ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2014.0118

ADAPTIVE CONTENT BASED TEXTUAL INFORMATION SOURCE PRIORITIZATION

Nikhil Mitra[1], Nilanjana Goel[2], S. Chakraverty[3] and Gurmeet Singh[4]
[1]*Division of Instrumentation and Control Engineering, Netaji Subhas Institute of Technology, India*
E-mail: nikhilmitra@nsitonline.in
[2, 3, 4] *Division of Computer Engineering, Netaji Subhas Institute of Technology, India*
E-mail: [2]nilanjana.goel@gmail.com, [3]apmahs.nsit@gmail.com, [4]gurmeet0108@gmail.com

Abstract

The world-wide-web offers a posse of textual information sources which are ready to be utilized for several applications. In fact, given the rapidly evolving nature of online data, there is a real risk of information overload unless we continue to develop and refine techniques to meaningfully segregate these information sources. Specifically, there is a dearth of content-oriented and intelligent techniques which can learn from past search experiences and also adapt to a user's specific requirements during her current search. In this paper, we tackle the core issue of prioritizing textual information sources on the basis of the relevance of their content to the central theme that a user is currently exploring. We propose a new Source Prioritization Algorithm that adopts an iterative learning approach to assess the proclivity of given information sources towards a set of user-defined seed words in order to prioritise them. The final priorities obtained serve as initial priorities for the next search request. This serves a dual purpose. Firstly, the system learns incrementally from several users' cumulative search experiences and re-adjusts the source priorities to reflect the acquired knowledge. Secondly, the refreshed source priorities are utilized to direct a user's current search towards more relevant sources while adapting also to the new set of keywords acquired from that user. Experimental results show that the proposed algorithm progressively improves the system's ability to discern between different sources, even in the presence of several random sources. Further, it is able to scale well to identify the augmented information source when a new enriched information source is generated by clubbing existing ones.

Keywords:

Textual Information Source Prioritization, Search Engines, Domain Specificity, Term-Source Matrix, Text Information Density

1. INTRODUCTION

Throughout the ages, human society has benefitted tremendously from large collections of textual information sources encoded in several analogue and digital formats. Most of these resources are now available online. They include an eclectic mix of valuable scientific and socio-cultural content hosted on specialized websites such as encyclopaedias, open access journals, e-books, news and entertainment portals, social networks as well as spam and junk. It is generally observed that some of these sources of information are more relevant as compared with others, either for learning about a particular domain or from a user's personal perspective.

Literature embodies a range of Information Retrieval (IR) techniques that first filter out several textual sources that match an input query and subsequently rank them based on their global popularity or by linking them to the user's personal web access patterns [1-6]. However, adequate attention has not yet been given to content-based ranking in a manner that utilizes the accumulative search experiences of several users while also adapting to an individual user's current search requirements.

In this paper, we propose a new iterative learning algorithm named Source Prioritization Algorithm (SPA) that adaptively prioritizes a given set of textual information sources in order to cater to a user's current exploration requirements. Furthermore, it utilizes the cumulative search experiences of past users to dynamically adjust the relative importance of these textual information sources. We demonstrate the feasibility of the proposed approach by conducting systematic tests on assessing the discernability, resilience and scalability of the text prioritization system.

The remaining paper is organised into the following sections. Section 2 gives a bird's eye view of related work done in the realm of ranking textual sources. Section 3 presents the detailed scheme for the proposed Source Prioritization Algorithm. Section 4 explains experimental procedures and analyses their results. We conclude our work in section 5 and give pointers towards future work.

2. PRIOR WORK

The task of ranking textual information is well understood and widely reported in the literature. In an interesting early work described in [2], Macskassy et al. outline an approach to prioritize large volumes of time-sensitive textual information based on certain prospective indicators. The prioritization is done in retrospect by using subsequent actions related to the text. Much work has since been done in text prioritization, especially in the field of E-Mail Management. In [3], Shinjae et al. propose a transductive learning algorithm for personalized e-mail prioritization based on social features. In [4], the authors outline an approach for content based email prioritization through unsupervised clustering, social network analysis, semi supervised feature induction, and supervised classification. The work in [5] presents a study of machine learning approaches to email prioritization into discrete levels. Semantic analysis to visualize unread emails in order to facilitate their prioritization is highlighted in [6] and [7]. Visual approaches to comparing textual data have been elucidated in [8]. Literature has evidence of significant work done in the field of ontology based document ranking. For example in [9], Guarino et al. define a retrieval system which hinges on ontology based distance measures. It uses WordNet ontology to rank and retrieve content on-line yellow pages and product catalogs. Ghose et al. propose ranking techniques for product reviews based on econometric analysis with text mining techniques and with subjectivity analysis [10].

