Email Thread Reassembly Using Similarity Matching

Jen-Yuan Yeh
Dept. of Computer Science
National Chiao Tung University
Hsinchu 30010, TAIWAN
jyyeh@cis.nctu.edu.tw

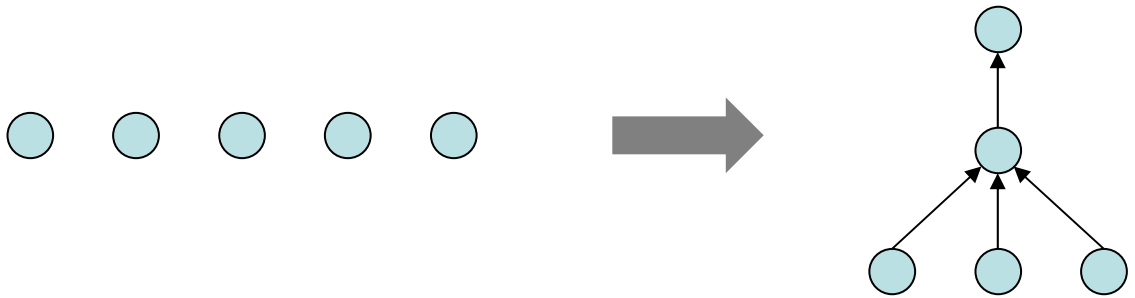
Aaron Harnly
Dept. of Computer Science
Columbia University
New York 10027, USA
aaron@cs.columbia.edu

Outline

- Introduction
- Related Work
- Proposed Methods
- Evaluation
- Discussion
- Conclusion

Introduction

- Email thread reassembly task
  - group messages together based on which messages are replies to which others (i.e., parent-child relationships)
- Email thread structure has been profitably employed
  - e.g., email search, email summarization, email classification, email visualization
  - however, thread structure is not always available



Related Work

- Zawinski (2002) used RFC 2822 header
  - "In-Reply-To" contains the Message-ID of its parent
  - "References" contains the parent's References followed by the parent's Message-ID
- Wu and Oard (2005) and Zhu et al. (2005) linked messages with identical subject lines (after removal of "re:", "fw:", etc.)
- Klimt and Yang (2004) groups messages if they have the same subjects and are among the same users (addresses)
- Lewis and Knowles (1997) exploited IR to email threading

Queries	Targets
Subject text	Subject text
Unquoted text	Unquoted text
Unquoted text	Quoted text
Quoted text	Unquoted text
Quoted text	Quoted text

Approach 1

Using Microsoft's Exchange Header – "Thread Index"

Header Example:

…
content-class: urn:content-classes:message
Subject: Message from Pug Winokur
Date: Tue, 27 Mar 2001 09:20:07 -0600
MIME-Version: 1.0
Content-Type: application/ms-tnef; name="winmail.dat"
X-MS-Has-Attach:Content-Transfer-Encoding: binary
Thread-Topic: Message from Pug Winokur
Thread-Index:
AcC20LeUM9ZkNCLDEdWw9ABQi+MJ2Q==
From: "\"Beth Grizzle\"
<bgrizzle@capricornholdings.com>@ENRON"
To: "Fastow, Andrew S." <Andrew.S.Fastow@ENRON.com>, "Buy,
Rick" <Rick.Buy@ENRON.com>, <rcausey@enron.com>
…

- Thread Index
  - computed from message references
  - can be used for associating messages into a thread
  - but no public information about how it is encoded and how to decode it

Approach 1 (con't)

- Observations
  - the initial message has a 32-byte index ending with "=="
  - a child message has an index which starts with the same string with its parent but an additional 4 or 8 bytes are appended and ends with 0 or 1 "="

Email	Depth	Index Length
E_1	0	L_1=32
E_2	1	$L_2=L_1$+4
E_3	2	$L_3=L_2$+8
E_4	3	$L_4=L_3$+8
…	…	… the 4-8-8 pattern repeats

