Latent Additivity: Combining Homogeneous Evidence

Shuming Shi, Ruihua Song, Ji-Rong Wen
Microsoft Research Asia, 49 Zhichun Road, Beijing, 100080, P.R. China
{shumings, rsong, jrwen}@microsoft.com

ABSTRACT
The relevance ranking problem in information retrieval and Web search is basically the task of computing aggregated scores from potentially large amounts of evidence. This paper focuses on computing an aggregated score for a homogeneous-evidence-set (HES), an evidence collection in which all evidence items are symmetric. Since the evidence items in an HES are typically highly dependent on one another, and the numbers of evidence items may vary from document to document, many existing techniques fail to properly deal with the problem. In this paper, we propose a simple, intuitive, and efficient approach for homogeneous evidence score combination. Our proposed approach can be derived in two different ways by utilizing two separate information retrieval models: The first way is to extend the BM25 formula by making a *latent additivity* assumption. The second is to adopt the recently proposed gravitational information retrieval model. The proposed approach could be seen as a generalization of some existing score combination formulas by considering the dependency between evidence items. We have tested our approach on both Text Retrieval Conference (TREC) collections and a dataset collected by a large scale commercial Web search engine. This approach could be a practical choice for homogeneous evidence combination, and act as a replacement for some of the existing heuristic formulas.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Search Process

General Terms
Algorithms, Experimentation, Theory

Keywords
Latent additivity, Homogeneous evidence set, Evidence fusion.

1. INTRODUCTION
A basic problem in information retrieval and Web search is computing the relevance score of a document when a query is given. The relevance relationship between a document and a query is normally determined by multiple pieces of evidence, each of which is an uncertain measure of how relevant the document is to the query. The main task of a ranking function is basically to compute an aggregated relevance score by combining information provided by all evidence available. For instance, it is common for most commercial or research Web search engines to rank documents by combining information from evidence such as title, URL, anchor text, body text, PageRank, etc.

Figure 1 lists some evidence items that are routinely exploited by various research and commercial search engines. Evidence items can have different levels and types, and one high-level evidence item can comprise some other lower-level evidence items. For example, evidence "*anchor phrase 1*" and "t_1^2" are clearly evidence items of different levels and types. "*Anchor*" (i.e. the aggregated text description of all links to this page) can be seen to be comprised of some "*anchor phrase*" (i.e. one piece of text description related to *one* link to this page) evidence items.

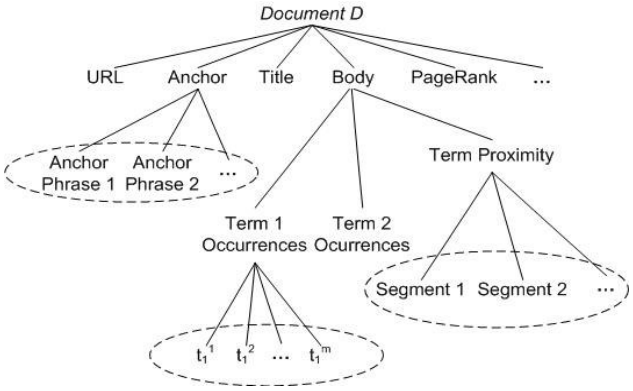

Figure 1. A sample evidence structure in document *D*, given a query $Q=\{t_1,t_2\}$. The evidence terms in the same dashed ellipse constitute a homogeneous evidence set.

For each piece of evidence, an evidence score can be defined, as a quantitative representation of the evidence, to indicate how relevant a document is to a query from the viewpoint of the evidence item itself. A score can be based on probability, certainty, or confidence. Evidence fusion is commonly (but not necessarily) the combination of evidence scores.

1.1 Homogeneous Evidence Set
Our interest in this paper centers on some specific evidence collections, as represented by dashed ellipses in Figure 1. The primary property of such kind of collections is that every two evidence items can be seen as interchangeable. To better illustrate this problem, let's give two examples,

Example 1: *Suppose we are about to retrieve books relevant to a query from a book collection. Assume each book comprises some chapters whose relevance scores (with respect to the query) have been computed. The problem is computing the overall relevance score of a book, given the score of all its chapters. Here is the chapter score distribution of three sample books,*

Book1: (<0.0, 5>, <0.6, 3>, <0.1, 2>)
Book2: (<0.0, 5>, <0.6, 3>, <0.1, 2>, <0.05, 1>)
Book3: (<0.1, 30>)

Here pair <s, n> means there are n chapters having score s. For example, <0.6, 3> represents that 3 chapters in the book get score 0.6. ▌

Example 2: *Consider the problem of computing the anchor score of a Web page given the scores of all its anchor phrases (see Figure 1). Assume we are given the following pages with their anchor phrase scores,*

d1: (<0.9, 3100>, <0.0, 1000>, <0.36, 50>)
d2: (<0.96>, 1), (0.95, 1)
d3: (<0.1, 65000>, <0.0, 46000>) ▌

In Example 1, each book chapter can be treated as an evidence item, and all chapters of a book constitute an evidence collection. Similarly, the evidence items in Example 2 are anchor phrases. In both the above examples, we do not differentiate between different evidence items, but their scores. For instance, in Example 2, we know that the two anchor phrases of page *d2* get score 0.96 and 0.95 respectively, but do not care about which one gets score 0.96.

This means the collection can be thought of as being comprised of homogeneous evidence items[1]. In this paper, we choose call such a collection as a *homogeneous-evidence-set* (HES).

Compared with other evidence collections, an HES has particular characteristics,

1. First, the size (i.e. number of evidence items contained) of an HES may vary from document to document. Take the *anchor phrase* evidence set in Example 2 for example. One document may have millions of anchor phrases that are related to a query, while another document may have no anchor text at all.
2. Second, the evidence items in a homogeneous-evidence-set are typically highly dependent. Take all chapters of a book as an example (see Example 1). If we have known that one book chapter is relevant to a given query, then the probability of another chapter of the same book being relevant to the query would surely be larger than the probability that a random-chosen chapter is relevant.
3. Last, maybe the most important one, the role of any two evidence items in an HES can be regarded as being interchangeable. That is, if the scores of two evidence items are exchanged, then the overall score of the evidence set would keep unchanged, as has been illustrated in Example 1 and 2.

The third property is the main reason why this kind of evidence set is called *homogeneous*. Please pay attention that it does *not* mean the scores of all evidence items in the same HES are the same. We are going to give a formal definition of *homogeneous* in Section 2.

1.2 Homogeneous Evidence Combination

This paper describes how to combine the scores of all homogenous evidence items into an aggregated score. We call this process as *homogeneous score combination* (HSC).

