CS 6740/INFO 6300 **Spring 2010**

Lecture 4 – February 9

Lecturer: Prof Lillian Lee *Scribes: Jerzy Hausknecht & Kent Sutherland*

Evaluation: Annotation and Experimental Design

1 Review: Pivoted Document Length Normalization

In the previous lecture we discussed pivoted document length normalization [Singhal et al. 96], a simple technique that applies a correction for the observation that document relevance correlates with document length. Through careful empirical verification of previous assumptions, they showed that the seemingly simple normalization term could have a big impact on results.

However, in our discussion of the analysis that led to pivoted document length normalization, we passed over a basic question: How were the relevance judgments in the TREC dataset made on the approximately 740,000 documents and 50 queries?

2 Generating Evaluation Data

As an aside, this lecture is potentially relevant to any setting where individual experiments are "expensive" (labeling items in image retrieval, systems research, etc.).

Given an evaluation corpus of 740,000 documents and 50 queries, it would require 37 million individual relevancy judgments to fully label the entire corpus. Assuming that the authoritative judgments were made by humans it would take over 610 weeks of manual labeling, under the generous assumption that a document-query judgment can be made in 10 seconds. Such a large task would seem infeasible, even if it were distributed among a large number of people. If these judgments couldn't be performed by people, how were they made? Moreover, how can they be trusted to be accurate if they were generated without direct human involvement?

3 Labeling Strategies

As we have seen, it would be a massive undertaking for humans to manually annotate 37 million document-query pairs. For even larger corpora, such as an index of all webpages on the internet, it would be simply impossible. As of July 2008, Google's search index contained over 1 trillion pages. In the face of massive datasets various methods have been proposed for annotating documents.

3.1 Cheap Labor

The availability of cheap labor on services such as Amazon Mechanical Turk has made it much easier to quickly distribute a huge number of tasks across a large workforce. While large collections can not be completely annotated, distributed labor can still be used effectively [Alonso 09].

3.2 Games

Rather than paying for annotations, other researchers have effectively used games such as the ESP game [von Ahn & Dabbish 04] to get people to annotate documents (images in the case of the ESP game). However, others have managed to exploit deficiencies in the scoring systems of such games, indicating weaknesses in their incentives; [Robertson, Vojnovic, Weber 09] found it was possible to earn points in the ESP game simply by predicting new tags based on the taboo words for each image.

3.3 Random Sampling

A straightforward approach to annotating a corpus is random sampling. However, the sparsity of relevant documents for a given query presents a considerable challenge to this strategy. For a given query most documents won't be relevant, which results in a boring task for labelers. This boredom and repetition primes labelers to label all documents as nonrelevant, regardless of the true relevance. Additionally, as most metrics are heavily influenced by the retrieval of relevant documents, having very few such documents can make distinguishing between systems difficult.

3.4 TREC Technique

Since random sampling is problematic due to the sparsity of relevant documents, a bias towards relevant documents would make annotation much easier. To achieve this, the TREC organizers use *pooling* to create a smaller subset of documents to annotate.

Pooling is based on the assumption that the "bakeoff" competitions each year contain a number of good (relevant documents are highly ranked) and diverse (different systems return different relevant documents) systems. Therefore, the pool of documents returned by all the systems for a given query should provide a good representation of all documents relevant to that query in the corpus. The end result is a much smaller set of documents to annotate, with a much higher proportion of relevant documents. For example, if 40 teams competed in the bakeoff, the top k (k is the pool depth, typically 100) documents returned by each team could be annotated and the performance of each system could be scored. Rather than having to perform 37 million annotations, the maximum number of documents to annotate is *pool depth* * *team count*. The document pool is therefore defined as:

$$\bigcup_{\text{Systems } S_i} \text{top } k \text{ documents according to } S_i$$

Unsurprisingly, this approach is not perfect:

1. Evaluation measures may depend on knowing all the relevant documents, not just the top ranked documents

   - average precision (over the relevant document ranks)
     $R^q :=$ set of all relevant docs for query q
     average precision $= \frac{1}{|R^q|} \sum_{d \in R^q}$ precision @ rank of d
     This requires knowing all relevant documents.
   - NDCG @ k: normalized by the score of optimal ranking.
     This requires knowing all relevant documents.

2. The annotated corpus should be reusable for future systems. These systems may retrieve documents that weren't retrieved by the original teams, and thus were unlabeled. How do we handle such documents when generating evaluation scores for future systems?

4 Assumptions About Relevance

In general, documents that were not retrieved by the original collection of systems are assumed to be nonrelevant. This assumption is based on the hope that the pool depth is large enough and that the systems that generated the original pool were both good and diverse enough to retrieve most of the relevant documents for a given query.

