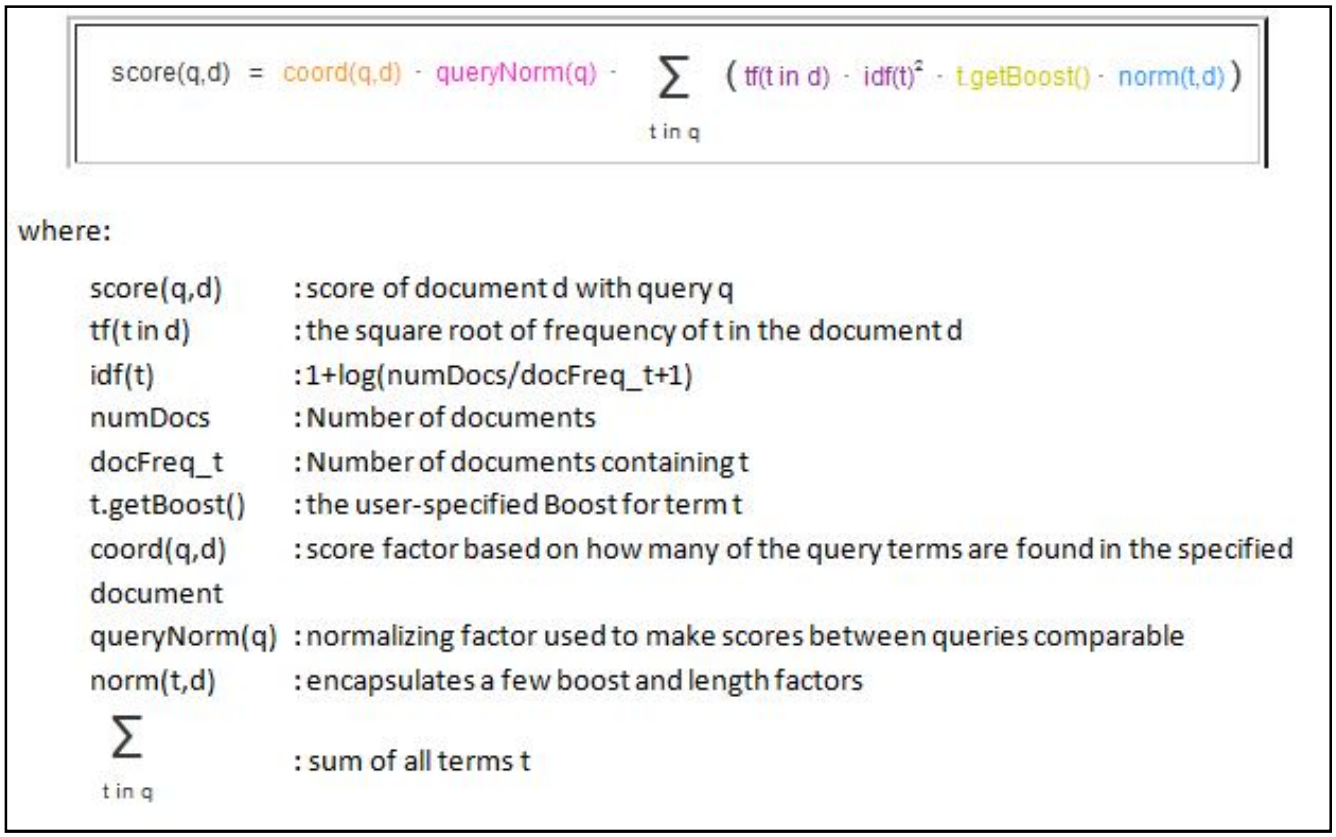

Figure 7. fomula used by Lucene to calculate score of documents [Lucene]

- *BooleanQuery:* Various types of query can be combined into a single query with Boolean expression.
- *PhraseQuery:* similar to TermQuery but allow multiple terms or a phrase to be matched in the query.
- *FuzzyQuery:* matches term similar to specified term.
- *SpanQuery:* a combination of several TermQuery, matched documents need to contain all terms regardless of their positions.

In email retrieval system, SpanQuery is the most appropriate querying method. The reason is people often search for an email contains multiple keywords regardless of their position. Another reason is SpanQuery provide MUST operation which implies all keywords in query must exist in email message.

3.4.2 Term Frequency and Inverse Document Frequency

Previous section of extracting keywords from documents concludes a set of key terms which is used to describe topics of emails. However, the number of key terms might become huge and users will be confused with large volume of information. It is very challenging because the efficiency of email retrieval system greatly depends on how to display information that triggers users' memory. In order to overcome this issue, the system should have capability to rank those predefined key terms based on their relevancy within mailbox. For ranking words based on documents, term frequency – inverse document frequency (tf-idf) is the prominent weighting scheme in today's information retrieval system. Essentially, tf-idf works by comparing 2 parts:

- The relative frequency of a particular word inside a specific document
- Inverse proportion of that word over the entire document corpus.

Intuitively, tf-idf calculation determines the relevancy or ranking of a word within a collection of documents. Words appear inside single or small group of documents will have higher relevance than words that appear in many documents, for instance prepositions, articles. The basic equation of tf-idf is as follows.

$$tf\text{-}idf = tf * idf$$

with

$$tf(t \text{ in } d) = (t \text{ frequency in } d)^{1/2}$$

$$idf(t) = 1 + \log (numDocs/docFreq+1)$$

where

t: term

d:document

numDocs: number of documents

docFreq: number of documents contain a term

This is the traditional method of implementing tf-idf in all information retrieval system. We will apply this equation to our list of predefined keywords. Result of this method is a collection of

terms describing email information ordered by relevancy. The most relevant keywords will be visualized onscreen in order to allow users remember the whole email context. Eventually it makes searching process easier and faster because users already knew how and where to get an email that they are searching for.

CHAPTER 4 – PROTOTYPE AND DESIGN

4.1 Prototyping

Key success of the system was reliance on our iterative design process. With design effort from Baris Serim of Helsinki Institute for Information Technology (HIIT), we were able to propose a series of candidate solutions for email search system design. We created many prototypes to capture the intended design aesthetic and evaluate weaknesses of the layout. Each one was built based on the preceding attempt. Several criteria were focused carefully during iterative design such as the ease of use, color scheme, interaction design, and graphical layout. Throughout the process, we were able to capture negative design factors and refine the interface in order to release better quality email search system. The following sub-sections describe each prototype in detail including design description, advantages and drawbacks.

4.1.1 First Prototype

The layout of first version is divided into four isolated semi-circle areas. Figure 8 shows the look of this version. First area is query area where users can drag and drop keywords into it to perform new search. Second area contains a collection of draggable keywords. Third area

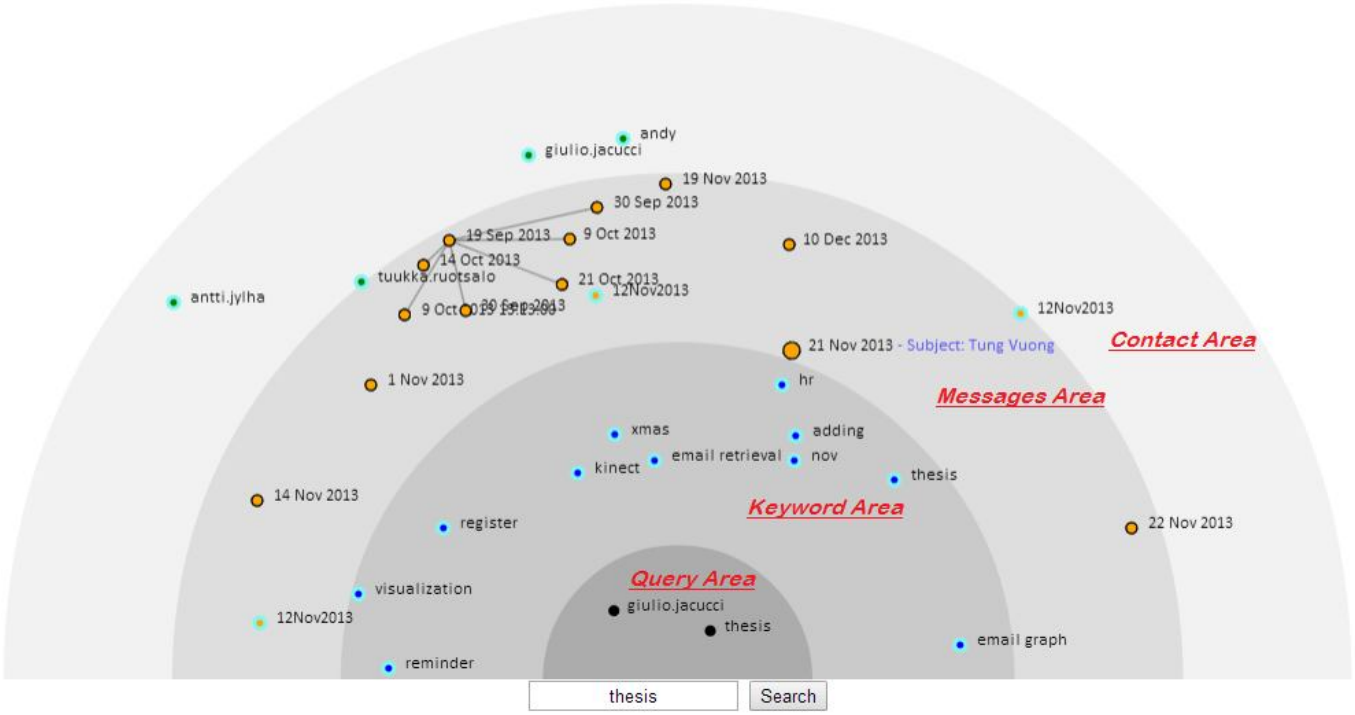

Figure 8. An example search result of First Prototype

contains messages resulted from search query. Last area is contact area containing related persons in search result. There are several types of interaction in this design.

- Users can double on each node on the screen to see the relationship between them. For instance, messages are grouped together belongs to the same thread or a conversation. A person in contact area connects messages because that person is either a sender or a recipient. A keyword connects to messages because messages contain that keyword.
- Drag-drop keywords and persons into query area to perform new search. Additionally users can also remove keywords from query area as well.
- Double click on message nodes to open full detail of messages.

Drawback of this design is messy look of nodes that confuses users. Even each node type has different color scheme but if search query returns too many hits then people can't find correct information. Additionally, when we expand all relationships, the interface becomes even more confusing.

4.1.2 Second Prototype

First prototype's weakness is messy look of the visualized objects' positions. Thus, we alleviate the problem by arranging them into a time pie. The pie indicates timeline with each portion or piece is a month period. Figure 9 shows how the interface of second prototype looks like. The

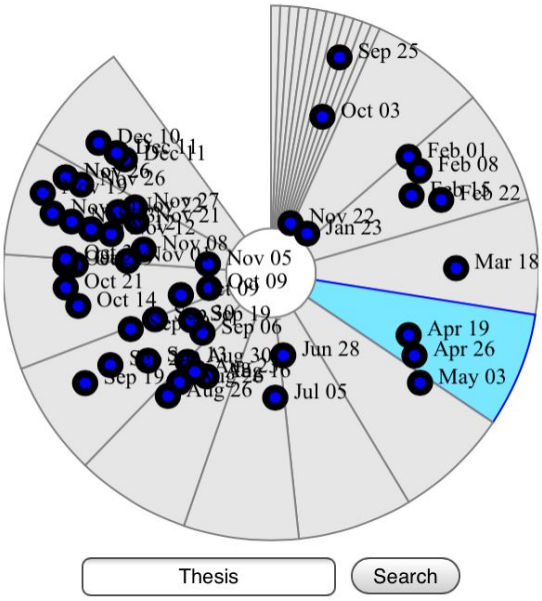

Figure 9. An example search result of Second Prototype

search query in the Figure is “thesis”, and all messages contain the keyword are displayed on the pie. The layout is organized in the following logic.