Evidence combination (or evidence fusion) plays an important role and has been extensively studied in information retrieval (and other fields of computer science). Croft [2] gives a comprehensive survey of different combination techniques. Existing efforts commonly result in one of the following fruits: 1) Simple heuristic formulas (e.g. summing up all evidence scores, taking the maximum score, Dempster's combination rule, etc); or 2) General formal frameworks with a lot of parameters for tuning (e.g. Bayesian networks, inference networks, linear or nonlinear regression using neural networks or SVM, etc). More detail analysis of existing evidence combination techniques will be given in the related work section.

Existing formal evidence fusion frameworks commonly provides a general-purpose way of combining evidence (not necessarily homogeneous evidence items). The main drawback of these frameworks is that most of them include too many parameters for users to tune, especially when there are a lot of evidence items participating in combination. Some techniques can only accept vectors of the same dimension as input, therefore are not feasible to be used for homogeneous evidence combination. That's because different homogeneous evidence sets may have different number of elements (Property 1), and the computed scores of any two evidence sets are required to be comparable.

Maybe due to the above reasons, most widely used evidence combination techniques now are still those simple and heuristic formulas. However, as we will illustrate in this paper, unreasonable results will be acquired when popular heuristic combination formulas are used to combine homogeneous evidence scores. Specially, because of the dependency of evidence items (Property 2), formulas relying on the "independent" assumption would not be feasible.

1.3 Contribution

Solving a general score combination problem in a simple and efficient way might be a challenging task. However, we argue that, because of the homogeneity of evidence items, the score of an HES could be computed efficiently by a simple formula.

In this paper, we present a simple, reasonable and effective approach to the HSC problem. In our approach, to combine homogeneous evidence scores, we first **sort them in descending order**, and then apply the following formula[2],

$$f(S) = W \cdot S = \sum_{i=1}^{m} w(i) \cdot s_i \qquad (1.1)$$

where $S = (s_1, s_2, ..., s_m)$ is the score vector after sorting, and W is a weighting vector determined by Formula 3.7 (Section 3).

The proposed approach is derived by two different methods and utilizes two different information retrieval models. The first extends the BM25 formula [12] by making a *latent additivity* assumption while the other adopts the recently proposed gravitational information retrieval model [16][17].

The proposed approach can be seen as a generalization of some existing score combination formulas by considering the dependency between evidence items. Just like many existing popular formulas, it is simple, easy-to-implement, and has a clear explanation. We provide a theoretical analysis of our approach and conduct some experiments to demonstrate its effectiveness. By this, we demonstrate that our solution outperforms standard baseline approaches, both in intuition and in practice.

The rest of this paper is organized as follows. In Section 2, we give a somewhat formal definition of the problem and discuss some baseline approaches to address it. Then, in Section 3, we propose our approach, analyze its advantages over some baseline

[1] Please note here it is only an intuitive and informal description of "homogeneous." Section 2 includes a formal definition.

[2] Please do not confuse the formula with ordinary linear combination (see Section 3 for an explanation).

solutions, and give two derivations of the approach. In Section 4, we discuss the relationship between our method and other approaches. The performance issues and possible impacts of the proposed approach are also discussed. We verify some of our discussions and test the effectiveness of our approach by conducting experiments in Section 4. Related work is discussed in Section 5. Finally, concluding remarks and future work is discussed in Section 6.

2. PROBLEM DEFINITION

In this section, we first give a formulation of the homogeneous evidence combination problem. Some basic and straightforward approaches to this problem are discussed.

2.1 Problem Formulation

We denote an evidence collection by a vector E containing all elements within as,

$$E = \{e_i\} = \{e_1, e_2, ..., e_m\} \quad (2.1)$$

where e_i ($1<=i<=m$) is the i'th evidence in E. In our information retrieval and Web search context, E can be a specific collection of evidence about the relevance relationship between a query and a document.

Example 3: *In Figure 1, the set of all occurrences of term t in document d is a evidence collection: E={t^1, t^2, …, t^m}, where t^i is the i-th occurrence of term t.*

For ease of further processing, a mathematical model is commonly used to assign a *quantitative expression* of the information contained in each piece of evidence. We call the quantitative expression of evidence e as its **evidence score**, denoted by $s(e)$. And the mapping from evidence items to their scores is called a **score assignment scheme**. We denote vector $S(E)$ as the collection of evidence scores of all the evidence items in E,

$$S(E) = (s(e_1), s(e_2), ..., s(e_m)) = (s_1, s_2, ..., s_m) \quad (2.2)$$

In information retrieval and Web search context, an evidence score can be defined to indicate how relevant a document is to a query from the viewpoint of this evidence item without considering the score of other evidence items. An evidence score can denote probability, log-odds, confidence (how confident when an evidence item says that a document is the answer to a query), and the like. We assume that for each evidence item e, its score $s(e) \geq 0$ [3]. The higher the score is, the more confidence the evidence provides in terms of the relevance between the document and the query. And a zero score ($s(e)=0$) means the score does not affect the aggregated score of the whole evidence set.

Evidence set E itself is actually a higher level and compound evidence item that should also have its own score $s(E)$, and is determined by the evidence scores of all its child evidence items. That is, there should be a function f, such that,

$$f(S(E))=s(E) \quad (2.3)$$

where f is called a **score combination function** (SCF) for evidence set E. It is clear that the score combination function of an evidence set is uniquely determined by its score assignment scheme.

[3] Actually a negative evidence score also has a distinct meaning. However, we choose NOT to consider this term in this paper.

Example 3 (cont.): *If a document can be considered as a bag of words, we can assign score $s(t^i)=1.0$ for each term t^i. That is, S(E)=(1.0, 1.0, …, 1.0). Score s(E) is the score of document d when all occurrences of t are considered. Score combination function f should satisfy f(S(E))=s(E).*

An evidence collection is called homogeneous under a score assignment scheme if the score exchange of any two evidence items does not affect the overall score of the evidence set. Formally,

Definition (**HES**) Given an evidence set E and its corresponding score vector $S(E)$. Assume $S'(E)$ is a score vector generated by exchanging two elements of $S(E)$. If equation $f(S)= f(S')$ holds for all such $S(E)$ and $S'(E)$, we say that E is a *homogeneous evidence set* (HES) under score assignment scheme $s(e)$.

For an HES, its score vector is called a homogeneous score set (HSS). Our goal in this paper is to find an appropriate function f satisfying Formula 2.3. Some basic terms and their abbreviations are listed in Table 1.

Table 1. Terms and their abbreviations

Term	Abbreviation
Homogeneous evidence set	HES
Homogeneous score set	HSS
Score combination function	SCF
Homogeneous score combination	HSC

2.2 Basic SCF Properties

The score combination function (SCF) of an HES should have the following properties.

Property-1 (Symmetry): Given a HSS $S_1=(s_{11}, …, s_{1,m})$, if S_2 is a permutation of elements in S_1, then $f(S_1)=f(S_2)$.
This property can directly be acquired from the definition of HES.