4

8

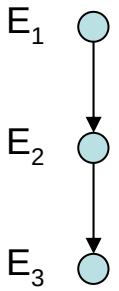
8

E_1: AcGPKD4/2h3YBL/6R9Cpa1YkzGzkaQ==

E_2: AcGPKD4/2h3YBL/6R9Cpa1YkzGzkaQAkldVU

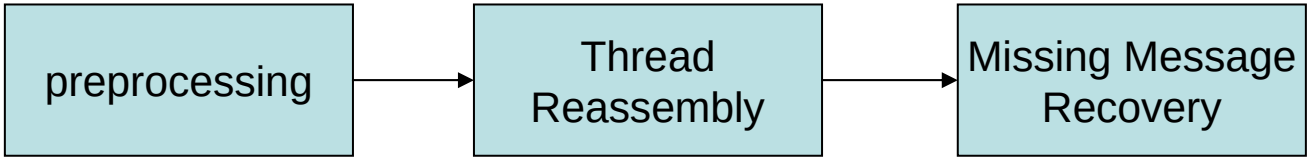
E_3: AcGPKD4/2h3YBL/6R9Cpa1YkzGzkaQAkldVUAAGA/ME=

Example:



Approach 2
Using Similarity Matching and Heuristics

- Mainly by measuring the content similarity between the quotation of a child and the unquoted part of a parent
- Exploit heuristics to reduce the search scope
  - time window
  - normalized subject line
  - sender/recipient relationships



Preprocessing

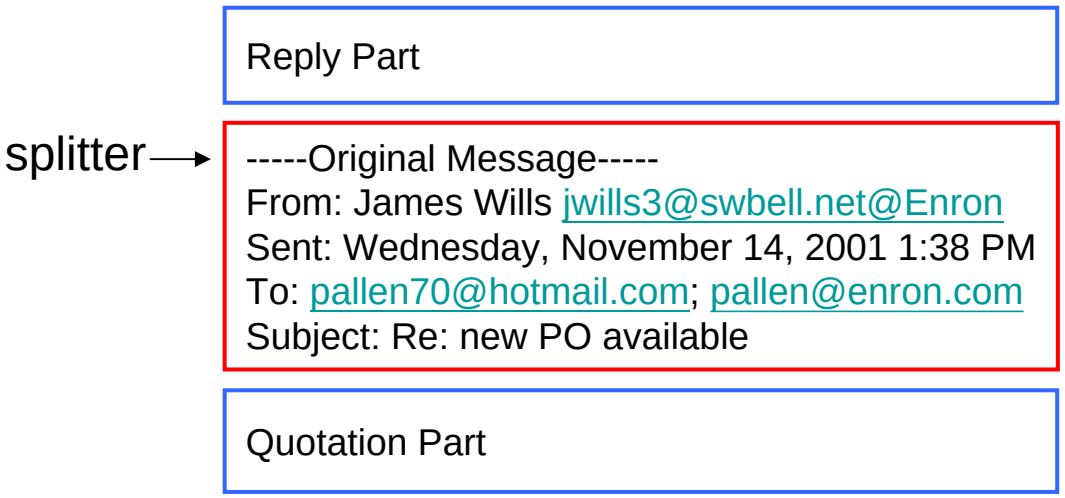
- Duplicate message grouping
  - group duplicate messages by looking for the same subject, datetime, message body, and headers information
- Datetime normalization
  - convert the timestamp of each message into a corresponding timestamp in the same time zone
- Subject normalization
  - remove common prefixes, e.g., ‘RE:’, ‘FW:’, ‘FWD:’, etc.

