IR-89

Measures in Collection Ranking Evaluation

Zhihong Lu James P. Callan W. Bruce Croft

Computer Science Department, University of Massachusetts
Amherst, MA 01003-4610
{zlu,callan,croft}@cs.umass.edu

Abstract

As a technique to hunt information on the Internet, ***collection location*** has received more attention. Several approaches have been proposed to solve this problem. All these approaches adopt the same procedure: ranking the collections and returning the top-ranks. But these approaches define different measures to evaluate ***collection ranking*** and the measures have significant weaknesses. In this paper, we survey the measures used in current research and propose a new pair of measures that are based on the concepts of precision and recall. The new measures overcome the problems found in the current measures.

1 Introduction

As hundreds or even thousands of collections become available on the Internet, it is practically impossible to query all of them when searching for information on a given topic: not only does such an exhaust search take a long time to complete, but network resource constraints such as limited bandwidth may also prevent the search from completing. Consequently the need to narrow the search to a few useful collections raises the issue of how to locate the most useful collections. Several approaches[1][3][2][6] have been proposed to solve this problem. All these approaches adopt the same procedure: ranking the collections and returning the top-ranks. Although collection ranking is similar to document ranking, these two problems are not identical, and thus the measures used in document ranking can not be used in collection ranking without modifications. Different measures have been defined to evaluate collection ranking. Some come from the measures used in conventional document retrieval and some do not. In this paper, we survey the measures for collection ranking used in the current research, discuss their strengthes and weaknesses, and propose new measures based on the concepts of precision and recall. In Section 2, we briefly survey the measures used in our previous research[1], NetSerf[2] and GLOSS[4][5]. In Section 3 we define the new measures, which are variations of precision and recall. In the final section, we summerize the discussions.

2 A Survey of Measures in Collection Ranking Evaluation

2.1 Ranking with INQUERY

In our previous research on collection ranking [1], the mean-square error was used to compare the effectiveness of variations to the basic collection ranking algorithms.

The mean-squared error (MSE) of the collection ranking for a single query is calculated as:

$$\frac{1}{|C|} \cdot \Sigma_{i \epsilon C} (O_i - R_i)^2$$

where:

O_i = optimal rank for collection i, based on the number of relevant documents it contained (the collection with the largest number of relevant documents is ranked 1, the collection with second largest number of relevant documents is ranked 2, and so on),

R_i = the rank for collection i determined by the retrieval algorithm, and

C = the set of collections being ranked.

This measure is easy to understand (an optimal result is 0), but it overemphasizes the absolute ranks of collections. This measure can not distinguish the minor errors caused by mixing up the ranks of the collections containing nearly equal numbers of relevant documents and the major errors caused by mixing up the ranks of collections containing very different numbers of relevant documents.

Example 2.1 *Consider a query q and 5 collections c_1, c_2, c_3, c_4 and c_5. Table 1 shows the optimal rank O and two estimated ranks $Rank_1$ and $Rank_2$ generated by different ranking algorithms.*

O	*Relevant Documents*	$Rank_1$	$Rank_2$
c_1	*20*	c_1	c_2
c_2	*19*	c_3	c_1
c_3	*8*	c_2	c_3
c_4	*1*	c_4	c_4
c_5	*0*	c_5	c_5

Table 1: The optimal and two estimated ranks for Example 2.1

$Rank_1$ and $Rank_2$ have the same mean square error, 0.667. However $Rank_2$ is better than $Rank_3$ in some sense, since $Rank_2$ only mixes up the ranks of c_1 (20 relevant documents) and c_2 (19 relevant documents), which might not be noticeable to users, while $Rank_1$ mixes up c_2 (19 relevant documents) and c_3 (8 relevant documents), which is a noticeable error. Considering the case that the system suggests to search only first two collections, $Rank_2$ can let the user get 39 relevant documents (the same as the optimal rank), and $Rank_3$ can only let the user get

28 relevant documents (here we assume that the search engine at each individual collection can retrieve all relevant documents).

2.2 NetSerf

NetSerf[2] defines its measure upon a set of queries: Given the set of relevant archives (collections in our context), count the number of queries for which any of the relevant archives were found in the first n top-ranked hits returned by the systems.

