

# Extended Performance Graphs for Cluster Retrieval

D. P. Huijsmans and N. Sebe

Leiden Institute of Advanced Computer Science  
Leiden University, NielsBohrweg 1, 2333CA Leiden,  
The Netherlands  
email: {huijsman,nicu}@liacs.nl

## Abstract

Performance evaluations in Probabilistic Information Retrieval are often presented as Precision-Recall or Precision-Scope graphs avoiding the otherwise dominating effect of the embedding irrelevant fraction. However, precision and recall values as such offer an incomplete overview of the information retrieval system under study: information about system parameters like generality (the embedding of the relevant fraction), random performance and the effect of varying the scope is badly missed.

In this paper three cluster performance graphs are presented. In those cases where complete ground truth is available (both cluster size and database size) the Cluster Precision-Recall (Cluster PR) graph and the Generality-Precision=Recall graph are proposed. In those cases where cluster sizes are unknown (and so recall) the double logarithmic Cluster Precision Window graph is proposed.

## 1 Shortcomings of presently used retrieval performance measures

Performance characterization of content-based image and audio retrieval often borrows from performance figures developed over the past 30 years for probabilistic text retrieval. Landmarks in the text retrieval field are the books [12] and [11] as well as the proceedings of the annual ACM SIGIR [7] and NIST TREC [14] conferences.

In the area of probabilistic retrieval the results of performance measurements are often presented in the form of Precision-Recall (or Recall-Precision) graphs and Precision-Scope graphs. Each of these standard performance graphs provides the user with incomplete information about how the IR System will perform for various cluster sizes and various embedding sizes. Generality (influence of the relevant fraction) as a system parameter hardly seems to play a role in performance analysis. Although generality may be left out as a performance indicator when competing methods are tested under constant generality conditions, it appears to be neglected even in cases where generality is widely varying (a wide range of cluster sizes in one specific database is the most frequently encountered example).

That generality for a cluster of relevant items in a large embedding database is often  $\approx 0.0$  does not mean that its exact low level no longer matters. A continually growing

embedding around a constant size cluster of relevant items will eventually lower the overall precision-result curve (for the user) to unacceptable low levels as is shown in Figure 1.

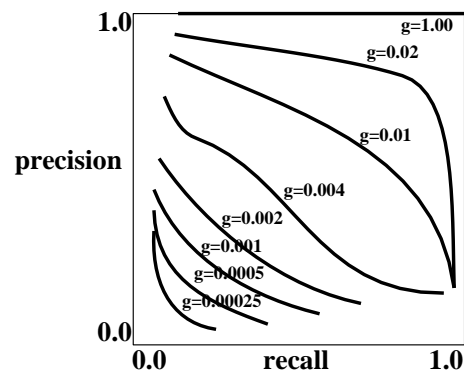


Figure 1: Typical Precision, Recall graphs for retrieval of a constant size cluster of 8 relevant items embedded in a growing number of irrelevant items (max 32,000)

Only when the size of the growing database results in a relative growth that is equal for both relevant items and embedding would the PR graph remain at the same level, but even then a constant retrieval recall rate would mean that the scope used to retrieve the relevant items would have to increase with the same percentage as well.

Precision and recall values from single or averaged query results are often presented when the size of the relevant cluster is known. It is unclear why in such cases explicit information about the generality (relevant fraction) is often hard to determine or is unrecoverable like in the papers [9], [13], [2].

Precision-Scope graphs are often presented when information about the size of the relevant fraction is missing. In this case a random retrieval method could have provided at least an estimate of this relevant fraction (and therefore of the recall value).

The lack of generality information in Precision-Recall and Precision-Scope graphs makes it difficult to make comparisons between different sized IR Systems and to find out how the performance will degrade with increasing database size. The recent overview of [8] does not even mention or propose the system parameter generality as one that would be needed

for performance evaluation!

We would also like to keep the effect of varying the scope (visible size of the query results) into our performance graphs, since it is the main parameter of economical effectiveness for the user of a retrieval system.

These considerations led us to re-evaluate the building blocks of information retrieval performance measurement and the way these performance measures are visualized in graphs. How can we make the performance measures complete so that results of specific studies can not only be used to select the better method but can also be used to make comparisons between different system sizes and different domains?