The diversity assumption is essential to the success of the pooling strategy. If all systems competing returned the same set of documents, then there would be a significantly smaller pool. However, the fact that TREC is a competition somewhat undermines this goal of diversity since it encourages teams to include techniques that have been previously proven effective.

This assumption raises two important questions about the TREC competitions. First, is it possible to verify the assumption that this is an effective and fair way to score systems? Second, even if the assumptions made aren't entirely correct, are previous comparisons based on TREC data still valid?

4.1 Checking Comparisons and Assumptions

In order to check the validity of pooling, [Zobel 98] compared systems with differing pool depths. While in general it isn't feasible to increase the pool depth of all previous TREC pools after the fact, the pool depth can be artificially shrunk. This increases the number of non-judged documents, simulating more missing data. Based on the TREC pooling assumptions, such documents are assumed to be nonrelevant. Zobel concluded that the inconsistencies in system performance were small with regard to shrinking the pool depth. These results were confirmed when earlier TREC pools were deepened after the fact [Harman 95]. Additionally,

similar experiments by other researchers have found that the statistical significance of ranking differences generally holds.

Zobel also artificially reduced the diversity of the pool by deleting systems. As a result, documents that were added to the original pool by the deleted system were considered "unlabeled." This would penalize systems that returned documents that no other system added to the pool. The results of these tests showed that the effect of what he calls "pooling bias" is small, and that the "effect is unimportant."

It is unclear what this conclusion actually means with regard to the effects of system deletion. It could imply that the TREC pool lacks sufficient system diversity, or that the ranking schemes are robust enough to handle incomplete data. The effect of system deletion was more dramatic using the TREC-3 dataset when compared with the TREC-5 dataset. The TREC-3 pool was generated using fewer systems, and for the 10 queries with the most answers, a system's score improved by an average of 19% when its uniquely retrieved documents were added back into the pool. In comparison, for the TREC-5 dataset, the average system score for the 10 queries with the most answers improved by 7%.

4.2 Recall

While the TREC pooling method was found to be reasonable when comparing retrieval systems, the effect of annotating only the top-ranked documents could be more dramatic if recall is important. Not all information retrieval applications are solely precision-focused; examples include patent searches and surveys of literature in a specific field. Many evaluation measures also have a recall component (see **Section 3.4).**

In his same paper, Zobel estimates that between 30 and 50% of relevant documents remain unjudged. This value was estimated by calculating the "rate of arrival" of relevant documents as the pool depth increased. The total number of relevant documents that remain unannotated can be estimated by first performing a regression to estimate the number of "new arrivals" at larger pool depths.

$$\begin{aligned}\text{Rate of arrival} = &(\text{relevant documents at depth } k-1)- \\ &(\text{number of relevant documents at depth } k-1)\end{aligned}$$

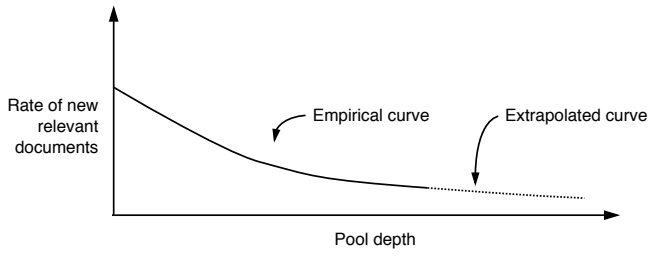

Figure 1: Rate of arrival.

The estimated number of relevant documents is then equal to the area under this curve [Figure 1].

5 Compensating for Incomplete Labeling

Effectively compensating for the fact that labels are incomplete is still an open question. Some of the proposals to solve this problem include:

- Use machine learning: Train a classifier on the pool to learn relevant/nonrelevant documents, then run the classifier on unlabeled data [Büttcher et al. 07]. However, if such an approach worked, why wouldn't it work as a complete IR system? Furthermore, systems could optimize their measured performance by simply reproducing the classifier's results, without regard to actual improvements in retrieval.

- Annotate "smarter" [Carterette et. al 06] (best student paper award and best paper award, SIGIR 2006). Starting with the assumption that you have two systems (yours and a baseline), for a given evaluation metric you annotate the next document(s) most likely to show that the two systems differ.

- Create a new evaluation metric, such as "bpref" [Buckley and Voorhees 04]. This metric is similar to average precision, but ignores unannotated documents. As a result, it is less sensitive to the assumption that non-judged documents are nonrelevant.

6 Questions

6.1 TREC Pooling

One of the effects that [Zobel 98] explored was the impact of pool depth on system ranking. Consider two systems being tested as if they were in a TREC bakeoff pool. The following table contains the top 10 documents returned by these systems for a single query, as well as annotations of relevance/non-relevance for that query. Assume there are a total of ten relevant documents in the entire corpus.

