Time, Relevance and Interaction Modelling for Information Retrieval

M. D. Dunlop

Computing Science, University of Glasgow, Glasgow G12 8QQ, Scotland.
mark@dcs.gla.ac.uk http://www.dcs.gla.ac.uk/~mark/

Abstract

The most common method for assessing the worth of an information retrieval (IR) system is through precision and recall graphs. These graphs show how precise an IR engine is when working at fixed levels of recall. This paper introduces *number-to-view graphs*, a new graphing method based on an early evaluation measure, which supplement precision-recall graphs by plotting the number of relevant documents a user wishes against the number of documents they would have to view to encounter them. The paper also proposes a step forward from number-to-view graphs that directly includes presentation, interface and temporal issues within the same framework as engine effectiveness: *time-to-view graphs*. Taken together, these graphs and models introduce a new evaluation approach called *Expected Search Duration*.

Introduction

Cleverdon (1966) identified the following list of six criteria that could be used to evaluate an information retrieval (IR) system:

Precision: fraction of retrieved documents that are relevant;

Recall: fraction of relevant documents that are retrieved;

Coverage: fraction of documents relevant to a query that are held in the document base;

Time lag: response time of the system to user requests;

Presentation issues and **User effort:** two closely related criteria covering the usability and interface design of the system.

Although there have been many experiments (e.g. Pejtersen [1996] and Robertson, Walker and Hancock-Beaulieu [1995]) that cover different aspects of this list, the major evaluation criteria in IR are precision and recall (focusing on retrieval effectiveness). While this focus has led to many improvements in the underlying IR engines (e.g. Harman 1995), there is an emerging consensus that more user oriented aspects of the retrieval process must now be included to achieve further improvements in IR system effectiveness for real world tasks (e.g. Dunlop [1996]). Research in IR can be separated into work on "user aspects" (e.g. user interface design, browsing based systems and understanding the meaning of the terms *relevance* and *information need* [e.g. Harman 1992, Campbell and Van Rijsbergen 1996, Agosti and Smeaton 1996, Mizzaro 1996]) and those working on "system aspects" (e.g. speed and effectiveness of a matching algorithm). This paper introduces a model of evaluation that includes aspects of both interface design and underlying performance, thus permitting a direct comparison of the benefits of an interface change with the benefits of changing the underlying engine.

System effectiveness is usually shown using precision-recall graphs, in order to plot these graphs it is necessary to have three sets of data:

1 a set of documents for the IR engine to search over;

2 a set of queries;

3 a set of relevance judgements stating which documents are relevant to each query.

Although these sets could be created by individual researchers for their own experiments this is costly and prevents direct comparison with the work of others. As such, a considerable effort has been put into creating standard test collections (e.g. CACM and TREC). In order to assess the quality of an IR engine a researcher would run the standard queries through their engine to produce a list of ranked documents that can then be *assessed* by noting where the judged relevant documents lie in this list. By working down the list a different recall and precision pair can be calculated at each relevant document to give a curve for a single query. The curves for all queries in the collection can then be averaged to produce an overall precision recall graph for the system [Van Rijsbergen 1979 Ch. 7]. Typically research hypotheses are tested by comparing the recall and precision graph for a system including the new research and that of an identical system bar the new approach.

Although this approach works well for evaluating underlying engines in a consistent manner, there are considerable assumptions made in the construction of such collections (e.g. experts will be able to assess relevance while not participating in a task that makes their decisions real). These assumptions have led to considerable criticism of recall-precision, especially when used for evaluating interactive IR systems (e.g. TREC discussion in [Dunlop 1996]). While the problems of users working with artificial queries and not their own information needs are significant, a more general problem is making evaluation in IR more realistic and closer to end users' tasks. In general, evaluation of human computer interaction (HCI) can be split into two broad categories: assessive and predictive.

Assessive evaluation is based around examining the performance of real users using a real system. Common techniques include think alouds, incident diaries, and statistically significant experiments (descriptions of these can be found in many HCI texts e.g. Preece et al [1994]). Although the "lightweight" approaches (e.g. think alouds and incident diaries) are very powerful during the formative stages of a project (in discovering "interaction bugs" and identifying aspects of the interface that commonly cause users

SIGIR 97 Philadelphia PA, USA

problems), they are of limited use in summative evaluation. On the other hand, to run experiments that achieve statistically significant results with real users using real systems is time consuming and expensive.

Predictive evaluation is concerned not with assessing user interaction, but in building models of interaction that allow system developers to predict user behaviour with a system, predict the change in behaviour given a change to the interface and use the model as a discussion tool for analysing the system's interface. Much of the work in predictive evaluation was carried out during the eighties at Rank Xerox's Palo Alto Research Center (PARC). Their approach to predictive evaluation is summed up by the following quote from Card and Moran [1986]:

> *Given a task (possibly involving several subtasks), the command language of a systems, the motor skill parameters of the user, the response time parameters of the system and the method used for the task, predict how long an expert user will take to execute the task using the system, providing he uses the method without error.*

This paper introduces a model of predictive evaluation in IR. This model should permit analysis of interface and engine performance characteristics within the same framework and should help in the development of design guidelines for IR interfaces. Initially a model of technical evaluation will be presented. This model will then be linked with an interface modelling approach to form the final predictive evaluation model.

