Examining the Feasibility of Metasearch Based on Results of Human Judgements on Thai Queries

Shisanu Tongchim, Virach Sornlertlamvanich and Hitoshi Isahara
Thai Computational Linguistics Laboratory
National Institute of Information and Communications Technology
112 Paholyothin Road, Klong Luang, Pathumthani, Thailand
{shisanu,virach}@tcllab.org, isahara@nict.go.jp

Abstract

This paper provides the results of public web search engine evaluation based on queries written in Thai. Statistical testing shows that there are some significant differences among engines. Besides comparing the effectiveness of web search engines, the returned results are compared in order to illustrate the relation and overlap among these results. The results reveal that the majority of returned results are quite unique. Since the results among engines differ greatly, this encourages the use of metasearch approaches to combine best search results from different engines. We examine metasearch models based on the Borda count voting scheme. We also propose the use of Evolutionary Programming (EP) to optimize the weight vector used by the Borda count algorithm. The results show that the use of metasearch approaches produces superior performance compared to any single search engine on Thai queries.

1. Introduction

This study addresses three aspects of the performance of public web search engines. We first evaluate the performance of web search engines using queries written in Thai. To our knowledge, most studies about the performance of public web search engines have been based on English queries. Nevertheless, the systematic evaluations of web search engines by using other languages are still important. These languages have certain characteristics which are not found in English. The evaluation results will encourage search engine companies to improve the performance of their search services on these languages. In this paper, we examine the quality of returned results of public web search engines based on a lesser-studied language, i.e. Thai. A challenge in developing information retrieval algorithms and other natural language processing techniques for Thai is that there are no explicit word boundaries. Therefore, efficient search engines should have the ability to deal with this ambiguity successfully.

After comparing the quality of search results, the second objective of this study is to examine the relation and degree of overlap among search engines. The results reveal that the results from different engines appear to have a low degree of overlap. That is, the majority of web documents are retrieved by only one engine. The results also show that the correctness of search results tends to improve in results with a high degree of overlap.

The findings from the investigation of the degree of overlap encourage the use of metasearch approaches to combine search results. Consequently, the third objective of this study is to examine some metasearch techniques based on the results of performance evaluation of search engines. The metasearch models are based on the Borda count voting scheme. Three different metasearch models are examined. The first two, called Borda-fuse and Weighted Borda-fuse, were proposed by Aslam and Montague [1]. We also propose the use of Evolutionary Programming (EP) to optimize the weight vector used by the Borda count. This algorithm is referred to as Evolutionary Borda-fuse.

The rest of this paper is organized as follows: Section 2 discusses related work which is roughly divided into three main topics, namely the evaluation of web search engines, the analysis of the degree of overlap and the metasearch approaches. Section 3 provides the description and the results of the blind evaluation of web search engines. Section 4 analyzes the relation and degree of overlap among the returned results. Section 5 evaluates the use of metasearch models to combine the results from different engines. Finally, Section 6 concludes our work.

2. Related Work

This paper is related to three research areas. This section provides a brief review of literature in each area.

2.1. Evaluation of web search engines

The comparisons among public search engines have regularly appeared in literature. The early studies conducted experiments by using the number of engines or the number of queries which sometimes can be considered to be significantly insufficient. As research in this area progresses, more systematic and well designed experiments have been carried out.

Leighton and Srivastava [8] compared five commercial search engines by using 15 queries in early 1997. They measured the precision on the first 20 returned results. They found that three search engines were superior to the other two. Gordon and Pathak [4] evaluated eight search engines by using 33 topics from faculty members. The top 20 returned results from each search engine were judged by the faculty members. The findings showed that there were statistical differences among search engines for precision, but not the retrieval effectiveness. Later, Hawking *et al.* [5] applied an extended TREC-8 Large Web task methodology [7] to compare 20 search engines. The experiment was based on 54 topics originated by anonymous searchers. The top 20 results of each engine were judged. They found that there was a significant difference in the performance of the search engines. They also compared 11 search engines using two different types of query, i.e. online service queries and topic relevance queries [6]. They found a strong correlation between the performance results on both types of query.

2.2. Measuring the degree of overlap

The research on the returned results from web search engines is not limited to only the relevance judgement. Some studies aimed to examine the properties of the ranked results, or to compare the results among search engines.