## 2 Performance Evaluation Elements

Essentially all Information Retrieval (IR) is about Cluster Retrieval: the user having specified a query would like the system to return some or all of the items (either documents, images or sounds) that are in some sense part of the same semantic cluster i.e., the relevant fraction of the database with respect to this query for this user. The Ideal IR System would quickly present the user some or all of the relevant material and nothing more. The user would value this Ideal System as being either 100 % effective or being without (0 %) error. Since being without error not necessarily means being completely effective, we would prefer the (0 to) 100 % normalized effectiveness measure for such a system, but this is contrary to the normalized measures advocated for instance in [11]. In this paper we will refer to the Ideal System that retrieves all the relevant material as the Total Recall Ideal System (TRIS).

In practice IR Systems are often far from ideal: the query results shown to the user (the finite list of retrieved elements) generally are incomplete (containing some retrieved relevant items but without some missed relevant items) and polluted (with retrieved but irrelevant items). The performance is characterized in terms of precision and recall.

$$p = \textit{precision} = \frac{\textit{relevant retrieved items}}{\textit{retrieved items}} \quad (1)$$

$$r = \textit{recall} = \frac{\textit{relevant retrieved items}}{\textit{relevant items}} \quad (2)$$

That these normalized fractions are commonly displayed in Precision-Recall (PR) graphs has as a disadvantage that the length (or scope) of the retrieved list is not displayed, whereas this scope is very important to the user because it determines the amount of items to be inspected and therefore the amount of time (and money) spent in searching.

Equally important is the degradation due to a growing database size (lowering the fraction of relevant items resulting in overall lower precision-recall values). A comparison between two information retrieval systems can only be done well when both systems are compared in terms of equal generality.

$$g = \textit{generality} = \frac{\textit{relevant items}}{\textit{all items}} \quad (3)$$

Although there is a simple method of minimizing the number of irrelevant items (by minimizing the number of retrieved items to zero) and a simple one to minimize the number of missed relevant items (by maximizing the number of retrieved items up to the complete database) the optimal length of the result list depends upon whether one is satisfied with finding one, some or all relevant items.

The parameterized  $E$ -measure of [11]:

$$E = 1 - \frac{1}{\alpha(1/p) + (1 - \alpha)(1/r)} \quad (4)$$

is a normalized Error-measure where a low value of  $\alpha$  favors recall and a high value of  $\alpha$  favors precision.  $E$  will be 0 for an ideal system with both precision and recall values at 1 (and in that case irrespective of  $\alpha$ ). Van Rijsbergen [11] favors the setting of

$$\alpha = 0.5$$

a choice giving equal weight to precision and recall and giving rise to the normalized symmetric difference as a good single number indicator of system performance (or rather system error):

$$\textit{Error} = E(\alpha = 0.5) = 1 - \frac{1}{(1/2p) + (1/2r)} \quad (5)$$

The problem with this  $E$ -measure is fourfold:

- An intuitive best value of 1 (or 100 %) is to be preferred; this can easily be remedied by inverting the [1,0] range by setting  $E$  to its range inverted and more simple form:

$$\textit{Effectiveness} = 1 - E(\alpha = 0.5) = \frac{1}{(1/2p) + (1/2r)} \quad (6)$$

- An indication of generality (database fraction of relevant cluster size) is missing completely. This will make comparisons of retrieval performance on different sized databases and for different cluster sizes impractical.
- An indication of expected random retrieval performance is missing. Since this random performance is directly tied to the generality a comparison with random performance would cover generality as well.
- An indication of expected result list size (or scope) is missing; for the user the length of the list to be inspected is very important and so knowing precision as a function of result list length is highly appreciated.

## 3 Cluster Retrieval Performance

Lets define all items, database or population size, as  $d$ ; the relevant items or cluster size, as  $c$ ; the retrieved items, the length of the result list, visible top of the ranking list or scope, as  $w$  (from window); the found items, the number of visible cluster members, as  $v$ . These four numbers, combined

in the quadruple  $\{d, c, w, v\}$ , are enough to completely specify system performance since all the other sizes, like missed relevant cluster members  $(c - v)$ , irrelevant retrieved items  $(w - v)$  and total of irrelevant items or embedding size  $(d - c)$ , can be derived from these.

Content-based text, image or sound retrieval is based on ranking database items with respect to a similarity (or dissimilarity) measure obtained from feature vectors that characterize both query and answers. In this paper we will restrict to the case of a query by example in content-based probabilistic retrieval resulting in a linear ordering of the database items of which a specific top portion is returned to the user for inspection.

Concepts worked out in this paper were experimentally verified on a database of  $\approx 20,000$  company logos that are manually grouped into 1858 clusters with an average size of almost 8; cluster size is in the range  $[2, 308]$ . Each of the cluster members in turn is taken as the query example; evaluations were based on the average number of remaining cluster members retrieved, within various scopes using different indexing feature vectors and using different similarity measures.