Number-to-view graphs

As a first step we revisit an early alternative to precision-recall graphs: the *expected search length* measure of Cooper [1968] (also described in detail in Van Rijsbergen [1979] and Salton and McGill [1983]). Although this measure came under some criticism, which will be discussed later, it does provide a method of plotting engine performance in a manner that, I argue, is closer to many task oriented considerations and complements recall-precision graphs. This paper introduces a new graph based variation on expected search length (ESL) which plots the number of relevant items (r) against the number of items a user would have to view on average (v_r) in order to encounter r relevant items. These number-to-view (NTV) graphs are later used as a basis for predictive evaluation in IR.

Calculation

Given a standard test collection, a set of rank lists L can be calculated by matching each query in the collection against the documents in the collection, such that L_q represents the ranked list of documents for query q (taken from the set of queries Q). Associated with each rank list is the set R_q of documents that are known to be relevant to the query (as specified by the collection constructors).

The items that a user needs to view to find r relevant items for query q is initially defined as the shortest prefix of L_q, P_{qr}, such that P_{qr} contains r relevant documents. The size of this list, $|P_{qr}|$, now defines the number of items that the user must view to find the required number of relevant ones for that query.

For a given number of required relevant items r we can define a subset, Q', of those queries that are capable of satisfying a request for r relevant items. More formally:

$$Q'_r = \left\{ q : \left| R_q \right| \geq r \wedge q \in Q \right\}$$

A simple definition of the average number of items a user needs to view, v'_r, can now be defined as:

$$v'_r = \frac{\sum_{q \in Q'_r} \left| P_{qr} \right|}{\left| Q'_r \right|}$$

Cooper's expected search length (ESL) measure is based around the observation that ranking algorithms do not, typically, produce a simple ordering of the collection but often allocate the same score to several documents – thus not specifying the rank position of relevant documents within a set of same score documents. The above measure would, therefore, have a random element associated with where in a set of equal scored documents the relevant ones are located. ESL treats the ranked list as a weak ordering with documents split into retrieval levels by score (and randomly located within each level). ESL also counts only non-relevant documents that the user has to view. Figure 1 shows a sample retrieval with ticked documents being relevant, all documents on one horizontal level having the same score and those scores reduce from top to bottom:

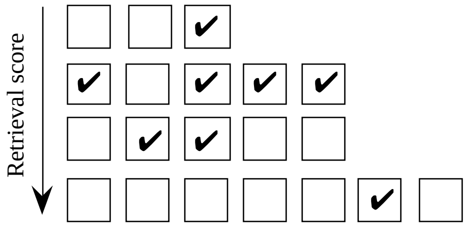


Figure 1: A weak ordering for a twenty document collection

If r=6 then the ESL for the given query q is:

$$esl(q,6) = \frac{4}{10} \times 3 + \frac{3}{10} \times 4 + \frac{2}{10} \times 5 + \frac{1}{10} \times 6 = 4$$

Since there is a 4/10 probability of a search length of 3 (the user has to look at 3 non-relevant documents before seeing 6 relevant ones if, in figure 1, one relevant document is at the left end of the third level), 3/10 probability of a search length of 4 etc. The measure can be calculated more directly as:

$$ESL(q,r) = j + \frac{i.s}{r+1}$$

where

- j = total number of documents non-relevant to q in all levels that precede the final level in the weak ordering (the final level being the level in that the rth relevant document is found);
- i = the number of non-relevant documents in the final level;
- s = the number of relevant documents required from the final level;
- r = the number of relevant documents in the final level.

The number of documents a user must look at on average can now be defined as:

$$v''_r = r + \sum_{q \in Q'_r} \frac{ESL(q,r)}{\left| Q'_r \right|}$$

Unfortunately a simple mean can be badly skewed by outlying documents (e.g. if a system retrieves the first relevant item at position 2 for 20 queries but fails completely for one query giving the first relevant at position 1000, a straight mean gives 49.5). A standard method for dealing with occasional outliers while still including them to some extent is to mean the logs of the individual items and take the exponential of the result (20 at position 2 and 1 and position 1000 now givens a mean of 2.7 which is more representative of the system's typical behaviour). This leads to the final definition *v*:

$$v_r = r + e^{\sum_{q\in Q'_r}\frac{\ln(ESL(q,r))}{|Q'_r|}}$$

This can also be viewed as the geometric sum (whereas V'_r is the arithmetic sum).

