Development and evaluation of clustering techniques for finding people

M. D. Dunlop[1]
Centre for Human-Machine Interaction
Systems Analysis Department., Risø National Laboratory, P.O. Box 49
4000 Roskilde, Denmark
mailto:mark.dunlop@risoe.dk http://www.chmi.dk/people/mdd/

Abstract

Typically in a large organisation much expertise and knowledge is held informally within employees' own memories. When employees leave an organisation many documented links that go through that person are broken and no mechanism is usually available to overcome these broken links. This matchmaking problem is related to the problem of finding potential work partners in a large and distributed organisation. This paper reports a comparative investigation into using standard information retrieval techniques to group employees together based on their web pages. This information can, hopefully, be subsequently used to redirect broken links to people who worked closely with a departed employee or used to highlight people, say in different departments, who work on similar topics. The paper reports the design and positive results of an experiment conducted at Risø National Laboratory comparing four different IR searching and clustering approaches using real users' web pages.

1 Motivation

This paper addresses the problem of automatically matching people to other people in an organisation based on already existing written documents describing the work of the people. Two possible scenarios where this form of matching would be helpful were used as the main motivation behind this work:

In any organisation a large amount of information concerning large projects is typically not documented but held only in the staff's memories. Such undocumented information typically includes details of who was involved in a project in consultation roles, what the problems of the project were and how these were solved (final project documentation often only records the final result and not the process, which is arguably the most important information for reuse). In the engineering domain, Hertzum & Pejtersen [1999] have identified that people typically interleave searching for people with searching for documents: "we find that engineers search for documents to find people, search for people to get documents, and interact socially to get information without engaging in explicit searches". Furthermore, they identified that "design documentation seems to be biased toward technical aspects of the chosen solution, while information about the context of the design process is typically not available. Hence, people become a critical source of information because they can explain and argue about why specific decisions were made and what purpose is served by individual parts of the design". A problematic implication of this observation is that considerable knowledge about the context of a project is lost when staff leave an organisation. One of the main aims of this work is to support finding of colleagues retrospectively, based on automatically keeping records of who work together. Thus, for example, when a key document is written by S Jones who has since left the company, a record will be available of whose work was closest to Jones at the time the document was written. This colleague should

Proc. of the Third Int. Conf. on Practical Aspects of Knowledge Management (PAKM2000)
Basel, Switzerland, 30-31 Oct. 2000, (U. Reimer, ed.)
http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-34

[1] Now at: Computer Science Department, Univerity of Strathclyde, Glasgow, G1 1XH, Scotland
mailto: mark.dunlop@cs.strath.ac.uk
http://www.cs.strath.ac.uk/~mdd/

then, hopefully, be able to act as a contact point for finding other people and documents concerning the project.

In large and distributed organisations, it is highly possible that two people will have similar interests or be working on similar projects without begin aware of each other. As an example, many Universities have several locations in which related work may be carried out in different departments and from different backgrounds. It is hoped that the models presented in this paper will help to bring together people who have similar research interests or work areas. This has been identified as one of the major social implications of a move from physical libraries to digital ones: one no longer fortuitously meets fellow researchers at the shelves. In a preface to their work on supporting interaction with and awareness of others in digital libraries, Robertson and Reese [1999] state that "libraries are hubs for social and intellectual interactions in communities and organisations. Virtual libraries should serve the same purpose, yet virtual libraries often focus simply on making their holdings available". Twidale and Nichols [1998] give a short overview of, so called, matchmaking systems and link these with other work on computer supported co-operative work in information retrieval (expanded description of their work on matchmaking in digital libraries can be found in [Twidale and Nichols 1997]).

This paper reports an investigation into the use of information retrieval (IR) techniques to automatically match people based on their web pages. The resulting matches could be used to highlight people who are working closely together and this information could be used to highlight potential collaborations now, or used historically to point searchers to colleagues of staff who have, say, left the company. The paper starts by describing the IR and clustering techniques used in the experiments, then it describes the experimental framework, the results of the experiment and, finally, presents a discussion of potential extensions to the model.