From a quantitative decision-support methodology this Query By Example (QBE) situation can be characterized by the well-known (see for instance [3])  $2 \times 2$  matrix of (relevant, irrelevant) versus (retrieved, not retrieved) items in Figure 2.

|              |            |              |               |
|--------------|------------|--------------|---------------|
|              |            | query result |               |
|              |            | retrieved    | not retrieved |
| ground truth | relevant   | TP           | FN            |
|              | irrelevant | FP           | TN            |

Figure 2: Decision table: TP=True Positive, FN=False Negative, FP=False Positive, TN=True Negative

This figure extended with the row- and column totals gives rise to the  $3 \times 3$  matrix in Figure 3.

|              |                    |              |              |
|--------------|--------------------|--------------|--------------|
| <b>v</b>     | <b>(c-v)</b>       | <b>c</b>     | <b>r=v/c</b> |
| <b>(w-v)</b> | <b>(d+v)-(c+w)</b> | <b>(d-c)</b> |              |
| <b>w</b>     | <b>(d-w)</b>       | <b>d</b>     |              |
| <b>p=v/w</b> |                    | <b>g=c/d</b> |              |

Figure 3: Retrieval sizes and ratios:  $v$ =found cluster members,  $c$ =cluster size,  $w$ =window or scope,  $d$ =database size;  $r$ =recall,  $p$ =precision,  $g$ =generality

To compare Information Retrieval System Performance between competing methods and among systems one would rather like to use a normalized triple like  $\{-\log_2(\mathbf{g}), \mathbf{r}, \mathbf{p}\}$  (log generality, recall and precision) with:

$$g = \frac{c}{d}, r = \frac{v}{c}, p = \frac{v}{w} \quad (7)$$

With these definitions the  $E$ -measure (see eqn 5) becomes

$$Error = 1 - \frac{2v}{(w+c)} \quad (8)$$

whereas the range inverted  $E$ -measure (see eqn 6) becomes

$$Effectiveness = \frac{2v}{(w+c)} \quad (9)$$

Generality information however is completely missing in eqn 9.

### 3.1 The addition of generality information

Whereas precision and recall are normalized measures on  $[0, 1]$ , the generality measure is chosen to be represented on a logarithmic scale because present day database sizes  $d$  are so much greater than the involved cluster sizes  $c$  that a normalized generality measure of  $g = c/d$  would become effectively nil in many graphs.

The associated 3-dimensional retrieval performance characterization can be presented in 2 dimensions as a set of Precision-Recall graphs (for instance at integer logarithmic generality levels to show how the  $p, r$  values decline due to successive doubling of the database to cluster ratio). In this paper another attractive plane in three-dimensional Generality-Precision-Recall space, the precision=recall plane (see Figure 8), will be advocated for the characterization of system performance.

The ratio triple  $\{-\log_2(\mathbf{g}), \mathbf{r}, \mathbf{p}\}$  would be enough to compare among methods for different cluster- and database sizes. Users aiming at a specific recall  $r = k/c$  can use precision values from the appropriate generality plane  $g = c/d$  to set a retrieval window size as  $w = k/p$  with  $k \leq c$ . To select the right generality plane the user must know cluster size and database size. The fact that these numbers are largely unknown to most users of information retrieval systems is no excuse for those claiming to investigate retrieval performance. For them such evaluations should be based on ground-truth in a well-defined experimental setting allowing the recovery of all essential parameters including generality.

In conclusion: a first addition needed for the Precision-Recall graph to become more complete is the addition of the generality level. The Total Recall Ideal System (TRIS) as described for the PR graph can be extended to cover generality by stipulating that the  $p, r$  values of TRIS should not depend on the generality level, so the effectiveness of TRIS does not depend on the size of the embedding.

### 3.2 The addition of window size or scope information

Information about the effect of changing the window size or scope on the measured precision and recall values can be added to the Precision-Recall graph by taking into account that possible precision, recall outcomes are restricted to lay on a line in the PR-graph. This is due to the fact that the definitions of the system parameters precision (see eqn 1) and

recall (see eqn 2) have the same numerator and are therefore not independent. If the size  $w$  of the retrieved list of items is denoted in terms of the cluster size  $c$  of relevant items as cluster-window-size  $w/c$  we can rewrite precision and recall of eqn 7 for  $w/c = 1$  (window size=cluster size) as:

$$p = \frac{v}{w} = \frac{v}{c} = r \quad (10)$$

This means that when retrieving cluster members with a cut-off window value of the ranking lists equal to the cluster size this will give  $p, r$  values that are restricted to the line  $p = r$  (the diagonal 0,0 – 1,1) in the Precision-Recall (or Recall-Precision) graph (Figure 4)!

