Enhancing Information Retrieval by Personalization Techniques

25November 2014

Research Supervisor

Dr. R. Shanmugalakshmi
Associate Professor,
Department of Computer Science & Engineering,
Government College of Technology,
Coimbatore, Tamil Nadu, India.

Scholar

Veningston .K
Senior Research Fellow,
Department of Computer Science & Engineering,
Government College of Technology,
Coimbatore, Tamil Nadu, India.

Thesis Outline

- Objectives of Research work
- Introduction
- Related Works
- Problem Descriptions
- Research Work Modules
  - Enhancing Information Retrieval using Term Association Graph Representation
  - Integration of Document topic model and User topic model for personalizing Information Retrieval.
  - Computational Intelligence for Document Re-ranking using Genetic Algorithm (GA)
  - Computational Intelligence for Search query reformulation using Ant Colony Optimization (ACO)
- Concluding Remarks
- References and Acknowledgements

Objectives of Research Work

- To study the personalized web information gathering approaches with an intention to satisfy a user's search context.
- To study the techniques that uses the content semantics of web documents in order to improve the retrieval effectiveness by employing Term Association Graph data structure to assess the importance of a document for the user query and thus web documents to be re-ranked according to the association and similarity exists among the documents.
- To devise an efficient mechanism to re-rank web search results by personalized ranking criteria which are typically derived from the modeling of users' search interests and preferences.
- To enhance personalized search system by employing Document topic model that integrates User topic model i.e. usage data with content semantics, in order to perform semantically enhanced retrieval task.
- To devise an approach to prepare personalized related query suggestions to enrich search queries to better represent the meanings of a query than the textual keywords.

Introduction

- Why Personalization in Search?
  - Personalization is an attempt to find most relevant documents using information about user's latent goals, knowledge, preferences, navigation history, etc.

Classifications of Typical IR systems

- Content-based approaches
  - Using language to match a query with results - this approach doesn't help users determine which results are actually worth reading
- Author-relevancy techniques
  - Using citation and hyperlinks - sometimes presents the problem of 'authoring bias' and/or 'ranking bias' (results that are valued by authors are not necessarily those valued by the entire population)
- Usage rank
  - This "leverages the actions of users to compute relevancy" - the usage rank is computed from the frequency, recency, and/or duration of interaction by users - usage ranks allow for changes in relevancy over time to be determined

Limitations in Typical IR systems

- Most of the techniques measure relevance "as a function of the entire population of users"
- This does not acknowledge that "relevance is relative" for each user
- There needs to be a way to "take into account that different people find different things relevant and that people's interests and knowledge change over time - "personal relevance"

General Approach for mitigating Challenges [1/2]

•In order to personalize search, we need to combine at least two different computational techniques

•Contextualization - “the interrelated conditions that occur within an activity..includes factors like the nature of information available, the information currently being examined, and the applications in use”

•Individualization - “the totality of characteristics that distinguishes an individual.. Uses the user’s goals, prior and tacit knowledge, past information-seeking behaviors”

General Approach for mitigating Challenges [2/2]

• Main ways to personalize a search are "Result processing" and "Query augmentation"

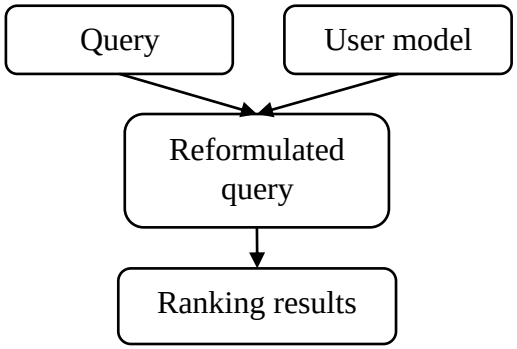
•Document Re-ranking – Another processing method is to re-rank the results based upon the "frequency, recency, or duration of usage. Providing users with the ability to identify the most popular, faddish and time-consuming pages they have seen"

•Query Reformulation - when a user enters a query, the query can be compared against the contextual information available to determine if the query can be refined to include other terms