For each query in the query set, this measure only uses a binary value to record the contribution of the query: if the query hits any relevant archive in the top n hits, return 1, otherwise return 0. It does not count how many relevant archives have been hit after the first relevant one is hit. All these characteristics make this measure unable to distinguish between the archive with a big number of relevant documents and that with a small number of relevant documents, and thus it ignores many significant errors.

Example 2.2 *Consider a query q and 5 archives a_1, a_2, a_3, a_4 and a_5. Table 2 shows the optimal rank O and three estimated ranks $Rank_2$,$Rank_3$ and $Rank_4$ generated by different ranking algorithms. Table 3 shows the values of the measure in the first n top-ranked hits for this query.*

O	*Relevant Documents*	$Rank_2$	$Rank_3$	$Rank_4$
a_1	*20*	a_2	a_4	a_4
a_2	*19*	a_1	a_3	a_5
a_3	*8*	a_3	a_1	a_1
a_4	*1*	a_4	a_2	a_2
a_5	*0*	a_5	a_5	a_3

Table 2: The optimal and three estimated ranks for Example 2.2

first n hits	$Rank_2$	$Rank_3$	$Rank_4$
1	*1*	*1*	*1*
2	*1*	*1*	*1*
3	*1*	*1*	*1*
4	*1*	*1*	*1*
5	*1*	*1*	*1*

Table 3: The values of the measure for the ranks in Table 2

The measure treats $Rank_2$, $Rank_3$ and $Rank_4$ as equally effective. However $Rank_3$ and $Rank_4$ are much worse rank than $Rank_2$, since $Rank_3$ and $Rank_4$ can not rank the archives with large relevant documents as the top ones. Considering the case that the system suggests to search two archives, $Rank_2$ can retrieve 39 relevant documents, $Rank_3$ can retrieve 9 relevant document, and $Rank_4$ can retrieve 1 relevant document.

2.3 GLOSS

GLOSS defines its measures based on the well-known precision and recall in conventional document retrieval. GLOSS has two versions: the Boolean version[4] and the vector-space version[5]. The definitions of the measures in these two versions have slight differences. GLOSS uses *databases* to refer to *collections* in our context.

2.3.1 The Boolean Version

In the Boolean version of GLOSS, GLOSS defines $Right(q, DB)$, where q is a query and DB is a set of databases. There are two definitions of $Right(q, DB)$.

- The first definition is that $Right(q, DB)$ is the set of all databases in DB containing at least one document that matches query q.
- The second definition is that $Right(q, DB)$ is the set of those databases containing the most matching documents, more formally,

$$\begin{aligned} Right(q, DB) &= \{db \in DB | RSize(q, db) > 0 \wedge \\ &\quad |\frac{RSize(q, db) - hreal}{hreal}| \leq \delta\} \end{aligned}$$

  where $RSize(q, db)$ is the number of documents that match q in database db, and $hreal = max_{db \in DB} RSize(q, db)$.

The first definition is a special case of the second definition (by setting δ big enough).

GLOSS defines the measures R_{Right} and P_{Right} as:

$$R_{Right}(DB) = \frac{|Chosen \cap Right|}{|Right|}$$

$$P_{Right}(DB) = \frac{|Chosen \cap Right|}{|Chosen|}.$$

$R_{Right}(DB)$ is the fraction of the $Right$ databases that are selected and $P_{Right}(DB)$ is the fraction of selected databases that are $Right$ ones.

These measures treat the databases in $Right(q, DB)$ as equally important. If the number of relevant documents in each database varies significantly, these measures can not distinguish between the databases with large numbers of relevant documents and those with small numbers.

Example 2.3.1 *Consider a query q and 5 databases db_1, db_2, db_3, db_4 and db_5. Table 4 shows the optimal rank O and two estimated ranks $Rank_2$ and $Rank_3$ generated by different GLOSS estimators. Table 5 shows the values of the measures in the top n databases for this query.*

O	*Relevant Documents*	$Rank_2$	$Rank_3$
db_1	*20*	db_2	db_4
db_2	*19*	db_1	db_3
db_3	*8*	db_3	db_1
db_4	*1*	db_4	db_2
db_5	*0*	db_5	db_5

Table 4: The optimal and two estimated ranks for Example 2.3.1

	R_{right}		P_{right}	
Top n dbs	$Rank_2$	$Rank_3$	$Rank_2$	$Rank_3$
1	*25%*	*25%*	*100%*	*100%*
2	*50%*	*50%*	*100%*	*100%*
3	*75%*	*75%*	*100%*	*100%*
4	*100%*	*100%*	*100%*	*100%*
5	*100%*	*100%*	*80%*	*80%*

Table 5: The values of the measures for the ranks in Table 4.