For window size  $w = c/2$ :

$$p = \frac{v}{w} = \frac{2v}{c} = 2r \quad (11)$$

and therefore  $p, r$  values are restricted to lay on the line  $p = 2r$  (see Figure 4).

For window size  $w = 2c$ :

$$p = \frac{v}{w} = \frac{v}{2c} = \frac{r}{2} \quad (12)$$

and therefore  $p, r$  values are restricted to lay on the line  $p = r/2$  (see Figure 4).

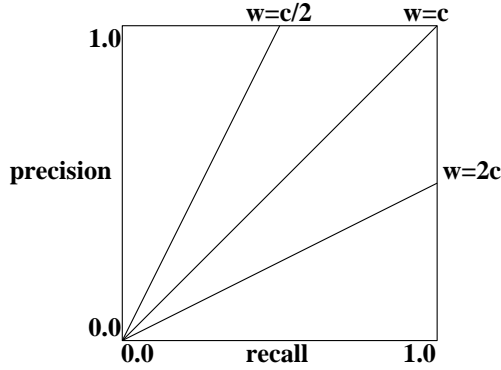


Figure 4:  $p, r$  values are restricted along lines defined by window size as a function of cluster size

So for each window size  $w = a \cdot c$  the  $p, r$  values are restricted to the line  $p = r/a$ . Therefore, window size or scope information can be displayed as a radiating set of lines from the origin of the PR graph. In Figure 4 the main cluster size related window sizes or scopes are displayed. In Figure 7 a number of constant scope lines for a retrieval of a cluster of four additional cluster members are shown.

With these window size lines drawn in the Precision-Recall graph one understands much better what the  $p, r$  values mean: the special meaning of the diagonal as delineating the lower-right half of the graph, where recall can become 100 % but precision will fall below 100 %, from the upper-left half of the graph, where precision can be as high as 100 % and recall will always be below 100 %. In the ideal

case (see Figure 5) precision  $p$  will run along  $p = 1.0$  for recall  $r \in [0.0, 1.0)$  and reach  $p, r = 1.0, 1.0$  (the TRIS point) when window size equals cluster size ( $w = c$ ); for window sizes above cluster size precision will slowly drop from  $p = 1.0$  along  $r = 1.0$  until the random level  $p = c/d$  at  $w = d$  is reached.

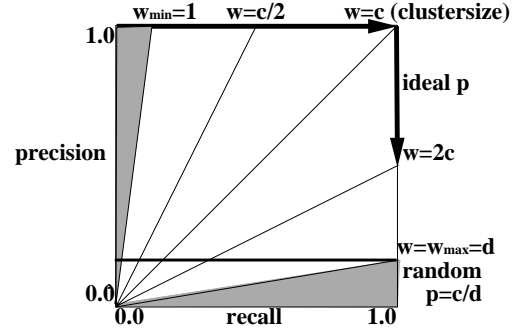


Figure 5:  $p, r$  values for ideal retrieval are  $1, r$  for  $r < 1$ ; for window size > cluster size  $p$  drops slowly towards random level  $c/d$

Also depending on cluster size the region to the left of  $p = r/c$  cannot be reached as well as the region below  $p = dr/c$ . This means that for the smallest clusters of 2 members, where 1 of the cluster members is used to locate its single partner, the complete upper-left half of the PR graph is out of reach (see Figure 6).

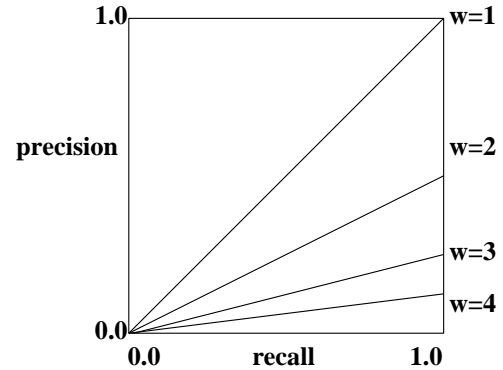


Figure 6: Lines along which  $p, r$  values for retrieved cluster size=1 (clusters of 2, 1 used for query, max 1 for retrieval) are located