Preprocessing (con't)

- Sender/recipient identification and normalization
  - pairs of email addresses are identified as belonging to the same individual if the pair meets:
    - in the same email, one address in the 'From' header and the other in 'Exchange-From' header
    - both addresses are in 'From' headers in different emails in a 'Sent Mail' folder
    - addresses are labeled with the same name

Preprocessing (con't)

- Reply and quotation extraction
  - based on manually defined splitters (see Table 2 in the paper)
  - didn't take into account cases, such as a reply interleaved with quoted material (because quite rare in the Enron corpus)
  - no signature identification (regarded as part of the message)
  - a small experiment showed 98% of 1,000 randomly selected emails were separated correctly



The Algorithm

Algorithm:

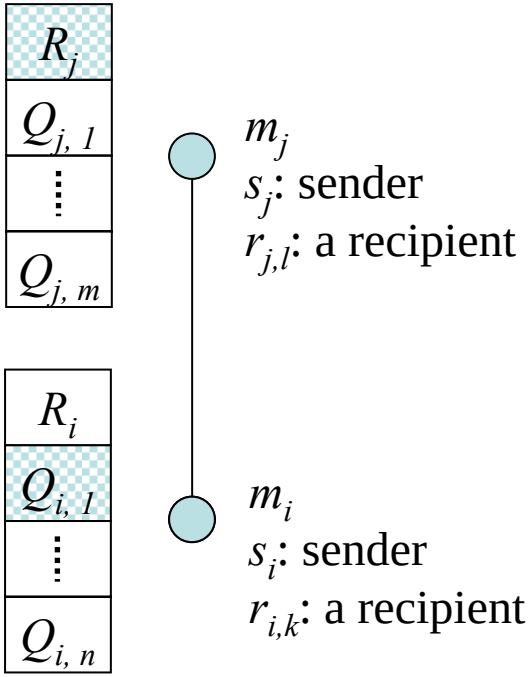
Input: a set of messages

Output: a set of email threads

1. Sort all messages in the chronological order.
2. Regard each message m as an initial thread T, and collect all messages into M, a) which fall within a pre-defined time window, and b) which have the same normalized subject with m's.
3. For each message m_i, $m_i \in M$, put it into T if *findParent*(m_i, T) != NULL. Go to Step 2 until every m_i in M is examined.

- The assumptions of *FindParent*
  - a child message can be either a reply or a forward to at most one parent message in the existing thread
  - missing messages could exist in an email thread

Case I



Conditions:

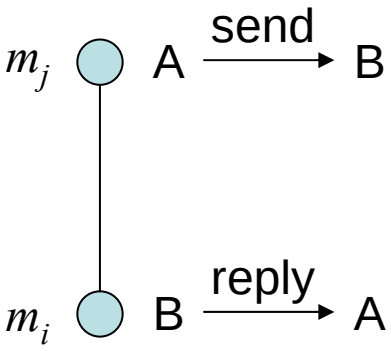
1) $s_i = r_{j,l}$ & $s_j = r_{i,k}$
2) $sim(Q_{i,1}, R_j) \geq \alpha$

Case I: for all m_j in T where exists a recipient $r_{j,l}$ of m_j, $r_{j,l}$ is m_i's sender and a recipient $r_{i,k}$ of m_i, $r_{i,k}$ is m_j's sender

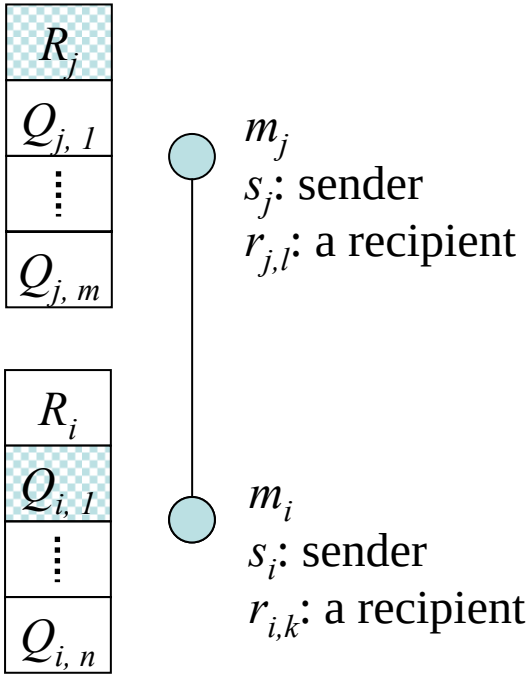
Find an m_j with the highest similarity of m_i's latest quotation (if there has multiple-level quotations) and m_j's reply:

- if the similarity between m_i and m_j is greater than a predefined threshold α, return m_j,
- otherwise, return an m_j with the closest timestamp to m_i.