Property-2 (V-Monotonicity): Given two score sets of HES E: $S_1=(s_{11}, …, s_{1,m})$, $S_2=(s_{21}, …, s_{2,m})$. If $s_{1,i} \leq s_{2,i}$ holds for every i $(1 \leq i \leq m)$, then $f(S_1) \leq f(S_2)$.
This property says that improving the relevance score of one or some evidence items would increase the aggregated overall score of the whole evidence set. It is apparently reasonable.

Property-3 (H-Monotonicity): Given two score vectors $S_1=(s_1, …, s_m)$, $S_2=(s_1, …, s_m, s_{m+1})$, if $s_{m+1}=0$, then $f(S_1) = f(S_2)$, else $f(S_1) < f(S_2)$.
This property says that when a new evidence item is discovered and added into the evidence set, then the overall score would be unchanged (if the new evidence item has a zero score) or increased (if a positive scoring item is added).

2.3 Baseline Evidence Fusion Techniques

In this subsection, we discuss some existing score combination methods that could possibly be borrowed to address the homogeneous score combination problem. We begin with some simple heuristic formulas which have been widely used for combining document representations, retrieval algorithms, and search system output). Followed by that, we illustrate that the BM25 [12] formula can actually be adopted for score combination when all scores are of equal value. In addition, many other well-studied evidence fusion approaches are available. However, because of the special properties of HES (symmetric, variant size,

dependent), some fusion techniques may not be appropriate to be used here. These approaches and some related work will be discussed in Section 5.

2.3.1 Simple Heuristic Formulas

The simplest way to combine a list of scores together is to sum them up, as expressed below:

$$f(S) = \sum_{i=1}^{m} s_i \tag{2.5}$$

And, another straightforward way is just taking the largest value as the combination score,

$$f(S) = \max_i s_i \tag{2.6}$$

Formula 2.5 and 2.6 have their applicable conditions. When the evidence items in an evidence set are completely independent and the evidence scores denote Log-Odds, then it is reasonable to sum up the scores to find the aggregate score of an entire evidence set. However, when evidence items are highly dependent, taking the top score (as the aggregated score) should be a better choice. Following existing work [4], we call Formula 2.5 and 2.6 as CombSum and CombMax, respectively.

Example 1 (cont.): *Using CombSum (Formula 2.5), the overall scores of the three books in Example 1 (see Section 1.1) are,*

*CombSum(book1) = 0.0*5+0.6*3+0.1*2 = 2.0*
*CombSum(book2) = 0.0*5+0.6*3+0.1*2+0.05*1 = 2.05*
*CombSum(book3) = 0.1*30 = 3.0*

While if CombMax (Formula 2.6) is used, the overall score of the three books will be,

CombMax(book1) = 0.6
CombMax(book2) = 0.6
CombMax(book3) = 0.1

Table 2 lists the ordering of the three books, according to their CombSum and CombMax scores respectively. ▌

Example 2 (cont.): *Similarly, the overall anchor scores of the Web pages in Example 2 can be computed as follows,*

CombSum(d1) = 2808.00; CombMax(d1) = 0.95
CombSum(d2) = 1.91; CombMax(d2) = 0.96
CombSum(d3) = 6500.00; CombMax(d3) = 0.1

The resultant ordering of documents is shown in Table 2. ▌

Linear combination [20] is another commonly used way of fusing evidence scores,

$$f(S) = \sum_{i=1}^{m} \lambda_i \cdot s_i \tag{2.7}$$

It is clear that the above linear combination function f is not a valid HSC function because the exchange of the values of s_i and s_j may result in a different aggregated score.

2.3.2 BM25 score combination

We demonstrate in this Subsection that, if all the scores participating in combination have the same value, then the BM25 formula [12] can be used to combine these scores.

BM25 formula is an effective way of computing the score of document D respect to a query term t,

$$score(tf) = \frac{(k_1+1)*tf}{K+tf} \cdot w(t) \tag{2.8}$$

where, tf is term frequency of t in D, and $w(t)$ is a function of inverse document frequency indicating how important the term in the whole collection. $K=k_1*((1-b)+b*|D|/avdl)$, where $|D|$ is document length, $avdl$ is average document length, and k_1, b are parameters.

Formula 2.8 can be rewritten as below,

$$\begin{aligned} score(m) &= \left(\frac{(K+1)\cdot m}{K+m}\right) \cdot \left(\frac{k_1+1}{K+1} \cdot w(t)\right) \\ &= \frac{(K+1)\cdot m}{K+m} \cdot score(1) \end{aligned} \tag{2.9}$$

where m is the number of times term t appearing in D. We can treat each occurrence of term t as an evidence item. Since BM25 does not distinguish different occurrences of the same term, all evidence items have the same score. Then the meaning of Formula 2.9 is to compute the combined score of m evidence items given the score of each evidence item.

For a homogeneous score set $S=(s_1, s_2, \ldots, s_m)$, we assume all the evidence scores have the same value s. By Formula 2.9, the score of S can be computed as below,

$$f(S) = \frac{(K+1)\cdot m}{K+m} \cdot s \tag{2.10}$$

Formula 2.10 is the evidence combination function derived from BM25 when all the evidence scores are the same. We call the above formula **BM25 score combination**. Compared with CombSum and CombMax, an advantage of the above BM25 score combination formula is that it can contains a parameter K, which makes it adaptable to different dependency degrees between evidence items. So, the BM25 formula actually offers a reasonable way of combining scores for HESs. However, it is unknown what 2.10 would be when the evidence items participating in combination hold different score values.

Table 2. The Ordering of books/documents according to the scores computed by various evidence fusion approaches

Approach	Ordering (Example 1)	Ordering (Example 2)
CombSum	book3 > book2 > book1	d3 > d1 > d2
CombMax	book1 = book2 > book3	d2 > d1 > d3
HSC3D (K=4)	book2 > book1 > book3	d1 > d2 > d3

3. OUR APPROACH

In this section, we first give our approach for homogeneous score combination, and then provide two derivations of it.

In Formula 2.10, if we denote,

$$\sigma(i) = \frac{(K+1)\cdot i}{K+i} \tag{3.1}$$

Then, Formula 2.10 becomes,

$$f(S) = \sigma(m) \cdot s \tag{3.2}$$

Our approach for HSC is shown in Figure 2. It is a two-step approach. The first step sorts all scores from largest to smallest. In the second step, Formula 3.3 is used to compute the aggregated score. At the moment, we treat Formula 3.1 as the expression of $\sigma(i)$ in our algorithm. A general form of $\sigma(i)$ will be given in Section 3.2.

Given the expression of $\sigma(i)$ in Formula 3.1, we call our approach as HSC3D (Homogeneous evidence combination in three-dimension space). It will be clear in Section 3.2 why it is got this name.