As the web is being increasingly populated by unstructured databases such as blogs, book reviews and dynamic web pages, researchers are actively directing their efforts towards ranking these sources textual as well. For example in [11], the authors present a new blog ranking algorithm, called B2Rank, wherein the focus is on blog-sphere users' online behavioural features such as comment putting, blog updating rate, different types of citation and time of citation. Techniques that entail multi-platform data integration techniques for comparing distributed data objects, are described in [12] and [13].

An analysis of the above works reveals that prior research has accorded due importance to prioritizing structured and unstructured textual sources. A significant body of work has accumulated on e-mail prioritization and ontology based prioritization. However, very little attention has been paid to content-based prioritization of textual information at the source level itself, i.e. by drawing upon the knowledge contained within the sources themselves without explicit reference to any ontology. Furthermore, there is an urgent need to rank information sources adaptively by learning from past search experiences while also attuning to the current search requirements.

Search Engines lie at the core of most Information Retrieval (IR) systems [14]. Most popular search engines are based on the PageRank algorithm [15]. PageRank makes an inherent assumption that the forward and backward links and their relative reputation are good indicators of the importance of a page. Hence, Search Engines based on the PageRank metric order documents with regard to their 'popularity' rather than their actual relevance to the current query. But the original PageRank algorithm completely ignores the dimension of domain specificity. In [16], the authors compute a set of Page Rank vectors which are biased by a set of representative topics, instead of computing a single vector using the link structure of the web. In [17], the authors propose a scheme to improve the PageRank algorithm by using a more intelligent surfer, one that is guided by a probabilistic model of the relevance of a page to a query.

The above survey on IR systems highlights the importance of the link structure for a given page in order to rank it. However, this does not bode well for the pages that have a rich amount of content, though they may not be well linked. Hence, there is an urgent need for refining content specific prioritization of textual sources. The main thrust of the work presented in this paper is content driven source prioritization enhanced with learning from past search experiences and aligning to a user's current search directions.

3. SOURCE PRIORITIZATION ALGORITHM

It is a well-accepted fact that the relevance of any source of information depends upon its contents, but its real utility is also a subjective consideration. It is therefore imperative to provide some form of personalized guidance to steer users' search for online information sources towards their own perceptions about what they may find to be more useful.

User Specificity: Users' individual linguistic tastes and browsing habits have a bearing on the ranking of various information sources. For example, a user may prefer getting information about certain topics, suiting his own reading habits, from 'Britannica' rather than from 'Wikipedia'. Even though both sources are encyclopaedias, SPA tailors the relative priorities of these two sources by judging their similarity with user-given seed-words.

Domain Specificity: A knowledge source comprises several documents belonging to different themes or domains. However, sources typically dwell on certain themes more elaborately as compared with others. For example, Wikipedia may host articles for the domains 'Technology' and 'History', but the breadth and depth of coverage would vary for them. Some sources dedicate their contents towards a broad domain. For instance, the source 'TechGuru' is a more exhaustive source of information for 'Technology' as compared to 'History' though it may host some historical information too. These observations point towards the need for a domain centric source prioritization.

Table.1. List of symbols used and their meanings

Symbol	Meaning
$\mathcal{S}$	A set of textual information sources $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{\lvert\mathcal{S}\rvert}\}$
$\mathcal{D}$(k)	A set of documents retrieved from $\mathcal{S}$: $\mathcal{D} = \{\mathcal{D}1, \ldots, \mathcal{D}_{\lvert\mathcal{D}\rvert}\}$ $\lvert\mathcal{D}\rvert=\lvert\mathcal{S}\rvert$
N($\mathcal{D}$)	Total number of terms in a document $\mathcal{D}$
Seed-Words $\mathcal{W}$	A set of seed-words input by the user to guide the training phase of SPA $\mathcal{W} = \{\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_{\lvert\mathcal{W}\rvert}\}$
$\mathcal{P}$[$\lvert\mathcal{S}\rvert$]	A vector representing priorities assigned to the members of sources $\mathcal{S}$: $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_i, \ldots, \mathcal{P}_{\lvert\mathcal{S}\rvert}\}$
$\mathcal{K}$[m][$\lvert\mathcal{D}\rvert$]	A matrix denoting the strength of key-words and key-phrases (Terms) in documents
$\mathcal{T}$[m]	A vector of cumulative Term strengths
$\mathcal{U}$[$\lvert\mathcal{D}\rvert$]	A vector of cosine distances between source Term strengths and cumulative Term strengths