Example: m_i replies to m_j



Case II



Conditions:

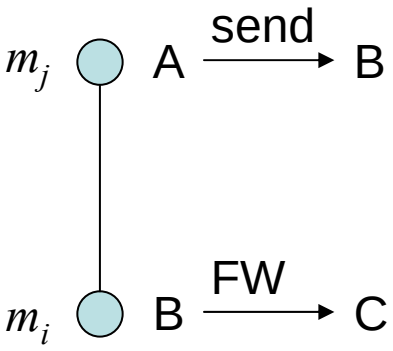
1) $s_i = r_{j,l}$
2) $sim(Q_{i,1}, R_j) \geq \beta$

Case II: for all m_j not satisfying Case I in T where exists a recipient $r_{j,l}$ of m_j, $r_{j,l}$ is m_i's sender

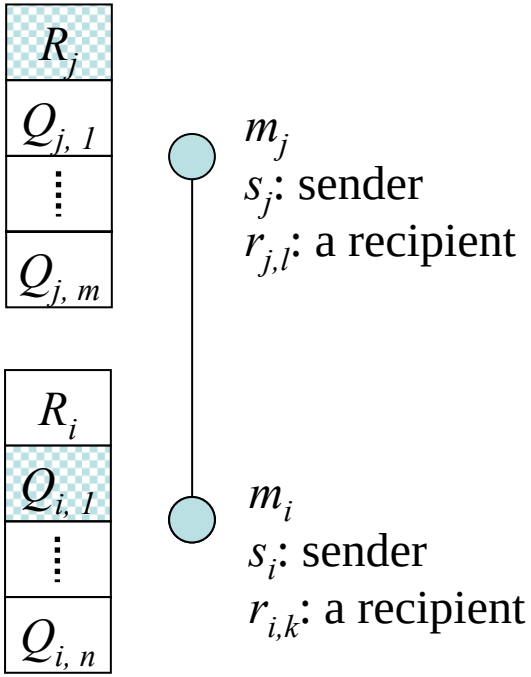
Find an m_j with the highest similarity of m_i's latest quotation (if there has multiple-level quotations) and m_j's reply:

- if the similarity between m_i and m_j is greater than a predefined threshold β, return m_j,
- otherwise, continue to examine Case III.

Example: m_i is a forward of m_j by *B*



Case III



Conditions:

1) $s_i = s_j$

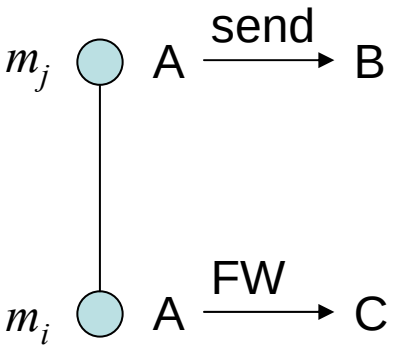
2) $sim(Q_{i,1}, R_j) \geq \beta$

Case III: for all m_j not satisfying Cases I-II in T where m_j's sender is m_i's sender

