Simple May Be Best - A Simple and Effective Method for Federated Web Search via Search Engine Impact Factor Estimation

Shan Jin, Man Lan*
Shanghai Key Laboratory of Multidimensional Information Processing
Department of Computer Science and Technology,
East China Normal University Shanghai 200241, P. R. China
51121201056@ecnu.cn, mlan@cs.ecnu.edu.cn*

Abstract

This paper reports our participation in the three tasks, i.e., vertical selection (VS), resource selection (RS) and results merging (RM) in TREC 2014 Federated Web Search track. In consideration of the connections between vertical and search engine (i.e., a vertical could contain multiple resources), we address the two tasks in an iterative way. Existing algorithms adopted relevance measures to calculate the semantic relatedness between query and resources or returned results. However they neglected the influence of search engine in itself. In this work, we propose a Search engine Impact Factor (SEIF) estimation approach to improve the performance of vertical and resource selection. The officially released results showed that our systems ranked 1st in RS task and 2nd in VS task.

1 Introduction

With the explosive development of Internet, a huge number of rich information resources and thousands of search engines have emerged. Web search is the most popular way for people to find information on the web. Federated Web search is a kind of information retrieval, which allows the simultaneous search of multiple disparate content sources with one query (Nguyen et al., 2012) (Demeester et al., 2013).

The TREC 2014 FedWeb track provides a common platform to evaluate approaches to federated search in a realistic setting, which consists of three tasks, i.e., vertical selection (VS), resource selection (RS) and results merging (RM). We participated in these three tasks. The first two tasks in Federated Web Search, i.e., VS and RS, are to select the right vertical (specialty or topical search engines) or resource (search engine) from a large number of independent search engines given a query. Since a vertical could contain multiple resources (search engines), we consider to address the first two tasks together in an mutual way by combining the outputs of one task for another.

The traditional approaches to vertical or resource selection treat results returned from one source as a single big document and estimate the relevance score by calculating the text similarity between given query and the big document, such as CORI in (Callan et al., 1995), or by building one language model for each resource and calculating the KL-divergence (Xu and Croft, 1999). Other methods, such as ReDDE (Si and Callan, 2003), CRCS (Shokouhi, 2007), estimate the relevance of between query and each document and combine these scores as a final relevance score. More recent methods take supervised classification features into consideration. For example, (Arguello et al., 2009a) used Category-based Similarity to rank the resources and (Arguello et al., 2009b) build a probabilistic model by combining multiple types of queries with the corresponding search engine types.

Almost all these existing methods are devoted to propose various measures to estimate the relevance score between query and sources and this kind of relevance is very closely related with the semantic content of query and results. However, almost all of them ignore one important factor for resource selection, i.e., the impact factor of information source itself. We state that each source itself has a significant impact on the users' selection intention of resource selection rather than the semantic similarity between query and results alone.

In our work, we proposed a concept of Search Engine Impact Factor (SEIF), which serves as a meaningful and indicative evidence to measure the impact power of search engine. Usually, users prefer to use and believe the search engines which

have more engine marketing share or have more good searching experience. And we observed that this evidence may be of great valuable for vertical selection and resource selection. To examine this idea, we propose two ways to calculate the SEIF. One is based on a economic exploration report regarding to the distribution of market shares of search engines, which is available to the public. Another is to use the existing TREC 2013 FedWeb track corpus to estimate SEIF.

The rest of the paper is organized as follows. Section 2 presents our two methods for SEIF estimation. Section 3 describes our VS system and results. Section 4 depicts our methodology for RS task and results. Section 5 simply reports our baseline system for RM task. Conclusions are provided in Section 6.

2 Search Engine Impact Factor (SEIF) Estimation

We propose two methods for search engine impact factor (SEIF) estimation, which is used for both vertical selection and resource selection tasks. Currently, large amount of search engines are widely used in the world. They are of different languages, such as Chinese, English and Russian, and of quite different focus areas such as videos, books, news, shopping, micro-blogs, music, network, jobs, etc. Since Search Engine Impact Factor to a certain degree is able to reflect the users' selection preference and the amount of information within the search engine, it is natural to take this impact factor as an important feature of FedWeb. Obviously, this SEIF estimation is independent of the user queries or the results returned from resources and verticals.

