Large Volume Email Retrieval

Shivaan Motilal
Computer Science Honours
University of Cape Town
mtlshi005@myuct.ac.za

ABSTRACT

When searching through large volumes of electronic mail(email) from an archive, email users can spend much time and effort trying to locate a specific email. This can be further aggravated when the user does not have much information on the sender of the email or the email contents themselves. This paper gives an overview of and compares various existing methods to extract email from large archives. These methods are differentiated in terms of their applications. Preservation of digital objects within the email archives, offline digital collections and longevity of email systems, are also mentioned as additional considerations in large volume email retrieval.

Keywords: email, search, volumes, retrieval, organize, methods, retrieval, efficiency, offline, enduring, preservation, longevity

1. INTRODUCTION

Email users make use of their email inboxes, in the process of communicating information. Sometimes the information is unimportant and can be deleted, whilst at other times, in the case of work-related emails, the information is important and needs to be stored by the user. Over time, a large amount of important information accumulates as an archive, and it becomes cumbersome to retrieve specific emails from that archive. Whittaker et al.[45] termed this as "email overload".

With email overload, a need arises for an efficient method to search through the archive. This need is addressed as part of the research field of information retrieval. Information retrieval is obtaining data (usually documents), in an unstructured form (usually text), that satisfies an information need from within large collections [34]. The purpose of this paper is to address the aforementioned need, by assessing information retrieval methods applied to large volumes of email.

2. EMAIL RETRIEVAL METHODS

2.1. Vector Space Model

The vector space model represents text as a vector of *terms*. Terms generally being words of phrases. A term is denoted as a dimension in a multidimensional vector space. Any text would be considered a vector in this dimensional space(*text-vector*). Terms that belong to a text receive a non-zero value, for each text-vector corresponding to the term [38].

A general vector space model attaches weightings to documents and queries, representing them as vectors [33]. Rankings are later calculated from these vectors. The ranking formulas proposed by Salton, G [33], are rarely used in recent times, but documents and queries are still commonly considered vectors in a high dimensional space[34]. See 2 for some general applications of the Vector Space Model in information retrieval.

If the vector space model is applied specifically to email systems, target email messages(emails from the corpus) can be represented as vectors with numeric weights:

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

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

Where , and *fik* is the number of times word *k* appears in email message *i*. Query messages were also represented as vectors:

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

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

Where , with *fk* as the number of times the word occurs in the query message, *N* as the number of messages in the database, and *nk* as the number of messages containing word *k* [19].

The target email messages are ranked using the cosine similarity formula and a variant of tf-idf weighting [33]. The *tf-idf weighting* is expanded to *term frequency-inverse document frequency weighting*, and is a numerical statistic intended to reflect how

important a word is to a document, in a collection or corpus [31]. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus. This adjusts for the case where words appear more frequently in general [19].

These rankings for the target email messages, are determined through the calculation of a score for each target email message *i,* with the score calculate as follows:

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

The *j* used in the above formula is the same as the *k* mentioned earlier, with the formula basically summing the product of the target email message and query message weights.

This approach to retrieving email was designed to deal with threaded emails, which can be broken into parent and child messages, where the parent is the message that the child is a reply to [19]. The approach however, ignores the order of words, when matching query messages against potential parents.

To determine what text from a child should be used as a query, and what text from target email 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

It was found that unquoted text queries and quoted text targets, performed the best [19].

According to Lewis and Knowles[19], a more advanced vector space model could be created, if one were to incorporate: 1. Indexing, matching, and term weighting on multi-word phrases or entire lines, 2. Use of timestamps on email messages to group them, 3. Use of authorship information to identify parent messages, 4. Categorization of messages into "calls for papers", "job ads" etc. 5. Sibling(another child with same parent) detection, through use of machine learning techniques.

2.2. Lazy Graph Walk

Minkov, Cohen and Ng [21] used the Lazy Graph Walk algorithm for their email retrieval system. The Lazy graph walk algorithm is an adaption of the well-known PageRank Algorithm [26], and involves traversing a graph possessing random resets, with a fixed probability of halting on each step during traversal.

The traversal or walk, uses a start node and compares it to edges in the graph for similarity, along multiple connecting paths. The graph nodes correspond to entity types like email addresses and dates, and edges correspond to relations, like who email was "sent-by" or the date the email was sent [21].

