Framework for Analysis and Improvement of Data-fusion Algorithms

Assistant professor Mohammad Othman Nassar

Arab Academy for Banking and Financial Sciences

Faculty of information system and technology

Website: aabfs.org

Jordan / Amman

Telephone: 0096278878059

mnassar@aabfs.org

Professor Ghassan Kanaan

Arab Academy for Banking and Financial Sciences

Faculty of information system and technology

Website: aabfs.org

Jordan / Amman

ghkanaan@aabfs.org

Assistant professor Hussain A.H Awad

Arab Academy for Banking and Financial Sciences

Faculty of information system and technology

Website: aabfs.org

Jordan / Amman

Telephone: 00962795134334

hawad@aabfs.org

Framework for Analysis and Improvement of Data-fusion Algorithms

***Abstract*—the data-fusion techniques have been investigated by many researchers and have been used in implementing several information retrieval systems. Introducing a new or improved data-fusion algorithm is an active research area for the researchers' community. We propose a framework for analyses and improvement of Data-fusion algorithms; this framework is going to be: First; a supportive tool for researchers when they are going to design a new Data-fusion algorithm by providing them with an extensive analysis and refinement for their new fusion algorithms, second; it can help researchers in understanding, analyzing, and improving existing Data-fusion algorithms.**

Keywords-Data-Fusion Algorithms; Information systems; information retrieval; Metasearch engines; performance analysis and improvement.

I. INTRODUCTION

Data-fusion is a problem-solving technique based on the idea of integrating many answers to a question into a single; best answer, while the data-fusion algorithm is the algorithm that achieves this integration. The task of fusing result sets were produced by using a number of information retrieval (IR) models to query the same document collection known as data-fusion [13]. whereas collection fusion also known as "distributed information retrieval (DIR)"; or "federated search" [14] is different from data-fusion since IR models query disjoint collections with a little or no overlap between them.

Metasearch engines are considered as an application of fusion to document retrieval; where a query is sent to a number of traditional search engines, each search engine returns a ranked list, Metasearch engine fuse them to produce a single ranked list that is hopefully better than any individual returned ranked list.

II. PROBLEM DESCRIPTION

Introducing a new or improved data-fusion algorithm is an active research area; many studies have been introduced to the literature such as [1, 2, 3, 4, 5, 6, 7, 10, 16], Despite the numerous data-fusion techniques available in the literature, most of the proposed data-fusion algorithms are competitive in performance, and there is no all-time winner [8, 10]. In [17] Nassar and Canaan studied the factors affecting the performance of Data-fusion algorithms; as a result for their study they introduced those factors in a complete and organized model; also they delivered recommendations which are related to how and when to deal with the factors that affect the performance. This paper will use Nassar and Canaan model, and the related available literature about data-fusion algorithms as a part of a new framework that can help in understanding, analyzing, and improving new and existing Data-fusion algorithms.

III. BACKGROUND RESEARCH

In [17] Nassar and Canaan introduced the factors affecting the performance of Data-fusion algorithms. They classify those factors in three types: first; factors related to the design of Data-fusion algorithms, second; factors related to the properties of individual systems, third; factors related to the features used as an input to the Data-fusion algorithm.

Figure 1 introduces the factors affecting the performance of Data-fusion algorithms as proposed by [17]. We will briefly discuss the factors mentioned in figure 1.

We will start by the factors related to the features used as an input to Data-fusion algorithm, usually Data-fusion algorithms take *N* input lists from *N* different retrieval systems to fuse them, as an output it computes a single ranked list, which is hopefully an improvement over any input list as measured by standard IR performance metrics. To compute the single ranked list data-fusion algorithm needs to use rank or score or both of them from input lists. According to the discussions in [17] we can say that; usually using score is better than using rank except in certain cases, using rank and score together is the best scenario.

The cases in which rank is better than score are:

- When multiple systems have incompatible scores, a combination method based on rank is the proper method for combination [17].
- When the runs in the combination have 'different' rank-similarity curves [9].