Table 5 shows that $Rank_2$ and $Rank_3$ are equally effective, but actually $Rank_3$ is a much worse one (considering the example in Example 2.2).

2.3.2 The Vector-Space Version

In the vector-space version of GLOSS (called gGLOSS in GLOSS terminology), the optimal rank is determined by sorting the databases according to their goodness $Goodness(l, q, db)$ for a given query q. $Goodness(l, q, db)$ is defined as:

$$Goodness(l, q, db) = \Sigma_{d \in Rank(l,q,db)} sim(q, d),$$

where $sim(q, d)$ is the similarity between query q and document d, and $Rank(l, q, db) = \{d \in db | sim(q, d) > l\}$.

It is noteworthy that gGLOSS does not use the relevance information in the definition of $Goodness(l, q, db)$. This definition does not offer benefit than using relevant information directly, since $Goodness(l, q, db)$ is highly correlated with relevant information ([5] gives some discussions for this).

GLOSS defines measures R_n and P_n for a given query as:

$$R_n = \frac{g_n}{i_n}$$

where:
i_n is the accumulated goodness of the top n databases in the ideal rank,
g_n is the accumulated goodness of the top n databases in the rank that gGloss generates.

$$P_n = \frac{|\{db \in Top_n(G) | Goodness(l, q, db) > 0\}|}{|Top_n(G)|}$$

where G is the rank gGLOSS generates.

The measure P_n suffers the same problem as P_{right} in the Boolean version. The measure R_n behaves much better than R_{right}.

Example 2.3.2 *Consider a query q and 5 databases db_1, db_2, db_3, db_4 and db_5. Table 6 shows the optimal rank O and two estimated ranks $Rank_2$ and $Rank_3$ generated by different gGLOSS estimators. Table 5 shows the values of the measures in the top n databases for this query.*

O	$Goodness$	$Rank_2$	$Rank_3$
db_1	0.9	db_2	db_4
db_2	0.6	db_1	db_3
db_3	0.3	db_3	db_1
db_4	0.2	db_4	db_2
db_5	0.0	db_5	db_5

Table 6: The optimal and two estimated ranks for Example 2.3.2

	R_n		P_n	
Top n dbs	$Rank_2$	$Rank_3$	$Rank_2$	$Rank_3$
1	66.7%	22.2%	100%	100%
2	100%	33.3%	100%	100%
3	100%	77.8%	100%	100%
4	100%	100%	100%	100%
5	100%	100%	80%	80%

Table 7: The values of the measures for the ranks in Table 6.

The measure R_n shows that $Rank_2$ is better than $Rank_3$ (compared with R_{right} in the Boolean version, this measure is better), but the measure P_n does not (the same as P_{right}).

3 New Measures

By analyzing the current measures, it is easy to see that the mean square error is in the sense of "too strict", because it does not distinguish between minor errors and major errors. The measures developed for NetSerf and GLOSS are "too loose", because they ignore many errors. New measures need to be defined to overcome these problems.

Like GLOSS, our new measures are based on the concepts of precision and recall. But we modify the conventional precision and recall by considering the numbers of the relevant documents in each collection.

As in our previous research[1], the optimal rank is determined by sorting collections according to the numbers of relevant documents they contain. The new measures are defined as:

For a given query and top n collections,

$$R_n = \frac{\sum_{i=1}^{n} r_i}{Rel}$$

$$P_n = \frac{\sum_{i=1}^{n} r_i}{n}$$

where:

r_i is the number of relevant documents in collection c_i,
c_i is the ith collection in the rank generated by a ranking algorithm;
Rel is the total number of relevant documents in all collections; and
n is the number of collections ranked.