2.1 SEIF Estimation Using Market Shares

The first simple and direct way to estimate SEIF is based on the search engines' market shares. Since the market distribution reflect the users' selection preference for search engine, it is quite natural to take this market share value as a reference of impact factor. We refer to the data source in *comScore* marketing search report, which is a global leader in measuring the digital world[1]. Figure 1 shows the distributions of top search engines in market value.

From Figure 1 we find that only a few of search engines cover more than 90% market shares.

[1]http://www.comscore.com/

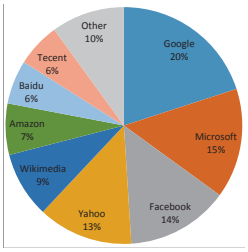

Figure 1: Market shares of top search engines from *comScore* report.

Meanwhile, many other search engines are missing in this market share list. To make a reasonable estimation for search engines with quite low market shares and new search engines not in this list, we adopt a discounting method to re-assign this distribution. The discounting method is widely used for probability estimation in many tasks in NLP, such as Language Modeling, Part-of-Speech task etc. The discounting formula is:

$$IF^*(x) = IF(x) - c \tag{1}$$

where we set $c = 0.2$. The remaining search engines not in this list would evenly share the missing probability mass.

2.2 SEIF Estimation Using TREC 2013 FebWeb Corpus

The second method is to use the existing TREC 2013 FedWeb track corpus to make estimation. In TREC 2013 FebWeb track corpus, for each *query-SE* pair, the gold truth file provides the human judgments of relevance score. Then we aggregate these scores grouped by search engine and perform the L_1 normalization for each query. After that, this normalized score is used as the SEIF value for each search engine.

Unlike the previous SEIF estimation based on market shares, the TREC corpus contains more than 100 search engines. Therefore, even for the search engines with quite low distribution, the second method still makes a more reasonable estimation than the rough discounting distribution estimation using market shares.

3 Vertical Selection

Vertical Selection (VS) is a new task in TREC 2014 FedWeb track, which is to predict the quality of different verticals for a particular query. To address this task, we present three methods to select appropriate verticals for each given query. The

first method is to simply match the keywords in queries and vertical labels. The second is to build a supervised machine learning model on labeled training data and to classify unknown input query. Unlike these two methods, the third method is to use the results of RS (Resource selection) task which takes the SEIF into consideration. To evaluate the system performance, the widely-used F_1 score is adopted in this task.

3.1 Machine Learning-based Classification

Unlike the first method which used direct keyword matching, (Shen et al., 2006) presented a machine learning method to perform query classification. Following their work, we used the KDD 2005 data set[2] which contains 911 query samples with manual annotation as training data.

Since the query is generally short and it is prone to produce search ambiguity, we performed query expansion by using Google search engine. For each query, we collected the titles and snippet descriptions from top 10 returned results from Google as expansion terms. By doing so, the averaged number of words of each query increased from 3.8 words/query to 222.4 words/query. It is obviously that this query expansion operation dramatically enriches the content of query.

After query expansion, we used Natural Language Toolkit (NLTK)[3] to remove stop words and to perform stemming. We also performed feature selection to select a subset of relevant features for model construction using χ^2 statistic (Tzeras and Hartmann, 1993; Schütze et al., 1995). Finally, we adopted the linear SVM algorithm from liblinear[4] to train the classification model, which is used for prediction.

3.2 SEIF-based Resource Selection

Different from the previous two methods which take the query into consideration, the third method is to use the results of resource selection based on SEIF. Since the estimation of SEIF is independent of query, the third method is considered to be independent of the semantic of queries and results.