2 The techniques

With the advent of large search engines over the World Wide Web, searching techniques are increasingly used to find people based on their name and, less frequently, based on similar research portfolios. The experiments reported here target finding people by similar topic and are based around testing four different solutions to the problem. These solutions can be classified into two categories: simple searching (one approach) and three approaches to cluster based matching. All of the approaches are based on indexing users' home pages using standard IR techniques. IR techniques have been developed over many years to support searching for documents [e.g. Van Rijsbergen 1979, Baeza-Yates & Ribeiro-Neto 1999].

2.1 Searching

The first technique used here to match people is based on indexing staff web pages then performing searches based on finding the most similar web pages for each user. For example, the content of user U_i's web page is used as the query for a search of all other staff home pages within an organisation, resulting in a ranked list, L, of those users who are closest to user U_i (i.e. L_1 is the closest person to U_i, L_2 the next closest, etc.) as defined by the content of their home pages.

The experiments were run using a baseline ranked IR engine developed in house using standard IR techniques [e.g. Salton & McGill 1983, Frakes & Baeza-Yates 1992, Sparck Jones & Willett 1997] and implemented in Java 1.1. Documents were indexed using:

- tf/idf weighting, which weights terms proportional to how often they occur in the current document but inversely to how often they occur in the collection as a whole [Sparck Jones 1972];
- a simple stop-word list based on the collection itself, the 30 most common words in the collection were not indexed;
- Porter's stemming algorithm, an algorithmic stemmer that conflates variants of a words into the same base form, e.g. *walking*, *walks* etc all conflate to *walk* [Porter 1980];
- and the cosine matching function, an IR standard that takes into account term weights and document lengths [Salton and McGill 1983].

The IR engine was designed to index web pages: it only indexes content baring sections, omitting HTML tags, and gives greater weight to words in the title of the page.

2.2 Clustering

Clustering techniques have long been used in IR to improve the performance of search engines, both in terms of timing and quality or results [e.g. Jardine and Van Rijsbergen 1971, Van Rijsbergen and Croft 1979 and Griffiths, Luckhurst and Willett 1986]. This work follows from the observation, known as *the cluster hypothesis*, that relevant documents are more like one another than they are to non-relevant documents [Van Rijsbergen & Sparck Jones 1973]. The work in this

paper investigated the use of clustering techniques to improve the performance of people matching. Three clustering techniques were used: balanced clustering, single link clustering and group average clustering. All these clustering algorithms are hierarchic agglomerative algorithms, meaning a hierarchical structure of clusters and sub-clusters is created by starting with small clusters and adding documents and merging clusters until a single cluster remains. The clustering techniques were used to produce a hierarchical clustering of the users, *H*, that, hopefully, has similar users grouped together on the lower levels of the hierarchy. Single link and group average were chosen for showing significantly different performance in comparative experiments for document retrieval [Griffiths, Luckhurst and Willett 1986] with balanced clustering being added following their observation that small clusters appear to be a strong factor in performance of clustering algorithms.

2.2.1 Balanced Clustering

In the balanced clustering approach each user U_i is grouped with another one or two users based on the similarity between the nodes (as defined by the same IR engine and indexing approaches used in the baseline searching method). These pairs or triples are then grouped together with the most similar other group based on the average vector for the groupings (this is essentially a balanced variation of group average clustering discussed later). Again these grouping are further grouped into pairs or triples until the second top level where a pair is forced. The process is more clearly described by the following pseudo code:

```
repeat
   take current set of documents or most recent set of clusters
   calculate all descriptor-descriptor comparisons
   insert descriptor-descriptor pairs into a list sorted by weight
   for each pair working down list
      if neither element has been assigned then
         record this pair as a new cluster
      else if both elements are assigned
          & both would prefer a higher cluster
          & those higher clusters are currently pairs
          & the higher clusters are different
         add both as a 3rd to appropriate higher clusters
      else
         ignore this pairing // they can't be assigned just now
until only one cluster remains
```

Although a relatively slow algorithm, this approach gives the following characteristics:

- the closest two documents are grouped together first, then the next available pair, and so on so that the strongest matches are honoured;
- documents are clustered in a 'stable' manner - clusters of three documents are permitted when both documents in a lower pair on the ranked list have stronger links with already paired documents;
- the resulting hierarchy is tight and fairly balanced with a maximum outage at any node of 3 and a normal minimum of 2 (occasionally single node clusters are formed)

2.2.2 Single Link Clustering

Single link clustering is based on creating a hierarchical tree by continually inserting an additional node that satisfies the following criteria:

- the new node is currently outside the hierarchy;
- of all similarities between nodes inside and outside the hierarchy, the new node is selected that has the strongest similarity. It is then added to the hierarchy at a level based on how strong the similarity is.