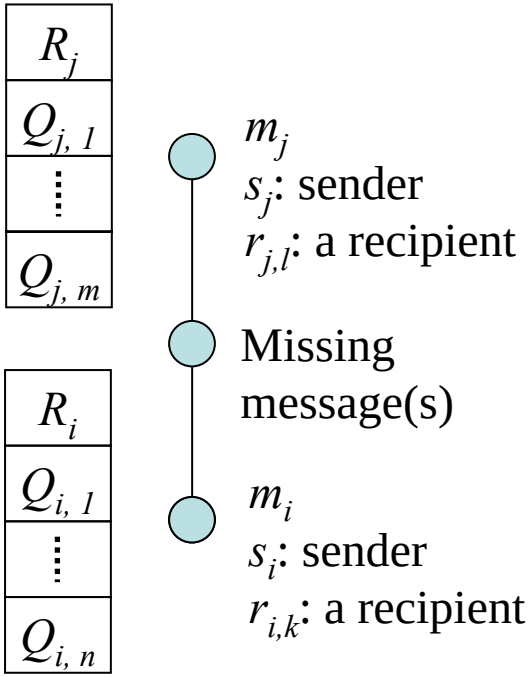
Find an m_j with the highest similarity of m_i's latest quotation (if there has multiple-level quotations) and m_j's reply:

- if the similarity between m_i and m_i is greater than a predefined threshold β, return m_j,
- otherwise, continue to examine Case IV.

Example: m_i is a forward of m_j by *A*



Case IV



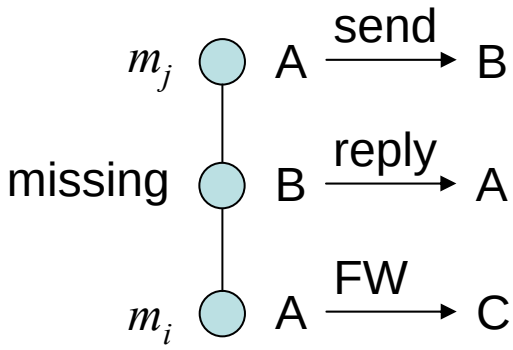
Conditions:
1) $sim(Q_{i,p}, R_j) \geq \gamma$ *or* $sim(Q_{i,p}, Q_{j,t}) \geq \gamma$

Case IV: for all m_j not satisfying Cases I-III in T

Find an m_j with the highest similarity of either m_i's part of quotations and m_j's reply or m_i's part of quotations and m_j's part of quotations:

- if the similarity between m_i and m_j is greater than a predefined threshold γ, return m_j but also add one missing message label between m_i and m_j,
- otherwise, continue to examine Case V.

Example: at least one missing message between m_i and m_j



Case V

Case V: No suitable parent in T for m_i

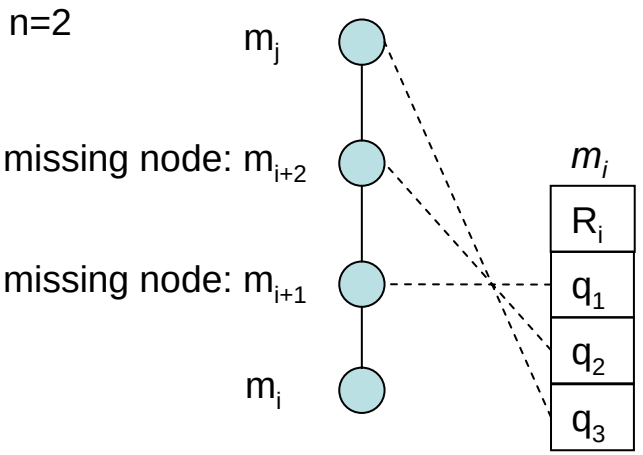
Return NULL.

Missing Message Recovery

Assumptions:

parent: m_j, child: m_i, n missing messages: m_{i+1}, …, m_{i+n}

- If a sequence of quoted text $q=\{q_1, \ldots, q_{n+1}\}$ in m_i can be found such that q_{n+1} is highly similar to the nonquoted text of m_j
- the sequence of quoted text q is assumed to contain a portion of each missing message