According to the SEIF value, we selected out the top 20 search engines and their corresponding vertical labels. Then we counted the occurrence of verticals and returned the top 2 vertical labels with maximum frequency to each query. Obviously, this is a query-independent method, which only considers the significance of search engines rather than the semantic relationship between queries and verticals. Moveover, for each given query, this method may assign the same vertical labels, i.e., the first two verticals with maximum frequency is of the same label.

3.3 Postprocessing

Since we are allowed to submit more than three systems, we also performed two postprocessing operations in some system configurations. The first operation is to recognize if the queries involve location information. For each query, we collected the returned results from Google and examined if any one word in the list $\{country, city, street, park\}$ exists in the results. If *yes*, this query is assigned a *Travel* vertical label.

Specifically, for the *Q&A* vertical, we cannot collect its synonym set from WordNet. The second operation is to manually collect a *Q&A* keyword list, i.e., $\{what, when, where, who, why, how\}$. Similarly, if a query contains any one word in this list, it is assigned a *Q&A* vertical label.

3.4 Experiments and Results

Based on the above mentioned three methods and postprocessing operations, we submitted the following five systems. The purpose of these experiments is two-fold. The first is to compare the performance of the three methods described above. The second is to examine the effects of two postprocessing operations involving manual intervene.

ekwma: This system is to use the synonym-based keywords matching method and postprocessing operation.

svmtrain: This system is to build an supervised machine learning model on KDD data to make prediction.

esvru This system is similar to the *svmtrain* system. Besides, it also preforms the postprocessing operations after prediction.

esevs: This system is to use the outputs of subsequent resource selection task. Firstly, the top 20 resources with maximal SEIF scores are collected. Since each resource has already assigned a vertical label, then the vertical labels with maximal counts are returned.

esevsru This system is similar to the *esevs* system. Also, it includes the postprocessing operations.

[2]http://www.sigkdd.org/kdd-cup-2005-internet-user-search-query-categorization

[3]http://www.nltk.org/

[4]http://www.csie.ntu.edu.tw/ cjlin/liblinear/

System	P	R	F1
ekwma	0.054	0.120	0.069
svmtrain	0.338	0.425	0.338
esvru	0.276	0.439	0.297
esevs	**0.398**	0.586	**0.483**
esevsru	0.388	**0.598**	0.440

Table 1: Officially released results of Vertical Selection in TREC 2014 FedWeb track

Table 1 shows the performance of different systems we submitted to the vertical selection task. It is interesting to find the following observations. Firstly, among the five systems, the two systems in combination of the outputs of resource selection task performed significantly better than the other systems. Specifically, the **esevs** system performed the best among these submissions and ranked $2nd$ in officially released results. This indicates that the SEIF based resource selection makes a great contribution to the vertical selection. Secondly, the two machine learning based systems, i.e., **svmtrain** and **esvru**, performed worse than the above the systems but outperformed the simple keywords matching method. Although the supervised machine learning method is widely used in NLP, in this task the data coverage may not be quite enough to build a reliable model. Thirdly, the baseline system **ekwma** performed the worst among these systems. Although this synonym-based keywords matching method is simple and direct, its performance is surprisingly quite low. We analyzed the synonym sets of queries and verticals and found that their common words are quite few. This may be the possible reason for this poor performance. Finally, we find that the post-processing operations impaired the system performance. A further analysis is needed in future work.

4 Resource Selection

The Resource Selection (RS) task is to predict the relevant relationships between queries and resources (search engines). To address it, we present four methods to rank the relevant resources for one given query (the most appropriate resources are ranked highest). The first is to use the surface similarity measurement between query and resources in bag-of-word representation. The second is to build a regression model in consideration of deeper semantic similarity between query and resource, which is expected to outperform the first method. The third method is to rank the resources based on SEIF estimation. The forth is to combine the outputs of the vertical selection task. The official evaluation measure for this task is *nDCG* (i.e., the normalized discounted cumulative gain), a variant introduced by Christopher Burges in (Burges et al., 2005).