This approach is fairly fast and results in hierarchies where the closest nearest neighbours are at lower levels of the hierarchy. However, it leads to non-balanced clusters and does not yield a binary hierarchy – many node-node comparisons can have the same strength of similarity thus many documents can be linked at the same level in the hierarchy.

The implementation was based on the pseudo code shown below. The pseudo code based closely on that from [Voorhees 1996], where the reader is directed for a more complete description.

```
// initialise hierarchy and insert document one into it
for (i=2 to collectionSize)
   info[i].sim = 0;
   info[i].inHierarchy = false;
   info[i].nn = UNDEF;
currentID = 1;

// place the document having maximum similarity with
// a document in the hierarchy into the hierarchy until
// all documents are in the hierarchy
while (currentID ≠ UNDEF)
   info[currentID].inHierarchy = TRUE;
   ComputeSims(currentID);
   maxSim = 0; nextID = UNDEF;
   // update nearest neighbour for docs outside hierarchy
   for (i=1 to collectionSize)
      if (not info[i].inHierarchy)
         if (sims[i] > info[i].sim)
            info[i].sim = sims[i]; info[i].nn = currentID;
         if (info[i].sim > maxSim)
            maxSim = info[i].sim; nextID = i;
   if (nextID ≠ UNDEF)
      currentID = nextID;
```

2.2.3 Group Average Clustering

Group average link clustering is based on creating a hierarchical tree by initially creating a singleton cluster for each document and marking these as "active". The clustering then repeats the following until only one cluster remains active:

- merge the two clusters with most similar cluster representatives. Where the cluster representative is the mean vector of all document vectors in the cluster (with singleton clusters being self representing);
- make the new pairing active and the two clusters which formed the pair non-active.

Again, pseudo code and implementation were based on that extracted from [Voorhees 1996]:

```
//initialise
maxSim = 0;
for (i = 1 to collectionSize)
   //create singleton clusters
   info[i].representative = document[i].representative
   computeSim(i, nn, sim)
   info[i].nn = nn, info[i].sim = sim; info[i].size = 1;
   if (sim > maxSim)
      id1 = i; id2 = nn; maxSim = sim;
numActive = collectionSize;
for (i = 1 to numActive) active [i] = i;

//merge clusters until only 1 left or remaining sims are zero
while (maxSim > 0 & numActive >1)
   smaller = min(id1,id2); larger = max(id1,id2);
   info[smaller].centroid = mergeCentroids(smaller,larger);
   info[smaller].size = info[smaller].size + info[larger].size;
   a = index of larger in active;
   active[a] = active[numActive]; numActive--;
   mergeClusters(smaller, larger, maxSim)
   maxSim = 0;
   for (each cluster a in active)
      if (info[a].nn = larger | info[a].nn = smaller)
         findMaxSim(a, nn, sim);
         info[a].nn = nn; info[a].sim = sim;
      if (info[a].sim>maxSim)
         id1 = a; id2 = info[a].nn; maxSim = info[a].sim;
```

Group average clustering is slower than single-link clustering, but is known to produce better clustering for document retrieval, guarantees to produce binary trees and keep closely related documents together in the initial pairs. Group average is essentially a non-balanced, purely binary, version of balanced clustering.

2.2.4 Evaluation

For consistent comparison with simple retrieval, a list of matching documents was required for each user. For each user U_i, each node U_j in the hierarchy H was scored based on how far U_j was from U_i (based on counting how many intermediate internal nodes there are in the hierarchy on the path through the hierarchy from one leaf node to the other, see figure 1 for an example). The list L was then based on these distances, smallest highest in ranking.

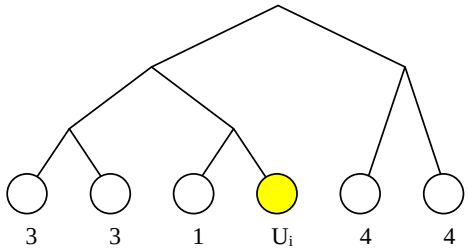

Figure 1: Distances from user U_i

The following four subsections describe in more detail the four approaches taken: searching, balanced clustering, single link clustering and group average clustering.