SPA(.)
INPUT : $\mathcal{S}$, $\mathcal{W}$
OUTPUT: $\mathcal{P}$
Initialize Static variables $\mathcal{P}_i$ to set all source priorities to be equal:
For i = 1..|$\mathcal{S}$|, let $\mathcal{P}_i$ = 1/| $\mathcal{S}$ |
Let $\mathcal{D}$ = ϕ, $\mathcal{D}'$= ϕ
For each seed-word $\mathcal{W}_k \in \mathcal{W}$
{
For each source $\mathcal{S}_j \in \mathcal{S}$ {
Extract the first document $\mathcal{D}$(j, k) from $\mathcal{S}_j$ containing $\mathcal{W}_k$
$\mathcal{D}= \mathcal{D} \cup \mathcal{D}(j, k)$ }
For each document $\mathcal{D}_j \in \mathcal{D}$ {
Derive $\mathcal{D}_j'$ by removing HTML tags and extracting content from $\mathcal{D}_j$
Extract all words and keyphrases (together called Terms) from $\mathcal{D}_j$'
$\mathcal{D}'= \mathcal{D}' \cup \mathcal{D}_j'$ }
Construct Term-Source Matrix $\mathcal{K}$ using all terms/phrases in the docs in $\mathcal{D}'$ (Eq. 1)
Generate $\mathcal{K}'$ by normalizing each Term strength in $\mathcal{K}$ using Text Information Density δ (Eqs. 2 and 3)
Generate $\mathcal{K}''$ by calculating Priority-weighted Term strengths (Eq.4)
Generate cumulative Term strength matrix $\mathcal{T}$: For each Term (row i) in $\mathcal{K}''$
Calculate cumulative Term strength $\mathcal{T}_i$ by summing across all documents (Eqs. 5, 6 and 7)
Generate cosine distance matrix : For each Source in $\mathcal{S}$ (Eqs. 8 and 9)
Calculate cosine distance between document-specific Term-Strength & cumulative Term-Strength
Update each source priority (Eq. 10)

$$\mathcal{P}'_j = \frac{Individual\ Cosine\ Distance}{\sum_{j=0}^{m} Individual\ Cosine\ Distance}$$

}

Fig.1. Pseudo-code for the Source Prioritization Algorithm

SPA alters the prior priorities of a given set of textual sources by attuning itself to a set of new "seed-words" that represent clues about the current user's specific quest. It is also a learning algorithm that discerns the domain specificity of textual information sources by repeatedly assessing their priorities using inputs from a series of users that averages out individual influences and makes apparent their global domain-centric features. Table.1 explains the symbols used in our explanation of SPA. The notation |.| denotes cardinality of a set.

The Fig.1 shows the pseudo-code describing the working of SPA. It takes as input (i) The set of the sources $\mathcal{S}$ that must be prioritized, (ii) a set of seed-words $\mathcal{W}$. It outputs the updated priorities $\mathcal{P}$ of sources $\mathcal{S}$ for each seed-word $\mathcal{W}_k$.

Step 1: *Initialization* (*Lines* 1-3)*:* The static variables $\{\mathcal{P}_i\}$, which are the initial priorities of all sources in $\mathcal{S}$, are initialized to 1/|$\mathcal{S}$|. Assuming no prior knowledge about of the content of any of the sources, it is only natural that we assign them equal importance. Being static variables however, they are initialized only for the very first invocation of SPA. At this stage, the only factor affecting the relative priorities of sources is the set of seed-words provided by the first user. For subsequent invocations of the algorithm, the priorities resulting from the previous search are taken as the initial priorities. Further, set variables $\mathcal{D}$ and $\mathcal{D}'$ are initialized to ϕ to be populated later.

Step 2: *Extract relevant documents for a seed-word* (*Lines* 4-8)*:* SPA now treats each of the given set of seed words in $\mathcal{W}$ turn by turn. Given a seed-word $\mathcal{W}_k$, the process scans through all sources to retrieve the first document from each source that contains this seed-word. Let us denote as (*i*,*k*), the particular document of source $\mathcal{S}_j$ that contains $\mathcal{W}_k$. In our current implementation, exactly one document is extracted per source for each seed-word. The documents thus retrieved from all sources form the set $\mathcal{D}$. Note that the documents in $\mathcal{D}$ bear a one-to-one correspondence with the sources in $\mathcal{S}$. Hence |$\mathcal{D}$|=|$\mathcal{S}$|. Since all of them contain the same keyword, we can drop the keyword index *k*, and identify each document in $\mathcal{D}$ by the source index *j* alone as $\mathcal{D}_j$.

