Threading Electronic Mail: A Preliminary Study

David D. Lewis* Kimberly A. Knowles†
AT&T Labs
Murray Hill, NJ, USA

Appeared, with different pagination, in *Information Processing and Management*, 33:2, 209–217, 1997.

Corresponding Author:
David D. Lewis
AT&T Labs
600 Mountain Ave., 2A-410
Murray Hill, NJ 07974; USA
lewis@research.att.com
ph. +1 908-582-3976
fax. +1 908-582-7550

*Address for Correspondence: David D. Lewis; AT&T Labs; 600 Mountain Ave., 2A-410; Murray Hill, NJ 07974; USA; *lewis@research.att.com*

†Current Address: Department of Mathematics; MIT; 77 Massachusetts Ave.; Cambridge, MA, USA.; *kknowles@mit.edu*

Abstract

Tools for processing e-mail and other electronic messages should be able to recognize and manipulate *threads*, that is, conversations among two or more people carried out by exchange of messages. While user clients typically insert in messages structural information useful for recovering threads, inconsistencies between clients, loose standards, creative user behavior, and the subjective nature of conversation make threading systems based on structural information only partially successful. We propose that this situation is unlikely to change, and that threading of electronic messages be treated as a language processing task. Preliminary experiments show that a significant level of threading effectiveness can be achieved by applying standard text matching methods from information retrieval to the textual portions of messages.

1 Introduction

The volume of electronic mail is huge and growing. Many users receive more messages than they can handle, which has sparked interest in better message handling software. Almost all mail readers now support separating messages into folders, and often allow rules to be defined to do this automatically. Tools for prioritizing and searching messages are also becoming available.

A problem with most such approaches is that they process each message individually. Many messages are parts of larger conversations or *threads* involving two or more participants, and treating them outside this context may lead to undesirable results. For instance, a system that sorts messages into folders based on their content is unlikely to be 100% accurate. The effectiveness of content-based text categorization systems varies considerably among categories (Lewis and Gale, 1994, Wiener et al., 1995), and accuracies over 90% are rarely reported. This means that threads with many component messages will almost always be broken up and distributed into multiple folders by such a system, making it difficult for a reader to follow the conversational structure.

On the other hand, a mail reading interface that understood threads could save users considerable effort. For instance, some programs for reading Usenet news (see Section 2) allow users to delete an entire thread at once, greatly reducing the number of messages the user must inspect.

In the following, we review current approaches to threading, and suggest that they are to some extent misconceived. Effective threading systems should rely on robust conventions in human communication, rather than (or in addition to) relying on rapidly changing conventions in software communication. We show experimentally that statistical information retrieval techniques can achieve a significant level of accuracy at identifying when one message is a response to another, and propose a variety of improvements beyond this basic level of effectiveness.

2 Threading in Electronic Messaging Systems

Messaging systems that are explicitly oriented to group discussion, e.g. the Usenet network and other bulletin board systems, provide the most support for threading. For instance, the reply command in most Usenet news posting programs inserts into a reply or *child* message two forms of information about the relationship between it and its *parent* message (the message it is a reply to). First, the chain of unique message identifiers in the References field of the parent is copied into the References field of the child, with the unique identifier of the parent added. Second, the Subject line of the parent is copied into the Subject line of the child, typically prefixed by *Re:*. Some Usenet news readers providing a threaded display use the structural links from the References field, while others organize a threaded display around Subject lines which are

identical or have identical prefixes.

Conversations, including group discussions, can also be carried out over electronic mail systems. The ability to send and reply to groups of people, as well as the use of centralized mail "reflectors" and mailing list management software, can informally support multiple large scale discussions. As with bulletin board systems, replying to an email message often inserts structural information into the reply. For Internet-based mail systems (Rose, 1993), the reply command may copy the Message-Id field or other identifying information from the parent, into the In-Reply-To field of the child. As in Usenet messages, the Subject line is typically copied to the Subject field, preceded by *Re:*.