4.1 Surface Text Similarity

The first method is to use the surface words to calculate the text similarity between query and resource. To do it, we first performed query expansion with the aid of Google as before. Then we extracted the content with *title* and *description* tags from *snippet* of resource provided by FebWeb track. Based on the bag-of-word representation and $tfidf$ weighting scheme, we calculated *cosine* similarity between expanded queries and the contents of resources. For each query, the resources (search engines) with higher similarity score would be returned. Specifically, the $tfidf$ is calculated on the TREC 2014 FebWeb corpus.

4.2 Semantic Similarity

Unlike the first method only considering surface words rather than their actual meaning, the second method is to adopt semantic similarity presented by (Zhao et al., 2013) to capture the semantic representations of sentences. In this method, the weighted textual matrix factorization (WTFM) (Guo and Diab, 2012) model is adopted to represent semantics of sentences due to its good quality of modeling short texts. Then we used *cosine*, *Manhattan*, *Euclidean* and *Pearson* measures to calculate semantic similarity between expanded queries and snippets, resulting in four features. Finally, a gradient boosting regression model trained on TREC 2013 FebWeb data is used to rank search engines.

4.3 SEIF-based Ranking

The second method is to adopt SEIF for resource ranking. In our preliminary experiments, the SEIF estimated by using market shares performs worse than that by using TREC 2013 corpus. In this work we only estimate SEIF by using TREC 2013 data. By doing so, each search engine has a SEIF score, which is independent with queries or independent with the semantic similarity between query and results. For each given query, we use this SEIFscore to rank search engines.

4.4 Outputs of VS System

Generally a query may be relevant to multiple search engines with the same vertical label. For example, given *Vera Pavlova* $\rightarrow$ {*General*, *Encyclopedia*}, query *Vera Pavlova* is assumed to be related to all resources with *General* or *Encyclopedia* vertical label. Therefore, we consider to use the outputs of vertical selection task to perform resource selection. To do so, for each query, we collected the top verticals returned from online test in VS task and then the search engines which are assigned to the best vertical are returned.

4.5 Experiments and Results

4.5.1 Date set and Preprocessing

We adopt TREC 2013 FebWeb corpus to estimate SEIF, which consists of 157 web search engines. The format of TREC data is XML and we extract the texts with $< title >$ and $< description >$ tags. That is, we use the data set only containing snippet rather than documents. Then for each source, we combine all results into a big document. After that, tokenization and lemmatization are performed and stop words are removed for each document.

4.5.2 Experimental Results

In resource selection task, we submitted the following six systems for the purpose of comparing the performance of above four methods. Furthermore, we also combine the results of these four methods in order to examine if the combination improves performance.

etfidf: This simple baseline is to use *cosine* similarity between query and resources in *tfidf* scheme.

esmimax: This system is to use semantic similarity score to rank search engines for each query.

eseif: This system is to use *SEIF* score estimated on TREC 2013 corpus.

ecomsv: This system combines the outputs of *SEIF* and outputs of VS system.

ecomsvt: This system is to combine the outputs of *SEIF*, VS and the first *tfidf* system.

ecomsvz: This system is to combine all outputs of four methods, i.e., the outputs of *SEIF*, VS, *tfidf* and semantic similarity system.

Table 2 shows the official released performance of six systems we submitted to the resource selection task. From this table we find the following observations. First, the **ecomsvz** system, which combines all above features, achieved the best performance among all these systems. Besides, the **ecomsvz** ranked 1st in officially released ranking. Second, **eseif** system, i.e., using SEIF only, significantly outperformed the first two systems **etfidf** and **esmimax** using semantic similarity feature. This indicates that the SEIF feature is quite effective and using SEIF only makes more contribution than using semantic feature alone. But it still performed worse than the last three systems with combination configuration. Third, the last three systems, i.e., **ecomsv**,**ecomsvt** and **ecomsvz**, significantly outperformed other three systems which only considered one single feature. It shows that the combination of all these features makes significant contributions to performance improvement in resource selection task. Fourth, in comparison with **etfidf** considering only *cosine* similarity using *tfidf*, the **esmimax** system in consideration of semantic similarity achieved a better result. This shows that semantic analysis on texts does outperform simple surface word similarity.