Example: system hypothesis testing

As an example of the number-to-view graphs, consider a standard IR system (sum of TF/IDF scores running on the CACM collection). The Porter's algorithm [Porter 1980] is a generally accepted stemming algorithm which will be treated as the research hypothesis in this example. Figure 2 shows two runs of an IR system – one with Porter's algorithm and one without. This shows that, for this IR engine on this collection, there is no difference to precision at high recall while there is a noticeable drop in precision without Porter in the 15-35% recall region.

Figure 3 shows the same comparison plotted as number relevant required (*r)* against number to view (v_r). This graph shows that for up to around 25 relevant documents required the two systems are performing roughly the same (with Porter being consistently slightly better). However, from this point onwards the effectiveness of the systems diverge and the use of Porter becomes more crucial (at 30 relevant documents required, making the difference between scanning through 560 and 990 documents). Figure 4 shows an exploded view of the graph for low values of *r*.

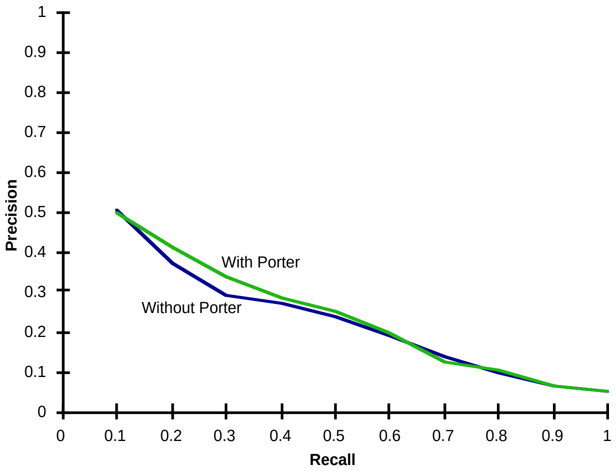


Figure 2: Porter v non-Porter recall-precision graph

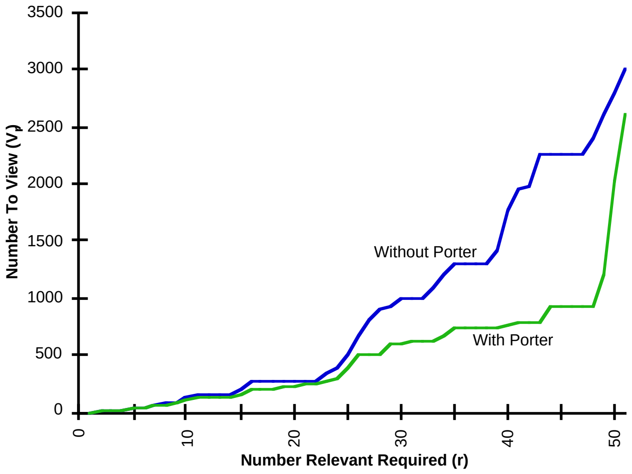


Figure 3: Number to view graph

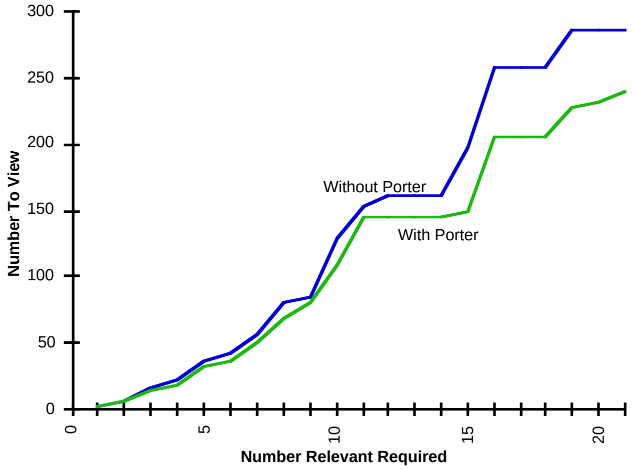


Figure 4: Number to view graph (zoomed)

Figure 3 also shows some of the strengths (and weaknesses) of the number-to-view graphing approach. I claim that "number relevant required" and "number to view" relate more closely to many user-task oriented considerations. Cooper [1966] characterises search tasks into five categories:

1. A user requires one relevant document in response to a very specific query;
2. A user may have some notion of the number of documents that (s)he wishes to read;
3. A user may want all documents on a topic;
4. A user may have some notion of the fraction of literature they wish to cover;
5. Some combination of the above (e.g. all relevant if there are five or fewer, otherwise precisely five).

For example, a user may well have some feel for how many documents are likely to satisfy his/her information need (type 2 query) or, conversely, how many documents (s)he are willing to search through. This complements precision-recall graphs, which discuss performance for different levels of recall (type 3 and 4 queries). NTV graphs also give a tangible and concrete representation of the quality of a system and of improvements – an algorithm that implies looking at 75 instead of 80 documents is an improvement, but one that is not likely to have an effect on task completion.