- Timeline order is in clockwise manner. When a year contains many messages matching keyword, then it will have portion’s size bigger than a year that contains fewer messages.
- The closest message to the center is the most relevant one.
- Users can select, expand each piece to display related messages.

In this version, the design tried to alleviate the issue of randomly organizing messages in the layout. Therefore, message nodes are now arranged in 2-dimensional grid with time and relevance dimensions. However, this design still could not satisfy our requirements. Firstly, the pie has only 360 degree angles which produce very limited space for positioning message nodes. Each piece of pie will become smaller when search result returns too many hits over many years. Hence, users are unable to select correct message information and the search process fails. Secondly, it is impossible to arrange predefined keywords and person names on the pie because keywords are treated based on relevancy not timeline.

4.2.3 Third Prototype

Third version eradicates layout weaknesses of two previous prototypes. The interface is still two-dimensional grid but it acts like a chart. Figure 10 depicts the example search result of Third prototype. X axis presents time period whereas Y axis presents relevance of messages. Since previous prototype we indicate the problem of keyword’s arrangement, thus we isolate keywords and persons into two bottom areas. Some additional interaction is added as follows.

- Users can select each time column to display corresponding messages’ contents
- Users can expand or zoom in each year column into columns of month and month column into columns of day.
- Keywords persons can be selected to see the relationship with messages.
- Users can drag keywords into search area indicated as plus sign to perform new search.
- Additional feature such keyword deletion is a plus, users to delete a keyword by dragging it into pink area at the bottom. New keyword will be retrieved to replace deleted one.

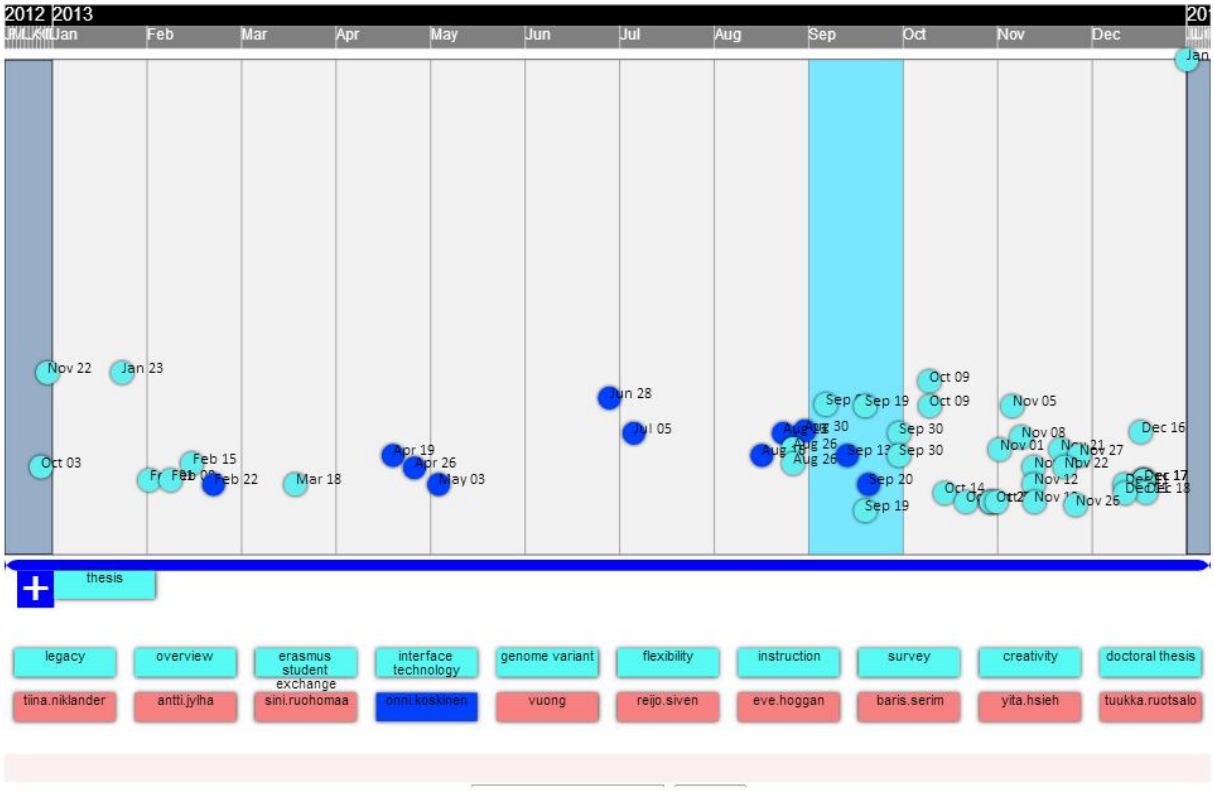

Figure 10. An example search result of Third Prototype

This prototype is considered as an acceptable version because layout has three separated areas of keyword, messages and contact person. However 2-dimensional grid with x and y axis does not comfort the initial requirements. The reason is email search engine is different from traditional search engine. People judge all messages equally because each of them might contain right information. Thus, y axis of relevance is not necessary in email context. Moreover, interaction in the layout also does not convince design team in the first place.

In conclusion, with three previous prototypes, we would be able to capture their characteristics and limitations. In order to create the user-friendly interface that effectively helps people to find their emails; we improved the design layout to create new version. Current version is defined as time-based layout with more graphics, color scheme and interaction behaviors. Next section, we will have more fruitful discussion about features included in the new version. Eventually, it seems to be a potential design candidate for email search application. However, we still continue to improve our design in order to adapt users' requirements.

4.2 Design Principles in Chronological Layout

In previous discussion, we already indicated the importance of memory and timeline that greatly affect email re-finding process. Additionally, three basic principles of design are time, person, and keyword have been discussed very well. In this section, we will have very deep analysis about these three principles and how interaction between them can improve the quality of searching. In order to provide an easy look and feel and to avoid scrolling behavior, all three principles will be organized in chronological manner.

Moreover, email search engine is not completely similar to traditional search engine. People want to see most recent emails in their mailbox rather than seeing nothing at all. The reason is email act as personal history to users whereas other search engine such as Google is considered as a public library for people to look up information. Therefore, initial view of the layout needs to be shown at first sight in order to benefit memory revival.

Furthermore, the main objective of Email Search Visualization is eliminating time-consuming process of typing keywords in search box. Specifically, we provide predefined keywords as a reference for email messages. Additionally, visual interaction is added to allow users to quickly retrieve their wanted information. Interaction behavior mainly occurs among keywords, person and messages. To support interaction more effectively, suggestions based on timeline are added to provide episodic cues.

4.2.1 Timeline as a main design theme

So far, our discussion concluded that our layout is based entirely on timeline. In section 3.5, we discussed that the application layout design based on time period. A time period is represented as a column. Messages received within each time period will be positioned in a corresponding column. Messages are visual nodes that align vertically on each column. Figure 11 shows the design of the application with initial view containing most recent email messages in chronological layout. By utilizing this new layout, eventually the interface can avoid scrolling behavior which consumes too much time.

In our design, four different areas present four different features and functionalities. The following is brief description of each area.

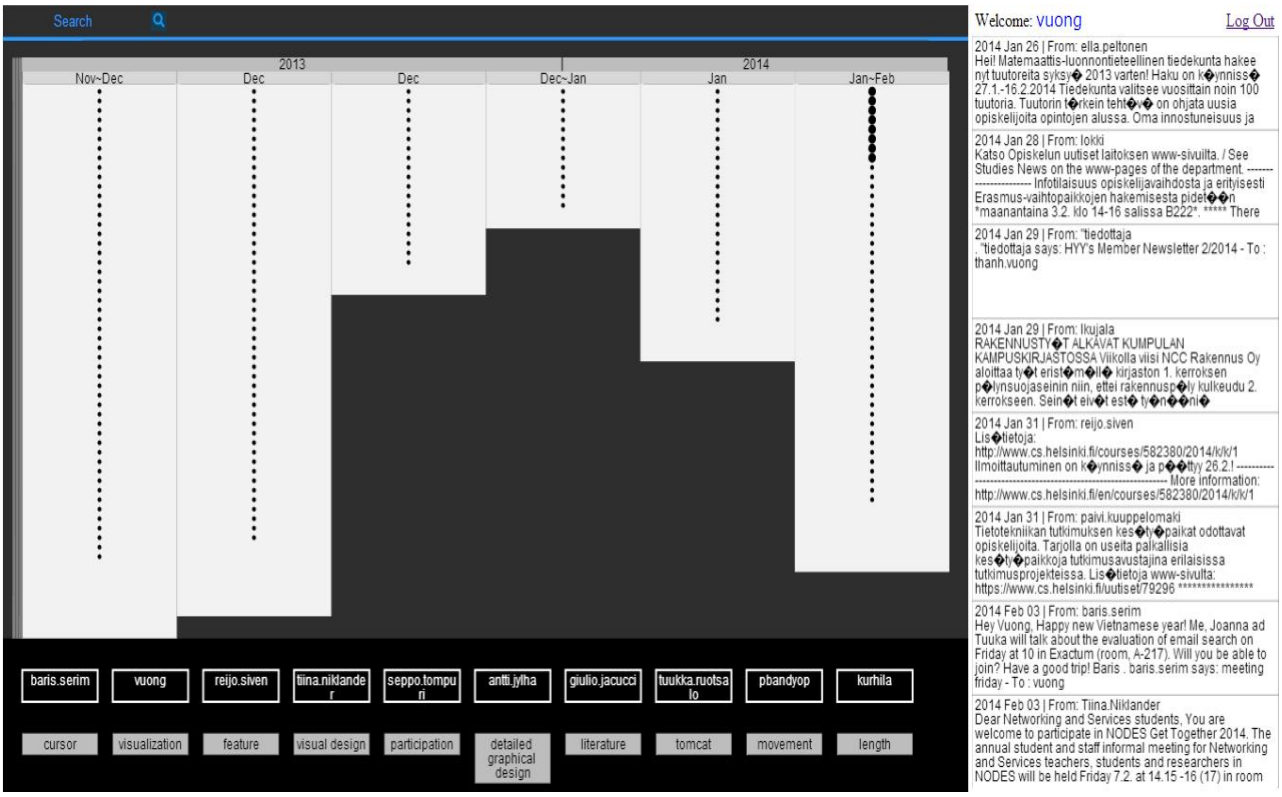

Figure 11. The example of initial view of Email Search Graph as chronological layout

Search Area

Figure 12 presents search query area. Search query area allows traditional search engine functionality. A suggested element is displayed on the right side when it is being selected, clicked and dragged. There are two operations that can be executed by users.

- Users choose to manually type keywords into search box and hit the button for new search result. This operation is not encouraged unless there is no other option. Reason is users might misspell the words or words do not exist in entire mailbox. Search query for this operation often returns too many hits or no hits at all.

Figure 12. Search Query Area

- Drag and drop predefined keywords and person name from the bottom of the interface for new search query. This is most reliable operation that we want users to perform. Reasons are keywords and person names are extracted directly from email body and they are ranked based on relevancy. Hence, search result always returns an adequate amount of matched messages that users can look up their information.