The second type of factors related to the design of data-fusion algorithm deals with the existence or the absence of the "three effects" in the design of any Data-fusion algorithm, the three effects are; skimming effect, chorus effect, and dark horse effect. Vogt and Cottrell [7] described those effects as:

- Chorus effect; this effect suggests that for a particular document if it is retrieved by two systems it will be "better" than another document retrieved by only one system, and if three systems retrieved a particular document so it will be "better" than another document retrieved by one or two systems, and so on. "Better" means the document has higher probability to be relevant. So any data-fusion algorithm takes this effect into account will be more efficient. This effect considered as being a very significant effect [9].
- Skimming effect; relevant documents are most likely to occur on the top of the retrieved list for each individual retrieval system, so any fusion algorithm that chooses the

top ranked documents from each individual retrieval system is expected to be more efficient.

- Dark horse effect; usually different retrieval systems retrieves different number of relevant documents. This effect assumes that the good fusion algorithm should treat the systems which retrieve larger number of relevant documents differently than other systems which don't retrieve large number of relevant documents. This means "listening" to one system more than the others based on the number of relevant documents retrieved by each individual system.

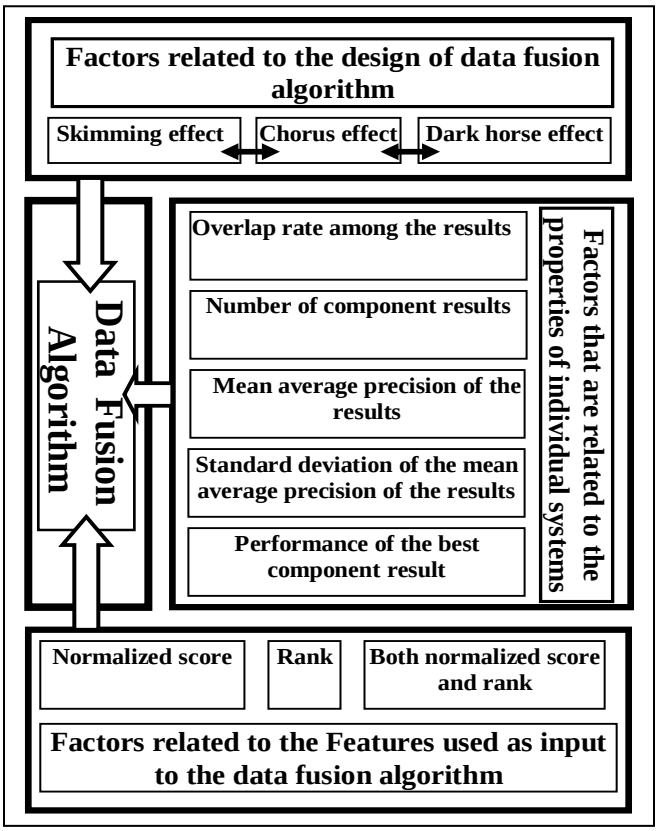

Figure 1. The factors affecting the performance of Data-fusion algorithms.

Two important phenomena concerning the three effects should be noted; those phenomena were discussed originally by [11], first; there is an apparent contradiction between chorus effects, which is trying to take as many of the input result sets as possible into account during fusion, and between the Dark Horse Effect which try to favor some input results over the others. The good Data-fusion algorithm should deal with this contradiction, second; if the combination model designed only to leverage documents in the intersection of results lists, then the chorus effect will cuts into the possible gain from skimming effect; and thus the relation between skimming and chorus effects should be controlled to achieve better performance. Nassar and Kanaan concluded that the good fusion algorithm is the one which have in its design, the ability to deal with all of the three effects, and also have in its design the ability to predict when these effects will occur and take advantage of them.

The third type of factors is related to the properties of individual systems, those properties are shown in figure 1; those properties will not be discussed since we are not going to use them in our framework because any Data-fusion algorithm can fuse any type of list from individual system no matter what their properties were, so ignoring them in our framework will not affect its ability to analyze and improve Data-fusion algorithms.

IV. THE PROPOSED FRAMEWORK

This section will introduce our proposed framework for analysis and improvement of data-fusion algorithms; this framework intended to be used as a supportive tool that can guide researchers in analyzing and improving existing and new Data-fusion algorithms.