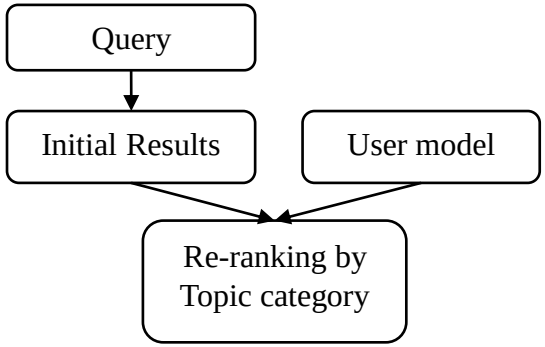
•The user model "can re-rank search results based upon the similarity of the content of the pages in the results and the user's profile"

Types of Personalization

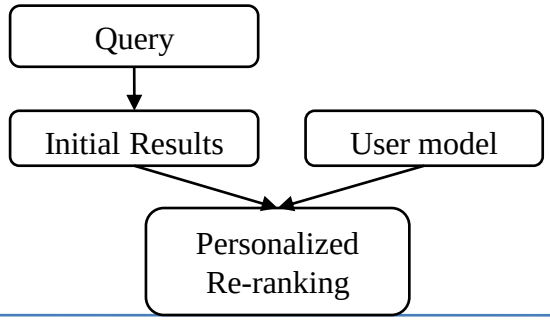
- Pre-retrieval Personalization



- On-retrieval Personalization



- Post-retrieval personalization



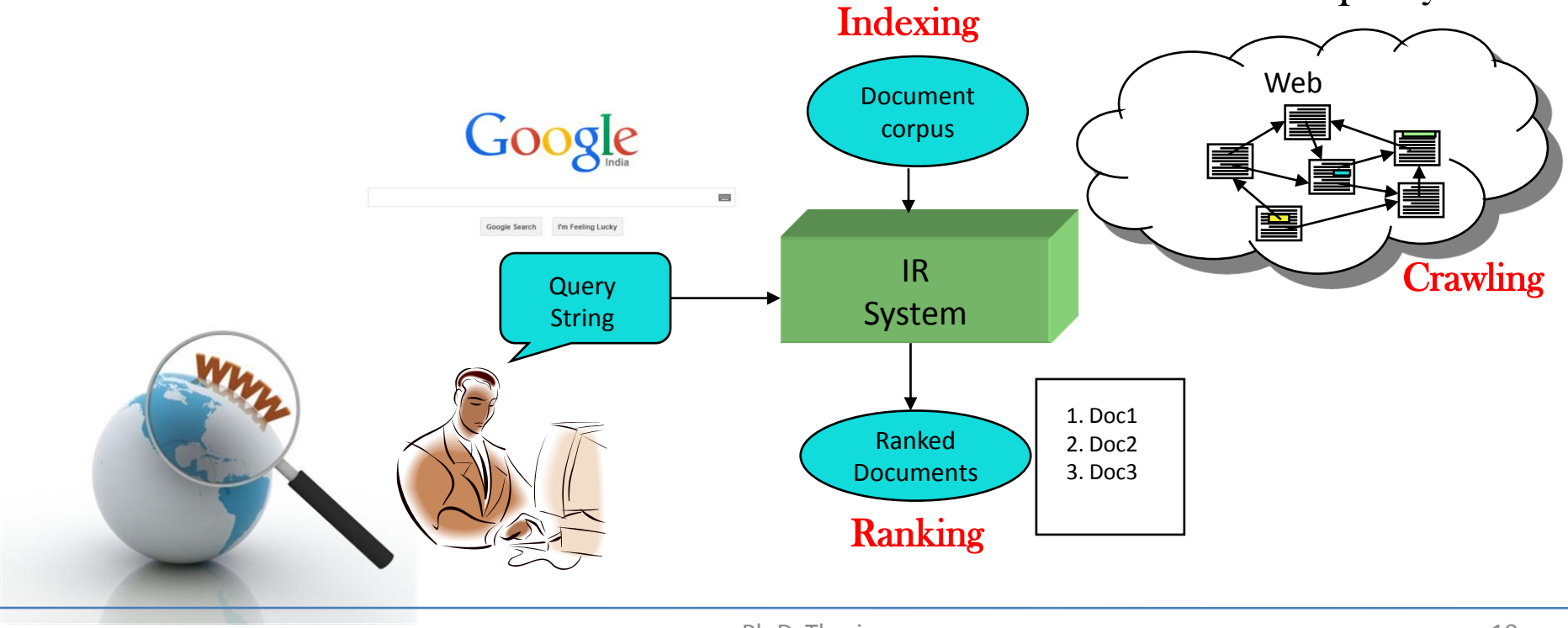
Typical Information Retrieval (IR) Task

Given:

- A corpus of textual natural-language documents.
- A user query in the form of a textual string.