Some mail clients provide threaded displays, though this is less common than in bulletin board systems. For instance, the *VM* mail reader (Jones, 1991) allows grouping of messages by any of several criteria, including having the same subject line text, the same author, or the same recipient. The mail archiving program *hypermail* (Hughes, 1995) marks up archives of email with a variety of links, including threading information. It attempts first to find a message ID in the In-Reply-To field and match it to a known message. Failing that, it looks for a matching date string in the In-Reply-To field, and finally tries for a match on the Subject line, after removing one *Re:* tag.

However, the error rate of the above approaches is considerable. While the References field is in theory required for replies to Usenet messages (Horton and Adams, 1987), threading is hampered by clients that delete portions of the References chain due to limitations on field length. In Internet electronic mail, the use of Message-Id and In-Reply-To fields are optional and their format and nature is only loosely constrained when they are present (Crocker, 1982). Subject lines for both Usenet messages and Internet mail are allowed to contain arbitrary text, clients are inconsistent in their use of *Re:* tags, and manual editing of Subject lines further confuses the issue.

One reaction to the above situation is to try to force clients to follow tighter standards for specifying threads. The increasing diversity of clients and the growing interconnection of only partially compatible messaging systems does not give one cause for optimism on this score. Tighter standards also do not help in recovering thread structure from archived messages, since deletion of fields such as In-Reply-To by archiving and digestifying programs is common.

It is also not clear that threads should be defined as trees of reply links. The reply command is often used to avoid retyping a mail address, rather than to continue a conversation. Further, users will disagree about what is on-topic in a thread, and off-topic responses can easily spawn subdiscussions. Conversely, on-topic contributors to a discussion may simply send a fresh message rather than using the reply command.

This suggests that the links we would like to display in a threading interface, and which result in structures we would like to process as a unit, are not objectively defined "pattern-matching" (Allan, 1996) or "structural" (Blustein, 1993, Ch. 3) links. The link semantics we want to capture is that of a *response* in an ongoing discourse. The fact that users are able to participate in online

discussions, despite the inadequacies of current threading software, suggests that messages contain the contextual information necessary to understand their place in an ongoing conversation. An automated system may be able to make use of this information as well to make this conversational structure explicit as a thread. How this might be done is discussed in the next section.

3 Cohesion and Thread Structure

Halliday and Hasan (Halliday and Hasan, 1976, pp. 299–300) have stressed the role of cohesion or linking between the parts of a dialogue. Language provides a variety of mechanisms for achieving this cohesion. One such mechanism is *lexical cohesion* and in particular *lexical repetition* (Halliday and Hasan, 1976, Ch. 6), that is the repeating of words in linked parts of a discourse.

The phenomenon of lexical repetition suggests that the similarity of the vocabulary between two messages should be a powerful clue to whether a response relationship exists between them. Measuring the similarity of vocabulary between texts is of course a widely used strategy for finding texts with similar topic to a query (Salton, 1989, Ch. 10). Indeed, similarity-based methods have been used to construct hypertexts linking documents or passages of documents on the basis of topic similarity (Bernstein, 1990, Salton and Buckley, 1991, Blustein, 1993, Salton et al., 1994a, Salton et al., 1994b, Myka and Guntzer, 1995, Salton et al., 1996).

Attempts have also been made to go beyond unlabeled linking to use similarity matching in detecting discourse relations. Hearst's *TextTiling* algorithm, which was motivated by the observations of Halliday and Hasan and other discourse researchers, uses vector space similarity to decompose a text into topically coherent segments (Hearst, 1994). Allan uses the graph structure of a network of raw similarity links to infer meta-links corresponding to discourse relations such as comparison and summarization (Allan, 1996).

These lines of evidence suggest text similarity could be a clue to the existence of a response relation between messages as well. We investigate this possibility in the next section.