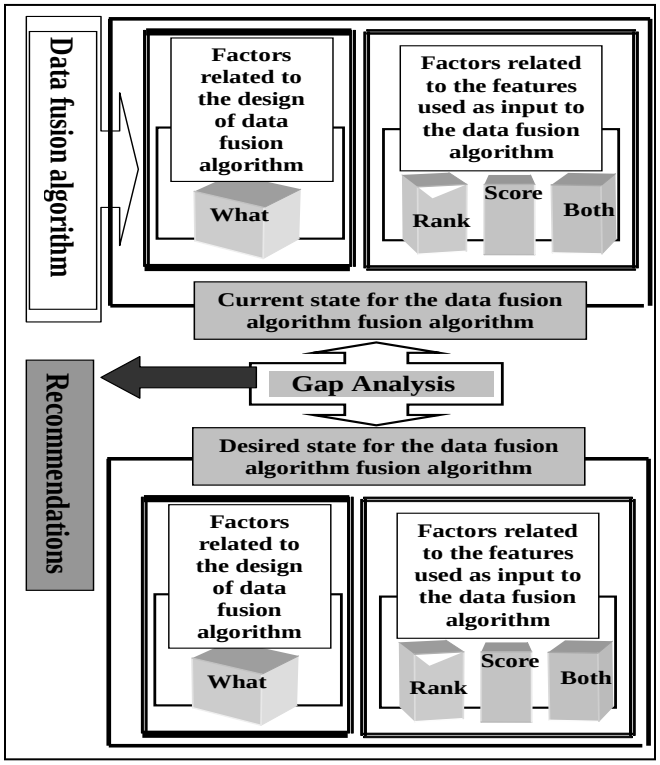

Figure 2. framework for analysis and improvement of Data-fusion algorithms.

Figure 2 present the proposed framework. To understand Figure 2 we will discuss its parts, those parts are:

1- Input: in this part we choose any existing Data-fusion algorithm, this algorithm will be analyzed and improved using the framework.

2- current state for the Data-fusion algorithm: in this part we analyze the current state for the chosen Data-fusion algorithm, this analysis will be based on:

a) Availability of the "three effects"; the Skimming effect, the Chorus effect, and the Dark horse effect. In this part we are trying to answer the following question; what effects are included in the Data-fusion algorithm design? So we are going to use "what" to refer to this point as shown in figure 2.

b) The type of features used as an input to the Data-fusion algorithm, this discussion answers the following question; which kind of features can be used as an input to the Data-fusion algorithm. The features which can be used as an input to the Data-fusion algorithm can be:

I. Score.
II. Rank.
III. Score and rank.

3- Desired state for the Data-fusion algorithm: in this part we will use golden, best known features to be implemented in Data-fusion algorithms, if these features are implemented in a given Data-fusion algorithm, its performance well be improved. Those features are primarily based Nassar and Kanaan recommendations [17], and secondarily on the available data-fusion algorithms performance issues in the literature.

4- Gap analysis: Gap analysis in general consists of defining the present state, the desired or `target' state and hence the gap between them, in other words the gap analysis compares what is currently there to what is required. In this part we have to do the gap analysis of the current state for the Data-fusion algorithm against the desired state for the Data-fusion algorithm. The goal of this is to find the performance gaps in a given Data-fusion algorithm.

5- Recommendations: once we define the performance gaps, we will provide recommendations about how to fix them, fixing the gapes will bring the current state for the Data-fusion algorithm up to the desired state.

V. USING THE FRAMEWORK TO ANALYSE AN EXISTING DATA-FUSION ALGORITHMS

To understand and prove the validity of our framework we choose an example from existing Data-fusion algorithms to analyze using our framework. This analysis will uncover the potential weaknesses in this algorithm. We will show how when other researchers handled these weaknesses well, they were able to improve the performance of the algorithm.

We chose CombSUM and CombMNZ Data-fusion algorithms; we are going to analyze these algorithms using the framework.