Step 3: *Extract document contents* (*Lines* 9-12): All documents in $\mathcal{D}$ are filtered to remove formatting tags and markup elements so that only their main textual content remains. Any content extraction algorithm can be employed for this purpose [18] [19]. The new set of filtered documents is denoted $\mathcal{D}'$.

Step 4: *Construct Terms-Source Matrix* (*Line* 13): Each document in $\mathcal{D}'$ is run through a Key Extraction Algorithm (KEA). This step assesses the relative strength of each term and each multi-word unit or key-phrase that occurs in a given document, by applying a key extraction technique. Individual words and key-phrases are together referred to as "Terms" here.

TextRank is an unsupervised algorithm which does not rely on corpuses and hence offer better results than supervised KEA algorithms such as TF-IDF. RAKE [20], another popular algorithm, extracts key-phrases instead of just keywords, and thus offers more accurate

results. However, due to the inherent scarcity of terms on the TDM, RAKE takes a larger number of seed words for the source priorities to converge to a stable value. A comparison of TF-IDF, TextRank and RAKE can be found at [21].

We employed the TextRank algorithm which is a graph-based ranking model for text processing, for the KEA step. When one vertex links to another, it is said to cast a vote for it. The strength of a vertex is determined by the number of votes cast for it [23]. In our adaptation of TextRank, we added the terms and key-phrases in a document as vertices in the graph. Two vertices (Terms) are connected if their corresponding lexical units co-occur within a window of 2 words.

The KEA step returns the Terms-Sources Matrix TSM, $(k)[m][\ |\mathcal{D}|]$, containing the strength of all Terms that are present in the set of filtered documents $\boldsymbol{\mathcal{D}'}$.

$$(k)[m]\ [\ |\mathcal{D}|]=\begin{pmatrix} \mathcal{K}_{1,1} & \cdots & \mathcal{K}_{1,|\boldsymbol{\mathcal{D}}|} \\ \vdots & \ddots & \vdots \\ \mathcal{K}_{m,1} & \cdots & \mathcal{K}_{m,|\boldsymbol{\mathcal{D}}|} \end{pmatrix} \quad (1)$$

Here $|\boldsymbol{\mathcal{D}}|$, the number of documents is the same as number of sources $|\boldsymbol{\mathcal{S}}|$, m is the total number of unique terms extracted from all the documents in $\boldsymbol{\mathcal{D}'}$, and k denotes the original seed-word $\mathcal{W}_{\mathrm{k}}$ that generated $\boldsymbol{\mathcal{D}}$. For compact representation, we have dropped the parameter k in the $\mathcal{K}$ terms. If the i^{th} Term is not present in the j^{th} document, then $\mathcal{K}_{i,j} = 0$.

Step 5: *Normalize* $\mathbb{K}$ *with Text Information Density* (*Line* 14)*:* It is observed that if we normalize Term-strengths by the length of the document, a small document with very few terms will yield high relative strengths. To remove this kind of imbalance, we use the metric Text Information Density (TID). The proportion of all words in a document that serve as connected Terms indicates the density of relevant information encapsulated within it. The factor Text Information Density $\delta(j)$ is defined as the ratio of the total strength of all key-phrases extracted from a document $\mathcal{D}_{\mathrm{j}}$ to the total number of words present in it. Thus:

$$\delta(j) = \frac{\sum_{i=1}^{m} \mathcal{K}_{i,j}}{\mathrm{N(D_j)}} \quad (2)$$

Using $\delta(.)$ to measure the relative strength of key-phrases amortizes the effect of varying lengths of the documents. The relative strength $\mathcal{K}'_{i,j}$ of each key-phrase is obtained by dividing its individual strength by the document's TID.

$$\mathcal{K}'_{i,j} = \frac{\mathcal{K}_{i,j}}{\delta(j)} = N(D_j)\frac{\mathcal{K}_{i,j}}{\sum_{i=1}^{m}\mathcal{K}_{i,j}} \quad (3)$$

Step 6: *Evaluate priority-weighted source strengths* (*Line* 15)*:* The normalized Term strength values are multiplied with their corresponding source priorities, as evaluated in the current iteration for seed word $\mathcal{W}_k$, to get the source weighted strength.

$$\mathcal{K}''_{i,j} = \mathcal{P}_j N(\mathcal{D}_j)\frac{\mathcal{K}_{i,j}}{\sum_{i=1}^{m}\mathcal{K}_{i,j}} \quad (4)$$