4 Experiment

Our goal was to test the ability of various similarity measures to indicate whether one message was a response to another. We investigated three types of textual material from messages: the Subject line, quoted material in the message body, and the unquoted material in the message body. (See Figure 1 for an example.)

Text from the Subject line is a good clue that a message belongs to a particular thread, though it may not directly indicate which message in the thread is being replied to. Quoting of material from the parent message, particularly quotes of several lines, is a much stronger form of context. Salton and Buckley

showed that text matching on a collection of Usenet messages which included substantial quoted material was highly effective at retrieving related messages, under a definition of relatedness that subsumed the response relationship we are interested in (Salton and Buckley, 1991).

Finally, the non-quoted content of the response can be expected, based on the coherence phenomena described earlier, to repeat words from the parent message. Since novel words will be present as well, we would expect this to be a somewhat weaker clue than the Subject line and quoted text.

4.1 Data Set and Preparation

A corpus of 2435 messages posted to the *www-talk* mailing list during the period February 1994 through July 1994 were obtained from the archives at

http://www.w3.org/hypertext/WWW/Archive/www-talk.

A total of 941 of these messages had an In-Reply-To field containing a unique identifier from the Message-Id field of another message in the corpus. While we have suggested that In-Reply-To links will not always correspond to the discourse response links we are interested in, they provide a reasonable initial test of the ability of text matching to find connections that are response-like. We therefore used these 941 child-parent pairs as ground truth against which we tested methods for finding parent messages.

Simple filters were written to extract three types of textual material from each message: the text of the Subject field, unquoted text from the message body, and quoted text from the message body. This resulted in three collections of 2435 document representatives, one for each type of textual material. Each of the three collections was indexed using Version 11.0 of the SMART experimental text retrieval system, obtained from directory *pub/smart* at *ftp.cs.cornell.edu*. Some messages had empty document representatives in some of the databases (for instance, a message might have no quoted material) and so could not be retrieved from that database.

4.2 Processing

Five text matching strategies were tested for their ability to retrieve the parent of a message, given text from the child message. For each strategy, all 941 document representatives of identified child messages were run as queries against one of the three databases of 2435 document representatives using the SMART system. This produced a ranking of all 2435 target (that is, potential parent) messages for each query message. (Messages which did not have any words in common with the query were not retrieved. We assigned them random ranks lower than that of any retrieved message.)

Target messages were ranked using the cosine similarity formula and a variant of $tf \times idf$ weighting (Salton and Buckley, 1988, Salton et al., 1996). Target messages were represented as vectors of numeric weights:

$$< w_{i1}, w_{i2}, ...w_{ik}, ..., w_{it} >$$

where

$$w_{ik} = \frac{f_{ik}}{\sqrt{\sum_{j=1}^{t} f_{ij}^2}}$$

and f_{ik} is the number of times word k appears in message i. Query messages were similarly represented as vectors:

$$< q_1, q_2, ...q_k, ..., q_t >$$

where

$$q_k = \frac{f_k \times \log(N/n_k)}{\sqrt{\sum_{j=1}^{t} (f_j \times \log(N/n_j))^2}}.$$

Here f_k is the number of times the word occurs in the query message, N is the number of messages in the database, and n_k is the number of messages containing word k.

The score of each target message i is then

$$\sum_{j=1}^{t} q_j w_{ij}.$$

Documents were ranked by this score.

Each strategy was a choice of what text from a child should be used as a query, and what text from target messages should be used to represent them in the database. The five combinations explored were:

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

4.3 Results

Figure 2 displays the distribution of ranks of the 941 parent documents with respect to each of the five forms of text matching. The value for rank 0 is the number of times a child retrieved its parent as the first document in the ranking, rank 1 indicates how often the parent was second in the ranking, and so on. In computing the rank of the parent, the child document (which was itself present in the database, though not necessarily in the same form as was

used in querying) was removed from the ranking, so that the ranks run from 0 to 2433 instead of 0 to 2434.