Algorithm *Homogeneous score combination*

Input: *Score vector $S=(s_1,s_2,\ldots,s_m)$ of an HES E.*
Output: *Aggregated score s(S).*

Step1. *Sort all scores in descending order, resulting in a vector* $S' = (s_1', s_2', \cdots, s_m')$.

Step2. *Compute f(S) by using the following formula*,

$$f(S) = \sum_{i=1}^{m} \sigma(i) \cdot (s_i' - s_{i+1}') \qquad (3.3)$$

where $\sigma(i)$ *is expressed by Formula 3.1. And we use the convention that* $s_{m+1}' = 0$.

Figure 2. Our HSC approach

Example 1 (cont.): *According to our approach in Figure 2, we first sort the chapter scores of each book in descending order. So the score vector of book1 becomes S = (0.6, 0.6, 0.6, 0.1, 0.1, 0.0,…). Using Formula 3.3, book1's overall score can be computed as follows (assuming parameter K=4.0 in Formula 3.1),*

$$\begin{aligned} HSC3D(book1) &= \sigma(1)(0.6\text{-}0.6)+\sigma(2)(0.6\text{-}0.6)+\sigma(3)(0.6\text{-}0.1) \\ &\quad +\sigma(4)(0.1\text{-}0.1)+\sigma(5)(0.1\text{-}0.0)+\sigma(6)(0.0\text{-}0.0)+\ldots \\ &= \sigma(3)(0.6\text{-}0.1) + \sigma(5)(0.1\text{-}0.0) \\ &= 1.349 \end{aligned}$$

Similarly,

$$\begin{aligned} HSC3D(book2) &= \sigma(3)(0.6\text{-}0.1)+ \sigma(5)(0.1\text{-}0.05)+ \sigma(6)(0.05\text{-}0.0) \\ &= 1.360 \end{aligned}$$

$$HSC3D(book3) = \sigma(30)(0.1\text{-}0) = 0.441$$

Table 2 shows the order of books when their scores are computed by HSC3D. ▌

Example 2 (cont.): *Similarly, the overall anchor scores of the Web pages in Example 2 can be computed using our approach,*

$$HSC3D(d1) = 4.494$$
$$HSC3D(d2) = 1.593$$
$$HSC3D(d3) = 0.500$$ ▌

By examining Table 2, we can see that our approach gives the most desirable order of documents in the two toy problems. The main problem of CombSum is that the combination of many tiny evidence scores would be inappropriately larger than the combination of a medium number of large scores. For CombMax, it cannot differentiate two evidence sets with the same highest evidence score values. Moreover, intuitively the combination of two same scores should get a higher score (except that one evidence item depends completely on another). However it is not the case for CombMax.

The above approach can be derived by two different methods and utilizing two distinct information retrieval models: The first extends the BM25 formula by making a latent additivity assumption while the other adopts the recently proposed gravitational information retrieval model [16][17]. We will describe the two derivations in the following subsections.

3.1 Derivation 1: BM25 + Latent Additivity

A description of the latent additivity assumption follows.

Latent Additivity Assumption: Given an HES *E*'s three *m*-dim score vectors U={s, s, …, s}, S_1={s_{11}, s_{12}, …, s_{1m}}, $S_2=U+S_1$={s+s_{11}, s+s_{12}, …, s+s_{1m}}. We say that *E* satisfies *latent additivity assumption* if for all such U and S_1, the following equation holds,

$$f(S_1)+f(U)=f(S_2) \qquad (3.4)$$

The following proposition guarantees that if an HES satisfies latent additivity and BM25 can be used to combine its uniform scores, then the only reasonable way of computing its aggregated score is by using our algorithm.

Proposition 1: Assume that an HES *E* satisfies the latent additivity assumption, and BM25 score combination (Formula 2.10) can be used to combine its scores when they hold the same value, then for *E*'s score set $S=(s_1, \ldots, s_m)$ ($s_1 \le s_2 \le \ldots \le s_m$), we have,

$$f(S) = \sum_{i=1}^{m} \sigma(i) \cdot (s_i - s_{i-1}) \qquad (3.5)$$

Please refer to Appendix A for the proof of the above proposition. The meaning and proof of Proposition 1 is depicted in Figure 3. The basic idea is to divide the score set into some sub-sets, each of which contains some equal-valued scores, such that BM25 can be applied to each sub-set.

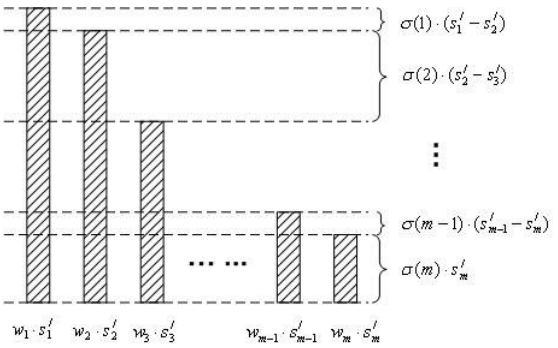

Figure 3. Proposition 1 illustration

Formula 3.5 can easily be transformed into the following equivalent one,

$$f(S) = W \cdot S = \sum_{i=1}^{m} w_i \cdot s_i \qquad (3.6)$$

where

$$w_i = \sigma(i) - \sigma(i-1) = \frac{(K+1)\cdot K}{(K+i)\cdot(K+i-1)} \qquad (3.7)$$

Figure 3 is a visual description of the transformation.

Formula 3.6 is actually an *inner product* of a score vector $S=(s_1,s_2,\ldots,s_m)$ and a weight vector $W=(w_1,w_2,\ldots,w_m)$. Please do not confuse it with ordinary linear combination. They are different in two ways: First, the evidence score in *S* must be sorted before computing the inner product. So by applying this kind of inner product, *the maximal score always get the largest weight*. Second, the length of vector *W* can potentially be infinitely large. That is, given a score vector *S* of any dimension *m*, we can construct an *m*-dim vector *W* for a linear combination with the score vector, by using the **only** parameter *K*.

3.2 Derivation 2: Gravitational IR Model

In this subsection, we try to address the HSC problem by using the GBM model [16][17]. In the model, documents and queries are modeled as physical objects with specific structures, and the relationship between a query and a document is modeled as the attractive force between them. Newton's theory of gravitation is used to compute the relevance of a document given a query.

For a homogeneous evidence set E, we model each evidence item as an ideal-cylinder-shaped object (just as in the continuous version of GBM). And, E is modeled by a list of evidence objects. As in [16][17], it is natural to define the score of evidence set E (given query Q) as the maximal gravitational attractive force between its corresponding evidence object and the query object. Apparently, the attractive force is maximized when E is in its *optimized evidence placement* state, where all evidence objects are sorted by the attractive forces between them and query Q.

