York University at TREC 2005: HARD Track

Miao Wen[1] , Xiangji Huang[2] , Aijun An[1] , YanRui Huang[1]

[1]Department of Computer Science
York University, Toronto, ON, Canada
[2]School of Information Technology
York University, Toronto, ON, Canada

Abstract

In an IR model, "user", "query" and "result" are three important components. Traditionally, a query is considered to be independent of the user. IR systems search documents without considering who issues the query and why the query is asked. However, those factors can affect the user's satisfaction about the result. The information about the user, such as the genre preference of the user, occupation of the user, location of the user, which are normally called personalized information, indicate users' preferences to the retrieval result.

We demonstrate the effectiveness of a dual indexing technique and a feedback method on the HARD 2005 data set. We also propose a non-content based method to measure user's familiarity to a query. A similarity model is built to utilize the familiarity information and to improve the overall performance.

1. Introduction

This year, we utilized YHSE2.0 (beta version), York HARD Search Engine 2.0, to query all topics on the AQUAINT corpus. Our system was okapi-based and built on a dual index technique, constructed based on our last year's work. A co-training technique was applied to the relevance feedback process as a plug-in module. The hardware specification was the same as last year's [1].

"Familiarity" is one of our major concerns in HARD 2005. Although the track does not provide the overall familiarity score for each topic, we consider it as an important type of personalized information of the query connecting the query with the user, and thus proper utilization of it should improve the retrieval performance. We collected familiarity information via clarification forms. To utilize that information, we implemented a plug-in module on top of our original system. It extracts the non-content information from the response of clarification forms and builds a model of a desired familiarity level. The model is used to adjust the original result list.

2. System Description

Fifty topics were given this year and each one contains three meaningful fields: "title", "desc" (description) and "narr" (narrative). All those three fields are used in our experiments in both baseline and final runs. Three baseline runs and five final runs were submitted and all of them were considered as automatic runs.

A total of 4 sets of clarification forms (CF) were generated automatically from the baseline results. First three were responded by assessors. The responses are used in the final run to reinforce the topic and generate the "familiarity" model. Two types of forms are created and each type contains two sets. Type I is designed to collect relevant information and familiarity information at the paragraph level. Type II is designed to collect relevant information at both paragraph and phrase levels. Top ranked paragraphs in baseline runs are selected to create CFs. The suggested phrases in type II

CFs are chosen by the entropy of the words in the selected paragraphs. Up to five phrases are chosen in each paragraph.

The work flow of our system is shown in figure 1.

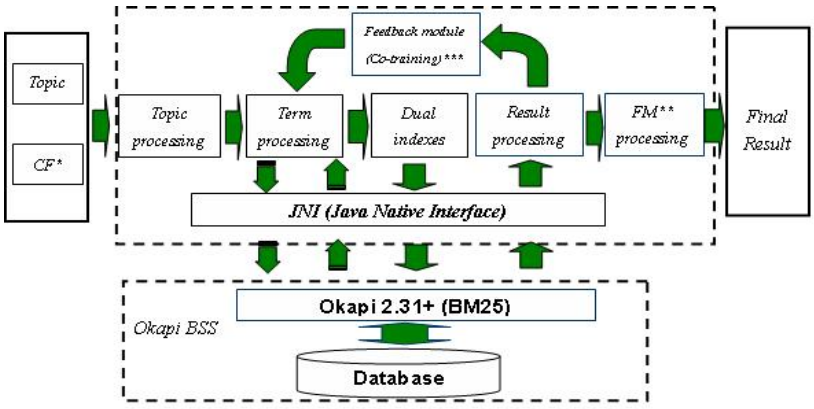

Figure 1

- *: "CF", response of clarification forms, used only in final runs.
- **: "FM", Familiarity Module, used only in final runs.
- ***: No feedback applied in baseline runs.

3. Algorithms

3.1 Dual Index

This year we do not need to pinpoint the paragraph within the retrieval documents. However, previous results show that dual index technique can also benefit the document level retrieval. The basic assumption that a document is more important if it is selected in both level indexes still holds in this situation. Thus, we applied an algorithm similar to last year's [1], which combined the results from both indexes to re-rank the document-level retrieval results. However, the document-level result may bring too much noise for the feedback process and degrade the final performance. To overcome this problem, we use only the paragraph-level result to do the relevant feedback. The final result is still the result from the document level retrieval.

3.2. Familiarity measure and adjustment

Familiarity, one type of personalized information, reflects the relationship between the user and the topic. In HARD 2005, familiarity indicates the assessors' level of experience with the topics they are judging [2]. This concept could be flexible in the real world. It can be either directly scored by the user, as HARD 2005 originally planed to do, or implied by other personalized information such as the occupation of the user or the education level of the user. For example, Hubble Telescope Achievements (topic-303), a medical doctor and a high school astronomical fan would show different preference in retrieved documents. The former user might just want to know how many new stars or any new things found by Hubble; on the other hand, the astronomical fan might want to know where the supernova is and what scientific meaning it is, etc.

The idea of familiarity is hard to represent completely from semantics aspect, especially in cross domain retrieval. In single domain retrieval, say molecular biology or astronomy, we may easily create distinguishable term lists, which respect different levels of familiarity. For instance, "with dwarf" might be in the high level list of astronomy, but "star" should be in low level list. Such type of term lists is generated from semantics aspect. However, in cross domain retrieval, it is almost impossible to pre-build such lists at run time because the query and domain are unknown. Then we try to investigate non-content based feature of document to distinguish its level of familiarity.