To Find:

- A ranked set of documents that are relevant to the query.



General Problem Description [1/2]

- Unknown of an Unknown of an Unknown

User Information Need	User search context	Results retrieved
Unknown	Unknown	Unknown

- Issues to be addressed
  - To understand the keyword query issued by the user which is short and ambiguous
  - To model user interest profile by using user's search history for ranking the documents
  - To identify the more relevant documents based on individual users interest

General Problem Description [2/2]

- Diverse interest of users

Original query	Intention on relevant documents		
	User 1	User 2	User 3
World cup	Web pages mainly dealing with the foodball championship	Web pages mainly dealing with the ICC cricket world cup	Web pages mainly dealing with the T20 cricket world cup
India crisis	Web pages dealing with the economic crisis in India	Web pages dealing with the security crisis in India	Web pages dealing with the job crisis in India
Data structures	Web pages about online book store to purchase data structures book	Web pages about data structure E-book downloads	Web pages about programming libraries to implement data structures
Apple	Web pages on Apple store	Web pages on varieties of apple fruit	Web pages on Apple OS updates and downloads
Job search	Web pages about student part time jobs	Web pages about government jobs	Web pages about engineering and IT job search
Cancer	Web pages relating to cancer astrology and zodiac	Web pages relating to lung cancer and prevention	Web pages relating to causes of cancer, symptoms and treatment
The ring	Web pages about Ornament	Web pages about the horror movie	Web pages about circus ring show
Okapi	Pages related to animal giraffe	Pages related to okapi African luxury hand bags	Pages related to Information retrieval model BM25

Related Work on Result Re-ranking

- Short term personalization
- Long term personalization
- Hyperlink based personalization
- Collaborative personalization
- Personalized web search using categories
- Session based personalization
- Location based personalization
- Task based personalization
- Tag based personalization
- Folksonomy based personalization
- Click based personalization
- Cluster based personalized search
- Result Diversification

Related Work on Query Reformulation

- Query refinement using Anchor text
- Query suggestion using bipartite graph
- Personalized facets
- Query reformulation using term association patterns
- Query reformulation using Rule based classifier
- Semantic models of query reformulation
- Clustering query suggestions
- Merging query reformulations
- Query expansion using Co-occurrence

Analysis on variants of Personalization Approach

Approach	User interest representation	User interest learning	User profile exploitation	Similarity measure	Source of user profile	Knowledge base	Remarks
Short term personalization	Terms	Implicit	Document weighting	Topic similarity using user interest hierarchy	Bookmark	Ad-hoc and Top ranked results	diverse result are not presented
Session based personalization	Terms and Contexts	Implicit	Document weighting	concept-based terms, User/Topic similarity	Search history	Ad-hoc and User's topical interest	Changes in the user interests across search sessions needs to be captured and adapted
Long term personalization	Terms	Implicit	Document weighting	Term-similarity and Term frequency	Browsing history, Query and Click-through log	Ad-hoc and Top ranked results	It requires more number of attributes to represent user model
Query ambiguity prediction	Terms	Implicit and Explicit	query expansion	Term-similarity and Query/Term similarity	surrounding context	Top ranked results	it is needed to analyze a large sample of the query logs
User Modeling	Terms, context, usage history, documents	Implicit	Document weighting	Term-similarity, User/Topic similarity	Click-through log	ODP, Top ranked results	User model needs to be optimized
Collaborative personalization	group context information	Implicit	Document Suggestion and weighting	Term-similarity, User/Topic similarity, Term frequency	Click-through log	Top ranked results	users with the same interest may share their profiles so to produce the result set
Search interaction personalization	usage history, documents	Implicit	Document weighting, ranking, filtering	Topic similarity, User/Topic similarity	click-through, browsing, and query-text features	Top ranked results	feedback model is derived from mining millions of user interactions
Ontology based personalization	Terms and context	Implicit	link based or association based	Topic similarity, User/Topic similarity	query-text features	Concept net, Term graph and association (Domain ontology)	This does not ensure that the ontological user profile is updated with changes that reflect in user interests