To be more clear about the problem we wish to solve, note Figure 4. Figure 4(a) shows the relationship between query Q and evidence e_i. Note that the score of a piece of evidence is the relevance of the document and the query *without* considering the score of other evidence items. Therefore, it is natural in the gravitational model to model an evidence score the gravitational force between the query and the evidence *without* considering other evidence objects, as shown in Figure-4(a). So, we have,

$$s_i = \int_1^{1+d} \frac{m_Q \cdot m_i \cdot \kappa(x)}{d} dx = \frac{m_Q \cdot m_i}{d} \cdot \int_1^{1+d} \kappa(x) dx \qquad (3.8)$$

where m_i is the mass of evidence e_i, and $\kappa(x)$ denotes the gravitational force between two unit point masses of distance x ($\kappa(x) = G/x^2$ according to Newton's law).

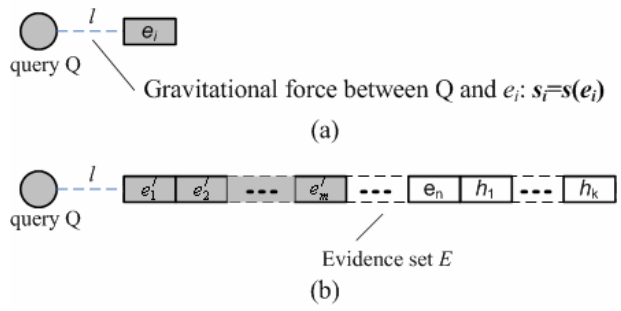

Figure 4. Using the gravitational IR model for HSC

Now consider how to compute the aggregated score of evidence set E. Figure-4(b) shows a sample evidence set and all its child evidence items. In the figure, m is the number of non-zero evidence items, and n is the total number of evidence items (zero scored or non-zero scored) being observable to users. And k is the number of hidden evidence items (i.e. evidence items that do exist but are not observable). We denote the mass of evidence object e_i by $m(e_i)$, and assume all evidence items have the same diameter[4] d. It is natural to define the score of evidence set E as the gravitational force between Q and E. When object E is in its optimized evidence placement state (as in Figure-4(b)), all its child evidence objects should be sorted according to their respective mass. The larger the mass, the larger the gravitational force between Q (since we have assumed all evidence objects have the same size), therefore the nearer the evidence to object Q. Without loss of generality, we assume that $m_1 \geq m_2 \geq \ldots \geq m_m$ (the mass of other evidence items is zero because of zero evidence score), as shown in Figure-4(b).

We can clearly see from Figure-4 that, when evidence set E is in its optimized evidence placement state, most evidence objects would be farther away from query Q, except for evidence e_1. As a result, the attractive forces suffered by Q would decrease accordingly. Now the *new* force between Q and e_i have changed to,

$$F(e_i, Q) = \frac{m_Q \cdot m_i}{d} \int_{1+(i-1)\cdot d}^{1+i\cdot d} \kappa(x) dx \qquad (3.9)$$

By combining Formula 3.8 and 3.9, we have,

$$F(e_i, Q) = u_i \cdot s_i \qquad (3.10)$$

where

$$u_i = \frac{\int_{1+(i-1)\cdot d}^{1+i\cdot d} \kappa(x) dx}{\int_1^{1+d} \kappa(x) dx} \qquad (3.11)$$

So the attractive force between query Q and evidence set E is,

$$f(E) = F(E, Q) = \sum_{i=1}^{m} F(e_i, Q) = \sum_{i=1}^{m} u_i \cdot s_i \qquad (3.12)$$

If we define

$$\sigma(i) = \sum_{j=1}^{i} u_i = \frac{\int_{1+d}^{1+i\cdot d} \kappa(x) dx}{\int_1^{1+d} \kappa(x) dx} \qquad (3.13)$$

then Formula 3.12 can be re-written as,

$$f(E) = \sum_{i=1}^{m} \sigma(i) \cdot (s_i - s_{i+1}) \qquad (3.14)$$

According to Newton's law, $\kappa(x) = G/x^2$. So, by applying it to Formula 3.13, we have,

$$\sigma(i) = \frac{(1/d + 1) \cdot i}{1/d + i} \qquad (3.15)$$

As has been illustrated in [16][17], $1/d$ in the gravitational model corresponds to K in BM25 formula. Therefore, the expressions of $\sigma(i)$ in Formula 3.1 and 3.15 are actually the same. As a result, we derived our HSC Formula 3.3 by providing Formula 3.14.

Here, we do not make the latent additivity assumption. Readers may also have realized that the role of Latent additivity assumption in the previous subsection is the same as that of the additivity of gravitational forces in this case here.

3.2.1 Going beyond inverse square

When Newton first discovered the law of universal gravitation, he did not give an explanation why the gravitational force is inversely proportional to the *square* of distance. Laplace (in his famous book, *Mecanique Celeste*) extended Newton's law by building a model and formulating a differential equation satisfied by the gravitational potential corresponding to a certain distribution of mass in space. By assuming that a gravitational potential must satisfy his equation, Laplace actually "proved" that gravitational force should be inverse-square in $\mathbf{R}^3$ (three-dimension space)[5]. From the same process, the gravitational force

[4] Please note that the diameter of an ideal-cylinder-shaped object is defined by its height in the GBM model.

[5] As Laplace's original treatise may be hard to acquire, please refer to Section 73.11 of [3] for an introduction to Laplace's work on gravitation.

in $\mathbf{R}^2$ can be proved to be inversely proportional to the distance (*without* the *square*). So the gravitational function can actually have different expressions in different dimensional spaces.

Since we do not know exactly which dimension should be taken in information retrieval and/or Web search tasks, the expression of $\kappa(x)$ is therefore not necessarily inverse-square, according to the Laplacian model. Therefore, the expression of $\sigma(i)$ in Formula 3.13 is in a more general form than 3.15.

According to the Laplacian model, the expression of $\kappa(x)$ in $\mathbf{R}^2$ is c/x (where c is a constant). By applying it into Formula 3.13, the expression of $\sigma(i)$ in $\mathbf{R}^2$ is,

$$\sigma(i) = \frac{\ln(1/d + i)}{\ln(1/d + 1)} \quad \left(or\ equivalently: \frac{\ln(K + i)}{\ln(K + 1)} \right) \qquad (3.16)$$

Since the above Formula is for a two-dimensional space, we call it as HSC2D. Accordingly, Formula 3.15 (or 3.1) is called HSC3D.

4. DISCUSSIONS

4.1 Relationships with Other Techniques

It is easy to verify that $\sigma(i)=1$ for each $i>0$ when $K=0$ in Formula 3.1 (or $d = +\infty$ in Formula 3.14). In this case, our approach (Formula 3.3) actually returns the maximal score. Therefore, CombMax (Formula 2.6) is a special case of our approach. Similarly, CombSum (Formula 2.5) is also a special case of our approach (when $K = +\infty$, or $d=0$). The BM25 score combination formula (Formula 2.10) is apparently a special case in which all evidence scores are the same. To sum up, the CombMax, CombSum, and BM25 score combination approach are all special cases in our approach.