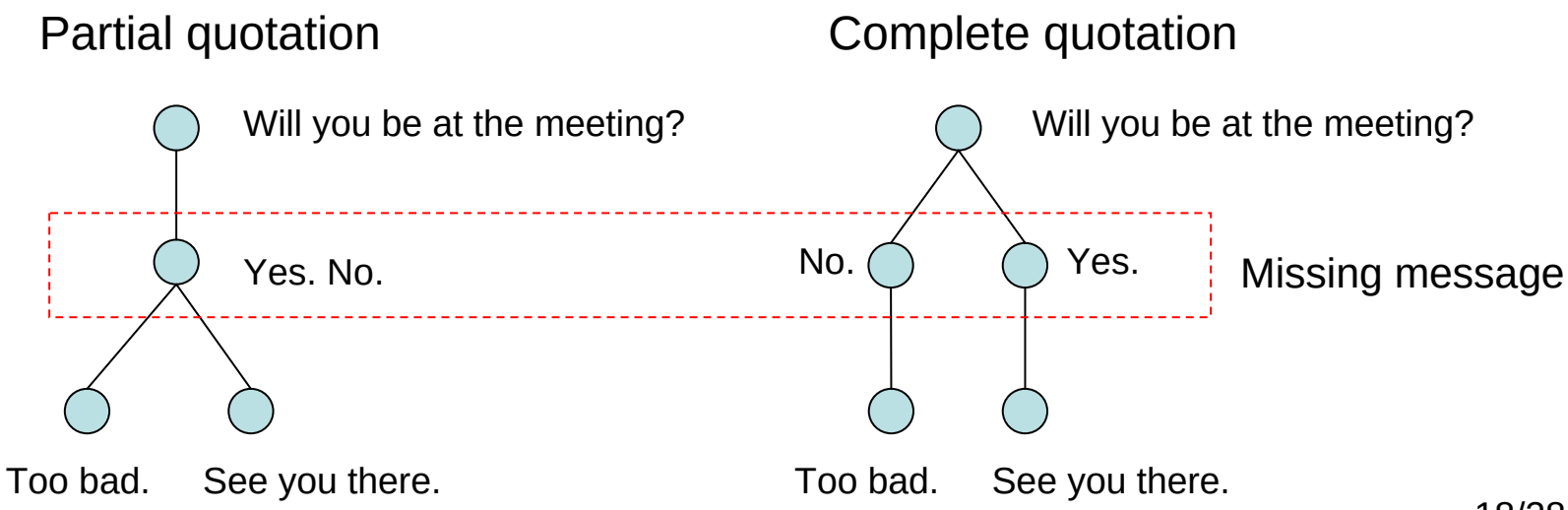


If $q_3=R_j$
$\Rightarrow q_1$ is regarded as m_{i+1}'s body
$\Rightarrow q_2$ is regarded as m_{i+2}'s body

Missing Message Recovery (con't)

When a missing message has multiple children

- Partial quotation assumption (Carenini et al., 2005)
  - the children are siblings – children of a single missing message?
- Complete quotation assumption (In this work)
  - "cousins", i.e., children of distinct missing messages?



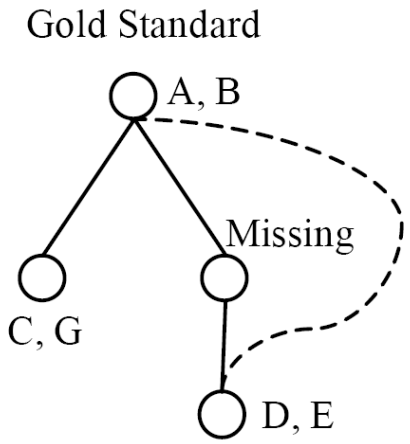
The Enron Corpus

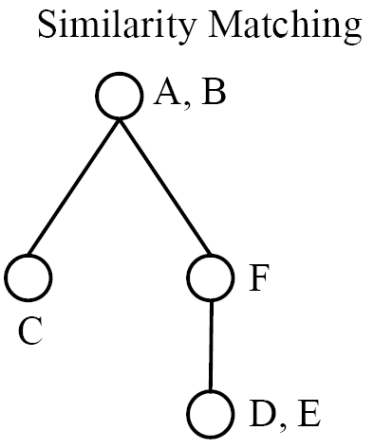
- Raw data
  - Downloaded from the website
  - 1,361,403 messages
  - 158 mailboxes owned by 149 people
- After cleaning
  - 269,257 unique messages
  - in average, 1,704 messages in a mailbox (max: 16,727; min: 2)
  - a large number of emails belong to a small group of users
    34.6% (93,187) messages belong to 10 largest mailboxes