Table 1 shows the number of times the parent was retrieved at rank 0, ranks 0 to 4, and ranks 0 to 10 for each of the methods. We compare this to the values that would be expected if the parent appeared at a random rank between 0 and 2433.

5 Discussion

As expected, using the quoted portion of a message as a query and matching against the unquoted portions of target messages was the most effective of the five strategies tried. The parent was the highest ranked message in 666 out of 941 trials or 71% of the time. Put another way, a system that simply assumed the highest ranked message under this matching strategy was the parent would have 0.71 recall and 0.71 precision at finding parent messages. Of course, these figures are for messages that are known to have a parent message. An operational system would need not only to distinguish among potential parents, but also to detect whether or not the message had a parent at all.

Our results can be roughly compared with the 0.90 recall and 0.72 precision in Salton and Buckley's experiments with Usenet messages containing quoted material (Salton and Buckley, 1991). However, Salton and Buckley were attempting to find related messages, not just parent messages, and defined all messages with the same Subject line as being related. This is a simpler task than finding the single parent of a message.

The other strategies we tried were not as effective as matching quoted text against unquoted targets, though all were far better than random at finding parent messages. Even matching unquoted text queries against quoted text targets, which preferentially retrieves the children of a message rather than the parent, still returns a nontrivial number of parents based on general content similarity. Similarly, quoted queries against quoted targets mostly should find siblings of a message, but gets some parents due to nested quotations that persist to the child.

Matching on Subject lines is an interesting strategy to examine, since variants on this method are used in several message processing agents. This strategy is only fourth best at finding the parent message at rank 0, but is second best at finding the parent somewhere in ranks 0 to 4 or 0 to 9 (Table 1). The reason is that the Subject line of a child matches that of other children better (due to *Re:* tags) than it matches the Subject line of its parent. Once the fellow children are passed in the ranking, the parent is quickly found. Better processing of *Re:* tags (see next section) would make the curve for this method look more like that for quoted query vs. unquoted target (Figure 2). The quoted vs. quoted case could similarly be sharpened up by proper processing of nested quotations. As expected, the unquoted vs. unquoted curve in Figure 2 shows a smooth curve similar to that seen with similarity-based text retrieval systems.

6 Future Work

By improving our processing of document text, as well as by making use of additional evidence, we believe that the above results can be greatly improved. Some strategies we plan to implement are:

Better Text Representation. Our experiments ignored the order of words when matching query messages against potential parents. This is sensible for detecting similarity of topic, as is our goal in matching unquoted text against unquoted text. A quotation in a child message, however, is likely to repeat a long sequence of words from the parent. Indexing, matching, and term weighting based on multi-word phrases or entire lines should greatly reduce the number and strength of spurious matches. Since header material (From lines, etc.) can appear in quotes as well, matching should be allowed on this material as well as on body text.

Nested Quotation. Multiple levels of quotation are common in electronic messaging, and are indicated by concatenated prefixes. For instance, if textual material is prefixed by ">> >", we would expect that the parent message has the material prefixed by "> >", or perhaps by ">", but probably not by nothing and certainly not by "|" or "*". Similarly, concatenated *Re:* tags play a role as quotation markers in Subject lines. All quotation markers should be characterized statistically rather than deterministically, since their use by mail agents is erratic.

Time. Most replies to a message occur within a window of a few days after the message is posted. A simple statistical model, perhaps similar to those used in analyzing citation patterns (Egghe and Rao, 1992), could be used to take this tendency into account. To use temporal information, however, anomalies introduced by unsynchronized clocks and nonstandard uses of time zones must be dealt with.