In 2005, Dogpile [3] which is a metasearch provider conducted a study about the degree of overlap in the first page results from search engines. They used their findings to support their claims about the importance of using metasearch. Some of their findings are as follows:

- By submitting 12,570 queries to major four search engines, the majority of results (84.9%) were unique to one of these engines. Only 1.1% of returned results were found by all four search engines.
- Since the majority of the first page results are unique to only one engine, using only one web search engine may miss desired results. They used the number of unique search results missed by using only one search engine to support this claim. The results showed that 68%-72% of the first page results will be missed when using only one search engine.

Spoerri [12] developed a visualization technique for showing the degree of overlap in search engine results. The article itself does not measure the degree of overlap. The proposed technique not only provides an overview of overlap in search engine results, but it can be used to perform filtering operations visually, e.g. assigning different weights to search engines in order to create a new ranking function.

2.3. Metasearch approaches

Some studies proposed metasearch techniques to combine the final results from several search engines or information retrieval algorithms. Aslam and Montague [1] categorized metasearch techniques by the data they require. Some techniques require training data, while some do not use any training data. Some techniques require relevance scores, while some use only ranks.

Our work is carried out on results from public web search engines. The relevance scores of these results are not available. Therefore, we will consider only the metasearch techniques that utilize ranks, rather than relevance scores.

Aslam and Montague [1] proposed the use of a voting system, called the *Borda count*, as a fusion algorithm for metasearch. In the Borda count, voters rank choices or candidates in order of preference. Each candidate gets a number of points, depending on the position ranked by each voter. In a simple implementation, where there are n candidates, the top ranked candidate receives n points, the second ranked candidate gets $n - 1$ points, and so on. Finally, the candidates are ranked according to the total points. In a single-winner election, the candidate with the most points wins. However, it is possible to use the Borda count in a multiple-winner election by selecting the candidates with the most points.

The fusion of ranked results from different search engines can be analogous to a multiple-winner election. Each search engine acts like a voter, while the returned results from each search engine are the ranked candidates. Thus, the Borda count can be applied to the problem of metasearch.

Aslam and Montague [1] developed two algorithms based on the Borda Count, namely *Borda-fuse* and *Weighted Borda-fuse*. Borda-fuse assigns the same weights to all engines, while Weighted Borda-fuse allows the use of different weights.

Later, Aslam and Montague [10] proposed a new algorithm based on another voting system, called the *Condorcet method*. Generally speaking, the Condorcet method

finds the winners by comparing each candidate against every other candidate. In each pairwise comparison, the winner is the candidate that is ranked in the higher positions by the majority of voters. The winners are determined from the results of every possible pairing.

3. Performance Evaluation of Search Engines

3.1. Blind evaluation

The first part of this study is to evaluate search engines based on user preference of returned documents. Seven public search engines are included in this study: SiamGURU[1], Sansarn[2], Google, Yahoo, MSN, AltaVista and AlltheWeb. SiamGURU and Sansarn are Thai-focused search sites since their services center on Thai web documents. Unlike the first two engines, the rest of engines have wider collections of web data and support other languages as well. These engines are referred to as global search engines in the rest of this paper.

Note that the number of engines used in this work is less than those used in some studies (e.g. 20 engines in [5]). The first reason is that only a small number of search engines have been found to support Thai when we conducted a survey. Moreover, several metasearch engines cannot handle Thai queries correctly, although these engines receive the results from Thai-supported search engines. The second reason is that the current search engine market seems to be shared by just few companies [9]. Many search providers now utilize services from other companies, rather than using their own engines. Moreover, some companies were acquired by others, while some companies (e.g. Northern Light) already closed their public search services. Therefore, our experiment is unable to cover all engines used in previously published articles. We also would like to include other Thai search engines in our evaluation. From our survey, however, only SiamGURU and Sansarn have actively operated. One of the biggest web portals in Thailand like Sanook[3] has just opened the search feature. However, the search function is based on the results provided by Google. Therefore, we decide not to include Sanook in our evaluation.

We developed a web-based user interface for the evaluation. This interface accepts keywords from judges. Then, it performs search operations by submitting the input keywords simultaneously to several search engines. The results from all search engines are merged into a single pool, and then presented to judges in random order. Therefore, judges do not know which each result originates.

[1] http://www.siamguru.com/
[2] http://www.sansarn.com/
[3] http://www.sanook.com/

We use 56 Thai queries in our evaluation. Each query is composed of selected keywords and a query description. We do not use natural language queries in this study since none of public search engines has been found to support natural language queries written in Thai. In our study, the length of queries ranges between 1 and 4 words.