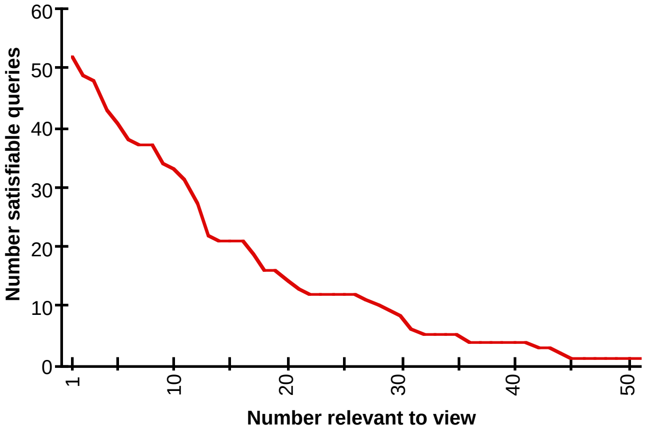


Figure 5: Distribution of number of relevant documents per query.

In the CACM collection the average number of relevant documents per query is low (figure 5 shows the distribution). Therefore, the representativeness of the values in figure 3 decreases towards the right as $|L_r\rceil$ reduces. With raw calculations this results in fluctuating graphs towards high values of r. Clearly, if $r_x < r_y$ then it should be the case that $v_{rx} < v_{ry}$, therefore the graphs shown here are smoothed pessimistically to show the highest value of v_r in the range v_1 to v_r (figure 6 shows the *smoothed* graphs in bold with fine lines representing underlying values). Furthermore, if the gap shown in figure 3 was not so large, it would not be safe to assume it is statistically significant when $|L_r\rceil$ is low[1]. The averaging issue is not discussed by Cooper, who also assumes that averaging is performed over queries that can achieve the r relevant matches. However, his work did not envisage graphing the results against a varying number of required relevant documents and focused more on providing a single measure of system performance.

NTV graphs are also, clearly, more dependent on the test collection than precision-recall graphs because of their more direct reliance on the number of documents relevant to a query, indeed this is one of the main criticisms of ESL [Senko 1969]. However, results using precision-recall graphs are rarely presented as an independent single graph but usually as an improvement over a standard graph. The ability to do this with precision-recall graphs is also questionable as different collections perform differently, but the translation is unarguably more stable than NTV graphs. Cooper acknowledges that “if the query (or query set) or the document collection differ for the different systems being evaluated, this simple basis of comparison can be grossly misleading”. He then goes on to define *the expected search length reduction factor* which gives a measure that can be used for comparison. However, this normalised method does not give the “physical” interpretation, in terms of number of documents, which is the strength of NTV graphs and the basis for the remainder of this paper on predictive evaluation in IR. Cooper states that “in a theoretical investigation of how varying a certain design parameter would affect retrieval effectiveness under certain assumed conditions regarding input queries and document collection makeup. The (mean) expected search length for a system also provides a *physically* interpretable statistic that may be of interest in its own right, even when there is no thought of using it as a basis for comparison with another system”.

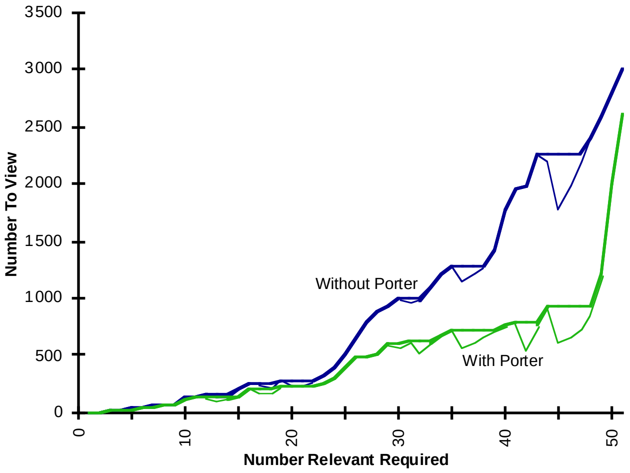


Figure 6: Smoothed and underlying NTV graphs.

Time-to-view graphs

Following the argument of Card and Moran, given a suitable model of an interface to an IR engine, it should be possible to predict how long a user would take to work through a certain number of documents with the IR interface. If we can plot the number of documents to view against time and, taking this in combination with number-to-view graphs, how many to view to find a given number of relevant documents, then we can plot a prediction of how long it would take to find a given number of relevant documents. As well as having the physical interpretation and user-task benefits claimed for NTV graphs, *time-to-view* (TTV) graphs also introduce a single presentation in which interface changes and effectiveness changes can be compared.

This section initially introduces the interface modelling approach that TTV graphs are based on, shows how these models can be converted into equations and then combined with NTV graph.