Main Area

Figure 13 shows the look of main area. Main area contains all visual attention and interactions of the layout. It is designed as chronological layout comprising of 6 different columns. A column represents a past time period from mailbox. Each column contains a fixed set of messages received within corresponding time period. Message nodes are positioned vertically and ordered by ascending time from top to bottom.

Most of interactions occur in main area including suggested keywords interaction, suggested person name interaction, and selection of messages. Message node is being selected will have bigger size compared to others. By default, all message nodes are black. If a message node is

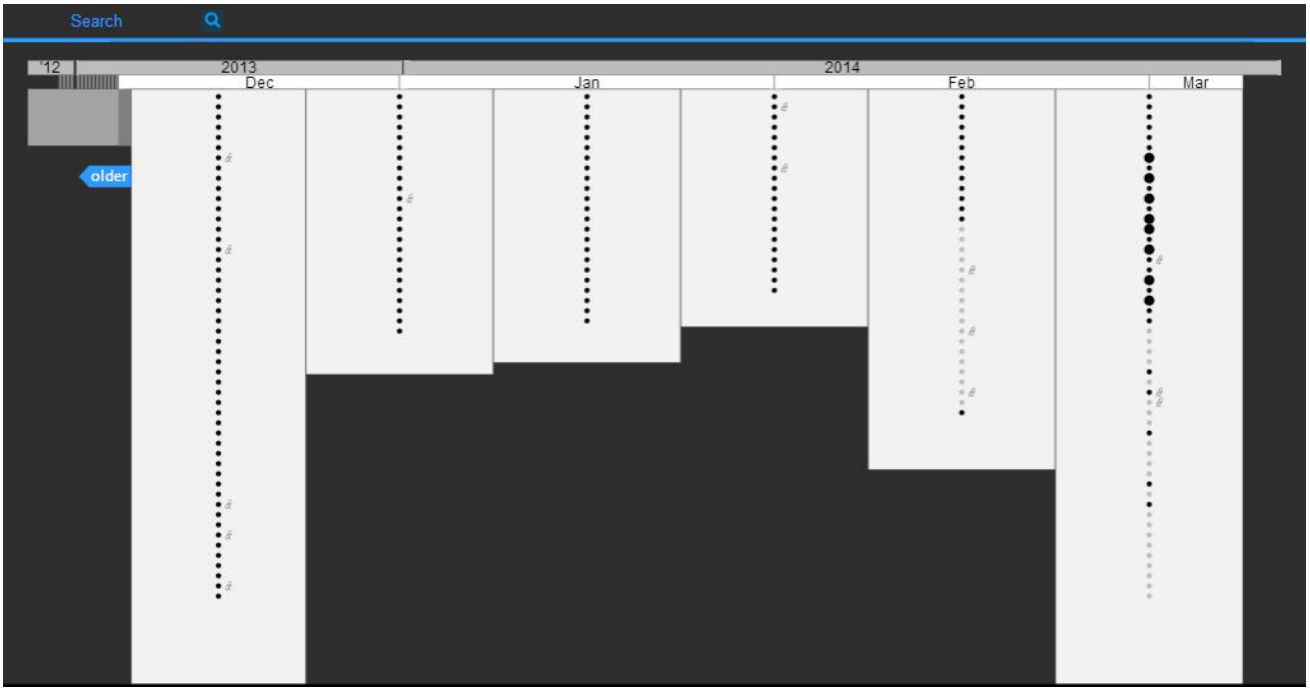

Figure 13. Main Area

selected and read by users once, its color will be changed into slight gray. This is the purpose to increase user perception toward which emails have been read and which have not.

There is also some expandable area marked with “Older” and “Newer” label. Those areas allow users to search emails based on specific timeframe. For instance in Figure 13, by clicking on the area highlighted “Older” and selecting specific time period, result will be a new search with time period before Dec 2013.

Suggestion Area

Figure 14 depicts suggestion area. Suggested elements including keywords and person names are organized in a meaningful way in this area. Left most element is the most relevant one compared to others. Users can drag them into either main area to ignite the interaction or into search query to perform new search query. Users can also double click on multiple suggested elements to see the relationship between multiple keywords and message nodes.

This area contains precious information that has capability to revive user memory toward messages’ content. Each element either describes exactly email topic discussed or person involved in email conversation. The moment users understand those topical words and related persons, searching process will become fast and easy.

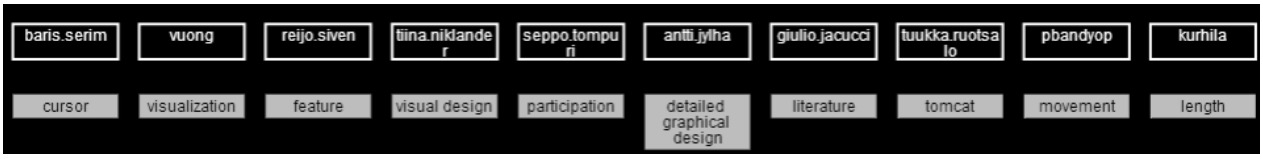

Figure 14. Suggestion Area

Figure 15. A example of Details of message Area

Details of message Area

Figure 15 show an example of details of message area. Details of message Area displayed selected messaged node on main area. Details include brief description of email information such as sender, date, subject, and short content. Users can double click on each row to expand full email information.

4.2.2 Suggestions based on timeframes to provide episodic cues

Contextual information provides rich support during cognitive and perceptual process. It removes ambiguity from visualized information perception and memory. In this implementation, date, keywords and persons provide rich contextual cues for retrieval because information has been accessed before.

- Person's name plays an important role in email search activities. Prior research indicates that 25% of email search queries person's names [Dumais03]. Therefore, suggestions on person's name are very powerful memory cues for retrieval personal content.

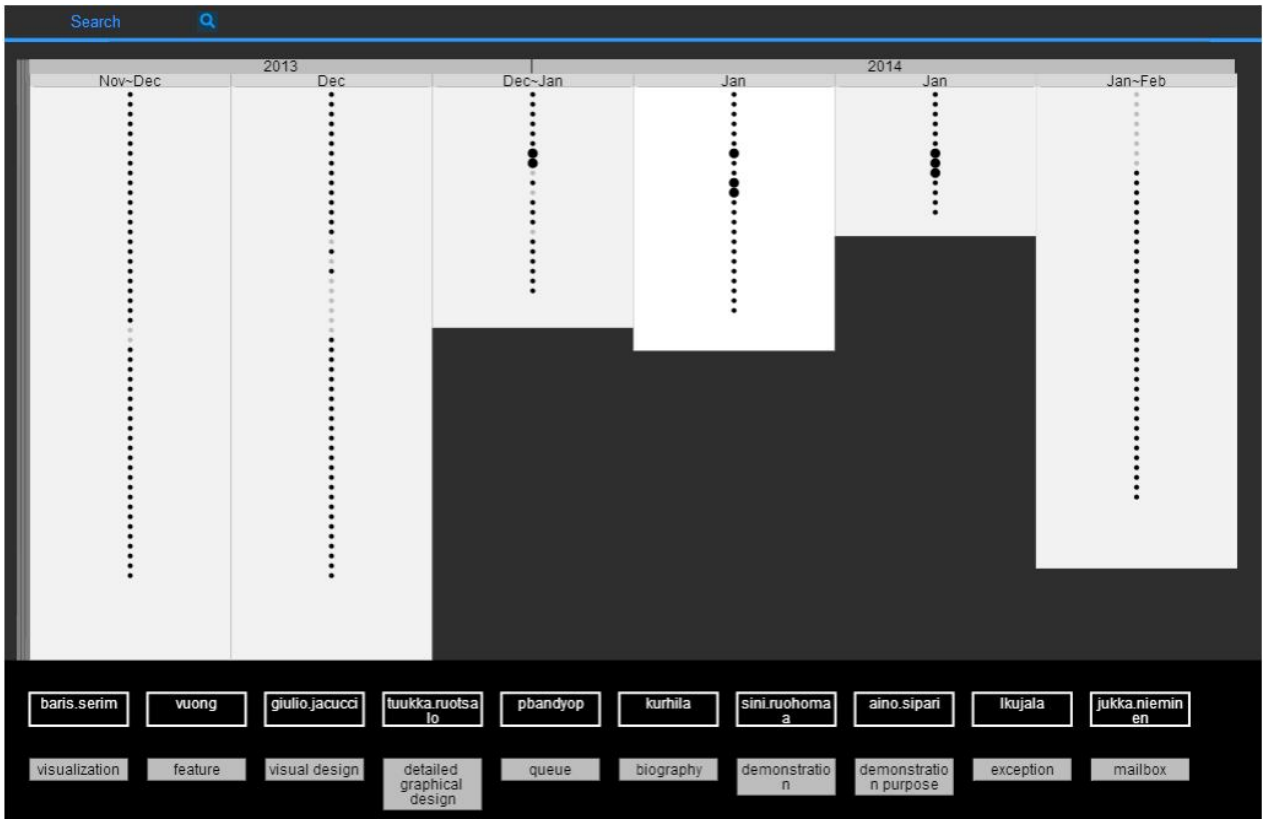

Figure 16. Suggestions based on timeframes with Common Selection

- Additionally, associative cues in email content highly increase quality of search activities. These cues present topics discussed in emails. They are suggested keywords that remind people of a conversation that they took part in.

In order to provide episodic cues based on timeframes, we propose filtered selection with low interaction cost to select relevant items in the visualization. Filtered selection is similar to common selection but it only highlights message nodes that can fulfill filtering criteria. The action can be achieved as following methods.

– Common selection: simply click on each timeframe or column to receive corresponding cues or suggestions at the bottom. Those suggestions are collected based on the timeframe and they ranked based on their relevancy within the mailbox. Figure 16 shows how suggestions based on timeframes to provide episodic cues and improve the quality of searching process. As you can see in the Figure 16, when users select a timeframe in main area, suggestion list is automatically updated to follow the interaction. The purpose