Step 7: *Calculate Cumulative Term Strength across documents* (*Lines* 16-17)*:* Let the row for the i^{th} term in $\mathbb{K}''(k)$ be denoted as $\mathcal{V}_{\mathrm{i}}$.

$$\boldsymbol{\mathcal{V}_{\mathbf{i}}} = \{\mathcal{K}''_{i,j} \mid \forall\, j = 1..|\boldsymbol{\mathcal{D}}|\} \quad (5)$$

The cumulative Term strength $\mathcal{T}_i$ represents the overall importance of the i^{th} Term extracted from these documents with regard to the source seed-word $\mathcal{W}_k$.

$$\mathcal{T}_i = \|\mathcal{V}_i\| \quad (6)$$

We utilize the Taxicab or Manhattan Norm [22] for evaluating the norm as it is a good indicator of the overall relevance of the i^{th} Term for a single user-defined seed-word $\mathcal{W}_k$ considering all available sources. Vector $\boldsymbol{\mathcal{T}}$**[m]** represents the cumulative strength for all Terms extracted from $\boldsymbol{\mathcal{D}}$.

$$\boldsymbol{\mathcal{T}}[\boldsymbol{m}] = \{\textstyle\sum_{j=1}^{|\boldsymbol{\mathcal{D}}|}\mathcal{K}''_{i,j} \mid \forall\, i = 1..m\} \quad (7)$$

Step 8: *Find cosine distances* (*Lines* 18-19): Each source vector, *i.e.* column of the normalized and weighted TSM $\mathbb{K}''$ is now compared with the cumulative Term strength vector using vector distance function cosine distance.

$$\mathcal{U}_{\mathrm{j}} = \textstyle\sum_{i=1}^{m}\mathcal{K}''_{i,j}\mathcal{T}_i(k) \quad (8)$$

This gives us the matrix of cosine distances $\boldsymbol{\mathcal{U}}[|\boldsymbol{\mathcal{D}}|]$, between the Term strengths of each and every source and the cumulative Term strength vector **[m].**

$$\boldsymbol{\mathcal{U}} = \{\mathcal{U}_{\mathrm{j}} | \forall\, j = 1..|\boldsymbol{\mathcal{D}}|\} \quad (9)$$

Step 9: *Calculate New Priorities* (*Line* 20-22)*:* A higher value of cosine distance signifies greater relevance of the source towards the input seed-word $\mathcal{W}_{\mathrm{k}}$. The matrix $\boldsymbol{\mathcal{U}}$ can thus be used to calculate the updated source priorities as the source vector closer to the cumulative term strength vector is indicative of a better source of information. The new importance weight of source $\mathcal{S}_j$ is:

$$\mathcal{P}'_j = \frac{\mathcal{U}_{\mathrm{j}}}{\sum_{j=0}^{m}\mathcal{U}_{\mathrm{j}}} \quad (10)$$

where, $\boldsymbol{\mathcal{P}'}$ is the vector of updated priorities.

The process enclosed within line 4 and line 22 including the steps 4 to 9 just explained, is again invoked for the next seed-word $\mathcal{W}_{k+1}$, this time using the just calculated source priorities. This re-evaluates their new priorities with reference to the new seed-word. The iterative process repeats till all seed-words in $\boldsymbol{\mathcal{W}}$ are exhausted. The final source priorities $\{\boldsymbol{\mathcal{P}}_i\}$ reflect the relative importance of each textual information source in terms of the entire set of seed-words input by a user.

At the end of this iterative process, the final source priorities $\{\boldsymbol{\mathcal{P}}_i\}$ reflect the learning garnered from the past search. These values are inherited by the next search request for their subsequent adaptation to the new set of search seed-words. For a fixed set of sources, the accumulated learning over a span of multiple searches drives the sources priorities towards their stable steady-state values. The learning continues when new sources are added.

4. TESTING AND RESULTS

We now present experimental results to illustrate the performance of the proposed SPA tool. Our objective is to test how well SPA can discriminate between a given set of textual information sources. Further, we will show how the system responds to "noise" in the form of random sources and when new sources with enhanced knowledge are added.

4.1 TEST ENVIRONMENT

Resources: We performed our experiments on Intel Core i3 processor running Windows 7, using the following computing resources:

- Programming Language: Python
- IR System: Commercial Search Engine Google Site Search with Python urllib downloader.
- Content Extractor tool: Goose Extractor.
- Key-phrase Extraction Algorithm used: TextRank [23]

Textual Information Sources: To prepare our test bench, we used popular websites that contain a rich repertoire of information on various topics, as textual information sources. The documents are html pages on these websites. A mix of these sources were chosen because they represent distinctive methods of curating knowledge. These are:

Source 1: Wikipedia [24]: Wikipedia is a free, open content online encyclopedia created through the collaborative effort of a community of users known as Wikipedians.