Recognizing Other Message Relationships. Duplicated, bounced, reposted, continued, and revised messages have strong textual similarity to other messages. If treated simply as nonreplies they are likely to distort statistical models distinguishing replies from nonreplies. A better approach is to model these relationships separately and explicitly, both to distinguish them from response relationships and to provide additional useful links between messages. For instance, a mail reader might display a revised message while backgrounding the original.

Authorship Information. Replies often refer to the author of the parent message, either in an automatically produced fashion:

> *lewis@research.att.com (David D. Lewis) writes:*
> > *I'd really like a threading email reader.*

or via a manually written salutation (e.g. *Dear Susan*). These may be matched against header information of messages and manually or automatically produced signatures.

Cue Phrases. In responses which do not directly quote the parent message, the author will often use linguistic cues to indicate the parent message, e.g. *I really like the suggestion that...* or *Your argument is....* Considerable research has been done on distinguishing what relationship a particular cue phrase is indicating (Litman and Passonneau, 1995).

Message Categorization. Certain types of messages such as calls for papers and job ads are unlikely to be replies to other messages and/or are unlikely to be replied to publically. Text categorization methods (Lewis and Hayes, 1994) can detect these and provide evidence against the presence of response links.

Detection of Siblings. A message without a clear connection to its parent may be similar to another child of the same parent, which does have a clear link. (For instance, two people may post similar responses objecting to an error in the parent message, with only one of the responders using the reply command.)

All of the above clues provide evidence toward the presence or absence of response links, but in all cases this evidence is uncertain. Our strategy is to implement each clue so as to reduce its uncertainty as much as is reasonable. We will then rely on machine learning methods to combine these multiple uncertain clues into a decision procedure. This is a widely used strategy in speech and language processing, and has been successfully applied to complex IR problems as well (Croft et al., 1994, Spertus, 1996). A machine learning approach also allows the system to be tailored to user preferences as expressed, for instance, through their overriding of system decisions. This is desirable, since the presence of a response link is to some degree subjective.

7 Summary

The usual response to problems with threading in messaging systems is to try to force more consistency in the use of structural links by client software. We instead suggest that threading in the fashion that users desire is unlikely to be achieved by enforcing consistency on a chaotic messaging environment. Instead, threading software should attempt to operate as human readers do, making use of a wide range of individually uncertain, but cumulatively compelling clues as to what is going on in a conversation.

Acknowledgments

We thank Jamie Blustein, Dan Dabney, Parni Dasu, Jay Glicksman, Mark Hansen, Jon Helfman, Julia Hirschberg, Alan Jaffray, Mark Jones, Tom Kirk, Leonid Libkin, Diane Litman, Albert Lunde, Tim Pierce, Ellen Riloff, and Don Swanson, along with the anonymous reviewers, for helpful comments and suggestions on this work.

References

Allan, J. (1996). Automatic hypertext link typing. In *Hypertext '96 Proceedings*, pages 42–52, Washington, DC. ACM Press.

Bernstein, M. (1990). An apprentice that discovers hypertext links. In *Hypertext: Concepts, Systems, and Applications. Proceedings of the European Conference on Hypertext*, pages 212–223, France. INRIA.

Blustein, W. J. (1993). An evaluation of tools for converting text to hypertext. Master's thesis, The University of Western Ontario, London, Ontario.

Crocker, D. H. (1982). Standard for the format of ARPA internet text messages. Request for Comments 822, University of Delaware, Newark, DE.

Croft, B., Callan, J., and Broglio, J. (1994). TREC-2 routing and ad-hoc retrieval evaluation using the inquery system. In Harman, D. K., editor, *The Second Text Retrieval Conference (TREC-2)*, pages 75–83, Gaithersburg, MD. U. S. Dept. of Commerce, National Institute of Standards and Technology. NIST Special Publication 500-215.

Egghe, L. and Rao, I. K. R. (1992). Citation age data and the obsolescence function: Fits and explanations. *Information Processing and Management*, 28(2):201–217.

Halliday, M. A. K. and Hasan, R. (1976). *Cohesion in English*. Longman, London.