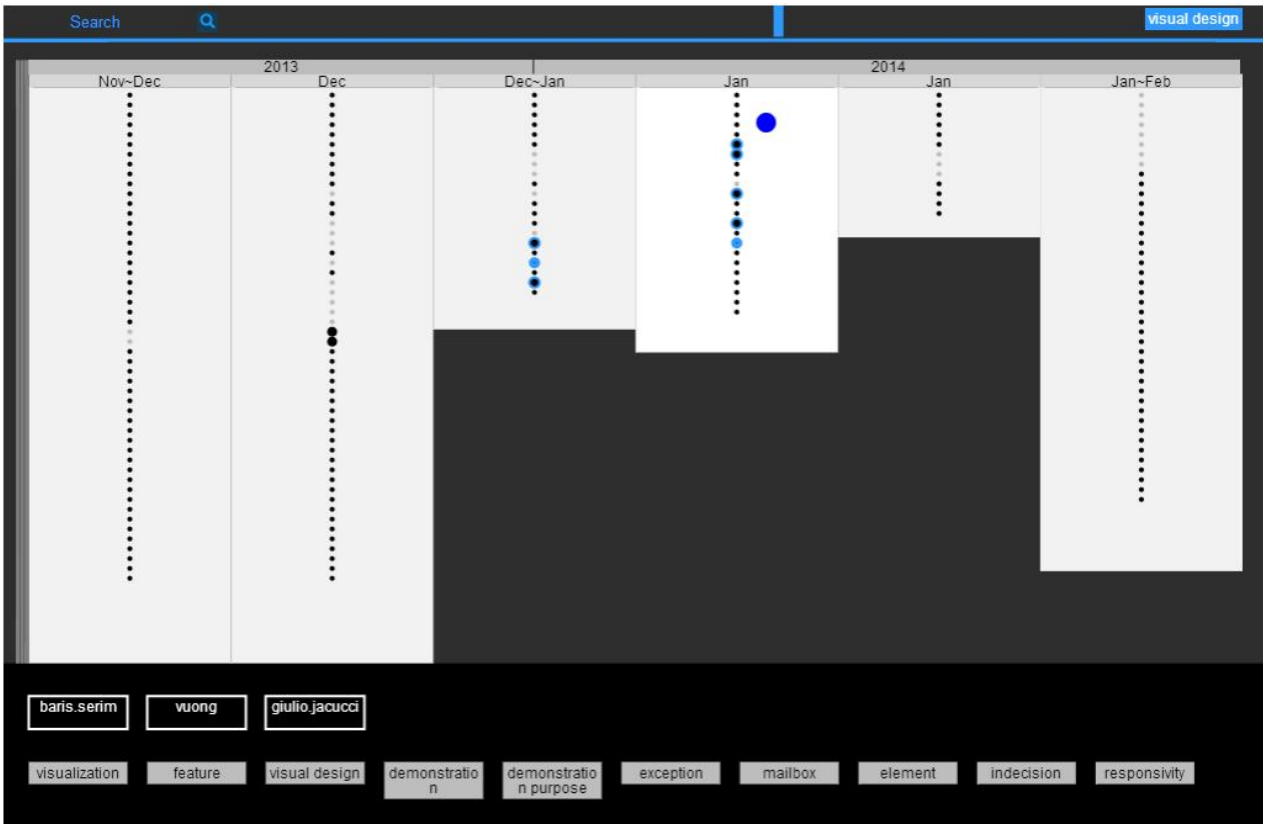

Figure 17. Suggestions based on timeframes with Keyword Filtered Selection

of this selection is providing good summary of all conversations within a specific timeframe. For instance, by clicking on column “January”, users can observe topical words, related persons at the bottom describing all conversations that they had within January.

– Filtered selection with keywords: clicking on a suggested keyword, users will see highlighted messages that contain the suggested term. Dragging that keyword into a timeframe, users also are able to receive a new set of suggested keywords and persons regarding each timeframe. Figure 17 presents episodic cueing based on suggested keywords. For instance, in Figure 17, the word “visual design” is selected and highlighted blue message nodes are messages contain the word “visual design”. The moment users drag the cursor over “January” timeframe, a set of new keywords and person names will be extracted from those highlighted messages in “January” and update the suggestions area. With this interaction, users can easily know and remember all conversation topics

relate to “visual design” such as “visualization”, “feature”, “demonstration”, “mailbox”, and so on.

– Filtered selection with person’s name: by clicking on a suggested person, users will see a collection of lines on the left and right of message nodes. Left lines indicate the cues that selected person is the one who sent those messages. Right line indicates the cue that selected person is the one who belong to the list of recipients. Similar to suggested keyword selection, users can drag person’s name into a timeframe, the same behavior is a new set of suggested keywords and persons are displayed on suggestion area. Figure 18 presents episodic cueing based on suggestion person’s name.

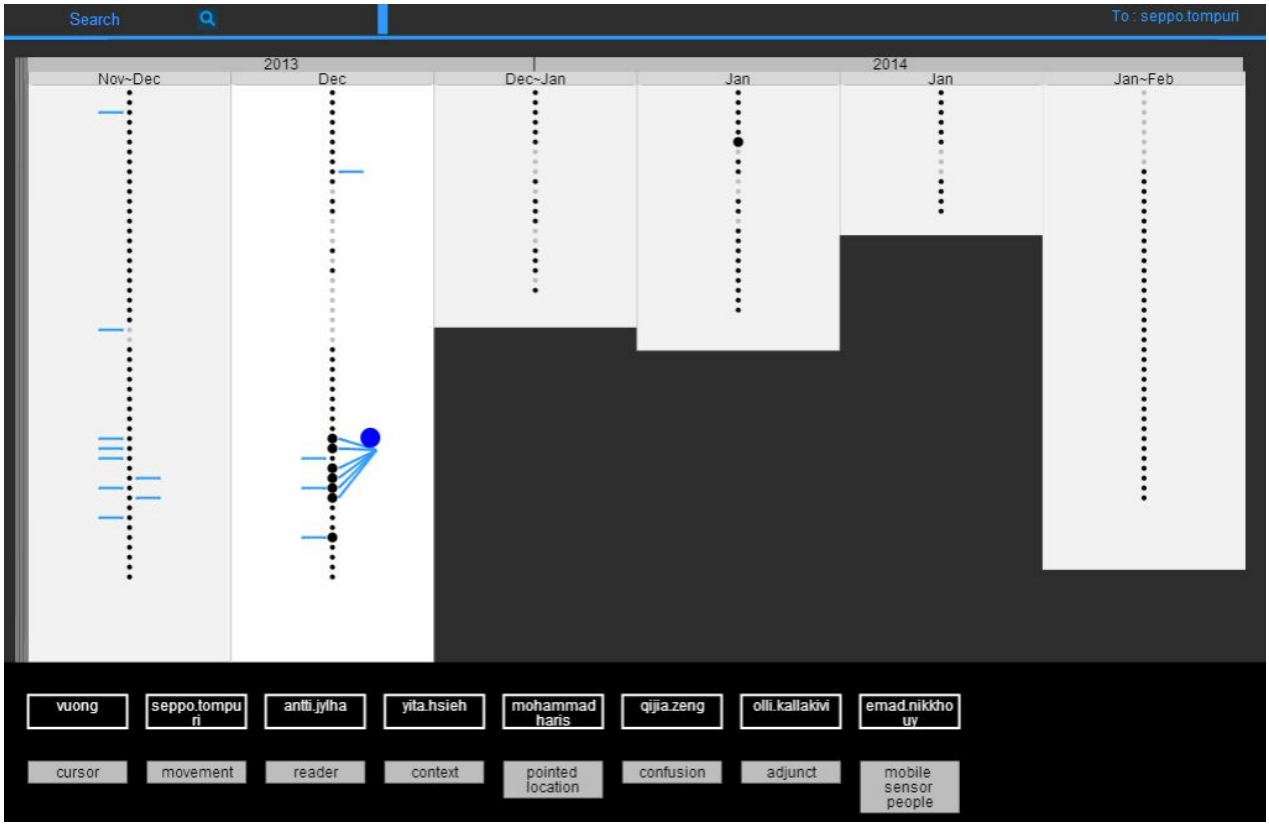

Figure 18. Suggestions based on timeframes with Person Filtered Selection

4.3 Filtering with different criteria

Users often don't know exactly spelling of keywords they should type query into search box. For instance, they remember to search for some information related to my name "vuong", but since my name is not an English name so they might misspell the word as "voung". Thus the result returns no hits at all because the spelling of word does not match any word inside mailbox. Table 2 shows prior research results of local-search engines for mistyped queries [Alsubaiee10]. Under this circumstance, the designed visualization interface already provided users with a set of suggested person name and keywords. They just simply drag those words and drop them into search query area to perform the new search. Search result will always return an adequate amount of matched messages because keywords and person names are directly extracted from email body. With this method, users need not to worry about misspelling of words when typing query into search box. Additionally, the system also allows users to narrow down search result by selecting multiple criteria simultaneously. The following discussion is show how filtering interaction with drag-drop behavior works.

4.3.1 Person name and Keyword filtering

First priority of the system is providing episodic cues for memory reinstatement. Second priority is reducing efforts in typing query in search box. This task could only be done by maintaining effortless transition between different variables such as keywords and persons. The system

Search Engine	Results of Mistyped Queries				
	aumatso restaurant	aomatso restaurant	aumatsp restaurant	amatsu restaurant	aumatso
Yahoo! Local	●	✓	●	●	●
Bing Maps	●	●	●	●	●
Yellow Pages	●	●	●	✗	●
MapQuest Maps	✗	✗	✗	✗	●

● : No results ✓ : Correct suggestion ✗ : Wrong suggestion/answer.

Table 2. Results of Local-Search Engines for Mistyped Queries (as of June 20, 2009) [Alsubaiee10]

provides a set of keywords and relevant time periods as possible different alternatives for users continue their search based on search results. Unlike Thunderbird which provides filtering technique to narrow down the initial result, our system enables users to interact with keywords in order to change search criteria directly. For instance, users can select multiple keywords to highlight message nodes that contain them. Figure 19 shows the example of selecting multiple criteria. In the figure, user narrows down search result by selecting multiple variables including "baris.serim" and "visual design".

In addition to suggesting multiple filters to narrow down search result, the system also provides conventional querying behavior without typing search query. Users can drag-drop multiple keywords and persons into search query area to begin new search. The result output is same with initial view which is chronological order of message nodes with suggestions. Moreover, users also have the ability to manually type in the keywords similar to traditional search engine. User typed keywords will be added to current search query, and each keyword can also be removed from search query area as well. Those keywords are treated equally while querying, thus

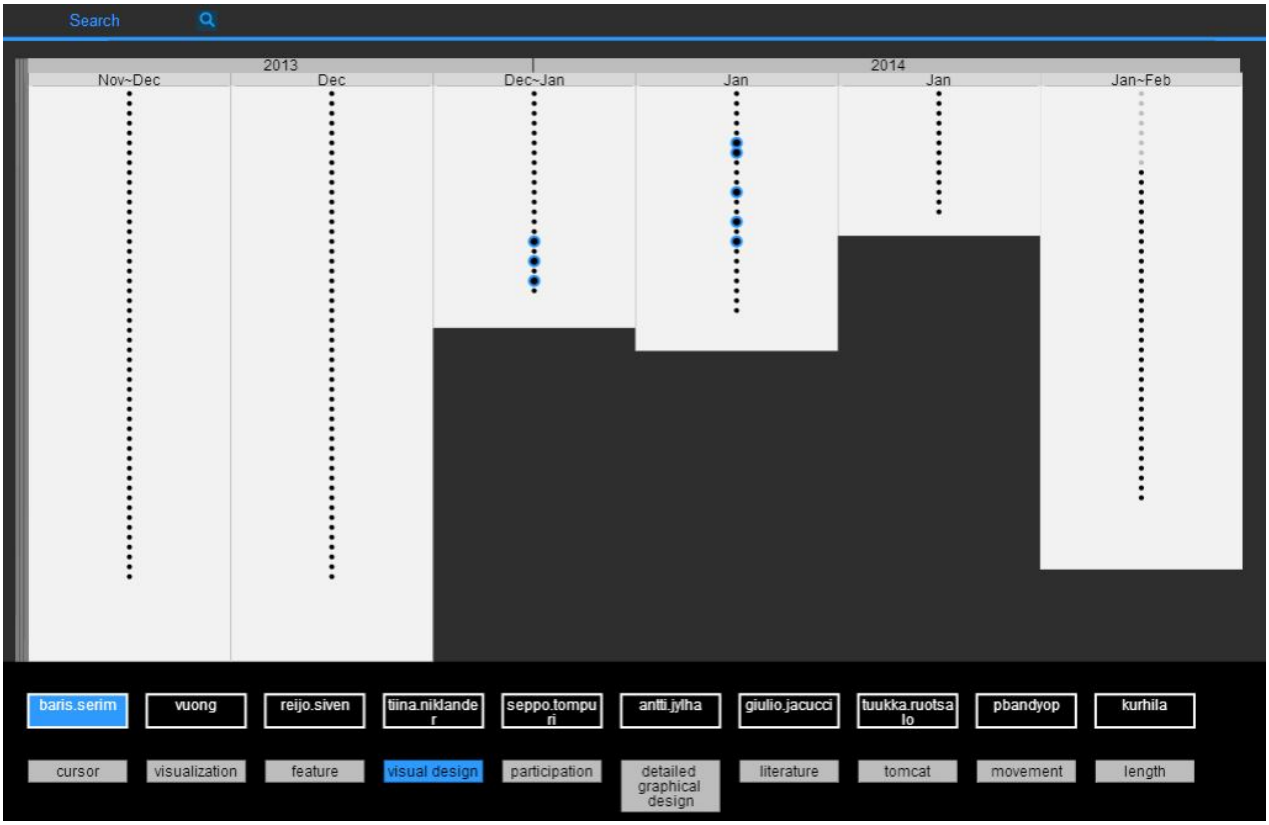

Figure 19. An example of filtering selection with multiple variables

matched messages must contain all of them. Figure 20 shows query of multiple keywords and corresponded search result.