Salton, Fox and Wu [14] developed the p-Norm model as one generalized way of processing Boolean queries. For a query $Q=\{t_1, t_2, \ldots, t_n\}$, assume the term score of document d for the query terms are $s_1, s_2, \ldots, s_n$, then the overall document score could be computed by the following OR-like formula,

$$S(d) = \left(\frac{{s_1}^p + \ldots + {s_n}^p}{n} \right)^{\frac{1}{p}} \qquad (4.1)$$

P-Norm and our approach share some similar characteristics. First, in both approaches, high score values can be seen as receiving larger weights. Second, they both are generalizations of some simple heuristic formulas. The two extremes (p=1 and p=∞) of p-Norm are average and CombMax, and the two extremes of our approach are CombSum and CombMax.

Despite the similarities, they have apparent differences. First, p-Norm is commonly applied to fixed dimensional evidence sets. For example, in computing the overall term scores of documents with respect to a four-term query, each document corresponds to a 4-dimension evidence set. However, for evidence sets of different lengths (i.e. numbers of items), their overall scores computed by p-Norm might not be comparable. For instance, in Example 2 (Section 1), p-Norm would always assign a larger score to *d2* than to *d1*, which is not reasonable. Second, our approach is linear in nature (see Formula 3.6), while p-Norm is non-linear for most p values. Third, for p-Norm, scaling evidence scores (multiplying by a factor c) does not affect the relative ordering of evidence sets. Our approach also has this property. Moreover, for our approach, score transformation (i.e. adding scores by a constant value s_0) also does not affect the relative ordering of evidence sets, as long as they have the same number of evidence items. Forth, p-Norm has two kinds of dual operations *AND*(p) and *OR*(p), while our approach has only one format for now. Finally, p-Norm is based on geometry and has an excellent geometric explanation, while our approach has a physical explanation (subsection 3.2).

4.2 Computational Complexity

Since evidence combination is often embedded in the online query processing module, its computational performance is crucial. Now let's analyze the time complexity of our approach.

Assume that an HES has M elements and K different score values. Implementing our approach in a naïve way would result in an algorithm with $O(M\log M)$ time complexity, because all scores should be sorted in the first step. When score with same values have been grouped together, the time cost can easily be reduced to $O(K\log K)$. This time complexity would be acceptable for most applications with medium-length evidences sets.

In the case of requiring linear time-complexity, an approximate algorithm is provided in Figure 5. In the algorithm, some score-range slots are utilized to record and summarize the scores in their ranges. For example, if all evidence scores are in [0, 1], we can generate 100 slots each of which is in charge of a score range of 0.01 length. The time complexity of this algorithm is $O(M+h)$, where h is the number of slots utilized.

Algorithm *Linear-time HSC*

Input*: Score vector $S=(s_1,s_2,\ldots,s_m)$ of an HES E.*
Output*: Aggregated score s(S).*

Step1*. Build h score-range slots $R=(r_1, \ldots, r_k)$, with each slot initialized to be empty.*
Step2*. For each evidence score s_i, put it into the slot whose score range contains the score.*
Step3*. By treating all scores in the same slot to be the same value, utilize Formula 3.3 to compute the aggregated score.*

Figure 5. A linear-time HSC algorithm

4.3 Impact Analysis

Homogeneous score combination is important for information retrieval and Web search, for two main reasons. On one hand, good HSC mechanism may improve search performance. Because of the challenges of computing a combined score for an HES, existing ranking algorithms ordinarily choose to bypass it and use a relatively non-optimal, alternative method. For example, due to the difficulty of computing the aggregated score of a collection of anchor phrases, most of (if not all) retrieval systems choose to merge all anchor phrases into an entire document and apply full-text ranking models or formulas (e.g. BM25 [12]). In this way, the anchor phrase structure can not be utilized.

On the other hand, HSC helps to ranking function design. If for each HES in Figure 1 we remove all the evidence items belonging to it and replace them with a piece of new aggregated evidence, the evidence structure of Figure 1 would be greatly simplified. Most importantly, it is possible that different documents would have the same number of evidence items by this way. Therefore, some machine learning techniques which can only accept fixed-dimension vectors could possibly be applied now.

There are surprising numbers of evidence collections in various fields which could be considered, at least approximately, as

homogeneous. Actually, many widely used heuristic formulas are implicitly assuming the symmetry of evidence items. A large number of problems would benefit from an efficient homogenous evidence combination approach.

Although the derivation process of our approach is somewhat not easy to understand (especially the derivation from GBM), the resultant formula is simple, efficient and intuitive. It power is showed and verified by our preliminary experiments (Section 5). We believe this approach could be a practical choice for homogeneous evidence combination, and act as a replacement for some of the existing heuristic formulas.

5. EXPERIMENTS

In this section, we verify the analysis of previous sections and test the performance of our approach through experiments.

5.1 Experimental Setup

We report results on the .GOV data set used in the TREC [18] Web Retrieval Track and a dataset acquired from MSN Search. We call the former dataset TREC and the latter CSE (Commercial Search Engine) hereafter. There are 1,053,111 web pages in TREC (not including PDF files) and about 12,500,000 web pages in the MSN dataset.

For the TREC data set, we use two query-sets: TREC2003 Mixed, and TREC2004 Mixed. They are queries used in Web track of TREC'2003 and 2004. Both of the query-sets mix three types of queries: topic distillation, homepage finding and named page finding. The CSE dataset comprises 1,000 queries. For each query, averagely 70 Web pages were manually labeled and assigned a relevance value from 1 (meaning "poor" match) to 5 (meaning "perfect" match). The reason for using five-judgment levels instead of the binary judgments widely used in information retrieval is that multiple judgments can more precisely evaluate the relevance of a Web page to a query. We shuffled the queries and used 1/10 (100 queries) for training and 9/10 (900 queries) for testing.

For the TREC dataset, we choose Mean Average Precision (MAP) as a primary measure for representing our experimental results. Similar experimental results have been observed using other measures (e.g. MRR, P@5).

Since we used five judgment levels for the CSE dataset, some common evaluation metrics (e.g. mean average precision, precision@10, etc) are not applicable any more. In order to study the performance of our approach, we adopt the nDCG [5] measure in the experiments to evaluate search results. nDCG has two kinds of parameters: discount factor b, and gains for all labeled relevance levels. In our experiments, the value of the discount factor b is fixed to be 2. And the gain value for the 5 relevance levels (from 1 to 5) are 0.01, 1, 3, 7 and 15, respectively. For completeness, we also transform the 5 judgment levels into binary judgments (with judgment level 1 and 2 treated as irrelevant, and other levels as relevant), and utilize traditional IR evaluation metrics to evaluate our results (see Table 6). For the CSE dataset, we chose nDCG@3 as a primary measure for representing experimental results. Similar experimental results have been observed at other nDCG ranks and binary judgments (e.g. MAP, MRR, P@5).