Hearst, M. A. (1994). Multi-paragraph segmentation of expository text. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 9–16, Las Cruces, New Mexico.

Horton, M. and Adams, R. (1987). Standard for interchange of USENET messages. Request for Comments 1036, AT&T Bell Laboratories.

Hughes, K. (1995). Hypermail 1.02. *http://www.eit.com/software/hypermail*.

Jones, K. E. (1991). VM User's Manual. Second Edition, VM Version 5. File vm.texinfo in version 5.95 (beta) VM distribution. Latest VM distribution available at *ftp.uu.net* in the *networking/mail/vm* directory.

Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In Croft, W. B. and van Rijsbergen, C. J., editors, *SIGIR 94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, London. Springer-Verlag.

Lewis, D. D. and Hayes, P. J. (1994). Guest editorial. *ACM Transactions on Information Systems*, 12(3):231.

Litman, D. J. and Passonneau, R. J. (1995). Combining multiple knowledge sources for discourse segmentation. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 108–115, Cambridge, MA.

Myka, A. and Guntzer, U. (1995). Automatic hypertext conversion of paper document collections. In Adam, N. R., Bhargava, B. K., and Yesha, Y., editors, *Digital Libraries: Current Issues. Digitial Libraries Workshop DL '94.*, pages 65–90, Berlin. Springer.

Rose, M. T. (1993). *The Internet Message: Closing the Book with Electronic Mail.* Prentice-Hall, Englewood Cliffs, NJ.

Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, Reading, MA.

Salton, G., Allan, J., and Buckley, C. (1994a). Automatic structuring and retrieval of large text files. *Communications of the ACM*, 37(2):97–108.

Salton, G., Allan, J., Buckley, C., and Singhal, A. (1994b). Automatic analysis, theme generation, and summarization of machine-readable texts. *Science*, 264:1421–1426.

Salton, G., Allan, J., and Singhal, A. (1996). Automatic text decomposition and structuring. *Information Processing and Management*, 32(2):127–138.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

Salton, G. and Buckley, C. (1991). Global text matching for information retrieval. *Science*, 253:1012–1015.

Spertus, E. (1996). Smokey: Automatic flame recognition. Manuscript.

Wiener, E., Pedersen, J. O., and Weigend, A. S. (1995). A neural network approach to topic spotting. In *Symposium on Document Analysis and Information Retrieval*, pages 317–332, Las Vegas, NV. ISRI; Univ. of Nevada, Las Vegas.

```
From maria@foobar.baz.com Sat Feb 5 20:33:13 1994
Return-Path:  <maria@foobar.baz.com>
To:  www-talk@www0.cern.ch
Subject:  Re:  pages in multiple languages (subject)
Date:  Sat, 05 Feb 1994 13:15:44 -0600
From:  Maria Jones <maria@foobar.baz.com>

Bob Smith <smith@somewhere.what.org> writes: (unquoted)

> Can anyone recommend some good examples of pages (quoted)
> with text in multiple languages? (quoted)

Sure, see the list of Japanese pages on my home page: (unquoted)
   http://www.baz.com/maria/index.html (unquoted)
Cheers, Maria (unquoted)
```

Figure 1: A hypothetical message from our data set, with comments in italics showing subject text, quoted text, and unquoted text.