Comparisons of Personalization Approaches

Type of Approaches	Techniques used	Dataset used	Evaluation measures	Limitations	Merits
Implicit preference	Collaborative algorithm, Term Frequency scheme	TREC WT10g test collection	Precision	Noisy browsing history is used	Considers persistent and transient preferences
Hyper link structure	Link structure analysis	Web URLs	Precision	Computes universal notion of importance	Identify high quality, authoritative web pages
Collaborative filtering	Groupization algorithm, k-Nearest Neighbor algorithm	Geographical group data, occupational group data, MSN query logs	NDCG	Scalability issue, user data are dynamic	Predicts user preferences based on other users
Using categories	Mapping queries to related categories	ODP category	Accuracy	Predefined categories are used	Categories are used as a context of the query
Long-term user behavior	Create profiles from entire past search history	Microsoft Bing search logs, TREC 2009 web search track	MAP, NDCG	Do not address the searcher needs for the current task	Historic behavior provides substantial benefits at the start of a search session
Short-term user behavior	Create profiles from recent search session	Microsoft Bing search logs	MAP	This lacks in capturing users long term interest	Capture recent interactions with the system
Using tag data	Creates Tag based and content based profiles	Data from social bookmarking sites	MRR, success at top N	Biased towards particular group	Provides consensus categories of interesting web pages
Location awareness	Joachim's, Spy-naïve Bayes method	Location ontology, click-through data	Precision at top N	Captures location information based on simple text matching	Extracts content, location concepts and its relationships
Task awareness	Determines the ambiguity level of a query using task language model	Real time search log of users	NDCG	The temporal features of user tasks are not considered	Decides whether a query need personalization or not
Contextual information	Modeling users interest using context based features	ODP Category labels	Precision, MRR, NDCG	Treat all context sources equally	User interest modeling has many representation, preserves privacy
Folksonomy-based	Social tagging	Social bookmarks by the users	MRR, precision at position N	Not scalable to web	User-based tag frequency is measured
Click-based	Assesses the web pages frequently clicked by the users, collaborative learning	MSN query logs	Rank scoring, average rank	Makes no use of the terms and its associated weights	Click entropy ensures whether to personalize or not
Result diversification	Relevance feedback approach, Greedy algorithm	Queries with associated returned documents, ODP Taxonomy	NDCG, MRR, MAP	Predefined categories are used instead the topics could be extracted from results set itself	Helps in the presence of ambiguous queries

Proposed Research Modules

- MODULE #1

  Enhancing Information Retrieval using Term Association Graph Representation
- MODULE #2

  Integration of Document topic model and User topic model for personalizing Information Retrieval
- MODULE #3

  Computational Intelligence for Document Re-ranking using Genetic Algorithm (GA)
- MODULE #4

  Computational Intelligence for Search query reformulation using Ant Colony Optimization (ACO)

Module #1: Introduction [1/2]

- Enhancing Information Retrieval using Term Association Graph Representation
- Approach
  - Associations among words in the documents are assessed and it is expressed in term graph model to represent the document content and the relationship among the keywords.
  - Ranking strategy which exploits term graph data structure to assess the importance of a document for the user query and thus web documents are re-ranked according to the association and similarity exists among the documents.
- Problem statements
  - How to represent document collection as term graph model?
  - How to use it for improving search results?