Because the diagonal  $w = c$  line presents the hardest case for a retrieval system (last chance of precision being max 1.0 and first chance of recall being max 1.0) and is the only line that covers all cluster sizes (see Figure 6) the best system performance presentation would be the  $p = r$  plane in the three-dimensional Generality-Precision-Recall (GPR) graph (see Figure 8 and Figure 9) showing the precision=recall level

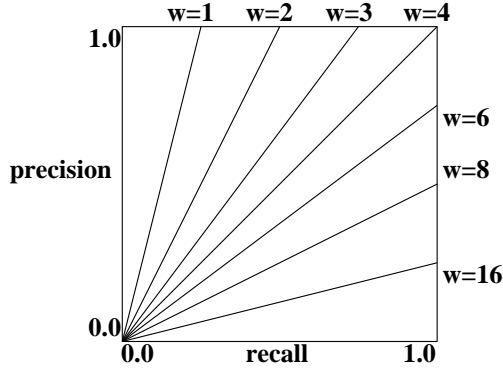


Figure 7: Lines along which  $p, r$  values are located for retrieval cluster size=4

as a function of generality; this also satisfies van Rijsbergen's range inverted  $E$ -measure with  $\alpha = 0.5$  (see eqn 6 and 9) as a function of generality, for in case of the  $p = r$  plane,  $w = c$  and  $p = r = v/c$ :

$$E(\alpha = 0.5) = \frac{1}{(1/2p) + (1/2r)} = \frac{v}{c} = p = r \quad (13)$$

System performance would then be given by  $E(g)$  or rather  $E(-\log_2 g)$  (*Effectiveness* as a function of generality).

In conclusion: a second addition to the Precision-Recall graph is the display of information about cluster related window sizes or scopes in the form of a set of lines radiating from the origin: the diagonal showing the  $p, r$  results for a result window or scope which has the same size as the size of the relevant cluster.

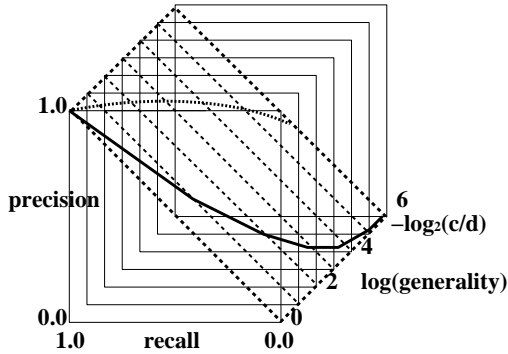


Figure 8: The 3-dimensional information system performance space with the  $p = r$  plane (with random and  $w = c$  results for different generality values)

### 3.3 The addition of random performance information

An objective evaluation of Information Retrieval System performance should always indicate how well the system performs with respect to a random retrieval method. In [4] system performance is even characterized by the reduction in search time offered, but for the growing number of very large

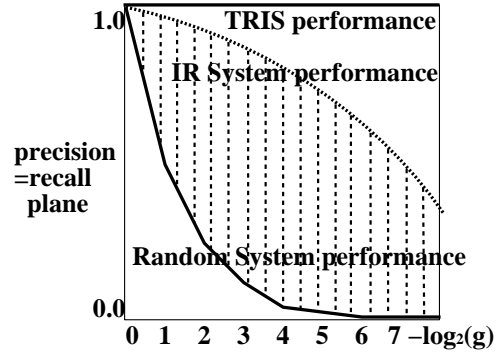


Figure 9:  $p = r$  values for window size=cluster size as a logarithmic function of generality

databases the gain of such a normalized performance number like  $\overline{ERSL}$  (mean Expected Reduction in Search Length) will be effectively 1.0, so that one cannot well differentiate between methods and systems.

The present use of Precision-Recall graphs where generality and random performance levels are not shown makes it possible to heighten the area under the average PR curve to almost any desired upper level by taking a small enough embedding. The mentioning of the generality level would already remedy this situation of incomplete knowledge, but it would even be better if a bottom line for the Random IR System performance level at

$$p_{random} = \frac{c}{d} = g \quad (14)$$

is drawn in the PR graph as well. Especially when the embedding of irrelevant items is relatively small this random performance level would visibly be above  $p \approx 0.0$ . For experimental studies on large databases one might demand that this random retrieval performance level remains below 1%. This would simply mean that the embedding of irrelevant items has to outnumber the cluster members by at least a factor of 100; the generality index in that case would have to be above  $\approx 6$  ( $-\log_2(1/128)$ ).