System 1	**System 2**
doc. 1 : relevant	doc. 1 : relevant
doc. 2 : relevant	doc. 3 : relevant
doc. 4 : nonrelevant	doc. 5 : relevant
doc. 5 : relevant	doc. 6 : nonrelevant
doc. 3 : relevant	doc. 7 : nonrelevant
doc. 9 : relevant	doc. 9 : relevant
doc. 6 : nonrelevant	doc. 10 : relevant
doc. 8 : nonrelevant	doc. 11 : relevant
doc. 10 : relevant	doc. 12 : relevant
doc. 13 : relevant	doc. 5 : relevant

a) What is the precision at 5 for both systems? What is the precision at 10?

b) Calculate the normalized discounted cumulative gain (NDCG) for both systems using both the top 5 and top 10 documents. Does the same switch in relative rankings occur? Assume binary relevance scores.

c) Which error metric would you expect to be more robust to changes in pool depth? Explain.

d) Pretend you have not yet classified the documents as relevant/nonrelevant. Considering the top ten results for both systems, how could you minimize the number of documents classified while still generating an accurate *relative* ranking with regard to precision at 10?

Another issue with the TREC pooling strategy is that the data may not necessarily be reusable for new systems, which could possibly retrieve many unlabeled documents. The following table contains the top 10 documents returned by a third system for the same query as above. This system wasn't included in the original pool, and therefore may highly rank some unlabeled documents.

System 3
doc. 1
doc. 9
doc. 5
doc. 10
doc. 6
doc. 2
doc. 3
doc. 4
doc. 11
doc. 14

e) If we assume that unclassified documents are nonrelevant, what is system 3's precision at 5, assuming a pool depth of 5 for systems 1 and 2? Precision at 10 with a pool depth of 10?

6.2 Evaluation Metrics

a) When testing new systems that weren't members of the original pool, two methods of evaluation can be used. The first is to consider the results returned as a "condensed list," where unjudged documents are ignored and judged documents are therefore shifted up the results list. The other is to assume all unjudged documents are nonrelevant, and score them as such ("assumed irrelevance") [Webber & Park 09].

What sort of problems do each of these approaches introduce when comparing the scores of new systems against the original pooled systems?

b) Given these two extremes, would it be possible to blend these two approaches such that unjudged documents are assumed to be "partially" nonrelevant? Documents would then either be relevant, nonrelevant, or unjudged. Systems returning an unjudged document for a query would therefore be penalized less than those that return judged nonrelevant documents. Would this be a reasonable approach, while still assuming binary relevance judgments? Were you to explore this approach, what would be a reasonable starting point?

c) Others have looked into compensating for these scoring biases in various ways [Webber & Park 09]. Another bias that emerges relates to the number of systems in the pool for a particular query. When considering the results using "assumed irrelevance," how does the number of systems in the pool affect the results? How about when treated as a condensed list?

7 Answers

7.1 TREC Pooling

a) Remember that precision at k is: $\frac{\text{\# of relevant documents in the top k results}}{k}$

Metric	System 1	System 2
prec@5	80%	60%
prec@10	70%	80%

Note that the change in pool depth changes the relative ranking of the two systems.

b) Remember that the formula for DCG at k is: $\sum_{i=1}^{k} \frac{2^{\text{relevance score of document i}} - 1}{\log_2(i+1)}$.

NDCG at k is the DCG at k score divided by the optimal score.

Sample calculations:

Optimal score at 5: $\frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5} + \frac{1}{\log_2 6} = 2.948$

DCG at 5 for system 1: $\frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 5} + \frac{1}{\log_2 6} = 2.448$

NDCG at 5 for system 1: $\frac{2.448}{2.948} = 0.830$

Optimal score at 10: $\frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5} + \frac{1}{\log_2 6} + \frac{1}{\log_2 7} + \frac{1}{\log_2 8} + \frac{1}{\log_2 9} + \frac{1}{\log_2 10} + \frac{1}{\log_2 11} = 4.544$

DCG at 10 for system 1: $\frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 5} + \frac{1}{\log_2 6} + \frac{1}{\log_2 7} + \frac{1}{\log_2 10} + \frac{1}{\log_2 11} = 3.395$

NDCG at 10 for system 1: $\frac{3.395}{4.544} = 0.747$

Metric	System 1	System 2
DCG@5	2.448	2.131
NDCG@5	0.830	0.723
DCG@10	3.395	3.726
NDCG@10	0.747	0.820

Scoring with NDCG also causes the ranking to change when changing pool depth.

c) In general, one would expect NDCG to be more robust to expanding pool depth than precision at k. This is because NDCG gives a larger weight to the highest-ranked results than lower-ranked results. In contrast, precision at k gives uniform weight to all results. Overall this means NDCG will be less affected by the tail end of a pool than precision at k.

d) Documents 1, 3, 5, 6, 9, and 10 do not have to be scored since both systems returned them in the top 10 results. It does not matter whether the documents are relevant or not, as it won't affect the relative ranking of the two systems. This assumption is true when scoring using precision at k, but it may not hold for other scoring metrics.