Module #1: Introduction [2/2]

- Methodology
  - Term graph representation
    - Pre-processing
    - Graph model construction
      - Frequent Item -set mining
      - Term graph construction
  - Ranking semantic association for Re-ranking
    - TermRank based approach (TRA)
    - Path Traversal based approach (PTA)
      - Naïve approach
      - Paired similarity document ordering
      - Personalized path selection

Module #1: Methodology [1/21]

- Term graph representation
  - Pre-processing
    - Stop words removal
    - Stemming
  - Graph model construction
    - The graph data structure reveals the important semantic relationship among the words of the document

Module #1: Methodology [2/21]

- Graph model construction
  - Frequent Item -set mining
    - Uses algorithm based on Apriori algorithm
    - After preprocessing, each document in the corpus has been stored as a transaction (item-set) in which each term/concept (item) is represented
    - First step of Apriori algorithm alone has been used to find all the subset of items that appeared more than a user specified threshold

$$Support_d = \frac{\sum_{i=1}^{n} f_d(t_i)}{\sum_{j=1}^{N}\sum_{i=1}^{n} f_{d_j}(t_i)} \qquad f_d(t_i) = \frac{term\_frequency_d(t_i)}{\underset{i}{MAX}(term\_frequency_d(t_i))} \qquad \textbf{Eq. (1.1)}$$

Module #1: Methodology [3/21]

- Frequent Item -set mining

Doc ID	Item-set	Support
54711	{Ribonuclease, catalytic, lysine, phosphate, enzymatic, ethylation}	0.12
55199	{Ribonuclease, Adx, glucocorticoids, chymotrypsin, mRNA}	0.2
62920	{Ribonuclease, anticodon, alanine, tRNA}	0.1
64711	{Cl- channels, catalytic, Monophosphate, cells}	0.072
65118	{isozyme, enzyme, aldehyde, catalytic}	0.096

Table 1.1: Frequent Item-sets and its corresponding Document support value

Module #1: Methodology [4/21]