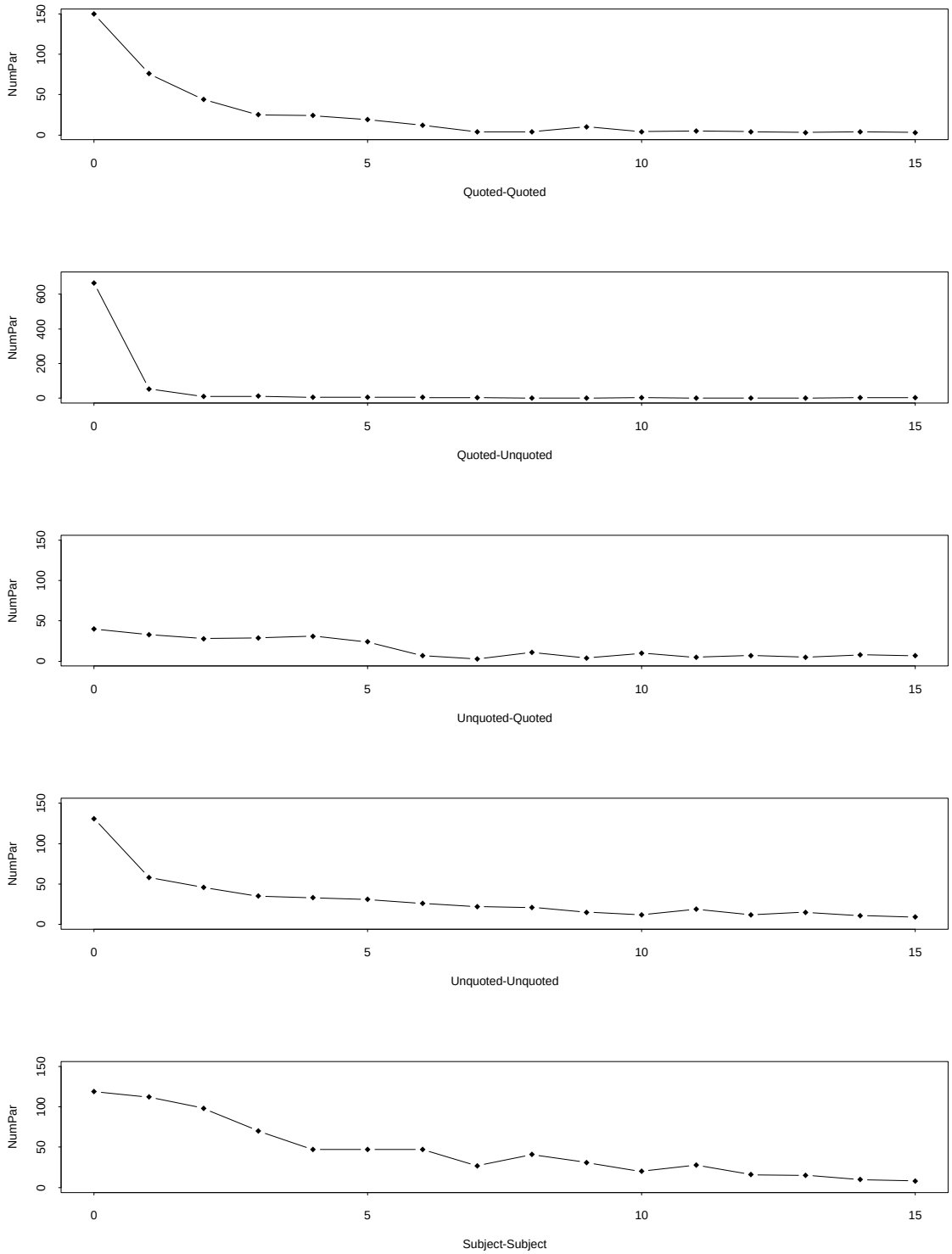

Figure 2: Number of times parent message is retrieved at ranks 0 through 15, for each of five search strategies. Note different scale on y-axis for second graph.

Ranks	Search Strategy					
	Random	Subj-Subj	Unquot-Unquot	Unquot-Quot	Quot-Unquot	Quot-Quot
0	0.39	119	131	40	666	150
0-4	1.93	446	303	161	745	319
0-9	3.87	639	418	210	759	368

Table 1: Number of parents retrieved at rank 0, ranks 0 to 4, and ranks 0-9 for each of the search strategies, over 941 trials. Figures for random retrieval are expected values.