A graph *G* incorporates a set of nodes, and has labeled directed edges. Nodes are represented by letters like *a, b, or c*, and an edge from *a* to *b* is labeled as $a \xrightarrow{\ell} b$. The entity types for an email corpus are shown in the far left of Table 1 [21].

source type	edge type	target type
file	sent-from	*person*
	sent-from-email	*email-address*
	sent-to	*person*
	sent-to-email	*email-address*
	date-of	*date*
	has-subject-term	*term*
	has-term	*term*
person	sent-from^{-1}	*file*
	sent-to^{-1}	*file*
	alias	*email-address*
	includes-term	*term*
email-address	sent-to-email^{-1}	*file*
	sent-from-email^{-1}	*file*
	alias^{-1}	*person*
	is-email^{-1}	*term*
term	has-subject-term^{-1}	*file*
	has-term^{-1}	*file*
	is-email	*email-address*
	includes-term^{-1}	*person*
date	date-of^{-1}	*file*

Table 1: Graph structure: Node and relation types

Every node *a* has a type, denoted *T(a)*, which we assume is a fixed set of possible types. There will be the assumption that there are no edges from a node to itself, for (modeling) convenience. An *inverse label* l^{-1} is also created for each edge label (relation). This means however, that the graph will be cyclic in nature[21].

2.3. Automatic Query Expansion(AQE)

Users often type short queries, when searching for information in information retrieval (IR) systems. In the case of an email system; a user might be looking to retrieve past correspondence with vague information on what they are searching for. This could be due to the correspondence having been done a long time in the past, or the brief nature of the correspondence making it harder to remember. These queries that contain only a few words, are termed *lazy queries* [27]. Query formulation and reformulation is regarded as one of the most difficult tasks in information retrieval.[34]

Automatic query expansion involves analyzing all the text related to the search query, to compile a specific thesaurus. The thesaurus is then used to expand the query, with words or phrases of similar meaning to those in the query [49]. This increases the chances of finding relevant documents that contain words of similar meaning to the query.

There are four main approaches to automatic query expansion [28]:

1. Using *plain co-occurrence* data:

   This involves calculating the similarities between terms(words; keywords or phrases in query or document), using the association hypothesis(correlation test) and thereafter classifying terms by their similarity threshold. This is finding the statistical co-occurrence of terms [8]. The set of index terms is subdivided into classes of similar terms. A query is then expanded by adding all the terms of the classes that contain query terms [28].

   This automatic query expansion method utilizing statistical co-occurrence data, can result in significant improvement in the retrieval effectiveness, when measured using both recall-precision and usefulness. A consistent performance improvement was noted [28].

2. Using *document classification:*

   In the case of an email system, this would refer to email classification or ranking. Documents are first classified using a document classification algorithm, then infrequent terms found in a document class are considered similar and clustered in the same term class (thesaurus class) [12]. The terms in the query are interchanged with a thesaurus class, or a thesaurus is added to the search query.

   Due the better quality of the thesaurus for large collections (in comparison to smaller ones), the number of additional terms searched for increases with the size of the archive [28]. This may result in a more specific search result, but there are issues that were noted in past papers with regards to retrieval time and efficiency. An illustration would be when El-hamdouchi, A [8] tested the CLINK algorithm(a document classification method), on large document collections, and found the retrieval efficiency to considerably poor.

3. Using *syntactic context*:

   The relations between terms in the queries are formed using linguistic knowledge and co-occurrence statistics [13].

   This method uses a defined grammar and a dictionary data structure, to extract a list of terms for each term t in the query(t is the query term). A list consists of all terms that relate to the query term t (called modifiers). Similarities between query terms and modifiers are then calculated. Subsequently, a query is expanded by adding those modifiers from the lists, most similar to any of the query terms [28]. This method however, produces only slightly better results, than when using the original unexpanded queries [13].

4. Using *relevance information*:

   This falls into the category of *relevance feedback methods,* where important terms or expressions relevant to the user, that are attached to previous retrieved documents, are chosen and their importance enhanced in future queries [32].

   Relevance information is used to construct a global information structure, like a pseudo thesaurus [30, 31] or a minimum spanning tree [31].

3. APPLICATION OF RETRIEVAL METHODS

In general information retrieval(IR), the vector space model is used for the retrieval of documents and in web search[3, 7, 10, 33, 47], with not many applications for large volume email retrieval systems[3, 13, 43]. The vector space model can be applied to threaded email messaging systems, and provides the potential to better model relationships among emails for search-related tasks, after the suggested advanced features of the model are added on [19].