Source 2: Citizendium [25]: Citizendium, like Wikipedia, is a free encyclopedia, with the constraint that the authors use their real, verified names.

Source 3: Encyclopaedia Britannica [26]: The Encyclopædia Britannica (Latin for "British Encyclopaedia"), published by Encyclopædia Britannica, Inc., is a general knowledge English-language encyclopaedia.

Source 4: Random Source - In order to test the efficacy of the system, we synthetically generated a random source. The random text was generated from a corpus of adjectives constructed manually. For fair comparison, the random text was made roughly equal in length to that of the other documents.

Seed words: we selected the following eight seed-words for testing the relevance of the above information sources for a wide spectrum of topics that frequently in several domains such as science, technology, geography, music and so on. We assume that these seed words are specific to a random user.

India, dog, metabolism, atom, Biology, United States of America, Led Zeppelin, Prime number

4.2 EXPERIMENTS

In order to test the SPA tool, we conducted three experiments as described below.

4.2.1 Test for Discernibility:

It is imperative that any information source prioritization algorithm be able to discern between different information sources. The test for discernibility was designed to test whether, given a set of user-input seed words, the SPA system is able to differentiate between them by assigning reasonably spaced apart priorities.

We input the three information sources 1, 2 and 3 and the random source and the given set of seed words to SPA.

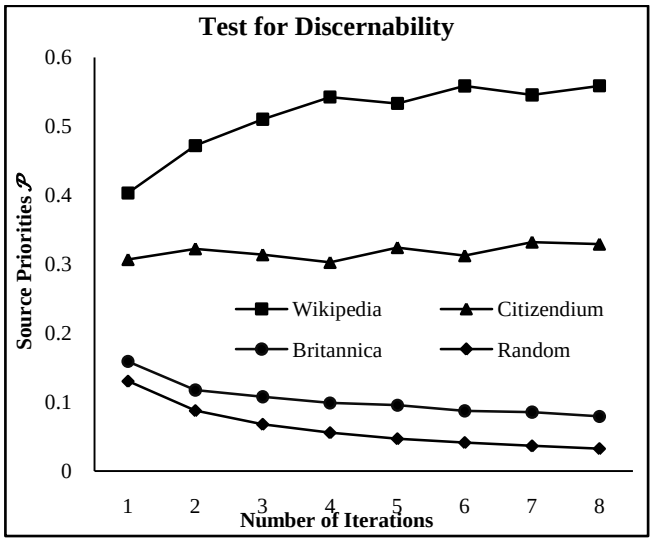

Fig.2. Evolution of source priorities with keywords

The Fig.3 shows the change of priorities that SPA assigned to the different sources as it progressed through its iterations, each iteration driven by a fresh seed word. The X axis represents the iteration number and the Y axis shows the source priority $\mathcal{P}$. The following observations may be noted:

1) The priorities converge to within 10% of their final stable values after 6 iterations when SPA has been trained with 6 seed words.
2) The difference between the priorities of the top ranked knowledge source Wikipedia and the least ranked random source improves steadily as the training progresses. This difference started at 0.27 after the first iteration and increased by 48% to 0.52 at the end of 8 training cycles. This demonstrates that SPA is able to enhance the degree of differentiation between different sources.
3) Wikipedia received the highest rank followed by Citizendium and then Britannica. Significantly, the random source was consistently ranked below all information sources. After 8 training cycles, the final relevance values and priorities of the sources were:
   - Citizendium: The relevance of 0.32 is 41% less than that of Wikipedia.
   - Britannica: The relevance value of 0.07 is 85% less than that of Wikipedia.
   - Random source: It was assigned a very low relevance value of 0.03 which is 8% lesser that the least ranked knowledge source (Britannica).

Thus SPA could successfully segregate the random source from the knowledge sources by ranking it the lowest.

4.2.2 Test for Resilience:

Any comparative knowledge ranking algorithm must be able to withstand the presence of "noise" which in this case, takes the form of several random sources. As the ranking is driven by aggregated key-phrase strengths, we expect that the addition of random sources will have a detrimental effect on the aggregation, and ultimately the ranking. This test was designed to test the resilience of SPA in the presence of such random sources and verify whether it is still able to discern between sources adequately. For this test, we added as many as 8 random sources.