3 Experimental setting

To evaluate the performance of the different people finding algorithms, an experiment was conducted based on the web pages for the Systems Analysis Department at Risø National Laboratory. The Risø S.A.D. web contains home pages for 60 staff within the department. To complete the test collection, each member of the department was given a form in which they were asked to assess, on a four point scale, how closely they worked with each other person in the department. They were given the instructions to tick:

- "3 boxes for those you work very closely with;
- 2 boxes for those you work with;
- 1 box for those whose work is related mildly;
- 0 boxes if your work is unconnected (or you don't know who the person is!)."

To prevent biasing the staff towards defining these terms closer to definitions that would match the clustering algorithms, no explanation of the terms in the instructions were given (e.g. the terms "work with", "closely" and "unconnected" were left undefined).

A total of 27 forms were returned with a mean of 11.8 people marked per form (minimum 1, maximum 33, mean 11.76, standard deviation 6.74) and 21.6 ticks per form (min 2, max 53, mean 21.60, standard deviation 13.80).

Following standard IR practice, precision and recall figures were calculated. However, these were based on

relevance weight rather than the more common approach of simply counting how many relevant documents were found (following definition in [Reid 2000] for non-binary test collections). For these experiments, relevance weight is defined as how many ticks were marked divided by the maximum number of ticks (e.g. 0.333 for 1 tick and 1 for 3 ticks). This allows the results to highlight how well the system is at finding those people who users work closely with over those they simply work with. For a ranked list *L* with the best match at position 1, second at position 2 etc., precision and recall at position *p* were defined as follows:

$$\text{recall}_{i,p} = \frac{\text{Relevance weight in } L_1 \ldots L_p \text{ for user } U_i}{\text{Total relevance weight for user } U_i}$$

$$\text{precision}_{i,p} = \frac{\text{Relevance weight in } L_1 \ldots L_p \text{ for user } U_i}{p}$$

For consistency of evaluation, each algorithm produced a full ranking of all users (i.e. *L* contains an entry for every user in the collection bar U_i). Individual recall precision graphs were then combined using standard macro-evaluation as defined in Van Rijsbergen [1979 pp 152-153].

4 Results

Figure 1 shows the results for the four algorithms. It clearly shows that for this collection clustering-based approaches are more effective at matching people than simple searching and that overall, for basic IR techniques, the performance of the system is good. Of the clustering algorithms, group average performs best for low recall (0..0.27 approximately). This is the region in which people matching programmes are most likely to have their main impact – usually looking for one or two substitute names rather than, say, 70% of colleagues. However, balanced retrieval is only slightly poorer and, probably because of the more balanced hierarchy leading to more normalised distances within the tree, is better than Group Average Clustering as the recall levels are increased.

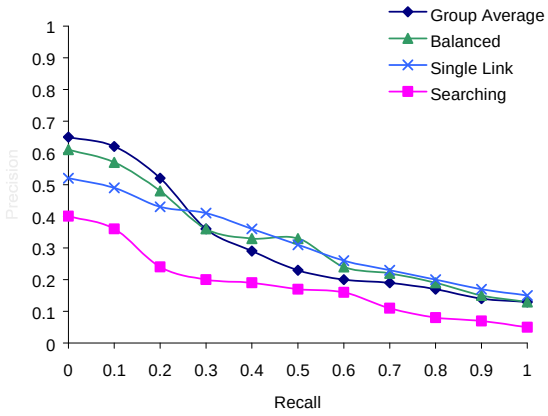

Figure 1: Results from experiments with full lists

Figure 2 shows the same results plotted by ranked position – showing how many ticks were accumulated, on average, by each rank position up to the tenth rank position. This shows that for the best approach, group average clustering, an average of 1.72 ticks were found at rank position one (57% perfect) whereas the worst approach, straight searching, was only achieving 0.88 ticks (29% accuracy). For group average clustering, the success rate rose to an average of 3.08 ticks by rank position 2 – which could be considered as "one close colleague equivalent" within the first two suggestions of the system.