This becomes clear when dividing the precision at k calculation into two parts:

precision at k for system $x = \frac{\text{\# of rel docs both systems returned + \# of rel docs only system } x \text{ returned}}{k}$
precision at 10 for system 1 $= \frac{5+2}{10}$
precision at 10 for system 2 $= \frac{5+3}{10}$

When considered under rank, the five documents relevant documents returned by both systems have no effect on the system rankings.

e) System 3's precision at 5 is 40%, assuming a pool depth of 5 on the original systems. Documents 9 and 10 are assumed to be nonrelevant since they did not appear in the top 5 results of any system in the original pool.

System 3's precision at 10 is 70%, assuming a pool depth of 10 on the original systems. Document 14 is assumed to be nonrelevant since it did not appear in the top 10 results of any system in the original pool.

7.2 Evaluation Metrics

a) Treating results as a condensed list is the assumption that bpref makes, which biases scores towards new systems when doing comparisons against the original systems in the pool. Assumed irrelevance has the exact opposite problem, biasing scores in favor of the pooled systems.

b) Such an approach could potentially reduce biases for or against systems not in the original pool based on the ranking metric used. This could increase the usefulness of annotated data after a bakeoff competition is completed.

A fairly basic starting point for implementing such a metric would be to impose a small penalty for returning nonjudged documents. This would reduce the bias against systems that weren't originally in the pool which could return unjudged documents. Retrieval of such documents wouldn't be penalized as heavily as if they were judged nonrelevant. It would also reduce the advantage a non-pooled system receives if scoring as a condensed list, since unjudged documents would still be penalized. To start, this penalty could be calculated as the average percentage of relevant documents to a given query in the pool.

c) There is a bias against new systems under assumed irrelevance, but the bias decreases as the number of systems in the pool increases. This occurs because system diversity is directly related to the number of systems in the pool, and greater system diversity leads to more judged documents. When treating results as a condensed list, the change in bias is "more complex. Fewer unassessed documents means that the condensed and true relevance vectors are more similar. At the same time, more pooled systems omitting a document strengthens the odds that the document is in fact nonrelevant, and therefore strengthens the bias resulting from excising it and replacing it with a pooled document, partially counteracting the effect of better accuracy. As a result, condensed lists have less bias than assumed irrelevance for small numbers of pooled systems, but assumed irrelevance has less bias for larger ones." [Webber & Park 09] (observation originally made in [Sakai 08]).

8 References

1. O. Alonso, S. Mizzaro. Relevance criteria for e-commerce: a crowdsourcing-based experimental analysis. *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval* (2009) pp. 760-761.

2. J. Alpert, and Nissan Hajaj. "We knew the web was big...." Official Google Blog. 07 Jul 2008. Google, Web. 15 Feb 2010. <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html>.

3. Amazon Mechanical Turk. http://www.mturk.com.

4. C. Buckley, E.M. Voorhees. Retrieval evaluation with incomplete information. *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval* (2004) pp. 25-32.

5. S. Büttcher, C.L.A. Clarke, P.C.K. Yeung, I. Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (2007) pp. 63-70.

6. B. Carterette, J. Allan, R. Sitaraman. Minimal test collections for retrieval evaluation. *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval* (2006) pp. 275.

7. D. Harman. Overview of the fourth text retrieval conference (TREC-4). *Proc. Text Retrieval Conference* (October 1995).

8. S. Robertson, M. Vojnovic, I. Weber. Rethinking the ESP game. *Proceedings of the 27th international conference extended abstracts on human factors in computing systems* (2009) pp. 3937-3942.

9. T. Sakai. Comparing metrics across TREC and NTCIR: The robustness to system bias. *Proceeding of the 17th ACM conference on information and knowledge management* (2008) pp. 581-590.

10. A. Singhal, C. Buckley, M. Mitra. Pivoted document length normalization. *Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval* (1996) pp. 21-29.

11. L. von Ahn, L. Dabbish. Labeling images with a computer game. *Proceedings of the SIGCHI conference on human factors in computing systems* (2004) pp. 319-326.

12. W. Webber and L. A. F. Park. Score adjustment for correction of pooling bias. *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval* (2009) pp. 444-451.

13. J. Zobel. How reliable are the results of large-scale information retrieval experiments?. *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (1998) pp. 307-314.