All 56 queries are assigned to a team of 7 judges (each is responsible for 8 queries). The experiments were conducted in June 2006. The relevance judgments are binary. In particular, each result is judged whether its content is relevant to the assigned keywords and the query description or not. The inaccessible results are treated as irrelevant answers. The first 20 results from each engine which are typical results in the first two pages are used in this study.

3.2. Performance evaluation

In the literature on information retrieval, many evaluation measures are based on *Precision* and *Recall*. Precision is the ratio of relevant documents returned to the amount of all returned documents. Recall is the ratio of relevant documents retrieved to the total number of relevant documents in the collection. Typically, precision is plotted as function of recall.

In the evaluation of public web search engines, the number of relevant documents to a particular topic is usually unknown in practice. Thus, it is impractical to calculate recall. For this reason, other measures have been used to measure the performance of web search engines. Among these measures, Precision at n documents (P@n) is one of common evaluation measures used in the annual Text REtrieval Conference (TREC) web track and other literature. P@n means the proportion of relevant documents returned, calculated from the first n results returned from each engine. In our case, it is questionable to adopt this measure since the numbers of retrieved results on some queries is less than the document cutoff value (20). We therefore use Mean average precision (MAP) and Mean reciprocal rank of the first correct answer (MRR) which are standard TREC measures [2]. MAP is the average of the precision value obtained when each relevant document is retrieved. It rewards systems that rank relevant documents high. Unlike MAP, MRR is calculated only from the first relevant document retrieved. Both measures are equivalent when there is just one relevant document. Between two measures, MAP is the most meaningful measure. Thus, the comparison is mainly based on MAP.

3.3. Results of performance evaluation

The results are shown in Table 1. Google is the top performer for both measures, while Sansarn achieves the lowest performance. SiamGURU is second only to Google in

Table 1. The results for the seven search engines

	MAP	MRR
Google	0.212	0.713
SiamGURU	0.194	0.585
AlltheWeb	0.171	0.634
AltaVista	0.150	0.603
Yahoo	0.128	0.540
MSN	0.111	0.617
Sansarn	0.022	0.151

terms of MAP, but not for MRR. To provide meaningful comparisons, statistical testing is used for comparing search engines based on MAP. Sanderson and Zobel [11] showed that the t-test is highly reliable for comparing IR systems. In our case, however, the use of several t-tests to compare all pairs of search engines will result in unacceptable familywise error rate. Thus, *repeated-measures ANOVA* is used for comparing MAP. The pairwise tests $(p < .05)$ are done by using the Bonferroni correction. The results of repeated-measures ANOVA show that there are significant differences among the performance of search engines, $F(2.90, 159.41) = 14.66, p < .001$.

The results of pairwise comparisons using the Bonferroni correction show that Google statistically outperforms three search engines (MSN, Yahoo, Sansarn), while SiamGURU outperforms only Sansarn. Moreover, Sansarn has statistically lower performance than all engines. Overall, there are some significant differences among these search engines.

4. Overlap and Relation Among Results from Different Engines

The results from all engines are compared to examine the degree of overlap and their relation to the correctness. The results are presented in Table 2. The first column, Degree of overlap (n), means the results in the second and third columns are based on returned results shared by n search engines. The second column shows the percentage of returned results found in n engines. The third column is the percentage of relevant results in the returned results found in n engines.

From the table, 75.72% of returned documents are unique to one of 7 engines, while among these unique results, 19.44% are relevant. Only 0.05% of returned results are found in all engines, while 50% of these results are judged to be relevant. Overall, the majority of returned results are unique to one of 7 engines. As the degree of overlap increases, the documents have a higher chance of being relevant. The findings from this analysis encourage the use of metasearch approaches to improve the performance by combining prospective results. In the next section, some experiments on metasearch approaches are carried out.

Table 2. Degree of overlap (n), the percentage of results found in n engines and the percentage of relevant results in the returned results found in n engines

Degree of Overlap (n)	% of documents	% of relevant documents
1	75.72	19.44
2	7.97	25.16
3	10.23	27.30
4	4.39	31.79
5	1.32	40.38
6	0.33	53.85
7	0.05	50.00

5. Metasearch Approaches

The previous analysis suggests the possibility of using some metasearch algorithms to improve the performance. In this work, we explore three algorithms based on the Borda count voting scheme. Aslam and Montague [1] proposed two algorithms based on the Borda count, i.e. *Borda-fuse* and *Weighted Borda-fuse*.