For IR, the lazy graph walk method can be used to estimate word dependency distributions[22]. Specifically applied to email, the lazy graph walk method has several advantageous applications for search-related tasks. Firstly, it can be used to preserve the entity type (linked to preservation of archived data), to handle a broad range of problems as typed search queries(including name disambiguation and threading). Secondly, it models relationships between structures, to provide a unified framework for integration of multiple types of information, including social network information, images, text and timelines within emails. Name disambiguation, and email threading are other applications of the lazy graph walk method [21].

AQE is fully automatic, meaning that the user is not involved in the decision of what addition terms are added to the search. It can be applied as the first run in an IR system, when no relevance information is yet available [34]. In case relevance information is available, feedback techniques could also be introduced to retrieve even more relevant documents [32].

There is however a side effect of AQE: as the retrieval results become more relevant and the archive larger (millions of emails),

the size of the thesaurus created becomes huge. The construction of the similarity thesaurus in this situation, could therefore be considered too computationally expensive [28].

4. RELATED WORK

5.1. Preservation of Digital Objects

Bellinger et al.[4] stated that a digital object is "an information object, of any type of information or any format, expressed in digital form." An information object encapsulates any piece of information or data.

Preservation of digital objects(including e-mails within archives), refers to preserving the structure and format of the original object. However, it does not entail maintaining all the digital attributes(eg. size, variety and complexity etc.) of the digital object. To preserve a digital object, the relationships between levels (in the structure), must be known or knowable by the system [4].

One particular way of preserving email consists of using simple email formats, including mbox, mdir, PST, NSF and Groupwise [40]. The most commonly used formats however, are Mbox and Mdir [18]. Mbox [17] stores the entire email archive as a single file, using one file lock to synchronize access to folders in that file. Mdir does not use locks, but instead represents mail folders as a directory and email messages as files. Metadata and mail flags are used in mdir to control access [18].

Another way involves the use of simple protocols to access email, such as the Post Office Protocol (POP) and Internet Message Access Protocol (IMAP). POP [24] is used to retrieve mail from a server and provides basic mail manipulation operations. IMAP is used by gmail [23] and expands on some the features of POP. It allows for offline email operations, that are later synchronized with the server and allows concurrent email clients (devices like laptops, smartphones etc.) to connect to the same inbox. With concurrent email access in IMAP, there is still the problem of duplicating or losing and email, from modifying a message or folder simultaneously.

For email archives, the original contents of the emails in the archive, needs to be preserved after retrieval. The retrieval should not alter the format or structure of the actual archive. The relationships between levels in the archive, would become unknown by the system thereafter [4].

5.2. Email System Longevity

Over time, the email system software becomes obsolete, as software aging [9] occurs. This can be attributed a number of factors including: insufficient updates; program size; reduction in performance; increasing user expectations and errors introduced through changes. Longevity of the system is thus a concern.

Some strategies to increase longevity of the email software, and decrease the rate of software aging include:

1. *Object orientation*: this involves having items that are most likely to change, constitute a small part of the code. In this way, when changes occur, less code would need to be discarded.

2. *Reducing restructuring:* whereby there is less reorganization of system components and less grouping of similar components [2].

3. *Modularity:* This refers to the degree of separability of the components of a system and their ability to be put back together[35]. If the email system could more easily be broken down into parts, then it would be simpler to reuse components of the system, even if the system as a whole no longer works.

4. *Improving Personal Information Management(PIM):* This refers to creating relevant categorizations of email such as: *active* (currently being worked upon); *dormant* (inactive but potentially useful); *not useful* and *un-assessed* (not viewed) [6].

The above longevity considerations should be factored in, when creating an email system aimed to be used in the long-term.

5.3. Offline Digital Collections

Poor internet connectivity in South Africa and other developing countries, has resulted in the attention of researchers being drawn to offline digital collections. Offline digital collections are any sort of offline information store, including backed-up email archives or databases [16]. These archives tend to need preprocessing, as offline collections may be slower than server-based systems [42].

The vector space model shown earlier, was tested on offline databases of email messages and considered the format of online internet email messages in its derivation [21]. The Lazy graph walk method was tested on offline collections only [15]. The various AQE methods, have been tested on both offline and online systems and is widely researched [8, 11, 39].

Suleman et al.[42] stated that online and offline systems present advantages and disadvantages, and a hybrid system(online-offline repository) would be a better alternative to both.

5. SUMMARY

There are many papers available on information or document retrieval, but not many of these papers touch on the topic of retrieval of emails. Even so, this paper presents a few effective solutions, majority of which requiring the creation of new data structures, like graphs, in their attempt to retrieve relevant results for the user. With the improvement in relevance of the retrieved results, the models themselves become more complex and issues later arise [1].