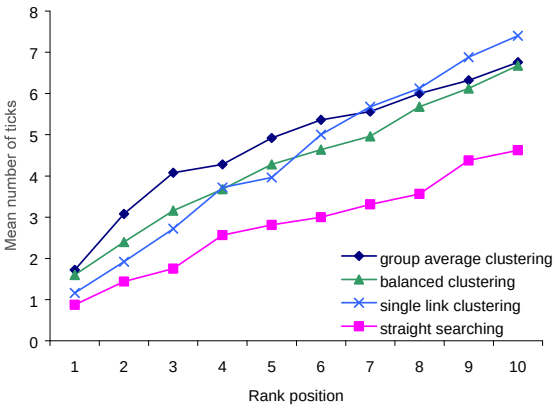

Figure 2: Mean number of ticks by rank position

To examine how resilient each approach was to storing limited information on people who each user works closely with, the lists *L* created by each of the four matching techniques were limited to nine elements each and the evaluation repeated (this simulates, say, a monthly recording of the nine closes people for each member of staff so involves storing 9*u* records as

opposed to u^2 records, where u is the number of users). Figure 3 shows that this results in a considerable drop in performance for all methods at higher recall levels, but relatively little drop in the 0...0.2 precision range. In particular the performance of group average clustering is almost unaffected in this range by storing limited information. Considering that the 0.2 recall point implies finding 20% of the colleagues and, in many settings, we are only likely to be looking for one/two - this is a promising result for reduced storage.

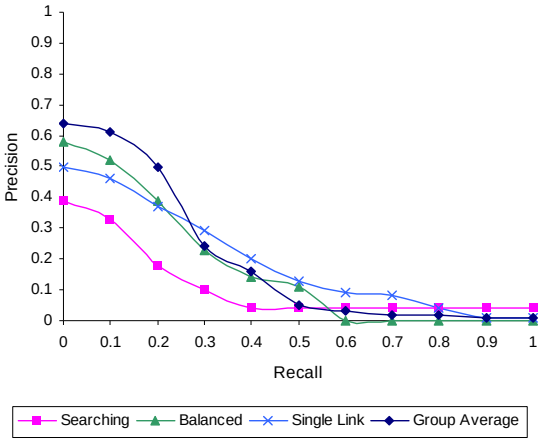

Figure 3: Results from experiments with lists limited to 10 entries

5 Extensions

As documents rarely exist in isolation, approaches to document retrieval that make use of the hypertext network in which documents are found [e.g. Dunlop 1991, Dunlop and Van Rijsbergen 1993, Frei, and Stieger 1992] could be used to augment home pages with connected documents (such as linked pages and subpages). These hypertext-IR techniques have been shown successful in improving retrieval for standard searching [Savoy 1996] and for accessing documents that cannot be indexed directly [Harmandas et al 1997]. As well as providing the usual benefits of hypertext-IR indexing, if used for indexing staff home pages, the approaches will also solve many of the problems of these pages. The Risø home pages used in this experiment were fairly consistent corporate style mini-CVs. Less consistent pages, such as those typically found in universities, have many problems such as frame based front pages (which don't actually have any content on the base home page URL), very simple front pages with linked pages and drastically different styles and quality and quantity of information provided. The use of hypertext-IR techniques could also bring in many other sources of evidence as to users' activities, such as documents written by the staff (possibly including weekly diaries, which are common in many engineering settings) and pages for organisations/activities the user is involved in.

One problem highlighted in the experiment reported here is that of assessing the different interpretations of "works with". For example, staff completing the data capture form were varied in how they reacted to secretaries being on the list of staff as well as other research colleagues and department managers. It would be worth investigation classification methods so that searches can be restricted, or at least rank positions affected by, matching users who are at roughly the same level in an organisation. This is likely to take place somewhat automatically as those users home pages are likely to contain more in common than users who are at drastically different levels in the organisation but these claims need further investigation.

As the motivational scenarios for this work did not require fast and frequent clustering of staff, only high quality clustering algorithms were considered. It may be worth investigating the use of faster and lighter methods, such as scatter/gather, to compare their performance with the tested algorithms.