TRIMIR Modelling

Green and Benyon [1996] introduced the use of entity-relationship (ER) diagrams for modelling information artefacts of an interactive system (ERMIA diagrams). As well as an interface modelling notation they link their ERMIA diagrams with work on modelling the cost of accessing data with an interactive system [Card, Pirolli and Mackinlay 1994] to allow comparisons of how many steps are required to reach a certain amount of information. The main difference between ERMIA and previous predictive evaluation work, e.g. GOMS [Card, Moran and Newell 1983], is the high level of abstraction and focus on interface objects rather than detailed modelling of user goals and tasks. While this reduces the user-centred nature of the modelling, and thus is likely to eliminate many benefits of GOMS in modelling different user characteristics and interaction behaviours, it does provide an accessible and easily constructed modelling notation that can be used to model many aspects of the user interface design. Figure 7 gives a simple example of ERMIA modelling for an object based drawing program.

[1] It may be worth graphically reinforcing this by plotting the number of satisfiable queries (figure 5) on number-to-view graphs (figure 4).

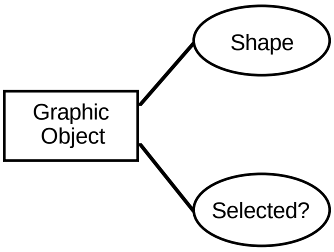


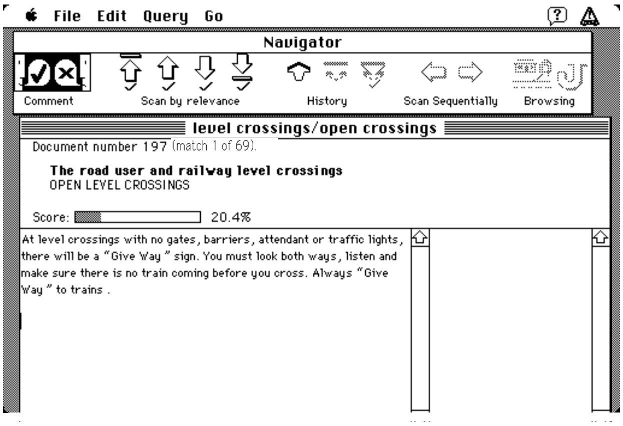
Figure 7: Example ERMIA model for drawing program.

In line with standard ER diagrams, *entities* are the prime elements which are connected by *relationships* and have *attributes*. Unlike traditional ER diagrams, ERMIA introduces a notion of representing the time complexity of searching for a particular entry; the annotation ø on an entity indicates that a collection is in a *two dimensional pile* (e.g. papers scattered on a desk), ↓ indicates that the collection is arranged linearly but not sorted (e.g. an unsorted bookshelf – both ø and ↓ can be searched in linear time, but a linear collection requires less memory during a search), and ⇓ indicates that the collection is arranged in linear order by key attribute (thus giving logarithmic search time). IR engines, essentially, sort the collection according to predicted probability of documents being relevant to the query. As such I introduce a new sort notation ∇ which indicates that a collection is sorted according to systems estimates of the probability of matching the query.

Example: different surrogates

As a working example, this section will model the time to access three different IR systems. Initially considering these IR systems to be based on the same underlying engine but differing on their interfaces, then combining these models with number-to-view information to incorporate engine performance into the model. The three systems chosen (MMIR, NRT and Alta Vista) present their ranked list to the user differently, as follows:

MMIR: this interface (initially developed to provide access to documents in a mixed IR / hypertext environment (Dunlop 1991/1993)) reacts to a query by presenting the user with the first matched document (see figure 8). The user can then move through the ranked list by go-top, go-higher, go-lower and go-bottom commands. MMIR was initially written to access The Highway Code (the UK's official guide to driving). An ERMIA model for MMIR[2] is given in figure 9 where *page* is a page of the collection which is made up of a *body*, a *title* and a record of whether the user has *marked* it as relevant for use in relevance feedback:



level crossings/open crossings

Document number 197 (match 1 of 69).

The road user and railway level crossings
OPEN LEVEL CROSSINGS

Score: 20.4%

At level crossings with no gates, barriers, attendant or traffic lights, there will be a "Give Way " sign. You must look both ways, listen and make sure there is no train coming before you cross. Always "Give Way " to trains .

Figure 8: Screenshot of initial result from MMIR

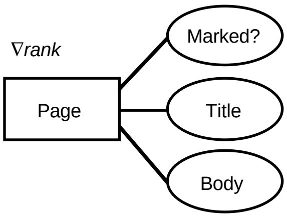


Figure 9: ERMIA model for a non-surrogate IR system (MMIR)

NRT: Initially devised for retrieval of news stories (Sanderson and Van Rijsbergen 1991), NRT presents a long list of surrogates in a retrieval window. The surrogates present summary information of the pages (e.g. headline, date and source newspaper name) and are ranked according to their likelihood of being relevant. An ERMIA model for NRT is given in figure 10.

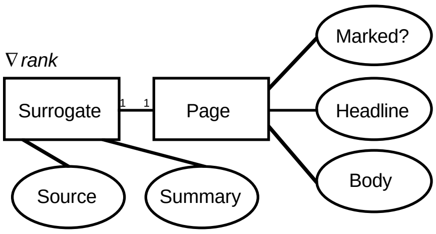


Figure 10: ERMIA model for surrogate based IR system (NRT)

Alta Vista: taken as an example of a World Wide Web search engine, Alta Vista[3] responds to a query by presenting a screen of surrogates for the top ten matched items. The user can either choose one of these items or move to the next page of items. The screens and the surrogates within a screen are ranked resulting in the model given in figure 11.