While large volume email retrieval methods are the focus of this paper, there are other considerations like preservation of the data within the email archive, and obsolescence of the email system, that also have an impact on the retrieval of email from large archives. In some cases the system

6. ACKNOWLEDGMENTS

I would like to acknowledge the contributions of Professor Hussein Suleman. He provided much guidance on what to focus on when doing research for this article.

7. REFERENCES

[1] Allan, J., Aslam, J., Belkin, N., Buckley, C., Callan, J., Croft, B., Dumais, S., Fuhr, N., Harman, D., Harper, D.J. and Hiemstra, D., 2003, April. Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, University of Massachusetts Amherst, September 2002. In ACM SIGIR Forum (Vol. 37, No. 1, pp. 31-47). ACM.

[2] Arnold, R.S., 1989. Software restructuring. *Proceedings of the IEEE*, *77*(4), pp.607-617.

[3] Bagga, A. and Baldwin, B., 1998, August. Entity-based cross-document coreferencing using the vector space model. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1 (pp. 79-85). Association for Computational Linguistics.

[4] Bellinger, M., Campbell, L., Hedstrom, M., Marcum, D., Thibodeau, K., Waters, D., van der Werf, T. and Webb, C., 2002. The state of digital preservation: An international perspective. Council on Library and Information Resources, pp.4-11.

[5] Berry, M.W. and Browne, M., 2005. Email surveillance using non-negative matrix factorization. Computational & Mathematical Organization Theory, 11(3), pp.249-264.

[6] Boardman, R. and Sasse, M.A., 2004, April. Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 583-590). ACM. Vancouver.

[7] Brin, S. and Page, L., 1998. The anatomy of a large-scale hypertextual web search engine. Computer networks and ISDN systems, 30(1-7), pp.107-117.

[8] Buckley, C., Salton, G., Allan, J. and Singhal, A., 1995. Automatic query expansion using SMART: TREC 3. NIST special publication sp, pp.69-69.

[9] Buschmann, F., Henney, K. and Schimdt, D., 2007. *Pattern-oriented Software Architecture: on patterns and pattern language* (Vol. 5). John wiley & sons.

[10] Castells, P., Fernandez, M. and Vallet, D., 2007. An adaptation of the vector-space model for ontology-based information retrieval. IEEE transactions on knowledge and data engineering, 19(2).

[11] Chum, O., Philbin, J., Sivic, J., Isard, M. and Zisserman, A., 2007, October. Total recall: Automatic query expansion with a generative feature model for object retrieval. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (pp. 1-8). IEEE. Vancouver

[12] Crouch, C.J., An approach to the automatic construction of global thesauri, Information Processing & Management, 26(5): 629-40, 1990.

[13] El-Hamdouchi, A. The Use of Inter-Document Relationships in Information Retrieval. PhD thesis, University of Sheffield, England; 1987.

[14] Fisher, D., Brush, A.J., Gleave, E. and Smith, M.A., 2006, November. Revisiting Whittaker & Sidner's email overload ten years later. In Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work (pp. 309-312). ACM.

[15] Grefenstette, G., Use of syntactic context to produce term association lists for retrieval, SIGIR'92, 15th Int. ACM/SIGIR Conf. on R&D in Information Retrieval, Copenhagen, Denmark, 89-97, June 1992.

[16] Gunter, B., Nicholas, D., Huntington, P. and Williams, P., 2002, August. Online versus offline research: implications for evaluating digital media. In Aslib Proceedings (Vol. 54, No. 4, pp. 229-239). MCB UP Ltd.

[17] Hall, E.A., 2005. The application/mbox media type.

[18] Kim, S., Lee, M.Z., Dunn, A.M., Hofmann, O.S., Wang, X., Witchel, E. and Porter, D.E., 2012, April. Improving server applications with system transactions. In Proceedings of the 7th ACM european conference on Computer Systems (pp. 15-28). ACM.

[19] Lewis, D.D. and Knowles, K.A., 1997. Threading electronic mail: A preliminary study. Information processing & management, 33(2), pp.209-217.

[20] Liebchen, C., Lübbecke, M., Möhring, R. and Stiller, S., 2009. The concept of recoverable robustness, linear programming recovery, and railway applications. In *Robust and online large-scale optimization* (pp. 1-27). Springer, Berlin, Heidelberg.

[21] Minkov, E., Cohen, W.W. and Ng, A.Y., 2006, August. Contextual search and name disambiguation in email using graphs. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 27-34). ACM.

[22] Minkov, E. and Cohen, W.W., 2007, August. Learning to rank typed graph walks: Local and global approaches. In Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis (pp. 1-8). ACM.