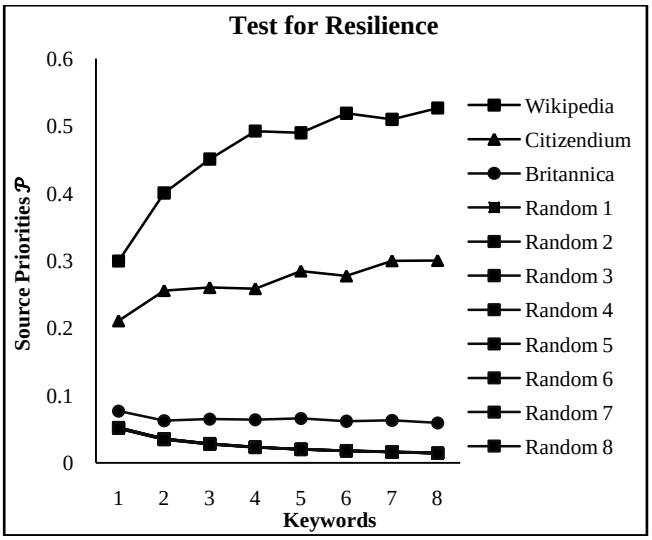

Fig.3. Evolution of source priorities with keywords in the presence of 8 random sources

The Fig.4 shows the change of priorities assigned to different sources as SPA progressed through its iterations, each driven by a fresh seed word. The X axis represents the iteration number and the Y axis shows the source priority $\mathcal{P}$. The following observations can be seen:

1) The random priorities are distinct from source priorities even when number of random sources is as high as 8. This shows that SPA is resilient to being influenced even by a number of random sources.
2) All Random Sources are very closely clumped together. SPA is able to club them all together as non-useful sources.
3) The maximum difference between the top-ranked source Wikipedia and least ranked random source reduced very marginally to 0.51 as compared with 0.52 when a single random source was used in experiment 4.2.1.

Thus, SPA could withstand multiple random sources with negligible degradation in its ability to discriminate between sources.

4.2.3 Test for Scalability:

In this test, the algorithm is tested for its ability to identify the improvement in the quality of information when similar sources are clubbed together. We created an augmented source by merging together the contents of Citizendium (C) and Britannica (B). This richer source of information (C+B) was made to compete with the three individual information sources described before.

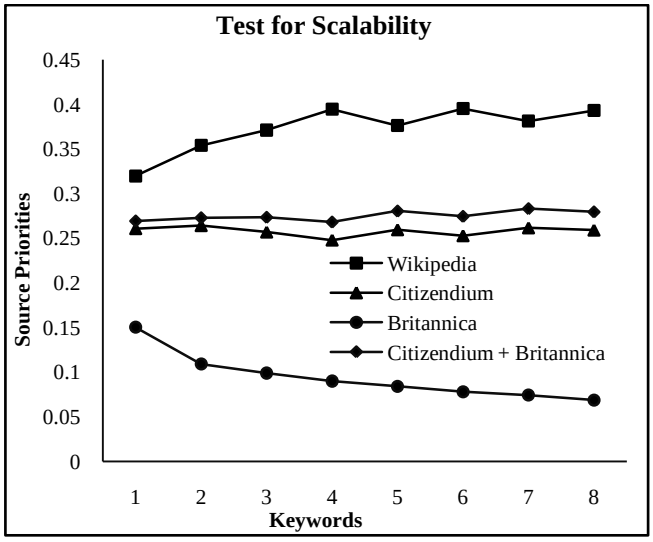

Fig.4. Evolution of source priorities when sources are enhanced

From Fig.4 we can observe that:

- The combined priority of [C+B] was always ranked higher than either C or B.
- Further, the priority of Britannica continues to decline considerably in the presence of a better source.
- Thus SPA ranked the combined knowledge source higher than either of the sources.

The above test proves that SPA is capable of sensing the improvement in the quality of the information sources when they are augmented with more information.

5. CONCLUSION

In this paper, we addressed the challenge of objectively ranking different textual information sources on the basis of the relevance of their contents to the underlying theme or domain of knowledge which the user is currently seeking to explore. The Source Prioritization Algorithm (SPA) has been developed and rigorously tested for its ability to discern between differently distinctly curated information sources as well as between genuine sources and random sources.

We are currently working towards applying SPA for implementing a recommender system and a search engine to make them reap the benefits of dynamic and adaptive source prioritization.

REFERENCES

[1] Vasanthan S. Dasan, "Personalized information retrieval using user-defined profile", U.S. Patent No. 5,761,662. 2 1998.

[2] Sofus A. Macskassy Haym Hirsh, Foster Provost, Ramesh Sankaranarayanan and Vasant Dhar, "Intelligent information triage", *Proceedings of the 24th Annual*

International Conference on Research and Development in Information Retrieval, 2001.

