VT at TREC-2003: The Web Track Report

Rui Yang, Li Wang[1], Weiguo Fan, Wensi Xi, Ming Luo, Ye Zhou, Edward A. Fox
Department of Computer Science
Virginia Polytechnic Institute and State University
{yrui, liwang5, wfan, xwensi, lming, yezhou, fox}@vt.edu

1. Introduction

This year, we participated in the Web Track in addition to the Robust Track. We submitted results on both topic distillation and home page/named page finding tasks. As our time and human resources were limited for taking two tasks simultaneously, in this task we only concentrate on testing our ranking function discovery technique, **ARRANGER** (**A**utomatic **R**endering of **RAN**king functions by **GE**netic p**R**ogramming) [Fan 2003a, Fan 2003b], which uses Genetic Programming (GP) to discover the "optimal" ranking functions for various information needs. From Web Track 2002, the training, testing and validation data sets are constructed in the same manner as in Robust Track. Our ARRANGER engine works on those data sets and automatically searches for the "best" ranking functions. The best runs are selected for submission according to their performance on queries in Web track 2002.

Our paper is organized as follows. Section 2 states our research objectives. Section 3 describes basic data processing steps. Section 4 summarizes the GP algorithm used in our system and detailed information about how we use GP to find ranking function. Section 5 shows the official submission results in comparison with the other TREC teams.

2. Research objectives

We have two objectives in this year's Web Track

1) We want to test the effectiveness of our ranking function discovery framework (ARRANGER) for other tasks (topic distillation, named page finding) and new collections. Previously, the framework is tested only on AP news collection.
2) We want to test whether the combing the structure information into the rank function can significantly improve the result. Unfortunately for this objective, due to the lack of time, we have not got any conclusive result yet.

3. Data processing

All our experiments were run on a dual-2.3GHz-processor Server running a Linux (Red-hat 7.3) operating system. Since our concentration in the Web Track is to test the significance of document structure as well as the GP ranking function, we made a lot of effort to parse the structure information of the documents. Those structured information then stored separately to form its own index. For instance, we stored the anchor information in one particular folder and use only the anchor text and properly split phrases of the URL as the text of this field, and index them into both forward index and inverted index format for our experimental purposes after removing stop words and stemming. We also parsed the in-link and out-link graph of the source document and hope to utilize it in the finding of the proper ranking function. No phrases were used in our experiments.

To speed up the process of the parsing, we used the standard HTML parser library from Perl. We modified the parsing codes. We also incorporated the parsing of all tags in one parsing step instead of separated parsing for each tag in the previous approach. This significantly increased the structure parsing processing speed. The parsed texts are then stored separately according the tag they belong to. We also removed the stop word and do stemming at this step. Then the output is sent to our indexer for indexing.

The structural elements we parsed are url (<a>), header (<h1> <h2> < h3>, <h4>, <h5>, <h6>, <th>) , title (<title>), meta (<meta>), anchor (the text in the <a> tag that point to the current document), strong (, <u>, <b>, <font>, , <i>), list (, <dl>, <ol>), and abstract (the first hundred non-stop-word from the body part of the document). We also include a plaintext part that is the union of the text parsed from all the tags. Then we separately store and index the parsed texts for each structural element.

[1] Li Wang is now at the University of Michigan (wang@umich.edu).

Although we spent substantial effort on categorizing all the web pages into different collections based on structural elements, in the end we only use plaintext collection to train our ARRANGER system and to prepare for the submissions because of lack of time.

4. Ranking function discovery using Genetic Programming

In Web track this year, we did not take advantage of the structural information of web pages. Instead, we construct a "surrogate" plaintext collection by merging full text content with all the anchor information for a page. Based on the plaintext collection, our ARRANGER engine, a Genetic Programming (GP) based ranking function discovery system, is used to discover the "optimal" ranking functions for the topic distillation task. For home/named page finding task, we simply plug various GP-based functions learned before as well as Okapi BM25 into our search engine and pick the best five runs for submission.