Borda-fuse assigns the same weights to all engines, while Weighted Borda-fuse uses the precision based on training data as the weights of engines. That is, Borda-fuse does not require training data. It can combine the ranked results directly. In contrast, Weighted Borda-fuse needs training data to determine the precision value of each engine. The precision values calculated from training data are used as the weights for engines in the Borda count. Aslam and Montague [1] pointed out that the use of precision values as weights may not always be optimal. It would be ideal if some techniques are used to fine-tune the weight vector used by the Borda count. The results will reveal whether the use of precision values as weights is still promising or it is not optimal.

In this paper, we explore the use of Evolutionary Programming (EP), which is a class of Evolutionary Algorithms (EAs), to optimize the weight vector used by the Borda count. Classical EP (CEP) uses Gaussian mutation as the primary search operator. However, CEP may converge slowly on certain classes of problems. The Improved Fast Evolutionary Programming (IFEP) [13] was proposed

Table 3. An example of voting profile

	2 engines	1 engine	2 engines
1st	a	b	d
2nd	d	a	a
3rd	b	c	b
4th	c	d	c

by Yao *et al.* to overcome this problem. IFEP utilizes two types of search operators: Gaussian mutation and Cauchy mutation. Algorithm 1 outlines the procedure of IFEP. The use of two different mutation operators is to balance between exploration and exploitation. This work uses IFEP to fine-tune the weight vector of the Borda count. This algorithm is denoted as *Evolutionary Borda-fuse*.

5.1. Borda count

In the Borda count, voters rank choices in order of preference, rather than just electing the most favorite choice. When applying the Borda count to the metasearch problem, voters are search engines. Each engine returns the ranked results in order of relevance scores. Each result gets a number of points, depending on the position ranked by each search engine. Then, the results are ranked according to the total points.

For example, assuming that there are 5 engines. The returned results of these 5 engines are given in Table 3. Each engine returns a list of four results ranked in order of relevance scores. The top ranked result receives 4 points, the second ranked result gets 3 points, and so on. From the table, the Borda score of 'a' is calculated as: (4×2)+(3×1)+(3×2)=17. By using the same calculation, the Borda scores of 'a', 'b', 'c', 'd' are 17, 12, 6, 15 respectively. The ranked results based on the total scores are 'a', 'd', 'b' and 'c'.

5.2. Borda-fuse, Weighted Borda-fuse and Evolutionary Borda-fuse

We use the topics and the assessment of returned results in Section 3 to examine the use of metasearch approaches. For each topic, each search engine returns the first 20 ranked results or fewer candidates. In our implementation, the top ranked result receives 20 points, the second ranked candidate gets 19 points, and so on. If the number of returned results is less than 20, only ranked results are assigned scores. Once the total scores of results have been counted, each metasearch approach selects the top 20 results ranked by the total scores as the final answer.

In the Borda-fuse algorithm, all engines are assigned their weights to one. In reality, there are some differences

Algorithm 1: IMPROVED FAST EVOLUTIONARY PROGRAMMING

initialize the population of μ individuals, $(x_i, \eta_i), \forall i \in \{1, \ldots, \mu\}$
evaluate the fitness of each individual, $(x_i, \eta_i), \forall i \in \{1, \ldots, \mu\}$

while *the halting criterion is not satisfied* **do**
  for *each individual* $(x_i, \eta_i), \forall i \in \{1, \ldots, \mu\}$ **do**
    create a single offspring $(x'_i, \eta'_i)_1$ from (x_i, η_i) by Gaussian mutation

    create a single offspring $(x'_i, \eta'_i)_2$ from (x_i, η_i) by Cauchy mutation

    evaluate the fitness of $(x'_i, \eta'_i)_1$ and $(x'_i, \eta'_i)_2$

    select the best offspring (x'_i, η'_i) out of $(x'_i, \eta'_i)_1$ and $(x'_i, \eta'_i)_2$

  end
  select the μ individuals out of (x_i, η_i) and $(x'_i, \eta'_i), \forall i \in \{1, \ldots, \mu\}$

end

Table 4. Parameter setting of IFEP

Population size	20
Number of Generations	100
Tournament size	3
Number of Objective Variables (n)	7
Range of Objective Variables	$[0, 1]^n$

in the performance of search engines. Thus, the use of different weights for calculating the total scores may improve the performance. Weighted Borda-fuse uses the precision values in terms of MAP as the weights of search engines. Note that the weights based on MAP range between 0 and 1. In order to calculate the weights in Weighted Borda-fuse, a training set is required. In our experiment, the 56 topics are randomly divided into two sets: (1) 40 topics as training data, (2) 16 topics as test data. In order to achieve statistically significant results, the experiments are repeated 100 times. That is, 100 sets of training data and test data are used in the evaluation.