[23] McGill, T.M., 2009. Gmail as institutional memory: Archiving correspondence in the cloud. College & Research Libraries News, 70(11), pp.638-645.

[24] Myers, J. and Rose, M., 1996. Post office protocol-version 3 (No. RFC 1939).

[25] Nour, M.M., Williams, K. and Suleman, H., 2012, November. Evaluating Simple Repository Deposit for Open Educational Resources. In International Conference on Asian Digital Libraries (pp. 289-298). Springer, Berlin, Heidelberg.

[26] Page, L., Brin, S., Motwani, R. and Winograd, T., 1999. *The PageRank citation ranking: Bringing order to the web*. Stanford InfoLab.

[27] Perkio, J., Tuulos, V., Buntine, W. and Tirri, H., 2005, September. Multi-faceted information retrieval system for large scale email archives. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM*

[28] Qiu, Y. and Frei, H.P., 1993, July. Concept based query expansion. In Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 160-169). ACM.

[29] Raghavan, V.V. and Wong, S.M., 1986. A critical analysis of vector space model for information retrieval. Journal of the American Society for information Science, 37(5), p.279.

[30] Salton, G., Experiments in automatic thesaurus construction for information retrieval, Information Processing 71, 1: 115-123, 1971.

[31] Salton, G., Automatic term class construction using relevance-a summary of work in automatic pseudoclassification, Information Processing & Management, 16(1): 1-15, 1980.

[32] Salton, G. and Buckley, C., 1990. Improving retrieval performance by relevance feedback. *Journal of the American society for information science*, *41*(4), pp.288-297.

[33] Salton, G., Wong, A. and Yang, C.S., 1975. A vector space model for automatic indexing. Communications of the ACM, 18(11), pp.613-620. Vancouver.

[34] Sanderson, M. and Croft, W.B., 2012. The history of information retrieval research. Proceedings of the IEEE, 100(Special Centennial Issue), pp.1444-1451.

[35] Schilling, M.A., 2000. Toward a general modular systems theory and its application to interfirm product modularity. *Academy of management review*, *25*(2), pp.312-334.

[36] Schuff, D., Turetken, O. and D'Arcy, J., 2006. A multi-attribute, multi-weight clustering approach to managing “e-mail overload”. *Decision Support Systems*, *42*(3), pp.1350-1365.

[37] Serban, G. and Czibula, I.G., 2007, November. Restructuring software systems using clustering. In *Computer and information sciences, 2007. iscis 2007. 22nd international symposium on* (pp. 1-6). IEEE.

[38] Singhal, A., 2001. Modern information retrieval: A brief overview. IEEE Data Eng. Bull., 24(4), pp.35-43.

[39] Smeaton, A.F., van Rijsbergen, C.J., The retrieval effects of query expansion on a feedback document retrieval system, The Computer Journal, 26(3): 239-46, 1983.

[40] Solutions, A.S. and Strength, A.E., 1999. Enterprise IT. Skin, 100, p.R35.

[41] Suleman, H., 2011. An African Perspective on Digital Preservation. In Multimedia Information Extraction And Digital Heritage Preservation (pp. 295-306).

[42] Suleman, H., Bowes, M., Hirst, M. and Subrun, S., 2010, October. Hybrid online-offline digital collections. In Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (pp. 421-425). ACM.

[43] Ullman, J.D., 2011. *Mining of massive datasets*. Cambridge University Press.

[44] Whittaker, S., Bellotti, V. and Gwizdka, J., 2006. Email in personal information management. *Communications of the ACM*, *49*(1), pp.68-73.

[45] Whittaker, S. and Sidner, C., 1996, April. Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 276-283). ACM.

[46] Willett, P., 1988. Recent trends in hierarchical document clustering: a critical review. Information Processing & Management, 24(5), pp.577-597.

[47] Wong, S.M., Ziarko, W. and Wong, P.C., 1985, June. Generalized vector spaces model in information retrieval. In Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 18-25). ACM.

[48] Wrobel, S., 1996. First order theory refinement. *Advances in inductive logic programming*, *32*, pp.14-33.

[49] Xu, J. and Croft, W.B., 1996, August. Query expansion using local and global document analysis. In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 4-11). ACM. Vancouver.

[50] Zeng, C., Lu, Z. and Gu, J., 2008, December. A new approach to Email classification using Concept Vector Space

Model. In Future Generation Communication and Networking Symposia, 2008. FGCNS'08. Second International Conference on (Vol. 3, pp. 162-166). IEEE.