System	nDCG @20	nDCG @10	nP@1	nP@5
etfidf	0.157	0.113	0.093	0.113
esmimax	0.299	0.261	0.222	0.265
eseif	0.651	0.623	0.306	0.546
ecomsv	0.700	0.601	0.525	0.579
ecomsvt	0.626	0.506	0.273	0.491
ecomsvz	**0.712**	**0.624**	**0.535**	**0.604**

Table 2: Officially Released Results of Resource Selection in FedWeb 2014

From these observations we conclude that SEIF is surprisingly effective. It is independent of query but it makes more contributions than other similarity features , i.e., surface similarity and semantic similarity. This is surprisingly good. A reasonable explanation for this is the resources (search engines) themselves have adopted more deeper and sophisticated explorations on search strategy before they returned the results. Using SEIF is standing on the shoulders of giants. On the other hand, the drawback of SEIF is it did not take the query into consideration and returned the same results for every query. This is not reasonable. Therefore, the combination of all above features, i.e., similarity features, SEIF, outputs of VS, which both benefits from SEIF and takes other effective features into consideration, performed the best.

5 Results Merging

The resource merging (RM) task aims to merge the snippet results returned from previously selected resources into a ranked list. In this task, we simply return the output of resource selection baseline provided by organizer. Table 2 lists the results on TREC 2014.

System	nDCG@20	nDCG@100
basedef	0.289	0.300

Table 3: Officially Released Results of Results Merging in FedWeb 2014

6 Conclusions

We employed several methods for vertical selection and resource selection tasks. The results showed that our proposed SEIF significantly improved the performance of both vertical selection and resource selection. In addition, the combination of multiple features can make up for each other and further improve performance. Our final results ranked 1st in RS task and 2nd in VS task.

Acknowledgements

This research is supported by grants from National Natural Science Foundation of China (No.60903093), Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of Things (ZF1213), and the Science and Technology Commission of Shanghai Municipality under research grant No.14DZ2260800.

References

Jaime Arguello, Jamie Callan, and Fernando Diaz. 2009a. Classification-based resource selection. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1277–1286. ACM.

Jaime Arguello, Fernando Diaz, Jamie Callan, and Jean-Francois Crespo. 2009b. Sources of evidence for vertical selection. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322. ACM.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 89–96, New York, NY, USA. ACM.

James P Callan, Zhihong Lu, and W Bruce Croft. 1995. Searching distributed collections with inference networks. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–28. ACM.

Thomas Demeester, Dolf Trieschnigg, Dong Nguyen, and Djoerd Hiemstra. 2013. Overview of the trec 2013 federated web search track. TREC.

Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 864–872. Association for Computational Linguistics.

Dong Nguyen, Thomas Demeester, Dolf Trieschnigg, and Djoerd Hiemstra. 2012. Federated search in the wild: the combined power of over a hundred search engines. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1874–1878. ACM.

Hinrich Schütze, David A Hull, and Jan O Pedersen. 1995. A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 229–237. ACM.

Dou Shen, Rong Pan, Jian-Tao Sun, Jeffrey Junfeng Pan, Kangheng Wu, Jie Yin, and Qiang Yang. 2006. Query enrichment for web-query classification. *ACM Transactions on Information Systems (TOIS)*, 24(3):320–352.

Milad Shokouhi. 2007. Central-rank-based collection selection in uncooperative distributed information retrieval. In *Advances in Information Retrieval*, pages 160–172. Springer.

Luo Si and Jamie Callan. 2003. Relevant document distribution estimation method for resource selection. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 298–305. ACM.

Kostas Tzeras and Stephan Hartmann. 1993. Automatic indexing based on bayesian inference networks. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 22–35. ACM.

Jinxi Xu and W Bruce Croft. 1999. Cluster-based language models for distributed retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 254–261. ACM.

Jiang Zhao, Man Lan, and Zheng-yu Niu. 2013. Ecnucs: Recognizing cross-lingual textual entailment using multiple text similarity and text difference measures. *Atlanta, Georgia, USA*, page 118.