In tuning parameters to optimize an evaluation measure, we use a grid search method. On the CSE dataset, parameters are tuned on the training set and then the optimized parameters are applied to the testing set. The experimental results reported are on the test query set.

We chose two types of HESs in our experiments to test the performance of our HSC approach and compared it with other techniques. The first is the set of all anchor phrases of a document. The second is the set of a document's all body-text fragments.

5.2 Anchor Text Experiments

For a Web page, its anchor text phrase is one piece of text description related to *one* link to page. Clearly all anchor phrases of a page can be thought of as an HES. Several techniques can be used to compute an aggregated anchor score for a document. The most straightforward way is merge all anchor phrases into a full-text and use common term weighting formulas (e.g. BM25) to compute a score. We call this method as anchor full text (AFT) in displaying the experimental results. If we have computed a score for each anchor phrase, then some methods mentioned in this paper can be used to combine them into an aggregated one. We will test the performance of the following approaches: CombSum (Formula 2.5), CombMax (Formula 2.6), HSC3D (our homogeneous score combination approach with Formula 3.1 or 3.15 as the expression of $\sigma(i)$), and HSC2D (our approach with Formula 3.16 as the expression of $\sigma(i)$).

Table 3. Search performance comparison between different anchor scoring approaches (Dataset: CSE; Metric: nDCG@3).

Method	Anchor Only	Imp. over CombMax (%)	Base+ Anchor	Imp. over Base (%)
Base	-	-	0.254	--
AFT	0.248	-23.7	0.359	+41.3
CombSum	0.207	-36.3	0.254	0
CombMax	0.325	--	0.353	+39.0
HSC3D	0.358	+10.2	0.366	+44.1
HSC2D	0.346	+6.46	0.37	+45.7

Table 4. Search performance comparison between different anchor scoring approaches (Dataset: TREC; Query-Set: TREC2003 Mixed; Metric: MAP)

Method	Anchor Only	Imp. over CombMax (%)	Base+ Anchor	Imp. over Base (%)
Base	-	-	0.292	-
AFT	0.461	+12.4	0.486	+66.4
CombSum	0.318	-22.4	0.367	-25.7
CombMax	0.410	-	0.449	+53.8
HSC3D	0.476	+16.1	0.48	+64.4
HSC2D	0.471	+14.9	0.505	+72.9

On the CSE data, the second column of Table 3 shows the search performance (on the test query set) of each anchor scoring approach, by using anchor text only. In the experiments, the score for an anchor phrase is computed by the BM25 formula with parameters tuned on the training query set. We can see that HSC3D and HSC2D outperform other approaches. AFT is the normal way of utilizing anchors in information retrieval. We observed that the CombMax approach and two of our homogeneous approaches achieve much better performances than AFT on the anchor field. Another observation is that the performance of the CombSum method is far worse than that of the others. The reason may lie in the high dependency between the anchor phrases. In addition to comparing the performance of these approaches on the anchor field, we need to investigate how they perform when their scores are combined with other scores. The last two columns of Table 3 show the combination of a base score with the anchor score, calculated by each approach. We can see

from the table that only the two variations of our approach outperform the basic AFT approach (i.e. merge all anchor phrases into a whole text). This may be the reason why existing approaches compute anchor scores using the AFT method.

Table 4 shows the search performance of each anchor scoring approaches on TREC2003 query-set. It indicates that HSC2D and HSC3D outperforms CombMax significantly when only anchor is used for ranking or anchor score is linearly combined with a base score that is computed by adopting the BM25 formula on title and body text fields of a Web page. Experimental results on TREC2004 Mixed data confirmed such conclusion.

5.3 Term Proximity Experiments

With term proximity, we mean how near query terms occur in a document. When query terms appear close together within a document, the document more likely to be an answer to the query. Given a query, a document can be split into segments with each segment acting as a piece of evidence. A score can be computed for each fragment according to nearness factors and order of terms in the fragment. The set of all segments is another example of HES (Figure 1)

Table 5. Search performance comparison between different term proximity score combination approaches (Dataset: CSE; Metric: nDCG@3).

Method	Proximity Only	Imp. over CombMax (%)	Base+ Proximity	Imp. over Base (%)
Base	-	-	0.254	-
CombSum	0.201	-3.83*	0.254	0
CombMax	0.209	-	0.267	+5.12
HSC3D	0.215	+2.87*	0.281	+10.6
HSC2D	0.240	+14.8	0.281	+10.6

Note: * means that the change is NOT statistically significant, i.e. when t-test is done to compare two ranking results, p-value is larger than 0.05.

Table 6. Search performance comparison between different term proximity score combination approaches (Dataset: CSE; Metric: MAP).

Method	Proximity Only	Imp. over CombMax (%)	Base+ Proximity	Imp. over Base (%)
Base	-	-	0.156	-
CombSum	0.114	1.572	0.156	0.000
CombMax	0.112	-	0.163	4.711
HSC3D	0.121	7.471	0.170	8.794
HSC2D	0.132	17.83	0.170	8.965

Table 7. Search performance comparison between different term proximity scoring approaches (Dataset: TREC; Query-Set: TREC2004 Mixed; Metric: MAP).

Method	Proximity Only	Imp. over CombMax (%)	Base+ Proximity	Imp.over Base (%)
Base	-	-	0.286	-
CombSum	0.175	+16.7	0.286	0
CombMax	0.150	-	0.317	+10.8
HSC3D	0.233	+55.3	0.319	+11.5
HSC2D	0.236	+57.3	0.315	+10.1

It is not difficult to split a document into segments. One natural query-independent method is splitting by sentence or paragraph boundaries. A query-dependent method is by the distribution of query terms, e.g. splitting when the distance between two adjacent query term occurrences are larger than a threshold. We used the latter method when generating segments. We use a heuristic formula in our experiments to compute the term proximity score for each segment. Then, the segment scores were combined by using four fusion approaches: CombSum, CombMax, HSC3D, and HSC2D (please refer to Section 4.2 for definitions).

The results on the CSE dataset are shown in Table 5 and Table 6. We can see that HSC2D and HSC3D behave better than CombSum and CombMax. Experimental results on TREC data indicate that both HSC3D and HSC2D outperform CombMax dramatically when term proximity score is considered only (see Table 7). On the other side, the gap between CombMax and HSC3D / HSC2D is slight if they are linearly combined with the base score. We observed similar results for the TREC2003 Mixed query-set.

6. RELATED WORK

Evidence combination and fusion is a basic problem not only for information retrieval and Web search, but widely throughout the research arena. There have been many theoretical or heuristic approaches to address this problem.