Fox and Shaw [2] presented the CombMNZ and CombSUM fusion algorithms. to understand these algorithms, let N be the number of result sets to be fused (number of input ranked lists), D^c is the normalized score of document d in result set c, and $|D^c > 0|$ is the number of non-zero normalized scores that were given to d by any result set (the number of ranked lists that return document d), $CombMNZ_d$ uses the following Equation to give the final score for each unique document:

$$CombMNZ_d = \sum_{c}^{N} D^c \times \left| D^c > 0 \right|$$

Note that D^c is calculated by equation two, since the normalization of scores is considered as very important and inevitable step to change the original scores in each individual result set into a common range. This method normalizes the scores to a range between zero and one. The normalized score for a document D^c is calculated by the following Equation:

$$D^c = \frac{S^d - D^c{}_{\min}}{D^c{}_{\max} - D^c{}_{\min}}$$

Where S^d is the score of document d in the rank list c before normalization. $D^c{}_{min}$ and $D^c{}_{max}$ are the minimum and maximum document scores available in the ranked list. $CombSUM_d$ data-fusion algorithm; uses the following Equation:

$$CombSUM_d = \sum_{c}^{N} D^c$$

The framework can analyze the chosen algorithm using the "current state for the Data-fusion algorithm" which allows the analysis to be conducted based on the following factors:

a) Availability of the "three effects": CombSUM and CombMNZ achieve the chorus effect and the skimming effect but they can not achieve the dark horse effect.

b) The type of features used as an input to the Data-fusion algorithm: both CombSUM and CombMNZ use normalized scores as an input.

After conducting the analysis for the chosen CombSUM and CombMNZ algorithms, the framework proposes a gap analysis between the current state for the chosen Data-fusion algorithm and the desired state for this algorithm. The desired state is defined by the results from [17], when we compared the analysis results for CombSUM and

CombMNZ the results from [17] we found the following performance gaps:

1) CombSUM and CombMNZ did not use any training data, and thus it cannot achieve the dark horse effect.

2) CombSUM and CombMNZ use scores but did not use the rank with it. As we knew before; using both of them together (rank and score) is better than using one of them.

3) CombMNZ and CombSUM did not include in their design anything to control the relation between the chorus effect and the skimming effect.

If the previous gaps are eliminated; the performance for the Data-fusion algorithms will be improved; this can be proved be reviewing the Data-fusion literature, in this regard we found in Vogt et al. experiments [7, 18, 19, 6] that they linearly combined the normalized relevance scores given to each document and use training to achieve the dark horse effect, this training improved their results over the CombSUM. The methods in [7, 18, 19, 6] are known as the linear combination model. The difference between linear combination model and CombSUM algorithm is that the linear combination model has considered the dark horse effect while the CombSUM did not; so linear combination model eliminated the performance gap that our framework found for CombSUM; thus its performance improved over CombSUM.

The recommendations from our framework suggests the need to control the relation between the chorus and the skimming effect, this gap if eliminated from CombMNZ and CombSUM; their performance will improved; this can be proved again by reviewing the literature, in this regard the researchers in [16] introduced fCombMNZ fusion algorithm that can achieve better results than the CombMNZ algorithm in most situations, fCombMNZ algorithm is considered as an improvement over the CombMNZ by adding a rule to CombMNZ called the fairness rule; this rule have been added to control the relation between skimming and chorus effects.

VI. CONCLUSIONS

We proposed a framework to analyze the current state for any given Data-fusion algorithm, uncover its performance gaps, and deliver recommendations for improvement. We showed in the previous discussions that our proposed framework can be used to uncover the performance gaps for any Data-fusion algorithm. We recommend using this framework as a supportive tool that can help the researchers in designing new Data-fusion algorithms, and in improving existing algorithms.

REFERENCES

[1] Milad Shokouhi, "Segmentation of search engine results for effective data-fusion," ECIR 2007, LNCS 4425, Springer Berlin / Heidelberg,2007, pp. 185–197.

[2] Edward Fox, and Joseph Shaw, "Combination of multiple searches," editor, D. K. Harman, In Proc. Second Text REtrieval Conf., National Institute of standards and technology (NIST) Special Publication 500-215, Gaithersburg, Maryland, 1993, pp. 243-252.