Taking random performance figures into account as well as the Total Recall Ideal System performance one could define  $E^*$  as the gain in effectiveness with respect to a random performance by subtracting eqn 14 from eqn 13, giving

$$E^*(g) = \frac{1}{(1/2p(g)) + (1/2r(g))} - g \quad (15)$$

in which formula  $p(g)$  and  $r(g)$  represent the precision and recall values at the specific generality level  $g$ . For comparison with the Total Recall Ideal System at  $w = c$ ,  $p(g) = r(g)$  and

$$E^*(g) = r(g) - g = p(g) - g \quad (16)$$

which corresponds to the shaded area in Figure 9.  $E^*(g)$  penalizes the use of small embeddings in retrieval tests but will approximate  $E(g)$  for large embeddings.

In conclusion: a third informative addition to the PR graph is the display of the level of a random retrieval performance method: this level at  $p = c/d = g$  should preferably be near nil (indicating a large enough embedding for test runs). By subtracting  $g$  from the  $E$ -measure a simple performance measure  $E^*$  as a fraction of Total Recall Ideal System performance is obtained that indicates the gain with respect to a random retrieval method; for large generalities in most cases  $E^* \approx E$  when  $p(g) \gg g$  and  $g \approx 0.0$ .

#### 4 Average Precision-Recall values

For system performance one normally averages the discrete sets of precision and recall values from single queries by averaging precision values at constant recall values to obtain the well-known monotonically decreasing average PR curves without paying attention to the generality or window size values associated with those measurements. In the critical review [10] the authors state with respect to averaging precision and recall values within the same database that precision values should be averaged by using constant scope or window values rather than using constant recall values.

The fact that results for equal cluster related window sizes lay along a radiating set of lines around the origin also has implications for the way average PR curves should be made up. Instead of averaging precision values within recall bins or window bins one should average precision values along  $p/r = constant$  lines and one should only average these  $p, r$  values if they have a common generality value!

An example of the way we obtain an average  $p, r$  curve out of 3 individual queries with different window sizes with respect to the cluster size in the same embedding is given in Figure 10.

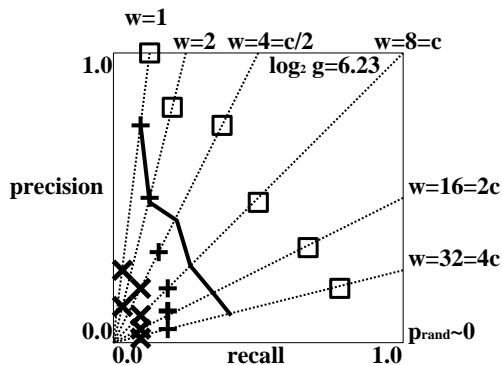


Figure 10: Average  $p, r$  curve from 3 individual cluster size=8 retrieval tests showing  $p, r$  results for different cluster size related window sizes

#### 5 Cluster or Database Size unknown

The extension of the Precision-Recall (or Recall-Precision) graph with generality, cluster window size and random performance level information and the use of the tightest case  $p = r$  plane in 3-dimensional Generality-Precision-

Recall space offers a complete overview of the set of independent information retrieval parameters. In practice however users trying to test an IR System often may not know either cluster size or database size, so that in their studies they can only plot precision versus window size like the values in Figure 11.

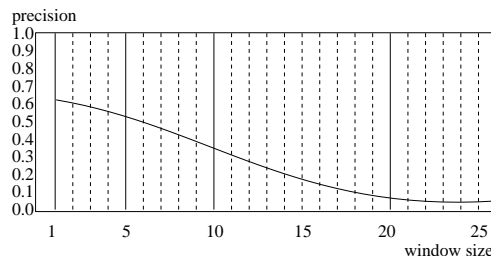


Figure 11: Conventional graph of precision values as a function of scope or window size

This section is about how to plot IR performance figures when cluster size is unknown but database size is known.

#### 5.1 The Cluster Precision-Window graph

If only cluster size is unknown two out of our three performance triple members  $\{-\log_2(\mathbf{g}), \mathbf{r}, \mathbf{p}\}$  i.e. generality and recall would not be available leaving us with  $p, w$  and  $d$ , (precision, window size and database size). For very large database sizes  $d \gg w$  and therefore a random performance level  $p = c/d = g$  with  $c \ll d$  like in the conventional Figure 11 would be near zero and hard to show in the graph. We therefore propose a double logarithmic Precision-Window-size (PW) graph for precision and window values, so that the effect of successive doubling/halving of window and database sizes becomes the unit of change in the graph keeping the effects of both  $w$  and  $d$  visible even for very large databases.