Theories developed for evidence fusion include Bayesian theory, Dempster-Shafer theory [15], Stanford certainty theory [8], etc. Bayesian networks [6] make use of conditional independence assumptions to simplify joint probability computation and, to some extent, make some evidence combination problems tractable. However, when the number of parents for a given random variable (network node) is large, inference network would not be feasible to adopt. The Dempster-Shafer theory of evidence [15] provides a rule, the Dempster's combination rule, which allows for the expression of the aggregated uncertainty from component uncertainties. And Stanford certainty theory [8] provides a simple way of combining some certainty factors. The Dempster-Shafer theory and Stanford certainty theory both make the independence assumption, so are not suitable for a general homogeneous score combination problem. All these theories provide basic frameworks for evidence combination. However, as have been pointed out, none of them is applicable to be used to solve a general HEC problem.

The p-Norm model [14] provides a remarkable way of combining evidence items. We have discussed the relationship between p-Norm and our approach in Section 4.

Heuristic approaches often provide simple and easy-to-understand ways of combining evidence scores. However, they are often ad-hoc and lack theoretic foundations. Moreover, their performances are not stable enough for different datasets and applications. We have demonstrated that our approach is superior to some commonly used heuristic approaches (CombSum and CombMax). In addition to the similarity value combination approaches tested by Fox et al [4] to combine TREC runs, Wilkinson [21] has conducted some empirical studies to evaluate different ways of combining the scores obtained from document fields.

There has been extensive work on evidence fusion in a wide area of research fields. In the meta-search field, many fusion algorithms have been developed and studied to address the problem of combining results from different retrieval systems [1][20]. Structured document retrieval has seen a strong demand for evidence fusion. Various kinds of approaches have been studied and tested [2][7][9] (e.g. language models [10], inference networks [19][11], and term frequency combination[13], etc).

7. CONCLUSION AND FUTURE WORK

In this paper, we have addressed the special case of the evidence fusion problem: homogeneous evidence combination. Generally speaking, a homogeneous-evidence-set typically have three properties: variant-size, mutual dependency, and homogeneity. The first two properties make it hard to combine homogeneous evidence scores by effectively using existing approaches. Our approach makes full use of the third property and provides a simple, effective, and somewhat formal way to address this problem.

There are a surprising number of evidence collections (in a wide area of fields) that can be regarded as homogeneous. Although our approach is derived from information retrieval models and formulas, it is hoped to be a general purpose way of fusing symmetric, mutually dependent, and variant-number evidence.

If would be perfect if our approach could be derived from a probabilistic framework by using Bayesian theory or other probabilistic theories. This will be left to future work. Another problem for further study is the combination of our approach with other approaches (e.g. regression, inference networks, and the like) to solve an array of evidence fusion problems.

8. REFERENCES

[1] J. A. Aslam and M. Montague. Models for Metasearch. In Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, p.276-284, 2001.

[2] W. B. Croft. Combining approaches to information retrieval. In Advances in Information Retrieval, p.1-36., Kluwer, 2000.

[3] K. Eriksson, D. Estep, C. Johnson. Applied Mathematics Body and Soul: Vol I-III. Springer-Verlag Publishing, 2003.

[4] E. A. Fox and J. A. Shaw. Combination of multiple searches. Proceedings of the 2nd Text Retrieval Conference, 1994.

[5] K. Jarvelin, and J. Kekalainen, IR evaluation methods for retrieving highly relevant documents. In Proceedings of the 23rd annual International ACM SIGIR Conference on Research and Development in Information Retrieval, p.41-48, New York, 2000

[6] F. V. Jensen. An introduction to Bayesian Networks. UCL Press, London, England, 1996.

[7] M. Lalmas. Uniform representation of content and structure for structured document retrieval. Technical report, Queen Mary and Westfield College, University of London, 2000.

[8] G. F. Luger and W. A. Stubblefield. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Third Edition. Addison Wesley Longman, Inc., 1998.

[9] S. H. Myaeng, D. H. Jang, M. S. Kim, and Z. C. Zhoo. A flexible model for retrieval of SGML documents. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998.

[10] P. Ogilvie and J. Callan. Combining document representations for known item search. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003.

[11] B. Piwowarski and P. Gallinari. A machine learning model for information retrieval with structured documents. In Machine Learning and Data Mining in Pattern Recognition (MLDM'03), p.425-438, 2003.

[12] S. E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC–7: automatic ad hoc, filtering, VLC and filtering tracks. In Proceedings of the 7th Text Retrieval Conference, 1999

[13] S. E. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 extension to multiple weighted fields. In Proceedings of Conference on Information and Knowledge Management (CIKM) 2004, 2004

[14] G. Salton, E.A. Fox, and H. Wu. Extended boolean information retrieval. Communications of the ACM, 26(11):1022--1036, 1983.

[15] G. Shafer. A Mathematical Theory of Evidence. Princeton Univesity Press, 1976.

[16] S. Shi, J.-R. Wen, Q. Yu, R. Song, and W.-Y. Ma. Gravitation-based model for information retrieval. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005.

[17] S. Shi, J.-R. Wen, Q. Yu, R. Song, and W.-Y. Ma. Gravitation-based model for information retrieval (extended version). Technique report, MSR-TR-2005-65, Microsoft Research, May 2005.

[18] TREC main page: http://trec.nist.gov/

[19] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. ACM Transactions on Information Systems, 9(3):187-222, 1991.

[20] C. C. Vogt and G. W. Cottrell. Fusion via a linear combination of scores. Information Retrieval, 1(3), p.151-173, 1999.

[21] R. Wilkinson. Effective retrieval of structured documents, Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, p.311-317, 1994.

APPENDIX

A. Proof of Proposition 1:

Proof: For each i ($1 \le i \le m$), construct the following i-dimension evidence score vectors,

$$U_i = (s_i - s_{i+1}, s_i - s_{i+1}, \cdots, s_i - s_{i+1})$$
$$R_i = (s_1 - s_i, s_2 - s_i, \cdots, s_i - s_i)$$
$$V_i = (s_1 - s_{i+1}, s_2 - s_{i+1}, \cdots, s_i - s_{i+1})$$

According to the Latent Additivity Assumption, we have

$$f(V_i) = f(R_i) + f(U_i)$$

As the last element of *Ri* is always zero, according to Property-3, we get,

$$f(R_i) = f(V_{i-1})$$

So,

$$\begin{aligned} f(S) = f(V_m) &= f(R_m) + f(U_m) \\ &= f(V_{m-1}) + \sigma(m) \cdot (s_m - s_{m+1}) \\ &= f(V_{m-2}) + \sum_{i=m-1}^{m} \sigma(i) \cdot (s_i - s_{i+1}) \\ &= \cdots \cdots \\ &= \sum_{i=1}^{m} \sigma(i) \cdot (s_i - s_{i+1}) \end{aligned}$$

Please note that we use the convention that s_0=0 and $\sigma(0) = 0$ in the process of proof.