R_n indicates the fraction of the relevant documents that can be retrieved when searching the top n collections. P_n indicates the average number of the relevant documents that each collection contains when searching the top n collections.

The advantages of these measures are that collections with many relevant documents can be distinguished from those with few relevant documents, and that the minor errors of ranks can also be distinguished from the major errors. The disadvantage is that the concept of P_n deviates from traditional concept of precision.

Example 3 *Consider a query q, 5 collections c_1, c_2, c_3, c_4 and c_5, and all rank examples used in previous examples (Table 8). Table 9 shows the values of R_n in the top n collections for a given query. Table 10 shows the values of P_n in the top n collections.*

O	*Relevant Documents*	$Rank_1$	$Rank_2$	$Rank_3$	$Rank_4$
c_1	*20*	c_1	c_2	c_4	c_4
c_2	*19*	c_3	c_1	c_3	c_5
c_3	*8*	c_2	c_3	c_1	c_1
c_4	*1*	c_4	c_4	c_2	c_2
c_5	*0*	c_5	c_5	c_5	c_3

Table 8: The optimal and the estimated ranks for Example 3

	R_n				
Top n colls	*Optimal*	$Rank_1$	$Rank_2$	$Rank_3$	$Rank_4$
1	*41.7%*	*41.7%*	*39.6%*	*2.1%*	*2.1%*
2	*81.3%*	*58.3%*	*81.3%*	*18.7%*	*2.1%*
3	*97.9%*	*97.9%*	*97.9%*	*60.4%*	*43.8%*
4	*100%*	*100%*	*100%*	*100%*	*83.3%*
5	*100%*	*100%*	*100%*	*100%*	*100%*

Table 9: The values of R_n for the ranks in Table 8.

	P_n				
Top n colls	*Optimal*	$Rank_1$	$Rank_2$	$Rank_3$	$Rank_4$
1	*20*	*20*	*19*	*1*	*1*
2	*19.5*	*14*	*19.5*	*4.5*	*0.5*
3	*15.7*	*15.7*	*15.7*	*9.7*	*7*
4	*12*	*12*	*12*	*12*	*10*
5	*9.6*	*9.6*	*9.6*	*9.6*	*9.6*

Table 10: The values of P_n for the ranks in Table 8.

Considering $Rank_1$, the data on the second and the third rows in Tables 9 and 10 indicate that $Rank_1$ makes some big mistakes between rank 2 and rank 3. Considering $Rank_2$, the mistakes between rank 1 and rank 2 are minor. Considering $Rank_2$, $Rank_3$ and $Rank_4$, it is easy to see that $Rank_2$ is much better than $Rank_3$ and $Rank_4$. Tables 9 and 10 show that the new measures make it possible to distinguish between major errors and minor errors.

4 Conclusion

This paper surveys the collection ranking measures used in current research. The measures used in NetSerf and GLOSS do not consider information about the relevant documents in each collection, which causes them to ignore significant errors in the rankings. The mean square error overemphasizes the absolute ranks of each collection, which make it count some minor errors as major errors. By analyzing the mistakes that the current measures made, a new pair of measures are proposed that are based on the concepts of precision and recall, and that include the information about relevant documents in each collection. The examples show that the new measures overcome the problems found in the other measures.

Acknowledgements

This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor.

References

[1] James P. Callan, Zhihong Lu and W. Bruce Croft. Searching Distributed Collections with Inference Networks. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pages 21-28, Seattle, July 1995. Association for Computing Machinery.

[2] NetSerf: Using Semantic Knowledge to Find Internet Information Archives. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pages 4-11, Seattle, July 1995. Association for Computing Machinery.

[3] Glossary of Servers *http://gloss.stanford.edu*

[4] Luis Gravano and Hector Garcia-Molina. Precision and Recall of GLOSS Estimator for Database Discovery. *Stanford Univerdity Technical Note Number STAN-CS-TN-94-10.*

[5] Luis Gravano and Hector Garcia-Molina. Generalizing GLOSS to Vector-Space Databases and Broker Hierarchies. *Proceedings of the 21st VLDB Conference,* Zurich, Switchland 1995.

[6] WAIS 2.0: Technical Description *http://www.wais.com/newhomepages/techtalk.html*