Figure 20. Search result with new query

4.3.2 Time filtering

As mentioned in previous chapters, time is the most important factor that contributes to the success of searching process. Knowing the key to success, we develop the extra function for the interface. This function allows users to search their mailbox based on timeframe. There is hidden compressed area or time period that users can select to perform new search with current selected keywords. The actual query in this search is current selected keywords plus time range specified by users. By pressing buttons marked with “Older” and “Newer” labels, users can select timeframe to generate new results. Figure 21 shows two compressed time period that users can select for new search process. In the Figure current query is “regard”, user presses “Older” button and start choosing timeframe “June 2013” to perform a new time range search with query “regard”. This feature has the following benefits:

– User is able to perceive other timeframes also contain messages having keyword “regard”.

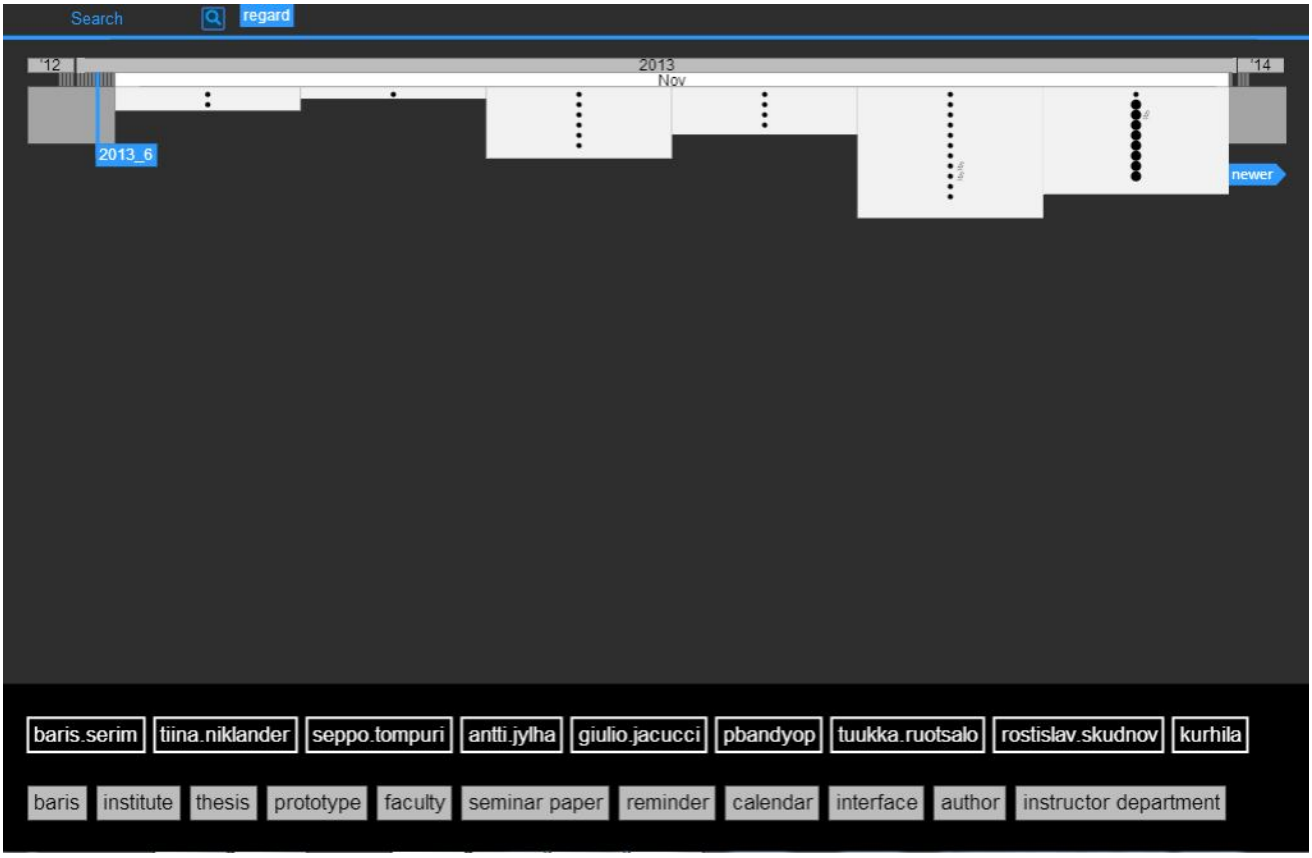

Figure 21. Two compressed time period that users can select for new search process

– Avoiding confusion by reducing the amount of matched messages on the interface. Specifically, when search result consists of huge number of email messages, it might confuse users. Reducing number of message nodes and hiding them into compressed timeframe create positive user experience.

CHAPTER 5 – APLICATION IMPLEMENTATION

This chapter outlines description of APIs will be utilized and the process of implementation of email search application. The goal of application is providing users the ability to search through their mailbox and retrieve accurate information that they want in efficient way. We want to develop the system support on multiple platform, thus the most suitable candidate would be HTML5 since it support cross-platform feature. Additionally, HTML5 also includes Canvas drawing, interaction and animation features to support visualization which matches requirements of the application. For server side scripting, we choose JavaServer Pages (JSP) and Servlet because they support Java Mail which can request data from IMAP server.

5.1 Tools and Libraries

HTML5:

HTML5 is Fifth generation of Hypertext markup language used to structuring and presenting content on web browser. It introduces many new multimedia features such as video, audio and canvas elements. It also is the potential candidate for cross-platform mobile applications. These features perfectly match the requirement of email search visualization. Canvas element will play the leading role in the application. Canvas allows dynamic rendering of 2D graphics on the fly via javascript language. In the application, canvas will be responsible for drawing shapes, background, animation, and provide interactions.

AJAX:

AJAX is Asynchronous javascript and XML which is used for sending asynchronous data request from client and receiving data reply from server. Prominent feature of ajax is the ability to exchange information with server and updating web content without reloading the whole web page. Purpose of ajax in email search visualization is requesting search result from server side.

KineticJS:

KineticJS is the JavaScript framework supports HTML5 Canvas. It enables high performance animations, interactions, and event handling for web applications. We use KineticJS to provide easy development process because it provides all classes support drawing graphics on canvas.

Apache Tomcat server:

Apache Tomcat is an open source software implementation of Java Servlet and JSP technologies. Tomcat server will be running and ready to take data request from users. Server will process data and reply back to clients a JSON string containing detail of keywords, persons, and messages.

Java Mail:

Java Mail is an API provided Oracle. It is a platform and protocol independent framework to build mail and messaging application. Java Mail API will take responsibility of retrieving all messages from IMAP server of users and import them into Lucene database for further processing.

Apache Lucene Core:

As mentioned in the section of Apache Lucene Overview, Apache Lucene Core is sub-project of Apache Lunce. It provides indexing and searching technology written entirely in Java. It will handle all activities of search engine.

Stanford CoreNLP:

Stanford CoreNLP is an integrated framework which provides a set of natural language analysis tools. It includes POS tagger used to read text and assign tags to words such as noun, adjective, verb, etc. We utilize Stanford CoreNLP for extracting key terms from email text.

JazzySpellChecker:

JazzySpellChecker is the API for checking whether a word is an English word or not. Current system only aims to support English users only.

Stopword:

Stopword is an API to identifying whether a word is inside stopword list or not. The API depends on stopword list which is editable.

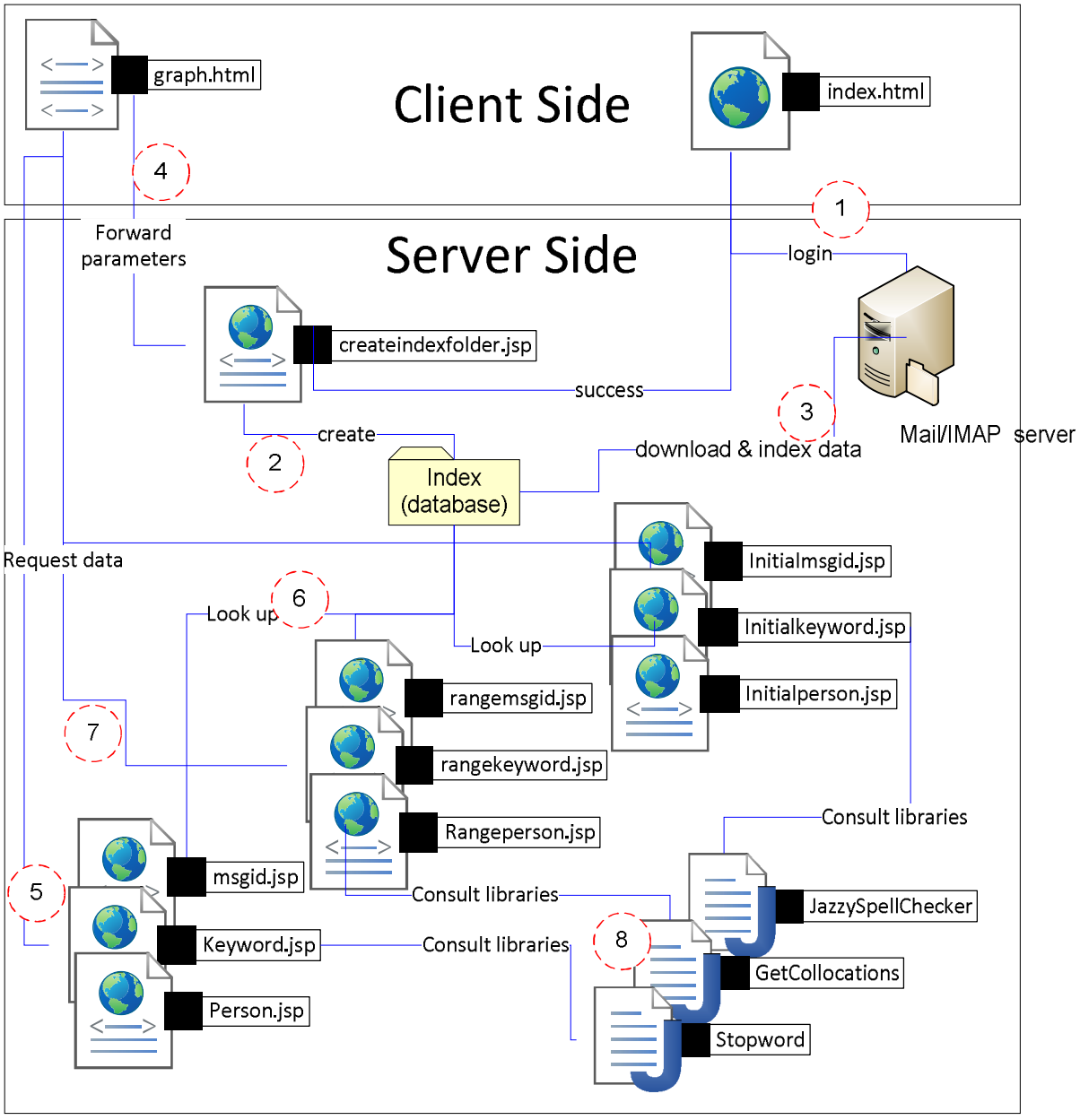

Figure 22. Architecture of Email Search Graph system

5.2 Implementation process

5.2.1 Application Architecture

Main structure of email search visualization system includes communication between client-side level and server-side level. Client side scripting language is Javascript, Html5, and css, whereas server-side scripting language is only Java for JSP and Servlet. Figure 22 depicts the workflow of the system infrastructure. Index page is designed only with username and password input. Index page will take parameters from users for IMAP authentication and extracting messages from mailbox. The following steps show how the whole system works:

1. Firstly, users fill in authentication parameter and send it to the server.
2. Server will forward authentication information from client to mail server of IMAP server. If the authentication passes, a specific index folder is created for each user.
3. After folders are created, server will start download all email messages from IMAP server and import them into Lucene database for indexing. Upon finish indexing, server will redirect to graph.jsp page.
4. Graph.jsp page contains all visual objects and activities of the email search system. Initial data will be loaded upon user's login. A request will be sent to request initial data by looking up email database in index folder. 3 scripts including initialmsg.jsp, initialkeyword.jsp, and initialperson.jsp are responsible to retrieve initial data. Initial data comprises of most recent 200 messages along with key terms and contact persons extracted from the mailbox. Replied data to client is a JSON string with standard format as follows:
    - JSON string from initial msgid:

      [{"<msgid>":{
      "date": <date>,
      "content": <content>,
      "relevance": <relevance>,
      "from": <sender>,
      "to": <recipients>},
      ……
      }]

- JSON string from initial keyword:
  [{“<keyword>": <relevance> ,
  ……
  }]
- JSON string from initial person:
  [{“<name>": <relevance> ,
  ……
  }]

5. After receiving JSON string of initial view from server, graph page will dynamically draw visualized objects on screen. Visualized objects include timeframes, keywords, persons, messages, and emails body.
6. When users perform new search with keywords, client will request for data again. 3 scripts including msgid.jsp, keyword.jsp, and person.jsp will handle new search query with keywords. Server will reply a JSON string similar to initial data. When new search query data received from server, graph page continue dynamically update visualized objects and content on the interface.
7. When users perform new search by pressing “Before” or “After” column, client will request for data within time range. 3 scripts including rangemsgid.jsp, rangekeyword.jsp and rangeperson.jsp will handle the request. Server also replies with a same JSON string format with initial data. When new search query data received from server, graph page will dynamically redraw visualized objects without reloading the whole page.
8. In addition to above steps of client server communication process, there is an extra execution from server side such as key term extraction process. This process comprises of consulting libraries such as JazzySpellChecker, Stopword, and GetCollocation. It will take up to 10 seconds to complete this extra process. All interactions are activated and processed locally on client side in order to reduce networking cost and improve the performance efficiency.

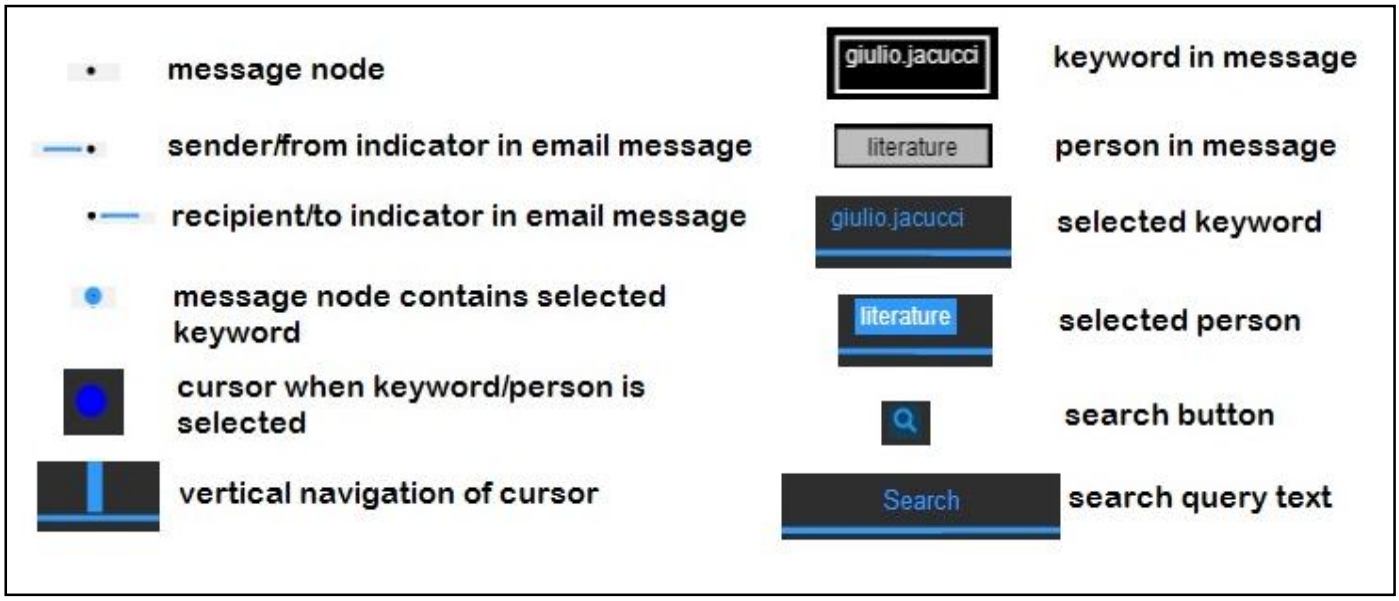

Figure 23. Visual objects in Email Search Graph system

In order to ensure privacy and security of users' mailbox, the system automatically removes all data in index folder. Removing data process is triggered based on two cases which are either users manually log out or timeout of the index folder.

5.2.2 Visual Objects

Visualization is the most crucial factor in the system, thus it is taken carefully into account when programming the application. Purpose of visualization is showing the networks of email attribute acquaintanceship and providing clues to user's memory. In the system, most of the programming tasks focus on creating aesthetic visualized appearance and impressive interaction. Figure 23 shows all visualized objects are drawn in email search system.

- *Message node*: A black circle indicates the email message in mailbox
- *Sender indicator*: A blue line connected to message node shows selected contact person is a sender in that email message.
- *Recipient indicator*: A blue line connected to message node shows selected contact person is one of recipients in that email message.
- *Message node contains selected keyword*: A black circle with blue border indicates email body or subject contain that keyword.

Figure 24. Color scheme of message node

- *Cursor when keyword/person is selected*: A blue circle indicates the keyword/person selection has been made.
- *Vertical navigation for cursor*: A blue vertical line presents x axis coordinate of cursor.
- *Keyword in message*: A black rectangle with white border and label shows keywords extracted from email body and subject
- *Person in message*: A gray rectangle with label shows person extracted from sender and recipient field in email message
- *Selected keyword*: A blue text indicates which keyword in currently selection.
- *Selected person*: A blue text indicates which person currently in selection.
- *Search button*: An image presents a button to start new search query.
- *Search query text*: HTML input field can receive search query.

Moreover, In order to improve user perception toward the system, objects such as message node, keyword, person, and visualized cursor have their own color scheme, event handling, and visual transition.

Color scheme

- Message node: different color presents different behavior of message node. For instance, black indicates message hasn't been viewed in a session of the system, gray show message already was viewed once in a session, and blue border indicates message contains selected keywords. Figure 24 shows color scheme of message node.

Figure 25. Color scheme for person/keyword object

- Person: this object has two colors indicating two behaviors. Gray shows normal person object and blue shows person object in double click mode. Figure 25 represents color scheme of person and keyword object.
- Keyword: this object has two colors indicating two behaviors. Black shows normal keyword object and blue shows keyword object in double click mode.

Event Handling

Event handling is added to the system in order to help users perceive information easily. Besides, event handler allows users to interact with the system. Visual interactions such as dynamic filtering keyword/person based on timeframe, display relationship between emails and persons, and visually provide new search query.

Events happen on visual keyword and person object:

- "Mouse down" on visual keyword, person object: temporarily select keyword/ person object, store their data string into temporary data for further processing, and display them on top right side of the interface.
- "Mouse up" on visual keyword, person object: remove temporary data, and drop their information on the top right side of the interface.
- "Double click" on visual keyword, person object: this event means selecting multiple filtering criteria. The interface will show email message nodes that contain multiple keywords and persons.
- "Drag and drop" keyword, person into search query area: activate the new search with provided keyword.

- "Drag and mouse over" keyword: dragging a particular keyword object over message nodes to select only messages containing that keyword.
- "Drag and mouse over" person: dragging a particular person object over message nodes to select only messages with sender or recipient is that person.
- Events happen on timeframe:
- "Mouse over" timeframe column: when users click and mouse over timeframe column, list of visual person objects and keyword objects are rearranged based on selection.
- "Mouse out" timeframe column: a list of person and keyword objects returns to their original list.
- "Mouse click" on "newer" column: expand time period to perform new search query based on time period which is newer than current time period.
- "Mouse click" on "older" column: expand time period to perform new search query based on time period which is older than current time period.

CHAPTER 6 – EVALUATION

Goal of the evaluation is to measure how interactive the interface can be. In order to do that, we need to prove the system can reduce the amount of typing users have to perform during search activities. Specifically, I investigated if timeline layout is helpful and suggested keywords are used during search activities. As part of the evaluation, I asked users to directly interact with the system using real email data. To guarantee the security and privacy of user's mailbox, I also implemented data encryption during authentication process and email data is erased immediately after user log out.

6.1 Method

Latest version prototype of the system is still not the final system of our research. However it includes several functions which are adequate for me to evaluate the system for this thesis. I recruited around 5 master students to perform 10 search tasks on the system within 2 weeks. Their mailboxes contain approximately from 1000 to 2000 email messages from past 2 years. Participants are all knowledge users, and have experience with information visualization. Besides, I also provided a demonstration video for users to easily understand the system prior to the test. In order to study the efficiency of the system, I collected both log data and subjective evaluation from users.