[2] ERMIA models can be refined to include varying levels of detail. The diagrams presented in this section represent high level non-detailed models of each interface.

[3] See http://altavista.digital.com/

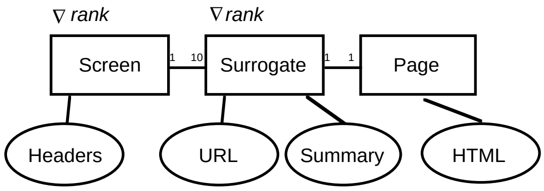


Figure 11: ERMIA model for web based IR system (Alta Vista)

In line with the work on the GOMS and keystroke [Card, Moran and Newell 1980] models of interaction it is possible to estimate a set of standard/average times which can be used in calculating the time taken to access a number of items. Table 1 shows the times required for the three systems mentioned above and gives initial guestimates[4] of their values – the list of parameters is based on analysing the entities and relationships in the about ERMIA models. The two sets of estimates are based around (1) a retrieval engine running on a single machine accessing local files and (2) an Internet based application with considerably slower loading times.

Value	Description	Est 1	Est 2
t_{ls}	Time to load surrogate	0.1	1.0
t_{rs}	Time to read surrogate	2.0	2.0
t_{ld}	Time to load document	0.3	10.0
t_{rd}	Time to read document	15.0	15.0
t_{lh}	Time to load screen header	0.3	5.0
k	Fraction of documents read	0.3	0.1

Table 1: times required to model mmIR, NRT and Alta Vista

By analysing the ERMIA diagram for MMIR and looking at the list constants, the following equation can be derived for MMIR:

$$t_{mmir} = v_r\,(t_{ld} + k\,t_{rd})$$

where

v_r = number of documents required to view in order to see r relevant ones.

This defines the time taken to process v_r documents in terms of the time to load a document, a constant time to read the document and an estimate of how many of the displayed documents the user will read. Similar models can be developed for NRT and Alta Vista[5]:

$$t_{nrt} = 100\,t_{ls} + v_r\,t_{rs} + v_r\,k\,(t_{ld} + t_{rd})$$

$$t_{alta} = (\lfloor (v_r - 1)/10 \rfloor + 1)\,(t_{lh} + 10(t_{ls} + t_{rs})) + v_r\,k(t_{ld} + t_{rd})$$

where

$\lfloor x \rfloor$ = floor of x, i.e. x rounded down to a whole number.

Figure 12 shows the graphs for estimates 1 (left) and 2 (right). As can clearly be seen, the MMIR approach is faster on a local machine (assuming these estimates) but is much slower over a network. However, for the form of question-answer tasks likely with the Highway Code this approach may still be quicker on slower connections (up to 5 documents accessed). The step function of Alta Vista helps performance over the web (at these estimated speeds) until the user needs to view more than roughly 50 documents.

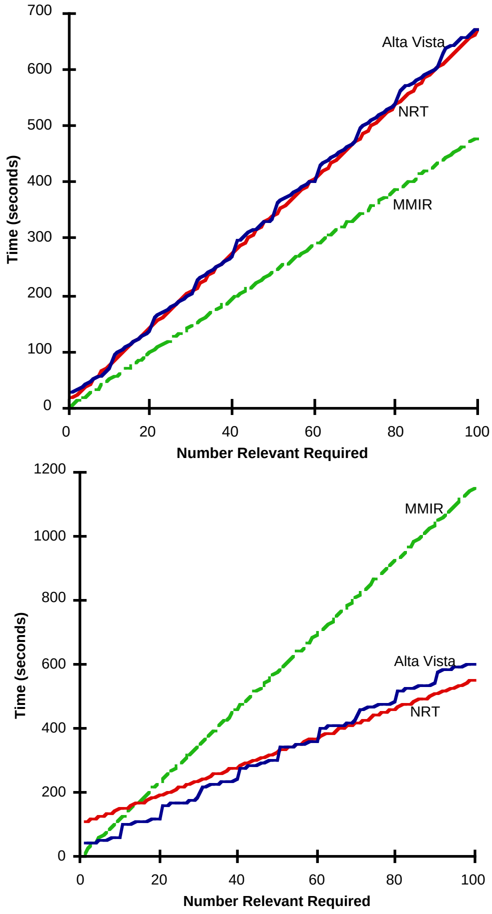


Figure 12: Predicted time to find v_r documents locally and over the Internet.

The equations, ERMIA models and timing estimates could all be further developed to represent real user behaviour model closely. For example, although the times in table 1 are fixed, which is clearly not representative of web behaviour, Card, Pirolli and Mackinlay integrated a probabilistic function to model unpredictable behaviour. A variant could easily be incorporated into these models and could be used to give a typical, best case and worst case for different network performances.

