Juru at TREC 2003 - Topic Distillation using Query-Sensitive Tuning and Cohesiveness Filtering

Einat Amitay, David Carmel, Adam Darlow, Michael Herscovici,
Ronny Lempel, Aya Soffer
IBM Haifa Research Lab
Reiner Kraft, Jason Zien
IBM Almaden Research Center

1 Introduction

This is the third year that our group participates in TREC's Web track, the second year in the topic distillation task. Our experiments last year, as well as those of other participants, indicated that sophisticated link-based measures did not significantly improve search results in comparison to standard text-based relevance scoring. We thus focused our experiments this year on improving the ranking algorithms of our core search engine, Juru, and on developing measures that are good indicators of topical pages.

In particular, realizing that one ranking flavor does not fit all queries [3][6], we developed a method, which fine tunes the parameters governing the ranking formula based on the nature of the query. This novel ranking method, called the *QUEry Sensitive Tuner* (or QUEST), tunes the ranking parameters according to the query type. QUEST classifies queries into "informational" vs. "navigational" by considering both the query's length and the expected number of documents containing all query terms (*edf*). For queries with a few expected results, each document's score is primarily determined according to the document's textual score, i.e. its similarity to the query. On the other hand, for queries with many expected results, document scores are determined by considering additional factors such as anchor-text data, number of in-links, etc.

In addition, we continued experimenting with some of the topic distillation filters we introduced last year [2], as well as with a new cohesiveness filter. The cohesiveness filter tries to identify pages that focus on the desired topic in contrast to pages than just mention it in passing, or which mention it in the context of a broader topic. This is achieved by identifying pages in which the query terms are uniformly distributed over the entire page.

The rest of this paper is organized as follows: Section 2 describes the QUEST algorithm and the query parameters tuned by the algorithm according to the query type. In section 3 we describe the cohesiveness filter that tries to distill pages that focus on the desired topic. Section 4 describes the results of the official runs submitted to TREC. Section 5 concludes.

2 QUEry Sensitive Tuner (QUEST)

The QUEST algorithm tunes the query parameters according to the query's characteristics, which in turn imply its type. QUEST classifies queries into "informational" vs. "navigational" by considering both the query's length and the expected number of documents containing all query terms (*edf*). The main rationale is that for short queries with many expected results (large *edf),* standard IR techniques based on textual scores cannot discriminate between topical and non-topical pages, therefore more factors, especially static scores and anchor-text scores associated with the documents, should be used in order to distill the best results. On the other hand, for long queries with few expected results, document's static scores, which are independent of the query, should only take a secondary role, as standard IR techniques are expected to return satisfactory results.

After query classification, the query parameters are tuned according to the query type. For "navigational" queries, parameters are set such that the number of in-links per page has stronger effect on the page's final score than for "informational" queries for which the textual score is dominant in determining the final score. Similarly, the anchor text associated with these in-links is weighted more heavily in navigational queries compared to informational ones. QUEST does not, however, assume just the two extremes; rather it tunes the parameters on a sliding scale ranging from purely navigational to purely informational.

We now describe in more details the QUEST algorithm. QUEST treats separately queries containing one, two and three+ terms. For each query length, it maintains a threshold on the query *edf*. In addition, it also maintains two sets of values for several ranking parameters, one set for informational queries and one set for navigational queries. A query with an *edf* lower than the threshold is classified as "informational" and its parameters are set using the informational set of parameters. A query with an *edf* higher then the threshold is considered "navigational" and its parameters are set using the navigational set of parameters. See Section 2.2 for details on the calculation of the *edf*.