Evaluation Metric

- No explicit gold standard thread structure information
  - use threads created by Approach 1 as a gold standard
- Test set: 3,705 threads
- Recall as the metric

$$R = \frac{\#\ \text{of correct parent/child relationships in Approach 2 threads}}{\#\ \text{of parent/child relationships in the gold standard}}$$





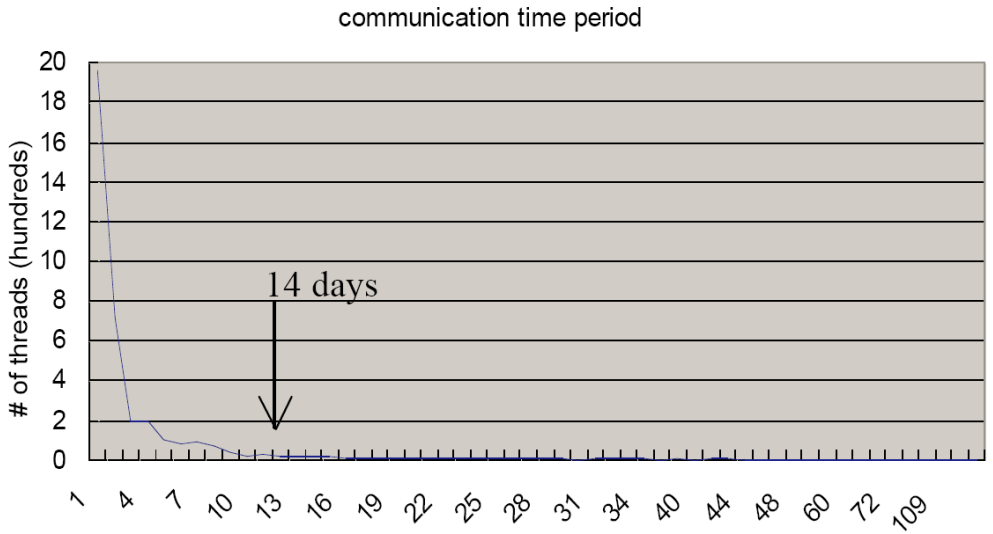
Gold standard: (A, C), (A, G), (B, C), (B, G), (A, D), (A, E), (B, D), (B, E)

Similarity Matching: (A, C), (B, C), (A, D), (A, E), (B, D), (B, E)

R=6/8=0.75

Results

- Settings for Approach 2
  - Time window: 14 days
  - α , β , γ : 0.9



Recall and Mean Per-Thread Recall

Type	Original	Time (T)	Subject (S)	T+S
# of Threads	3,705	3,608	3,122	3,045
Recall	0.5976	0.6421	0.8428	**0.8739**
MPT Recall	0.7184	0.7407	0.8706	**0.8949**

Thread Statistics

- 32,910 email threads, consisting of 95,259 unique messages
- Mean thread size: 3.14
- median thread size: 2
- Mean thread depth: 1.71

Distribution of email threads (without missing message recovery) on the thread size

Thread Size	2	3	4	5	6
# of Threads	19,941	6,753	2,868	1,378	770

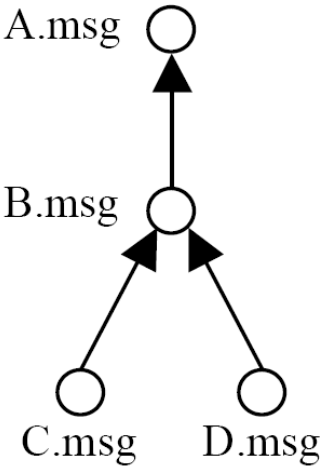
≈	7	8	9	10	(10-20)	20+
≈	406	241	170	121	221	41