Time to view graphs

Taking interface-based predicted-time modelling of how long it takes to view a set of documents with underlying-performance modelling of how many documents must be viewed to find a

[4] The times in this table are based on analysing the ERMIA diagrams and estimates of user behaviour. The final section of this paper discusses future work to refine these estimates.

[5] NRT can be set to retrieve a variable number of documents, for this paper the number is set to 100.

number of relevant ones, it is now trivial to plot the number of relevant documents required against the time to find them. Figures 13 and 14 show comparison of different interfaces and engine characteristics within the same graph (in a sense, combining figures 3 and 12), thus giving an impression of where most gain is likely to be found: with an interface improvement or an engine improvement.

These graphs clearly show that Porter's algorithm gives a bigger improvement in performance, after around 10 documents required, than using a more appropriate interface. For up to 5 relevant documents required Alta Vista is performing better than NRT and giving a larger improvement than Porter, thereafter the results are unclear until above 12 where NRT consistently out performs Alta Vista. However for both variants of the engine, Porter is never slower than that without (again assuming the time estimates given in table 1).

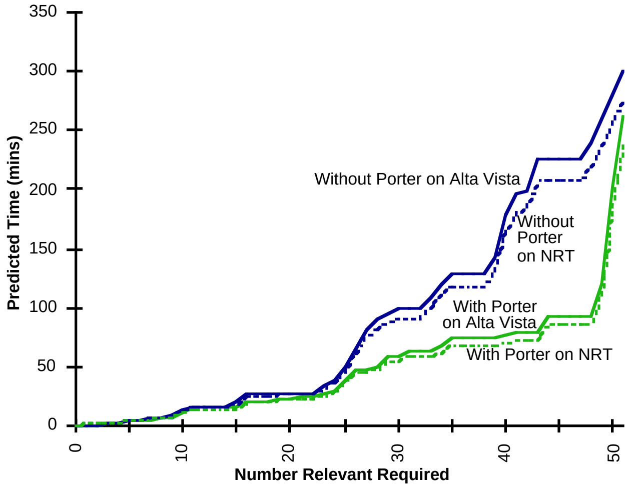


Figure 13: Time to view graph

Conclusions and future work

This paper has presented a new graph-based version of Cooper's Expected Search Length measure, it is argued that this graph-based method provides not only a more physical interpretation than recall-precision graphs but also complements recall-precision graphs by presenting different aspects of retrieval performance. The graphing model was then integrated with user interface models based on Green and Benyon's ERMIA diagrams. This provided a single integrated model of evaluation in which engine performance and interface features can be discussed within the same framework. Throughout the paper the results of experiments on a basic IR engine were shown using the graphing techniques as they were introduced.

Future work needs to be carried out to assess how well the initial estimates made, using the models presented here, hold out for real users of real IR systems. Work also needs to be carried out on introducing different user strategies into the models.

In summary, this paper has presented *Expected Search Duration*: a new graphing, modelling and evaluation approach that should act as the core for predictive evaluation in IR and thus be usable to analyse and discuss trade-offs between underlying engine performance and different interface features.

Acknowledgements

Many thanks are due to Ian Ruthven, Mark Sanderson and Mounia Lalmas for reading drafts of this paper and making it much better as a result. Thanks are also due to the Glasgow IR and HCI groups and The Computing Science Department as a whole for support while carrying out this work and for asking lots of nasty questions (in that respect thanks also to those from City and Robert Gordon Universities, attendees at the Mira Dagstuhl meeting and the paper referees for comments and criticisms on this work).

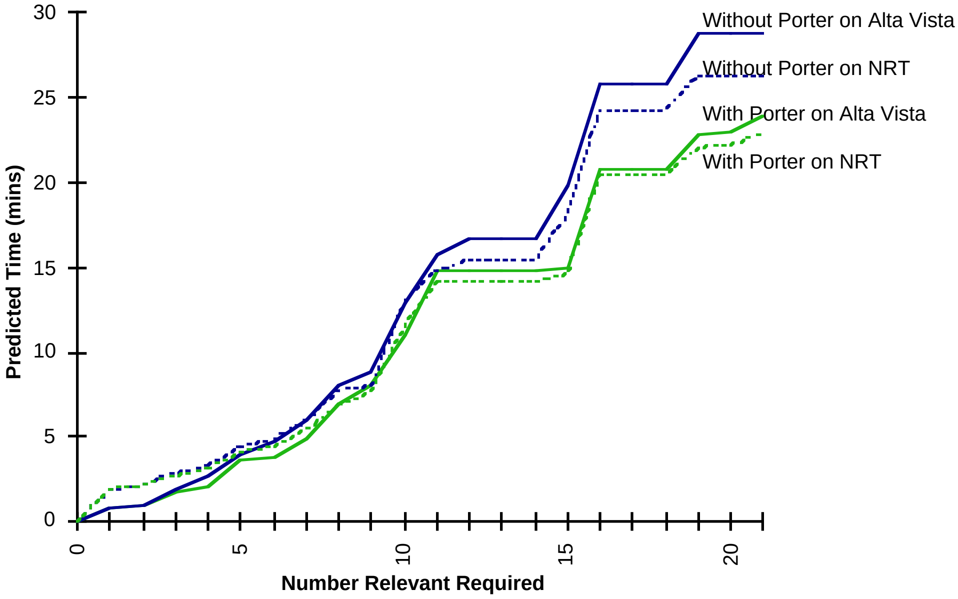


Figure 14: Time to view graph (zoomed)