2.1 Query Parameters tuned by QUEST

QUEST tunes three sets of parameters as described below:

I. ***Boosts for different token types***. The tokens of a document are classified into several types, and the significance of a token and its contribution to the document's textual score is determined by the boost associated with its type. Thus, the occurrence of tokens with a high boost in the document's content significantly affects its textual score, while tokens with a low boost contribute much less. The token types include:

   a. **Textual tokens**: tokens extracted from the document's raw text which are differentiated into:
      i. Title tokens – extracted from the document's title.
      ii. Strong tokens – extracted from the document's headers.
      iii. Mid tokens – extracted from the document's emphasized text (colored, bold, etc.).
      iv. Regular tokens – all the rest.

b. **Anchor tokens**: tokens extracted from the anchor text of the document's in-links. These tokens are differentiated according to the relation between the source and target of the link:
   1. Different site anchor: anchor tokens where the source site differs from the target site
   2. Same site anchor: anchor tokens where the anchor and the target pages are from the same site but in different directories.
   3. Same dir anchor: anchor tokens where the source and the target pages reside in the same directory.

c. **URL tokens**: tokens extracted from the document's URL.

d. **Snippet tokens**: Tokens extracted from the document's snippet. We compute for each document a snippet based on its anchors, using the method described in [1].

For informational queries, textual tokens are given the highest boosts while for navigational queries anchor tokens, URL tokens, and snippet tokens receive higher boosts.

II. ***Lexical Affinity weight (LA-Weight).*** Our ranking algorithm takes into account lexical affinities common to the query and the document, in addition to simple query terms. Lexical affinities are pairs of closely related terms frequently found in proximity to each other [7]. Each query term, a simple keyword or a lexical affinity, contributes to the textual score of the document according to its term frequency and to its inverse document frequency (following the *tf-idf* formula). The LA-weight determines the relative contribution of lexical-affinities to the document's score compared to simple keywords. Experiments have shown that the LA-weight should be smaller for longer queries [4]. In accordance, QUEST assigns a lower LA-weight for informational queries as compared to navigational queries.

III. ***Static Score coefficient:*** The final score of a document is computed by linearly combining its textual score with a static score. The static score is based on the number of its in-links. The Static Score coefficient determines the relative weight of the static score with respect to the weight of the textual score of the document. QUEST assigns a higher value to the static score coefficients for navigational queries.

2.2 Approximating the expected document frequency (*edf*) per query

The main feature used by QUEST for query classification is the expected document frequency *edf*. For one-term queries, the document frequency (*df*) can be precisely determined since the *df* of each term is stored within the index. For multi-term queries, the *edf* must be approximated since the only way to derive the precise *edf* is to process the query.

Given a query with k terms $q = q_1..q_k$. The *edf* of the query is approximated based on the *df* values of the individual query terms. Assuming independence between query terms, the number of documents containing all of the query terms can be estimated by multiplying the occurrence probability of all query terms. The occurrence probability of a query term q_i can be approximated by $Pr(q_i) = df(q_i)/|D|$, where $df(q_i)$ is the document frequency of term q_i and $|D|$ is the total number of documents in the collection. Thus, the *edf* of a query *q* with *k* independent terms is:

$$edf(q) = \frac{\prod_{i=1}^{k} df(q_i)}{|D|^k} |D|$$

Since query terms are usually not independent, but are rather expected to co-appear in documents, we heuristically multiply the above by the number of query terms *k*:

$$edf(q) = \frac{k \prod_{i=1}^{k} df(q_i)}{|D|^{k-1}}$$

3. The Cohesiveness Filter

The relevance score computed above finds good individual candidates for topical pages. However, given that the goal of the topic distillation task is to find a *set* of topical pages, we apply some additional filters that influence the final ranking. The goal of these topic distillation filters is to identify pages that exhibit features of a good topical page, and to boost their query relevance score. We applied the following sequence of filters to the initial search results: 1) duplicate-elimination filter, 2) site-compression filter, and 3) the new cohesiveness filter. The first two filters were already reported last year in [2]. The new cohesiveness filter tries to identify pages that focus on the desired topic in contrast to pages that just mention it in passing, or which mention it in the context of a broader topic. This is achieved by identifying pages in which the query terms are uniformly distributed over the entire page.