Log data: I was granted the permission from users to log all their actions during each search task. The log file contains information as follows:

- Which method users prefer to search their mailbox: search query with predefined

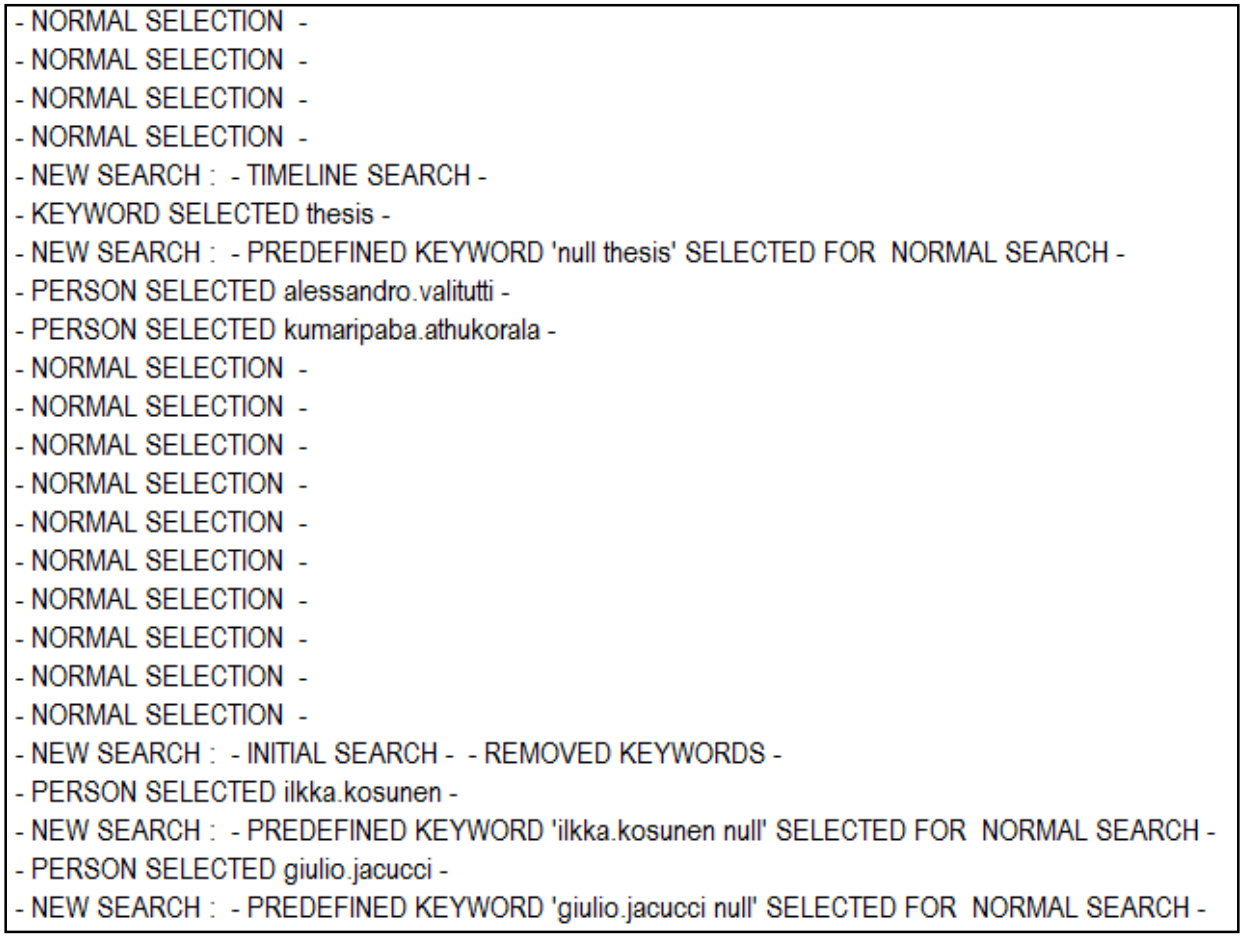

Figure 26. An example of logging from user's action

keywords, search query based on timeline, or manually type query in a search box?
- Whether users use suggested keywords and contacts as episodic cues during search activities?

Purpose of quantitative data is to learn which strategy or sequence of actions users uses to look up their email. Figure 26 show the example of loggings taken from user's actions.

Subjective evaluation: To measure the efficiency of the system, participants are assigned to fill in the questionnaire. They also provided difficulties every time they perform a search task. Questions were asked to conclude whether visualization can provide associative clues to reinstate memory of email information. The following questions are used to measure quality of each search task.

1. Messages are arranged in timeline behavior. How useful was this layout to you during search activity? (Rate scale 1-5)

2. Were you able to relate suggested person name/ keywords with the email? (Yes/No)
3. By clicking on keywords/person names and dragging them over timeframe, were you be able to remember message context related to those keywords/persons? (Yes/No)
4. Did you use suggested keywords/persons as search query? (Yes/No)
5. How often did you have to type in search box when suggested keywords/persons are not information that you want? (Rate scale 1-5)
6. How would you rate the difficulty for this search task? (Rate scale 1-5)
7. What difficulties did you encounter? (open ended question)
8. Do you think this visualized interface helps you remember email content and find email easier? (Rate scale 1-5)

In general, users must clarify if suggested keywords and contacts are useful to them during re-finding emails. Additionally, based on the answer, I am able to identify if the timeline layout is easy for users to look up their information.

6.2 Result

Logging

For better understanding about usage data, I divided search activities into tasks. A task includes several queries issued by users in order to complete one search task. During the study, users issued 128 queries in 50 tasks, average of 2.56 queries per task. Figure 27 shows the number of queries per task issued by users. The bar chart indicates the amount of queries decrease over task. First task begins with an average of 4.2 queries per task and 10^{th} or final task ends with an average of 0.8 queries per task. It appears that users issued high amount of queries in the first five tasks and lower the amount of queries in later task. This might dues to the fact the users got familiar to the system over time. Highest amount of queries was 5 for the first task; this also reflected the difficulty participants had during first time using the system. However after

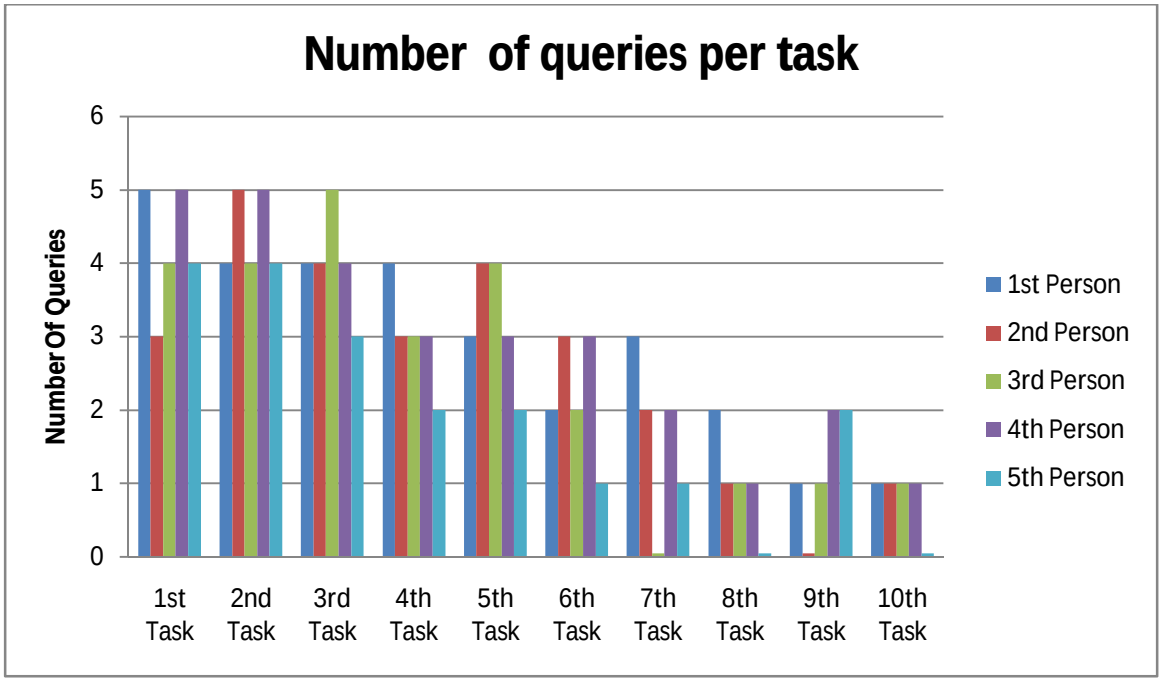

Figure 27. The number of queries per task

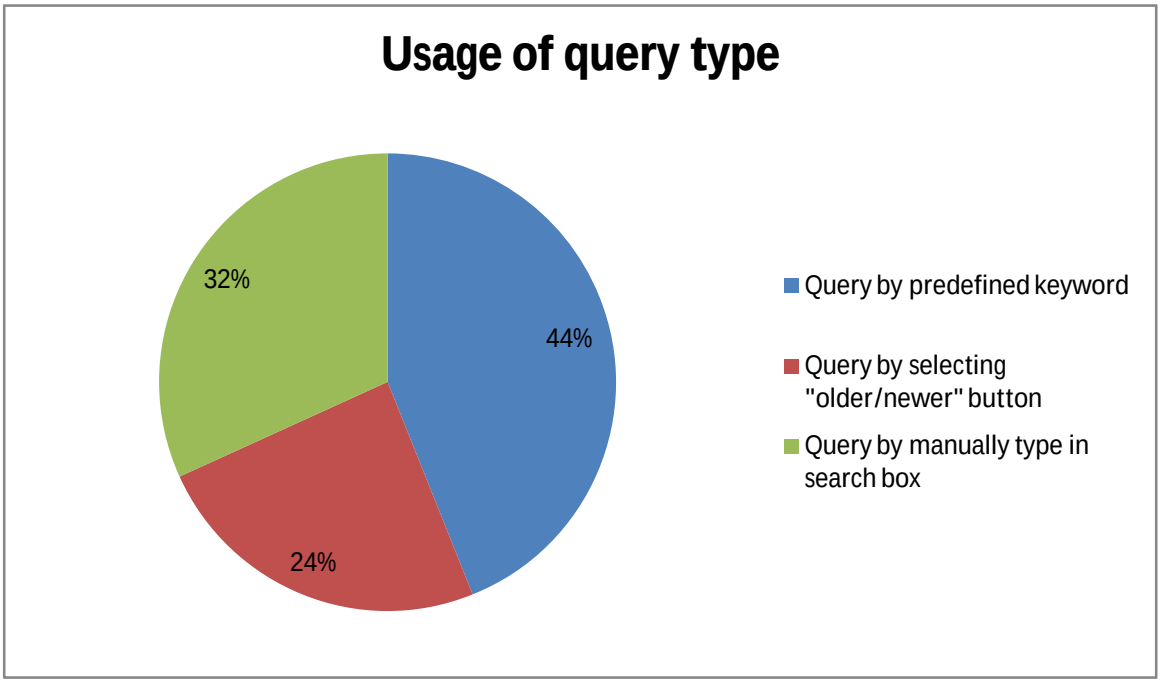

Figure 28. Percentage of usage of query type in the system

understanding and getting used to the interface, the number of queries decreases significantly to 0 or 1 query for a task. From this data, we can see an average of 2.56 queries per task is acceptable. After users familiarize with the system, they eventually only took 1 or no queries to complete the task. The purpose of this column chart is to show the learning process of users toward the interface.

Further, the log file also showed that there were 123 queries using predefined keywords and contacts, 68 queries by using "older/newer button or selecting specific time period and 89 queries by typing query in search box. Figure 28 shows the percentage of each query type is used during search activities. As we can see Query by predefined keyword is dominated with 44% whereas Query by manually typing keyword has only 32%. This data shows that users preferred to use predefined keywords to perform each search task. This is what I expected the system to behave during the test. Moreover there were 24% timeline searches issued by users. This is a desirable outcome due to the fact that people need to search based on timeline when they are clueless what query should be issued.