References

AGOSTI, M., AND SMEATON, A. *Information Retrieval and Hypertext*, Kluwer Academic Publishers, ISBN 0-7923-9710-X, 1996.

CAMPBELL, I., AND VAN RIJSBERGEN, C.J., “The Ostensive Model for developing information needs”, *Proceedings of CoLIS 2 Conference*, Copenhagen, 13-16 October 1996.

CARD, S., AND MORAN, T., “User Technology: From pointing to pondering”, *Proc. ACM Conference on History of Personal Workstations*, pp. 183-198, {reproduced in Baecker, Grudin, Buxton & Greenberg (Eds), *Readings in Human-Computer Interaction: towards the year 2000,* Morgan Kaufmann, 1995} (1986).

CARD, S., MORAN, T., AND NEWELL, A. *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, 1983.

CARD, S., MORAN, T., AND NEWELL, A. “The keystroke-level model for user performance time with interactive systems”, Communications *of the ACM,* vol. 23(7), pp. 396-410, 1980.

CARD, PIROLLI AND MACKINLAY, “The Cost-of-Knowledge characteristic function: display evaluation for direct-walk dynamic information visualisations”, *proceedings of ACM CHI-94*

COOPER, W.S. “Expected search length: a single measure of retrieval effectiveness based on the weak ordering action of retrieval systems”, *American Documentation*, vol. 19, January 1968.

DUNLOP, M.D. (editor). *Proceedings of the Second Mira Workshop (Monselice, Italy).* University of Glasgow Computing Science Research Report TR-1997-2, http://www.dcs.gla.ac.uk/mira/workshops/padua_procs/ 1996.

DUNLOP, M.D., AND VAN RIJSBERGEN, C.J. Hypermedia and probabilistic retrieval. *Proceedings of RIAO ’91 (Universitat Autònoma de Barcelona),* pp. 337-356. April 1991.

DUNLOP, M.D., AND VAN RIJSBERGEN, C.J. Hypermedia and free text retrieval. *Information Processing and Management*, vol. 29(3), pp. 287-298. May 1993.

HARMAN, D. “User-friendly systems instead of user-friendly front-ends”, *Journal of the American Society for Information Science*, vol. 43(2), pp. 164-174, 1992.

HARMAN, D. “Overview of the fourth Text REtrieval Conference (TREC-4)”, *Proceedings of TREC-4,* Gathersburg, 1995.

GREEN, T.R.G., AND BENYON, D.R., “The skull beneath the skin: entity-relationship models of information artefacts”, *International Journal of Human-Computer Studies*, **44**(6), pp. 801-828, June 1996.

MIZZARO, S. "A cognitive analysis of information retrieval." In P. Ingwersen and N. O. Pors (editors), *Information Science: Integration in Perspective - Proceedings of CoLIS2,* pp. 233-250, Copenhagen, Denmark, October 1996.

PEJTERSEN, A.M. “Empirical work place evaluation of complex systems”, *Advances in Applied Ergonomics. Proceedings of the 1st International Conference on Applied Ergonomics (ICAE'96),* Ozek, Ahmet and Salvendy (Eds.)Istanbul, Turkey, May 1996.

PORTER, M.F. “An Algorithm for Suffix Stripping”, *Program*, vol. **14**(3), pp. 130-137. July 1980.

PREECE, J., ROGERS, Y., SHARP, H., BENYON, D., HOLLAND, S., AND CAREY, T. *HUMAN-Computer Interaction,* Addison Wesley, ISBN 0-201-62769-8, 1994.

ROBERTSON, S.E., WALKER, S., AND HANCOCK-BEAULIEU, M.M. “Large test collection experiments on an operational, interactive system: OKAPI at TREC”, *Information Processing and Management*, vol. 31(3), pp. 345-360, 1995.

Salton, G., and McGill, M.J.. *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

SANDERSON, M., AND VAN RIJSBERGEN, C.J. NRT (News Retrieval Tool). *Electronic Publishing Origination, Dissemination & Design*, vol. 4(4), pp. 205-217, 1991.

SENKO, M. E. “Information storage and retrieval systems”, *Advances in Information Systems Science,* J Tou (editor), volume 2, 1969.

VAN RIJSBERGEN, C.J. *Information Retrieval (second edition).* Butterworths, London, http://www.dcs.gla.ac.uk/Keith/Preface.html . 1979.