More specifically, for each document in the result set we measure the uniformity of the query terms along the document's content. This is done by measuring the entropy of the occurrence distribution of the query terms within the document. The entropy is maximal when the term occurrences are uniformly distributed over the document's content. The entropy is minimal when all term occurrences are close to each other. We conjecture that the larger the entropy of the term distribution, the higher its uniformity.

Given a query term *t* with a list of positions $o_1, o_2, \ldots, o_k$ within document *d* of length $|d|$, the entropy of the term occurrence distribution within *d* is measured by:

$$entropy(t,d) = -o_1 \log o_1 - \sum_{i=2}^{k} (o_i - o_{i-1}) \log(o_i - o_{i-1}) - (|d| - o_k) \log(|d| - o_k)$$

The cohesiveness of the document *d* for query *q* is defined by the weighted average entropy of *q's* query terms within *d*:

$$cohesiveness(d,q) = \sum_{t \in q} idf(t) * entropy(t,d)$$

The cohesiveness filter computes for each document in the result set a new score based on its previous score and its cohesiveness. It then re-ranks the search results based on the new score. The new score is a linear combination of the previous score and the cohesiveness score. The cohesiveness filter weight determines the relative weight between the two scores. This weight is set by QUEST according to the query type. For purely informational queries this weight is low, while for purely navigational queries the weight is high.

The cohesiveness filter is especially useful for queries with high frequency terms. In such cases, the cohesiveness filter will prefer pages where the query terms occur throughout the entire document over pages where query terms appear only in part of the document.

4. Results

We used the Juru search engine [5] to index and search the pages in the ".gov" domain. Each page was indexed based on its content as well as its anchor descriptions, its URL, and its snippet (see Section 2). Each page is scored by a linear combination of its textual score and its link topology score (a static score). The static score of page *p* is based on the number of links *n* pointing to *p*:

$$St(p) = \begin{cases} 1.0 & n \geq N \\ \sqrt{n/N} & otherwise \end{cases}$$

The constant *N* determines an upper bound on a page's in-link number; each page with more than *N* in-links receives the maximum static score of 1. The *N* parameter is also set by QUEST according to the query type, low value for informational queries and high value for navigational queries.

The combined scores are used to rank the set of pages. The top 200 pages are re-ranked using the sequence of filters described above designed to guarantee a mixture of good sources in the top-10 list returned by the system. The top 100 pages were submitted to TREC.

We submitted 5 runs for the topic distillation task. The JuruFull run scored pages based on both a textual and a topological score. The query parameters were tuned separately for each query using the QUEST algorithm as described above, and all filters were invoked on the search results. In the JuruNoAnchor run we zeroed the boosts of all anchor tokens, thus, textual ranking is based only on the document content. In the JuruNoCohes run the cohesive filter was ignored by zeroing the cohesiveness filter weight. In JuruNoQueryDiff the QUEST algorithm was ignored by fixing the values of the ranking parameters for all the queries. In JuruNoSS the document static scores were ignored.

Table 1 shows the average P@10 and average R-precision of our runs and the average-best and median P@10 of all participants. While the results of all our runs are much higher than the median, the results are somewhat disappointing. For 16 topics

JuruFull could not find pages marked relevant by the assesors in its top 10 results, among them 3 topics for which all participants completely failed. On the other hand, for 7 topics JuruFull achieved the best result among all participants. Both the JuruNoAnchor and the JuruNoSS runs achieved significantly lower results than the other runs, indicating the significance of link analysis, contradicting our findings from previous year about the relatively insignificance of link analysis for the topic distillation task. There was however no difference between the runs applying QUEST and the cohesiveness filter, and thus the experiments we hoped to achieve by participating in this task are inconclusive.

	Best	Median	JuruFull	JuruNoAnchor	JuruNoCohes	NoQueryDiff	JuruNoSS
P@10	0.28	0.064	0.122	0.088	0.122	0.122	0.086
R-Precision			0.110	0.100	0.106	0.117	0.099

Table 1 -- Average P@10 and R-precision of our runs and the average-best and median P@10 of all participants.