[3] Alexandre Klementiev, Dan Roth, and Kevin Small, "An unsupervised learning algorithm for rank aggregation," in Proc. European Conf. on Machine Learning, LNAI 4701, Springer Berlin / Heidelberg, 2007, pp. 616-623.

[4] Javed Aslam and Mark Montague, "Models for Metasearch," In Proc. ACM SIGIR 2001 Conf., ACM press, New Orleans, Louisiana, 2001, pp. 276-284.

[5] Mark Montague and Javed Aslam, "Condorcet fusion for improved retrieval," In Proc. Eleventh International Conf. on Information and Knowledge Management, ACM press, Virginia, USA, 2002, pp. 538–548.

[6] Christopher Vogt, "How much more is better? Characterizing the effects of adding more IR systems to a combination," In Content-Based Multimedia Information Access, Paris, France, 2000, pp. 457–475.

[7] Christopher Vogt and Garrison Cottrell, "Fusion via a linear combination of scores," Information Retrieval, 1(3), Oct. 1999, pp. 151-173.

[8] Shengli Wu and Sally McClean, "Performance prediction of Data-fusion for information retrieval," Information Processing and Management, Vol. 42, Issue 4, Elsevier, 2006, pp. 899-915.

[9] Joon Ho Lee, "Analyses of multiple evidence combination," In Proc. of the 20th ACM SIGIR conf., editors, Nicholas J. Belkin, A. Desai Narasimhalu, and Peter Willett, , ACM press, Philadelphia, USA, 1997, pp 267–276.

[10] David Lillis, Fergus Toolan, Rem Collier, and John Dunnion, "ProbFuse: a probabilistic approach to Data-fusion," in Proc. 29th ACM SIGIR conf., ACM press, Seattle, Washington, USA, 2006, pp. 139-146.

[11] D. Harman, "Overview of the third Text REtrieval Conference (TREC-3)," In Proc. the third Text REtrieval Conf., NIST, Gaithersburg, Maryland, 1994, pp. 1–19.

[12] Ellen Voorhees and D. Harman, "Overview of the fifth Text REtrieval Conference (TREC-5)," In Proc. fifth Text Retrieval Conf., NIST, Gaithersburg, Maryland,1996, pp. 1–28.

[13] Javed Aslam and Mark Montague, "Bayes optimal Metasearch: A probabilistic model for combining the results of multiple retrieval systems," editors, N. J. Belkin, P. Ingwersen, and M.-K. Leong, Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM press,July 2000, pp. 379–381.

[14] Luo Si, Jamie Callan: "A semisupervised learning method to merge search engine results," ACM Transactions on Information Systems (TOIS), Volume 21, Issue 4, ACM press, October 2003, pp. 457-491.

[15] Shengli Wu, Fabio Crestani, and Yaxin Bi, "Evaluating score normalization methods in Data-fusion," editors, H.T. Ng et

al., AIRS 2006, LNCS 4182, Springer Berlin / Heidelberg, 2006, pp. 642–648.

[16] 1. Mohammad Othman Nassar, Ghassan Kanaan, "fCombMNZ: An Improved Data Fusion Algorithm," icime, pp.461-464, 2009 International Conference on Information Management and Engineering, published by IEEE press, 2009, ISBN: 978-0-7695-3595-1.

[17] 2. Mohammad Othman Nassar, Ghassan Kanaan, "The Factors Affecting the Performance of Data Fusion Algorithms," icime, pp.465-470, 2009 International Conference on Information Management and Engineering, published by IEEE press, 2009, ISBN: 978-0-7695-3595-1.

[18] Christopher Vogt, "Adaptive Combination of Evidence for Information Retrieval," PhD thesis, University of California, San Diego, 1999.

[19] Christopher Vogt, G. Garrison Cottrell, R. K.Belew, and B. T. Bartell, "Using relevance to train a linear mixture of experts," In Ellen Voorhees and D. Harman, editors, The Fifth Text REtrieval Conference (TREC-5), Gaithersburg, MD, USA, U.S. Government Printing Office, Washington D.C., 1997, pages 503–515.