6 Conclusions

The results of this experiment show that IR techniques can be used to match users home pages with those of other users to find colleagues who work in similar areas with a fair level of success. In the case of the experiment reported here, precision approximately 0.6 can be achieved for low levels of recall where the approach is most likely to be used. At rank position 1 an average of 1.72 ticks were found, where 2 ticks represents a staff declaration of someone that they work with (but not closely) – this rises to a total of 3.01 ticks by rank position two, equivalent to finding one close colleague in the first two suggestions from the system. Furthermore, use of limited length lists shows that storing only the nine closest predictions has little effect on the performance of the system at low recall while drastically reducing storage requirements for historical recording. The experiment compared straight IR searching to match users with potential colleagues with three different clustering approaches (balanced, single link and group average), all clustering approaches performed better with group average being the best overall (and noticeably more stable at low recall when using reduced length lists). This indicates that it is better to connect people based on the best overall arrangement (clustering approaches create a single best cluster hierarchy for the whole department then rank for individual users) rather than the best arrangement for

each individual person as performed in straight searching.

Further work is planned to investigate the results here in a corporate setting using different sources of documentation, over a longer timescale and using more users as the test base. Hypertext-IR approaches will also be investigated to see if they improve the effectiveness of the clustering approaches and make them more amenable to highly variable styles of home pages visible in some organisations.

Acknowledgements

Many thanks are due to the staff of Risø's Systems Analysis Department who completed questionnaires as part of the evaluation work.

The project was supported by the Danish Research Foundation's Centre for Human Machine Interaction.

References

Baeza-Yates, R., & Ribeiro-Neto, B., *Modern Information Retrieval*, ACM Press 1999.

Dunlop, MD, *Multimedia Information Retrieval*, PhD Thesis, Glasgow University Computing Science Research Report 1991/ R21, October 1991.

Dunlop, M.D., and Van Rijsbergen, C.J., "Hypermedia and free text retrieval", *Information Processing and Management*, vol 29(3), May 1993.

Frakes, W.B., and Baeza-Yates, R. (Eds), *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, 1992.

Frei H.P., and Stieger D., "Making use of hypertext links when retrieving information", *Proceedings ACM-ECHT'92*, Milan, Italy, pp. 102-111, 1992.

Griffiths, A., Luckhurst, C., and Willett, P., "Using inderdocument similarity information in document retrieval systems", *Journal of American Society for Information Science*, v37, p3-11, 1986 (reproduced in Sparck Jones and Willett 1997).

Harmandas, V., Sanderson, M., and M. D. Dunlop, M.D:, "Image retrieval by hypertext links", *Proceedings of SIGIR-97*, 1997.

Hertzum, M. and Pejtersen, A.M., "The information-seeking practices of engineers: Searching for documents as well as for people". To appear in *Information Processing and Management.*

Nichols, D.M., and Twidale, M.B., "Matchmaking and privacy in the digital library: striking the right balance", In *Proceedings of the 4th UK/International Conference on Electronic Library and Visual Information Research (ELVIRA 4), Milton Keynes*, Aslib: London, UK, pp 31-38, May 1997.

Porter, M. F., "An algorithm for suffix stripping", *Program*, v 14(3), pp130-137, July 1980 (reproduced in Sparck Jones and Willett 1997)

Reid, J., "A task oriented non-interactive evaluation methodology for information retrieval systems", *Information Retrieval*, v 2(1), 2000 to appear.

Robertson, S., and Reese, K., "A virtual library for building community and sharing knowledge", *International Journal of Human-Computer Studies*, vol 51, pp 663-685, 1999.

Salton, G., and McGill, M.J., *Introduction to modern information retrieval*, McGraw-Hill, 1983.

Savoy, J., "An Extended Vector-Processing Scheme for Searching Information in Hypertext Systems", *Information Processing and Management*, v32(2), 155-170, March 1996.

Sparck Jones, K., "A statistical interpretation of term specificity and its application in retrieval", *Journal of Documentation*, v28, pp 11-21, 1972.

Sparck Jones, K., and Willett, P. (Eds), *Readings in Information Retrieval*, Morgan Kaufmann, 1997.

Twidale, M.B., and Nichols, D.M., "Computer supported cooperative work in information search and retrieval", in Williams, E.M. (Ed), *Annual Review of Information Science and Technology (ARIST),*v33, pp259-319,Association of Information Science, 1999.

Van Rijsbergen, C.J. *Information Retrieval* (second edition). Butterworths, 1979.

Van Rijsbergen, C.J., and Sparck Jones, K., "A test for the separation of relevant and non-relevant documents in experimental retrieval collections", *Journal of Documentation*, v29, pp251-7, 1973.

Voorhees, E.M., *Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval*, Technical report TR 86-765, Department of Computer Science, Cornell University, July 1986.