Figure 1 shows the difference between P@10 of our runs and the median P@10 of all participants. For almost all topics (except 3 for the JuruFull run) our runs achieved a better result than the median.

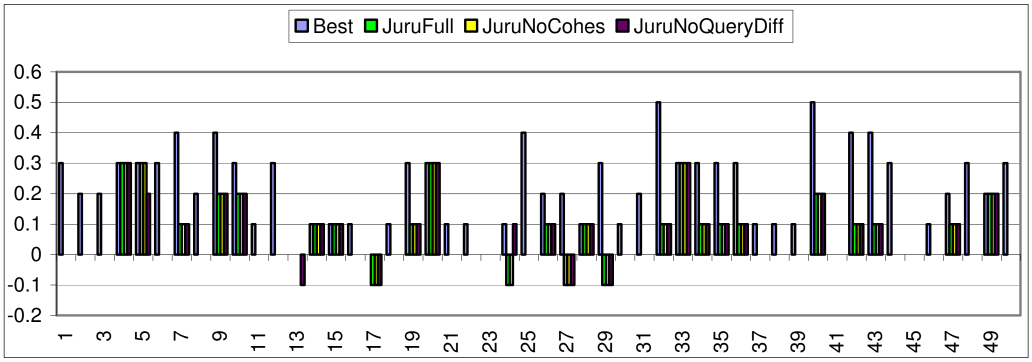

Figure 1 -- The difference between P@10 of the best result, some of our runs, and the median P@10 of all participants

5. Summary

Our experiments this year focused on improving the ranking algorithm of our core search engine, and on developing measures that are good indicators of topical pages. We experimented with the QUEST algorithm that tunes the query parameters according to the query's characteristics. We also experimented with the cohesiveness filter that tries to find topical pages by identifying those in which the query terms are uniformly distributed over the entire page. Our results demonstrate that link analysis and anchor-text data slightly improved the results this year, in contrast to last year. However, our results do not indicate any advantage for QUEST or the cohesiveness filter. One reason for this is the apparent disparity between our understanding and the assessors understanding of the notion of a 'topical page". The topic distillation task, in our opinion, is still not well defined. Consequently, our system in several cases returned many good pages (according to our judgment) that were rejected by the assessors as non-relevant. We believe that QUEST and cohesiveness can indeed make a difference - more exhaustive experiments are needed to study their effectiveness.

References

[1] Amitay E., Paris C. (2000). *Automatically Summarizing Web Sites - Is There A Way Around It?* CIKM 2000, pp. 173-179.

[2] Amitay E., Carmel D., Darlow A., Lempel R., Soffer A. (2002). *Topic Distillation with Knowledge Agents*. In Proceedings of the Eleventh Text REtrieval Conference (TREC 2002), National Institute of Standards and Technology (NIST).

[3] Broder A. (2002). *A taxonomy of web search.* ACM SIGIR Forum 36 (2), pages 3--10, 2002.

[4] Brown E.W., Chong H.A. (1998). *The GURU system in TREC-6*. The Sixth Text Retrieval Conference (TREC-6), pages 535–540, 1998. National Institute of Standards and Technology (NIST).

[5] Carmel D., Amitay E., Herscovici M., Maarek Y., Petruschka Y., Soffer A. (2001). *Juru at TREC 10 - Experiments with Index Pruning*. In Proceedings of the Tenth Text REtrieval Conference (TREC 2001), National Institute of Standards and Technology (NIST).

[6] Kang I.H., Kim G. (2003). *Query type classification for web document retrieval.* Proceedings of the 26th annual international ACM SIGIR conference on Research and Development in Information Retrieval, pages 64—71, 2003. ACM Press.

[7] Maarek Y., Smadja F. (1989). *Full text indexing based on lexical relations: An application: Software libraries*. In Proceedings of the 12th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 198--206, 1989.