Evolutionary Borda-fuse uses IFEP to find the weights for the seven engines. Therefore, the number of objective variables is 7. The parameters of IFEP are shown in Table 4. The fitness calculation is based on the 40 topics of training data. The best individual after 100 generations of each run is evaluated on the 16 topics of test data.

Table 5. The results of the metasearch approaches and the top two search engines

	MAP
Google	0.208
SiamGURU	0.191
Borda-fuse	0.256
Weighted Borda-fuse	0.287
Evolutionary Borda-fuse	0.285

5.3. Experimental results

The results averaged over 100 runs are shown in Table 5. In addition to the results of three metasearch approaches, the results of the top two search engines are presented as a baseline. Clearly, the use of metasearch approaches significantly improves the performance. All metasearch approaches outperform the top two engines from Section 3. Moreover, the use of different weight assignments can further improve the performance of Borda-fuse. No performance improvement can be gained by Evolutionary Borda-fuse. This suggests that the use of the MAP values as weights is quite optimal. No performance can be improved by IFEP.

The results of repeated-measures ANOVA show that there are significant differences among the performance of metasearch models and search engines, $F(2.07, 205.19) = 224.60, p < .001$. The pairwise comparisons ($p < .05$) using the Bonferroni correction reveal that all pairwise comparisons are significantly different, except for the comparison between Weighted Borda-fuse and Evolutionary Borda-fuse. This means that all metasearch models statistically outperform the top two search engine from the previous search engine evaluation.

6. Conclusions

This research conducts an evaluation of public web search engines by using Thai queries. The results show that there are statistically differences among seven search engines. We also compare the returned results to measure the degree of overlap. The comparisons among the returned results show that the majority of results are unique to just one of the search engines. Further, the results shared by several search engines are likely to be relevant. These findings encourage the use of metasearch to improve to the performance.

This study explores three metasearch approaches based on the Borda count voting scheme. The first two approaches are existing metasearch techniques. We introduce the use of evolutionary programming to optimize the weight vector of the Borda count as the third approach. The experiments of these metasearch techniques are conducted on the results of relevance judgments from the earlier search engine evaluation. The results show that all metasearch approaches statistically outperform the top search engines. Moreover, no improvement in the performance can be achieved by using evolutionary programming. This suggests that the use of average precision as weights is quite robust.

References

[1] J. A. Aslam and M. H. Montague. Models for metasearch. In W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *SIGIR*, pages 275–284. ACM, 2001.

[2] N. Craswell and D. Hawking. Overview of the TREC-2004 Web Track. In *Proceedings of TREC-2004*, Gaithersburg, Maryland USA, November 2004.

[3] Dogpile. Different engines, Different results. http://comparesearchengines.dogpile.com/OverlapAnalysis.pdf (accessed August 1, 2006), 2005.

[4] M. Gordon and P. Pathak. Finding information on the world wide web: The retrieval effectiveness of search engines. *Information Processing and Management*, 35(2):141–180, 1999.

[5] D. Hawking, N. Craswell, P. Bailey, and K. Griffiths. Measuring search engine quality. *Information Retrieval*, 4(1):33–59, 2001.

[6] D. Hawking, N. Craswell, and K. Griffiths. Which search engine is best at finding online services? In *Poster Proceedings of the Tenth International World Wide Web Conference*, 2001.

[7] D. Hawking, E. M. Voorhees, N. Craswell, and P. Bailey. Overview of the trec-8 web track. In *TREC*, 1999.

[8] H. Leighton and J. Srivastava. First 20 precision among world web search services (search engines). *Journal of the American Society for Information Science*, 50(10):870–881, 1999.

[9] D. Lewandowski. Web searching, search engines and information retrieval. *Information Services and Use*, 25(3-4):137–147, 2005.

[10] M. H. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *CIKM*, pages 538–548. ACM, 2002.

[11] M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In R. A. Baeza-Yates, N. Ziviani, G. Marchionini, A. Moffat, and J. Tait, editors, *SIGIR*, pages 162–169. ACM, 2005.

[12] A. Spoerri. Metacrystal: visualizing the degree of overlap between different search engines. In S. I. Feldman, M. Uretsky, M. Najork, and C. E. Wills, editors, *WWW (Alternate Track Papers & Posters)*, pages 378–379. ACM, 2004.

[13] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Trans. Evolutionary Computation*, 3(2):82–102, 1999.