[3] Shinjae Yoo, Yiming Yang, Frank Lin and II-Chul Moon, "Mining social networks for personalized email prioritization", *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[4] K. Harini and Uppe Nanaji, "Implementation of personalized email prioritization- A content based social network analysis", *International Journal of Computer Science and Communication Networks*, Vol. 1, No. 3, pp. 218-221, 2012.

[5] Shinjae Yoo, Yiming Yang and Jaime Carbonell, "Modeling personalized email prioritization: classification-based and regression-based approaches", *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 2011.

[6] Steven L. Rohall, Dan Gruen, Paul Moody, Martin Wattenberg, Mia Stern, Bernard Kerr, Bob Stachel, Kushal Dave, Robert Armes and Eric Wilcox, "ReMail: A reinvented email Prototype", *Proceedings of the CHI '04: CHI '04 Extended Abstracts on Human Factors in Computing Systems*, pp. 791-792, 2004.

[7] Danyel Fisher, Bernie Hogan, A. J. Brush, Marc A. Smith and Andy Jacobs, "Using social sorting to enhance email management", *Human-Computer Interaction Consortium*, 2006.

[8] Dixon Ip, Ken Lau Ka Keung, Weiwei Cu, Huamin Qu and Helen Shen, "A visual approach to text corpora comparison", *Proceedings of the First International Workshop on Intelligent Visual Interfaces for Text Analysis*, pp. 21-24, 2010.

[9] N. Guarino, C. Masolo and G. Verete, "OntoSeek: Content Based Access to the Web", *IEEE Intelligent Systems and their Applications*, Vol. 14, No. 3, pp. 70-80, 1999.

[10] Anindya Ghose and Panagiotis G. Ipeirotis, "Designing novel review ranking systems: predicting the usefulness and impact of reviews", *Proceedings of the ninth international conference on Electronic commerce*, 2007.

[11] Mohammad A. Tayebi, S. Mehdi Hashemi and Ali Mohades, "B2Rank: An algorithm for ranking blogs based on behavioral features", *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 104-107, 2007.

[12] Jerome Euzenat and Pavel Shvaiko, "*Ontology Matching*", Springer-Verlag, 2007.

[13] A. Smeaton and I. Quigley, "Experiment on Using Semantic Distance Between Words in Image Caption Retrieval", *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, pp. 174-180, 1996.

[14] Boris Chidlovskii, Natalie S. Glance and Antonietta Grasso, "System and method for collaborative ranking of search results employing user and group profiles derived from document collection content analysis", U.S. Patent No. 6,327,590, 2001.

[15] Sergey Brin, and Lawrence Page, "The anatomy of a large-scale hypertextual Web search engine", *Computer networks and ISDN systems*, Vol. 30, No. 1, pp. 107-117, 1998.

[16] Taher H. Haveliwala, "Topic-sensitive pagerank", *Proceedings of the 11th International Conference on World Wide Web*, 2002.

[17] Matthew Richardson and Pedro Domingos, "The intelligent surfer: Probabilistic combination of link and content information in pagerank", *Advances in neural information processing systems*, Vol. 2, pp. 1441-1448, 2002.

[18] Suhit Gupta, Gail Kaiser, David Neistadt and Peter Grimm, "DOM-based content extraction of HTML documents", *Proceedings of the 12th international conference on World Wide Web*, 2003.

[19] Tomaž Kovačič, "Evaluating Web Content Extraction Algorithms", EngD thesis, 2012.

[20] Stuart Rose, Dave Engel, Nick Cramer and Wendy Cowley, "Automatic keyword extraction from individual documents", *Text Mining: Applications and Theory*, pp. 1-20, 2010.

[21] Martin Dostál and Karel Jezek, "Automatic Keyphrase Extraction based on NLP and Statistical Methods", *DATESO*, pp. 140-145, 2011.

[22] Eugene F. Krause, "*Taxicab geometry: an adventure in non-Euclidean geometry*", Dover Publications, 1986.

[23] Rada Mihalcea and Paul Tarau, "TextRank: Bringing order into texts", *Association for Computational Linguistics*, 2004.

[24] Wikipedia.org, "Wikimedia Foundation", Accessed on: 10 July 2014.

[25] Wikipedia contributors, "Citizendium", *Wikipedia, The Free Encyclopedia*, Accessed on: 29 July 2014.

[26] Wikipedia contributors, "Encyclopædia Britannica", *Wikipedia, The Free Encyclopedia*, Accessed on: 29 July 2014.