Thread Statistics (con’t)

- The number of children of a message was only very weakly correlated with the number of recipients
  ($r = 0.0395$, $p << 0.001$)
- 7.3% (8,077/103,183) threads nodes are missing message
  - 4,850 messages were recovered
- 7.4% (359/4850) nodes contain more than one distinct recovered message
  - generated 430 additional sibling nodes

Discussion: Approach 1

- Advantages
  - simple to implement
  - never makes a “false positive” inference
- Disadvantages
  - doesn’t necessarily reflect the structure of *topic* relations
  - Thread-Index header is not always available
  - suffers “false negatives” in a common case: external exchange



Thread-Index

A: AcE**pvGkhOGBnNJWu**EdWxFgBQi+MJ2Q==

B: N/A (withou header because in the *Sent* folder)

C: AcE**rWenblTYNZZdM**EdWxFgBQi+MJ2Q==

D: AcE**qgwbv2o4HJpZz**EdWxFgBQi+MJ2Q==

→ : reply-to relation

Discussion: Approach 2

- Advantages
  - general applicability, even when there is no header
  - capability to recover missing messages
- Disadvantages
  - doesn’t necessarily reflect the structure of *topic* relations
  - potential for false positives: short parent message
  - suffers false negatives: if no quoted material in the child messages

Approach 1 vs. Approach 2

- Impact of normalized subjects

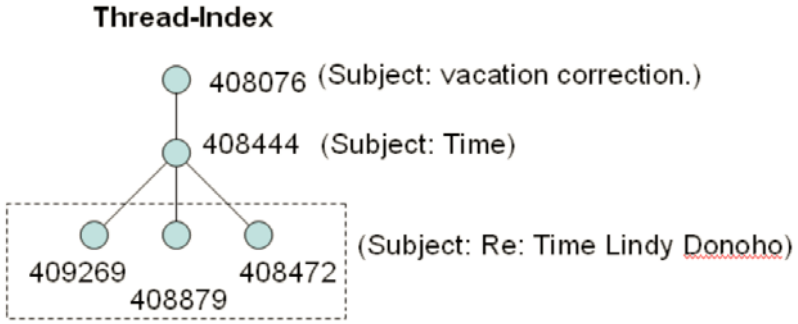


Figure 5. An example thread within which messages have different normalized subjects

- Missing messages

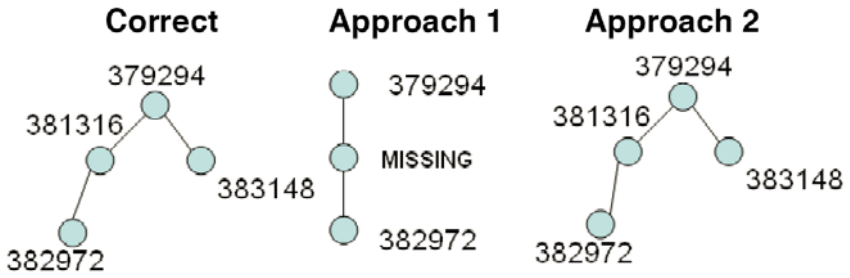


Figure 6. An example to explain the superiority of Approach 2 against Approach 1 when missing messages exist

Small Manual Evaluation

- 20 randomly selected initial root messages
  - manually constructed 20 threads as a gold standard
- A mean average recall
  - Approach 1: 0.7475
  - Approach 2: 0.9338

Conclusion

- Two methods to email thread reassembly were proposed
  - The first exploits Microsoft Exchange Protocol
  - The second links messages by similarity matching between the quoted material of a child message and the unquoted part of a parent message
- Both approaches aim to reconstruct parent-child relationships formed by reply or forwarding
  - might not shed adequate light on the *topic* structure of a thread
  - Approach 2 may be extended to address topic structure by more sophisticated lexical cohesion measures
- A combination of both approaches is an obvious possibility