For known cluster size one can easily indicate on the double logarithmic PW graph what the limits of an ideal IR system would be (constant  $p = 1.0$  until  $w = c$ , halving of this max value for every successive doubling of the window until the random level is reached at  $w = d$ ). Figure 12 shows the accessible area of the PW graph for a cluster of size 8 in a database of  $\approx 20,000$  items.

In Figure 13 the precision values for the most compact cluster in a database of  $\approx 20,000$  company logos is shown to closely follow the ideal line in the double logarithmic PW graph.

In case the cluster size is unknown one can add additional lines to the  $\log_2(p) - \log_2(w)$  graph to indicate both a lower bound of the expected random performance level and upper bounds on performance as a function of (unknown) cluster size. With this graph one can easily make an estimation of cluster size by either following the performance curve for growing window size and/or by plotting random performance results, that according to eqn 14 is an estimate for the gener-

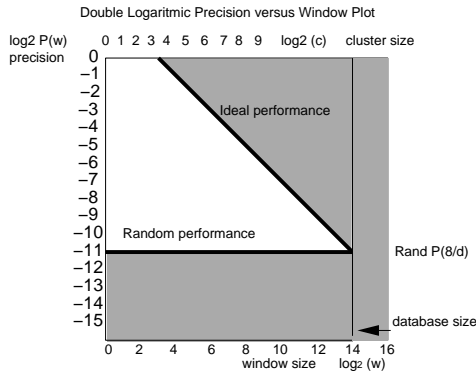


Figure 12: Precision value bounds for database size  $d \approx 16,000$  and for cluster size=8

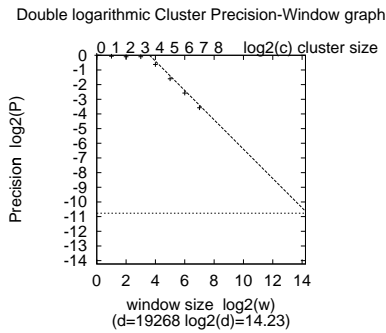


Figure 13: An almost ideal performing compact cluster of 11 members in the large database of company logos

ality  $g$  and given the database size  $d$  can provide an estimate for cluster size  $c$  and hence recall  $r$  can be estimated as well.

Although it is possible to use random retrieval methods to estimate the cluster size from the random precision level as  $c = d \cdot p_{random}$  this method is not very precise; tests with our ground truth test set (see Figure 14) only produced accurate estimates when the window size was at least 500.

We suggest that when random sampling is used to estimate precision like in [5] the same method is also used to estimate generality and recall.

## 6 Final remarks

The addition of the third normalized IR system parameter, generality, in performance studies makes that single number characterizations like van Rijsbergen's  $E$ -measure have become a function of generality. In general for a small embedding the precision, result curves will be close to the ideal curve with a high valued  $E$ -measure and for a growing size of the embedding the  $p, r$  curves will gradually turn into the well-known hyperbolic form of very large databases with an associated low valued  $E$ -measure. So for large enough databases the *Effectiveness* may well drop to a value no longer acceptable for users unless the fall in effectiveness is

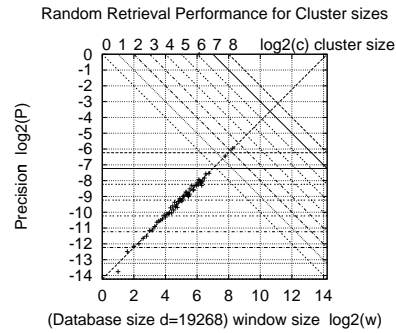


Figure 14: Random performance level as a function of cluster size in the company logo database

encountered by a rise in feature vector effectiveness (like a better or longer feature vector per item). The fall in effectiveness of the Alta-Vista [1] text search engine on Internet for instance is momentarily successfully repaired by the approach followed by the search engine Google [6] that uses additional features (in this case the number of links on the Internet to the retrieved items for a weighted sorting of the results) to regain an acceptable effectiveness at the top of the result list. This process of slow degradation caused by the continuous growth of the database and sudden improvement gains due to more successful arrangements of indexing and clustering methods will probably be seen and be needed more often in the years to come.