- Term graph Model

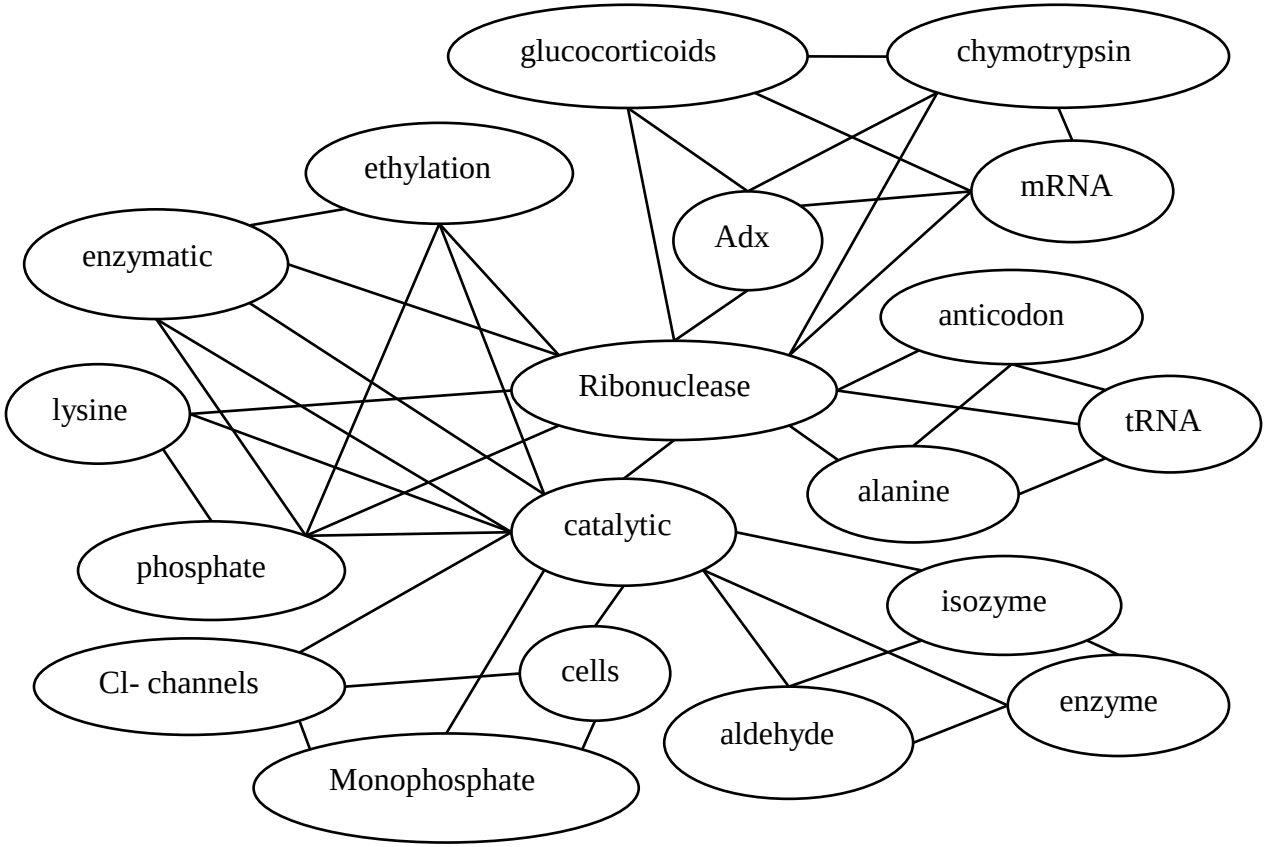


Figure 1.1. Term association graph for Table 1.1

- Term Graph for Document re-ranking : Block Diagram

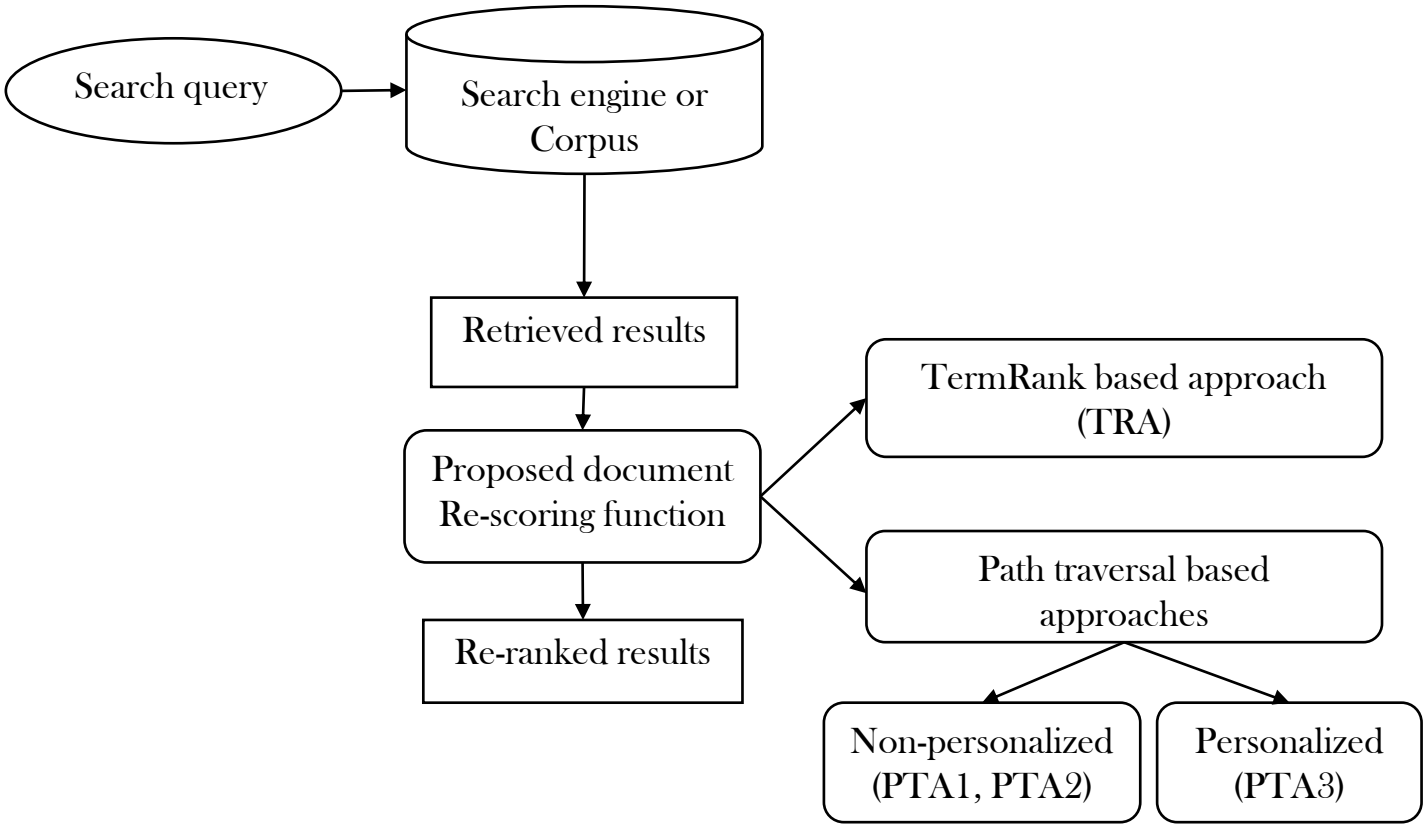


Figure 1.2. Process flow diagram

Module #1: Methodology [6/21]

- TermRank based approach (TRA)
  - The notion of PageRank is employed in this approach.
  - The PageRank scores for the nodes in the term graph is computed.
  - The rank of each term is computed

$$Rank(t_a) = c \sum_{t_b \in T_a} \frac{Rank(t_b)}{N_{t_b}} \qquad \text{Eq. (1.2)}$$

- t_a,t_b are nodes
- T_b is a set of terms t_a points to
- T_a is a set of terms that point to t_a
- $N_{t_b} = |T_b|$ is the number of links from t_a

Module #1: Methodology [7/21]

- TermRank based approach (TRA)
  - List the words that are linked with the node i.e. query term in the order of higher TermRank value [Depth may be specified]
  - Assume the top *k* terms {*k*=10,15,20,...}and identify the documents which contains these top words.
  - Order the documents according to the TermRank associated with top *k* terms in descending order.

Module #1: Methodology [8/21]

- Graph representation for Document re-ranking
  - Path traversal based approach: Introduction
    - Depth Fist Search (DFS) graph traversal is employed to re-order document.

Algorithm 1.1. DFS_Path_Traversal(Query_term q)

Input: Term_graph T_G

Output: dfs_paths $p_1, p_2, p_3, \ldots$

```
visit q;
for each neighbor_term_node w of q
    if w has not been visited then
        dfs_Path_Traversal(w);
```

Module #1: Methodology [9/21]

- Graph representation for Document re-ranking