Readability, borrowing from the study of communication theory and linguistics, describes the ease with which a document can be read. The general approach is to analysis the non-content based features of the paragraph, such as the length of the sentence, the number of syllables of the word, etc. Many readability tests were proposed in those areas since 1920s. Those tests calculate a score of the document and in most case the higher the score, the harder the document to be understood by human. Based on our intuitive assumption that user with high familiarity with a topic would prefer to

read more complex and hard-reading retrieval result, vice versa, we convert the familiarity measurement to readability scores of documents.

We chose some of the readability scores to buildup our familiarity module. Scores and other features were used to create multidimensional model. The information obtained from CF response is used to train the model. Following are readability scores we used (detail see appendix) [3]:

- Fog Index:
- SMOG-Grading [4]
- Kincaid
- Automated Readability Index
- Coleman-Liau Formula
- Flesh reading easy formula
- Lix formula

Some other non-content based features used in our familiarity module:

- Total number of sentence
- Total number of word
- Average number of Syllables per word
- Average number of word per sentence

Due to the lack of familiarity information collected from CF response, we were not able to build a classifier model, as our original designed. Instead, we built a similarity model over the mean and standard deviation and calculated absolute z-score on each feature of test document. The accumulation of those z-score is the final score of familiarity module (FM). Two different functions are tested in our experiments to adjust original score with FM score.

$$S_{final} = S_{ori} \times (1-\lambda) + \lambda \cdot \frac{h}{S_{FM}} \qquad (1)$$

$$S_{final} = S_{ori} \times \left[(1-\lambda) + \lambda \cdot \frac{h}{S_{FM}} \right] \qquad (2)$$

Sfinal : the final score of a document.

Sori : the original score of a document from dual index system.

SFM: the familiarity module score of a document.

We set λ= 0.5 to balance the effect of both scores and h = 6, an experimental value.

4. Experiments

No feedback was applied in baseline run. Run “yorkhb1” used simple indexing on full document level only, working as a contrast run. Run 2 and run 3 used dual indexing techniques with different combination type. All final runs applied dual indexing techniques and feedback. Following table shows detail of final runs:

Run ID	Dual indexing	CF response	Feedback	CoTraining (in feedback)	FM (1)	FM (2)
york05ha1	**Y**	**Y**	**Y**	**Y**	**N**	**N**
york05ha2	**Y**	**Y**	**Y**	**Y**	**Y**	**N**
york05ha3	**Y**	**N**	**Y**	**N**	**N**	**N**
york05ha4	**Y**	**Y**	**Y**	**N**	**N**	**N**
york05ha5	**Y**	**Y**	**Y**	**Y**	**N**	**Y**

5. Conclusion and Future work

After analyzing the results of all runs, we found that the dual indexing technique and relevance feedback improve the performance by 27% (york05hb1 0.1836 vs. york05ha3 0.2344). Applying the CF response improves the performance by 21% (york05ha3 0.2344 vs. york05a4 0.2849). The co-Training technique used in the feedback process improves about 2% (york05ha4 0.2849 vs. york05a1 0.2907). However, our familiarity module does not improve the overall performance from the standard evaluation. Especially for FM2, it even degrades the performance by 9% (york05ha1 0.2907 vs. york05a5 0.2634).

From the above results, we observe that dual indexing, feedback and our usage of CF (relevant information collection) benefit the overall performance. However, by further analysis of the result for each topic, we observe that the feedback process degrades some topics' performance. The performance drop is caused by the noise in the feedback. One of our future works is to improve the feedback algorithm to identify and filter out the noise and improve the quality of feedback paragraphs.

The results also show our design of FM is not mature yet, but we are confident that our design will work better if more information about familiarity is available. Another future work is to adjust FM and find an efficient way to collect enough familiarity information from the user and build a full machine learning model of FM. We also noticed that our current design of FM is bounded by the original system performance, since it only adjusts the score of the original ranking list. We will design a new familiarity model to handle this problem.

6. Acknowledgement

This research is supported in part by research grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[1] X. Huang, Y.R. Huang, M.Wen and M.Zhong (2004). York University at TREC 2004: HARD and Genomics Tracks, the proceedings of the Thirteenth Text REtrieval Conference, 2004.
[2] HARD 2005 Guideline. http://ciir.cs.umass.edu/research/hard/guidelines.html
[3] http://www.readability.info/info.shtml
[4] Adapted from McLaughlin, G. (1969), SMOG grading: A new readability formula. Journal of Reading, 12 (8) 639-646

Appendix:

Fog Index

The Fog index has been developed by Robert Gunning. Its value is a school grade. The ``ideal'' Fog Index level is 7 or 8. A level above 12 indicates the writing sample is too hard for most people to read.

SMOG-Grading

The SMOG-Grading for English texts has been developed by McLaughlin in 1969. Its result is a school grade.

Kincaid

The Kincaid Formula has been developed for Navy training manuals, that ranged in difficulty from 5.5 to 16.3. It is probably best applied to technical documents, because it is based on adult training manuals rather than school book text.

Automated Readability Index

The Automated Readability Index is typically higher than Kincaid and Coleman-Liau, but lower than Flesch.

Coleman-Liau Formula

The Coleman-Liau Formula usually gives a lower grade than Kincaid,ARI and Flesch when applied to technical documents

Flesh reading easy formula

The Flesh reading easy formula has been developed by Flesh in 1948 and it is based on school text covering grade 3 to 12. It is wide spread, especially in the USA, because of good results and simple computation. The index is usually between 0 (hard) and 100 (easy), standard English documents averages approximately 60 to 70.

Lix formula

The Lix formula developed by Bjornsson from Sweden is very simple and employs a mapping table as well.