As reported in the Robust track paper, we achieved significant performance improvement by using new ranking functions discovered by our ARRANGER system. In the Web track, our main goal is to test if the same ranking function discovery framework could work well under the Web context for other information needs (named/home page finding and topic distillation needs). There are substantial differences between Robust track and Web track. Besides collection and query property differences between two tracks, the objectives of Web track are totally different from Robust track. In Robust track, the document providing the most sufficient information for a query should be ranked at the top of the returned document list. In Web track, a different strategy must be employed to find the most likely home/named page (for home/named page task) or find the key resources for a topic (for topic distillation task). However for these two tracks, the same ranking function discovery system (ARRANGER) is used and so are the training, testing and validation processes. We want to demonstrate that our ARRANGER system is effective under various contexts and could satisfy distinct information needs, provided that training data are appropriately prepared. As GP can easily over-train the data, we use three independent data sets for training, testing and validation purposes. 150 queries and relevance information are obtained from TREC 2002 topic distillation task for training, testing and validation processes. The details of our system and methodology for Genetic Programming (GP) are discussed in our Robust track paper. Interested readers can reference that paper or [Fan 2003a, Fan 2003b].

5. Results

In the end, we submit ten independent runs for this year's Web Track – five for the topic distillation task, five for the Name/Home page finding task. Our submissions do not involve any human intervention, so they are all automatic runs. Tables 1, 2 give the detailed description of our submissions. Tables 3, 4 summarize the final evaluation results from TREC for all 5 runs.

Run Number	Description
VTnhpgp33	This run is for the name / home page finding task, using the GP discovered function.
VTnhpgp42	This run is for the name / home page finding task, using the GP discovered function.
VTnhpgp55	This run is for the name / home page finding task, using the GP discovered function.
VTnhpgpd4	This run is for the name / home page finding task, using the GP discovered function.
VTnhpok1	This run is for the name / home page finding task, using the OKAPI function

Table 1 - Description of our five official submissions for named/home page finding task

Run Number	Description
VTtdgp33	This run is for the topic distillation task, using the GP discovered function.
VTtdgp41	This run is for the topic distillation task, using the GP discovered function.
VTtdgp5055	This run is for the topic distillation task, using the GP discovered function.
VTtdgp52	This run is for the topic distillation task, using the GP discovered function.
VTtdok4	This run is for the topic distillation task, using the OKAPI function

Table 2 - Description of our five official submissions for topic distillation task

Run No.	P10	#Best	#>= Median	P20	#Best	#>= Median	P30	#Best	#>= Median
VTtdgp33	0.0540	6	43	0.0560	8	42	0.0460	3	40
VTtdgp41	0.0620	6	41	0.0550	7	43	0.0473	3	40
VTtdgp5055	0.0760	7	43	0.0550	5	38	0.0520	4	36
VTtdgp52	0.0660	6	42	0.0560	7	42	0.0487	3	40
VTtdok4	0.0620	7	43	0.0530	7	39	0.0447	2	37
Total		**10**			**9**			**11**	

Table 3 - Official submission results for the topic distillation task.

Run No.	#not found Named Page	MRR Named Page	#not found Home Page	MRR Home Page
VTnhpgp33	16	0.5129	59	0.2317
VTnhpgp42	17	0.5084	58	0.2391
VTnhpgp55	18	0.4971	58	0.2216
VTnhpgpd4	18	0.4919	66	0.1660
VTnhpok1	20	0.4929	61	0.2024

Table 4 - Official submission results for the page finding task.

As can be seen from Table 3, we did relatively well in the topic distillation task. Almost 90% of our results are equal or above the median performance. This indicates the relative advantage of ranking function optimization using GP.

Since we did not do any optimization for the named/home page finding task, our results in this task is a little bit disappointing. One of the main reasons is that we use the same ranking strategy for both home page and named page finding tasks. This proves to be wrong if we look at the performance results from Table 4. We did well for named page finding task, but poorly on home page finding task. This indicates that home paging requires some additional evidence such as URL, Link information for effective ranking. Our future strategy is to design a query classification scheme to automatically classify queries into two different types and apply different ranking strategies based on the type of queries.

References

W. Fan, M.D. Gordon, P. Pathak, "A generic ranking function discovery framework by genetic programming for information retrieval", *Information Processing and Management*, in press, 2003a.

W. Fan, M. D. Gordon, P. Pathak, "Discovery of context-specific ranking functions for effective information retrieval by Genetic Programming", *IEEE Transactions on Knowledge and Data Engineering*, in press, 2003b.