  - Path traversal based approach: Introduction
    - As the result of DFS algorithm, the different dfs_paths $p_1, p_2, p_3, \ldots$ are returned
    - Find an optimized path from these paths

dfs_ paths	*Term/Document$_i$ (T/D_i)*					
p_1	T_2/D_{11}	T_{26}/D_1D_{48}	$T_3/D_9\ D_{62}$	$T_{37}/\ D_3D_5D_{32}$	$T_{29}/\ D_1D_6D_{22}$	T_9/D_9D_{11}
p_2	$T_6/D_{11}D_1$	$T_7/\ D_{14}D_{23}$	T_{13}/D_9D_7	T_{31}/D_6D_{17}	$T_{23}/D_6D_{41}D_1$	T_{19}/D_{41}
p_3	T_{21}/D_1	T_{17}/D_{11}	T_2/D_{11}	T_{12}/D_{37}	T_{21}/D_{17}	$T_{59}/D_{22}D_5$
p_4	T_{61}/D_{16}	$T_{29}/\ D_1D_6D_{22}$	T_{14}/D_{11}	$T_7/D_{14}D_{23}$	T_2/D_{52}	T_2/D_{11}
p_5	T_{34}/D_{71}	T_{43}/D_{31}	$T_4/D_{58}D_{17}$	T_{11}/D_{16}	$T_8/D_{48}D_{12}$	T_{30}/D_1
p_6	T_{13}/D_9D_7	$T_9/D_{57}D_{41}$	T_{44}/D_9	T_{71}/D_{11}	$T_8/\ D_{48}D_{12}$	$T_{37}/\ D_3D_5D_{32}$

Table 1.2: Possible *dfs_paths* for the Query_term T_1 with depth = 6

Module #1: Methodology [10/21]

- Path traversal based approach
  - PTA 1: Naïve approach
    - Each dfs_path is a possible set of relevant documents based on the link structure exists among the terms in Term_graph T_G.
    - The solution path can be chosen in such a way that its total cost is as higher as other possible paths.
    - The cost is defined as the *support* between two terms in T_G.
    - Example: if the cumulative support of paths ($p_{i=1,2,...,6}$) shown in Table 2 [Slide No. 29] are 0.61, 0.55, 0.72, 0.38, 0.24, and 0.32 respectively, then the sequence of relevance documents are chosen from path p_3 i.e. D_{11}, D_1, D_{37}, D_{17}, D_{22}, and D_5. D_{11} will be the top ranked document.