### 6.1 Wide- or Narrow-Domain?

Although we have shown in this paper that all the relevant parameters of IR systems can be brought to light with additions to the PR graph, there remains a problem with generality for IR performance tests. How the performance degrades with rising generality greatly depends upon how similar or dissimilar the embedding items are with respect to the relevant items and how compact clusters of relevant items are. In this light one often makes the distinction of wide-domain versus narrow-domain databases: a wide domain database has a more uniformly filled feature space and since feature space in typical IR systems is high-dimensional, performance will remain quite high even for a very large embedding (millions of items) especially when the clusters are quite compact; a narrow-domain database on the other end of the spectrum is characterized by embedding items that are highly similar and therefore IR performance may degrade much more rapidly. Most commercial applications of content-based image or sound retrieval (with a notable exception for Internet search engines) are more likely to be narrow- than wide-domain databases; it is therefore crucial to develop features that remain distinctive in narrow-domain systems. Displaying generality information is therefore only part of the solution in comparing different IR systems: even equal generality level systems may be difficult to compare if one of them is

a wide-domain system (like an Internet image search engine) and the other one is a narrow-domain system (like a worldwide company logo search system). But even for this difficulty one may introduce additional statistical characteristics.

## 7 Conclusions

We have extended the traditional Precision-Recall graph to the Cluster PR graph by adding generality and random levels plus a set of radiating lines indicating the paths along which the results for specific (cluster size related) scopes will lay. For total recall System performance we advocated a comparison with the Total Recall Ideal System performance as the Generality-Precision=Recall graph showing the Precision=Recall values, equal on the diagonal of the PR graph, as a logarithmic function of generality which gives a much more true evaluation of the performance degradation as a function of generality, so that statements can be made about what to expect from the system for the retrieval of specific cluster sizes in a range of database sizes.

Finally in case some relevant retrieval results are aimed at rather than total recall we suggested to use a double logarithmic Cluster Precision Window graph with additional lines that indicate both ceilings for perfect retrieval and floors according to random levels as a function of cluster size.

The extensions to performance graphs suggested in this paper make it possible to better compare performance figures between IR Systems (by explicit mentioning of the generality level) and make it possible to infer precision and recall as a function of window size or scope, reducing the need for additional precision or recall versus scope graphs next to Precision-Recall graphs. For the evaluation of the performance in relation to continually growing database sizes the Generality-Precision=Recall graph offers the best overall IR System performance overview since this graph shows how well the system in question approaches the Total Recall Ideal System. A simple variant of van Rijsbergen's  $E$ -measure is shown to describe retrieval effectiveness well in this case.

## References

- [1] [http://www.altavista.com/sites/help/search/adv\\_help](http://www.altavista.com/sites/help/search/adv_help)
- [2] C. Baumgarten, A probabilistic solution to the selection and fusion problem in distributed information retrieval, in: Proceedings SIGIR'99, Berkeley, August 1999, 246-253.
- [3] J. H. van Bommel, M. A. Musen (editors), Handbook of Medical Informatics, Springer, Heidelberg, 1997.
- [4] W. S. Cooper, A definition of relevance for information retrieval, Information storage and retrieval, Vol 7, 19-37, 1971.
- [5] G. V. Cormack, O. Lhotak, and C. R. Palmer, Estimating precision by random sampling, in: Proceedings SIGIR 99, 273-274, 1999.
- [6] <http://www.google.com/technology/index.html>
- [7] M. Hearst, F. Gey, and R. Tong (editors), Proceedings of the 22nd International Conference on Research and Development in Information Retrieval SIGIR'99, Berkeley, August 1999.
- [8] H. Muller, W. Muller, D. McG. Squire, S. Marchand-Maillet, and T. Pun, Performance evaluation in content-base image retrieval: Overview and proposals, Pattern Recog. Letters, Vol 22, 593-601, 2001.
- [9] K. Porkaew, K. Chakrabarti, and S. Mehtotra, Query Refinement for Multimedia Similarity Retrieval in MARS, in: Proc. ACM Multimedia '99, Orlando, 235-238, 1999.
- [10] V. V. Raghavan, G. S. Wang, and P. Bollmann, A critical investigation of recall and precision as measures of retrieval system performance, ACM Trans. Inf. Syst., Vol 7, No 3, 205-229, 1989.
- [11] C. J. van Rijsbergen, Information Retrieval (second edition), Butterworths, London, 1979.
- [12] G. Salton (editor), The SMART retrieval system, Prentice Hall, Englewood Cliffs, 1971.
- [13] N. Vasconcelos, and A. Lippman, A Probabilistic Architecture for Content-based Image Retrieval, in: Proc. IEEE CVPR, Hilton Head Island, Vol 1, 216-221, June 2000.
- [14] E. M. Voorhees and D. Harman (editors), Proceedings of the 8th Text REtrieval Conference, TREC-8, Gaithersburg, November 1999.