Questionnaire

	1st person	**2nd person**	**3rd person**	**4th person**	**5th person**
Rating of usefulness of chronological layout	5	1	4	4	3

Able to relate keyword/contact with email	Yes	Yes	Yes	Yes	Yes
Able to remember email by dynamic switching between different filtering criteria	No	No	Yes	Yes	No
Use suggested keyword/contact as search query	Yes	Yes	Yes	Yes	Yes
Rating of typing query into search box	2	3	1	2	2
Rating of difficulty for all search task	2	4	1	2	3
All difficulties encountered during search activities	- Difficult to use if there is no instruction - Don't know how to go back to initial page - keywords are useful but don't know why some words are keywords - Can't remember whole email by keyword/contact	- The interface is complicated - Confused regarding the keywords - So many nodes displayed on the system	- Took 20 minutes to familiarize the system - All email messages node color are all black make it hard for him to look up information - Color scheme of selected email nodes	- Visual keywords are useful, but it would better if keywords are displayed on each message node to describe the email topic.	- Difficult to identify which content belongs to 1 of 8 selected email nodes
Rating of email search system	5	2	4	3	2

Table 3. Answers from users

Table 3 presents the answer collected from users. Data indicates that 3 out of 5 people agree timeline layout is useful to them. Reasons were similarity of the system with traditional email clients in which all messages arranged based on time.

Additionally, they all were able to relate suggested keywords and contacts within email message. Keywords were extracted from email content that's why it was possible for them to relate

keyword and contact with email. However, participants agreed that it was hard to remember some keyword because participants almost forget email content since access was long.

However, only 2 out of 5 people think episodic cues can help them to remembering email content. This might due to the fact that participants forget most of email information.

The important thing is they all agree to use predefined keywords as search query. Only 1 participant used high amount of typing keyword in the search box. The reason was participants believed predefined keywords and contacts reflected information within email messages.

Two participants rated search tasks were difficult to complete with the system. There were difficulties during search activities and most of them were caused by the complexity of the visualization. 2 participants had issues with color scheme of email message nodes. It seems displaying 200 email visual nodes with same color might confuse their color perception. Other 2 people suggested that if there were no instructions, it would be very hard for them to use. The last participant agreed suggested keywords and contacts are useful. However it will be beneficial if keywords are presented in the way that it can describe topic of each message node.

Despite of difficulties encountered, 3 out of 5 participants rated the system is quite useful to them, the other 2 just felt a bit hard to use the application.

In general, from the above analysis of logging and questionnaire, I can observed that if participants could relate keyword and contact in email information, then they less likely used search box. In addition, the usefulness of chronological layout also reflected the difficulty of search task. For instance the more they rated for usefulness of chronological layout, the less rating they gave for difficulty of search task. Participants seemed to use predefined keyword and contact very often, and this is what I want to achieve. Moreover by interacting with the system, users are able to remember email content and it is helpful during search activity. Nevertheless, there still exists some difficulty with using the visualization system, thus improving the interface is required.

CHAPTER 7 – CONCLUSION AND FUTURE WORK

The objective of this work is to show how visual and interactive search paradigm is with our new interface. We did 3 prototypes, and found out many crucial design principles in the 4th prototype. The current system prototype was designed in the way to create novel visual search interface that support interactive email seeking beyond query and response. User is now able to interact with the interface to search emails visually. This is a promising technology because user no longer feels hard to control and perceive very large textual information. Colors and graphics on the interface draw much more attention from users. It allows user to perceive information faster and more effective. With the interactive layout, scrolling behavior is excluded and user no longer skims through every email to search for their information. It saves lots of efforts and time during search process regardless of how big the size of user's mailbox is.

Another objective of email search visualization is to investigate several retrieval techniques that support visual interactive interface. Those techniques help reducing the amount of typing users have to perform and allow them to fully recover memory of email information. In order to achieve that goal, we focused on two crucial techniques including applying chronological layout and presenting suggested keywords to users. Prior to developing the application, we suggested the nature of mailbox was the personal history and people already accessed that information at least once. They search for email messages because they need to re-access only email partial information. Therefore, searching email was defined as a re-finding process [Whittaker11]. Information needs to be re-found because users lose memory of partial email content. If the whole content is remembered, re-finding task will be very much easier. Moreover, we also integrated several technologies into our system such as tf-idf calculation, and NPL. NPL was used to extract the right keyword and contact name. We want the system to display most meaningful keywords to users that's why we applied Tf-idf into the system. Tf-idf is responsible for ranking the relevance of keywords based on the frequency of word appearance within the whole mailbox. Those keywords are used as search query to avoid misspelling words. It is the fundamental factor that helps users to search faster since they don't spend too much time thinking of the right keyword.

During evaluating the system, feedback and logging were retrieved and recorded. They help us to understand if application design arguments were valid. Result of the evaluation indicates the

system is very interactive because people choose to interact with graphical objects instead of typing. The interface helped reducing the amount of typing users must perform and memory of email content is recovered during search activities. However, the system still remains some difficulties which could not be solved in the current prototype. We need to extend the capability of visualization to accommodate the process of re-finding email information. For instance, we need to focus more on color scheme of message nodes, and visual keywords should be rearranged in the way that it helps users remember email information. Therefore, future research concerns better use of suggestions and some additional visual features which are needed to create more user-friendly interface.

Acknowledgements

I wish to thank very much Professor Giulio Jacucci who guided me throughout the entire thesis, as well as Baris Serim who revised my thesis and contributed great design effort for the application interface. I would like to specially thank to Ruotsalo Tuukka who taught me about search engine functionalities and help me many other things during the study. Lastly, I also would like to thank people who shared their thought about my thesis, especially friends who helped to test the application.

References

[Alsubaiee10] Sattam Alsubaiee and Chen Li, "Fuzzy Keyword Search on Spatial Data," *Lecture Notes in Computer Science*, vol. 5982, pp. 464-467, 2010.

[Bellotti05] Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, Ian Smith, and Rebecca E. Grinter, "Quality Versus Quantity: E-Mail-Centric Task Management and Its Relation With Overload," *HUMAN-COMPUTER INTERACTION*, vol. 20, pp. 89-138, 2005.

[Chowdhury03] Gobinda G. Chowdhury, "Natural language processing," *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 51-89, 2003.

[Daille03] Béatrice Daille, "Conceptual structuring through term variations," in *MWE '03 Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, 2003, pp. 9-16.

[Baillie08] Mark Baillie , Ian Ruthven, David Elsweiler, "Exploring memory in email refinding," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 4,

pp. 1-36, September 2008.

[Diab04] Mona Diab, Kadri Hacioglu, and Daniel Jurafsky, "Automatic tagging of Arabic text: from raw text to base phrase chunks," in *HLT-NAACL-Short '04 Proceedings of HLT-NAACL 2004: Short Papers*, 2004, pp. 149-152.

[Dumais03] Susan Dumais et al., "Stuff I've seen: a system for personal information retrieval and re-use," in *SIGIR '03 Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, New York, 2003, pp. 72-79.

[Robbins06] Daniel Robbins , Susan Dumais , Raman Sarin, Edward Cutrell, "Fast, flexible filtering with phlat," in *Proceeding CHI '06 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* , New York, 2006, pp. 261-270.

[Elsweiler11] David Elsweiler, Morgan Harvey, and Martin Hacker, "Understanding re-finding behavior in naturalistic email interaction logs," in *SIGIR '11 Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, New York, 2011, pp. 35-44.

[Elsweiler07] David Elsweiler and Ian Ruthven, "Towards task-based personal information management evaluations," in *SIGIR '07 Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, 2007, pp. 23-30.

[Fox89] Christopher Fox, "A stop list for general text," *ACM SIGIR Forum*, vol. 24, no. 1-2, pp. 19-21, 1989.

[Frau05] Simone Frau, Jonathan C. Roberts, and Nadia Boukhelifa, "Dynamic Coordinated Email Visualization," in *WSCG05 - 13th International Conference on Computer Graphics, Visualization and Computer Vision'2005*, 2005.

[Giménez04] Jesús Giménez and Lluís Màrquez, "Svmtool: A general pos tagger generator based on support vector machines (2004)," in *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004, pp. 43-46.

[Gwizdka02] Jacek Gwizdka, "Reinventing the inbox: supporting the management of pending tasks in email," in *Proceeding CHI EA '02 CHI '02 Extended Abstracts on Human Factors in Computing Systems* , New York, 2002, pp. 550

- 551.

[Healey12] Christopher G. Healey and James T. Enns, "Attention and Visual Memory in Visualization and Computer Graphics," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 7, pp. 1170 - 1188, July 2012.

[Jovicic00] Sasha Jovicic, "Role of memory in email management," in *CHI EA '00 CHI '00 Extended Abstracts on Human Factors in Computing Systems*, New York, 2000, pp. 151-152.

[Kerr03] Bernard Kerr, "Thread Arcs: an email thread visualization," in *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, Seattle, WA, 2003, pp. 211 - 218.

[Kudo01] Taku Kudo and Yuji Matsumoto, "Chunking with support vector machines," in *NAACL '01 Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, 2001, pp. 1-8.

[Liddy98] Elizabeth D. Liddy, "Enhanced Text Retrieval Using Natural Language Processing," *Bulletin of the American Society for Information Science and Technology*, vol. 24, no. 4, p. 14016, 1998.

[Lucene] Apache Lucene. (2014, Feb) Apache Lucene. [Online]. HYPERLINK "www.lucene.apache.org" www.lucene.apache.org

[Bikson92] M. L., Bikson, T., El-Shinnawy, M., & Soe, L. Markus, "Fragments of your communication: Email, vmail, and fax," *The Information Society: An International Journal*, vol. 8, no. 4, pp. 207-226, 1992.

[Milosavljevic10] Branko Milosavljevic, Danijela Boberic, and Dušan Surl, "Retrieval of bibliographic records using Apache Lucene," *Electronic Library, The*, vol. 28, no. 4, pp. 525 - 539, 2010.

[Sudarsky02] Sandra Sudarsky and Rune Hjelsvold, "Visualizing electronic mail," in *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, 2002, pp. 3-9.

[Cutrell03] Edward Cutrell , JJ Cadiz , Gavin Jancke , Raman Sarin , Daniel C. Robbins Susan Dumais, "Stuff I've seen: a system for personal information retrieval and re-use," in *Proceeding SIGIR '03 Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* ,

New York, 2003, pp. 72-79.

[Tansley03] Robert Tansley et al., "The DSpace institutional digital repository system: current functionality," in *JCDL '03 Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, Washington, 2003, pp. 87-97.

[Penn Treebank] Penn Treebank. (2014, Feb) Penn Treebank POS tags. [Online]. HYPERLINK "https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html" https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[Viégas06] Fernanda B. Viégas, Scott Golder, and Judith Donath, "Visualizing email content: portraying relationships from conversational histories," in *CHI '06 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, 2006, pp. 979-988.

[Ware12] Colin Ware, *Information Visualization: Perception for Design*, 3rd ed.: Morgan Kaufman, 2012.

[Whittaker11] Steve Whittaker, Tara Matthews, Julian Cerruti, Hernan Badenes, and John Tang, "Am I wasting my time organizing email?: a study of email refinding," in *CHI '11 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New Year, 2011, pp. 3449-3458.

[Whittaker96] Steve Whittaker and Candace Sidner, "Email overload: exploring personal information management of email," in *Proceeding CHI '96 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, 1996, pp. 276-283.

[Wolfe89] Jeremy M. Wolfe, Kyle R. Cave, and Susan L. Franzel, "Guided search: an alternative to the feature integration model for visual search," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 15, no. 3, pp. 419-433, August 1989.

[Zebra] Zebra. (2014, Feb) IndexZebra. [Online]. HYPERLINK "http://www.indexdata.com/zebra" http://www.indexdata.com/zebra