Module #1: Methodology [11/21]

- Path traversal based approach
  - PTA 2: Paired similarity document ordering
    - To find the closest term for the query word from the keywords extracted from web pages retrieved, using Wu and Palmer similarity measure.

$$sim(T_1, T_2) = 2 \times \frac{depth(LCS)}{depth(T_1) + depth(T_2)} \qquad \text{Eq. (1.3)}$$

    - where, T_1 and T_2 denote the term nodes in a term graph T_G to be compared, LCS denote the Least Common Sub-Sumer of T_1 and T_2, and $depth(T)$ is the shortest distance from the query node q to a node T on T_G.

- Path traversal based approach
  - PTA 2: Paired similarity document ordering
    - The possible *dfs_paths* shown in Table 1.2 [Slide No. 29] are ranked according to the Eq. (1.3) by computing the similarity between all pairs of T_9, T_{19}, T_{59}, T_2, T_{30}, T_{37}. For example, $sim(T_9, T_{19})$ is computed as $2*(5/6+6)=0.83$ thereby the depth based similarity matrix D_{sim} is constructed as shown in Table 1.3.

sim(T,T)	T_9	T_{19}	T_{59}	T_2	T_{30}	T_{37}
T_9	1	0.83	0.66	0	0.17	0.33
T_{19}	0.83	1	0.5	0.83	0.17	0
T_{59}	0.66	0.5	1	0.66	0.5	0.17
T_2	0	0.83	0.66	1	0.33	0.83
T_{30}	0.17	0.17	0.5	0.33	1	0
T_{37}	0.33	0	0.17	0.83	0	1

Table 1.3: Depth based Similarity Matrix

Module #1: Methodology [13/21]

- Path traversal based approach
  - PTA 2: Paired similarity document ordering
    - Then the documents are re-ranked according to the paired similarity obtained using the algorithm 1.2.

Algorithm 1.2. Paired_Similarity_Re-ranking (Documents in *dfs_path*, Query_term *q*)

Input: Depth based Similarity matrix DS_{sim}

Output: Re-ordered documents

(1) *k*=number of pairs on *N* terms

(2) Map the upper/lower triangle of the symmetric matrix DS_{sim} into an array *A*

(3) Sort the array *A* in descending order

(4) Compare the DS_{sim} with sorted array *A* in order to identify the indices of the pair of words possesses higher depth based similarity

(1) From the list of term indices with duplication, Generate ranked list of terms by removing duplicate entry of term indices.

(1) Order the documents according to the term sequence in the ranked list

(2) Display the re-ranked list of documents

- Path traversal based approach
  - PTA 2: Paired similarity document ordering
    - For the depth based similarity matrix shown in Table 1.3, the Paired_Similarity_Re-ranking algorithm prepare the terms in the following sequence as $T_9, T_{19}, T_2, T_{37}, T_{59}, T_{30}$.

T_9	T_{19}	T_2	T_{37}	T_{59}	T_{30}
D_9, D_{11}	D_{41}	D_{11}	D_3, D_5, D_{32}	D_{22}, D_5	D_9, D_{62}

Table 1.4. Ranked list of terms and its associaed documents

    - Accordingly, documents possesses these terms has been re-ranked as shown in Table 1.4.

Module #1: Methodology [15/21]

- Path traversal based approach
  - PTA 3: Personalized path selection
    - The personalization concept has been adopted in this method.
    - **Steps:**
      - Document Topics
      - User search interest value table
      - Session identification
      - Search context weight

Module #1: Methodology [16/21]

- PTA 3: Personalized path selection

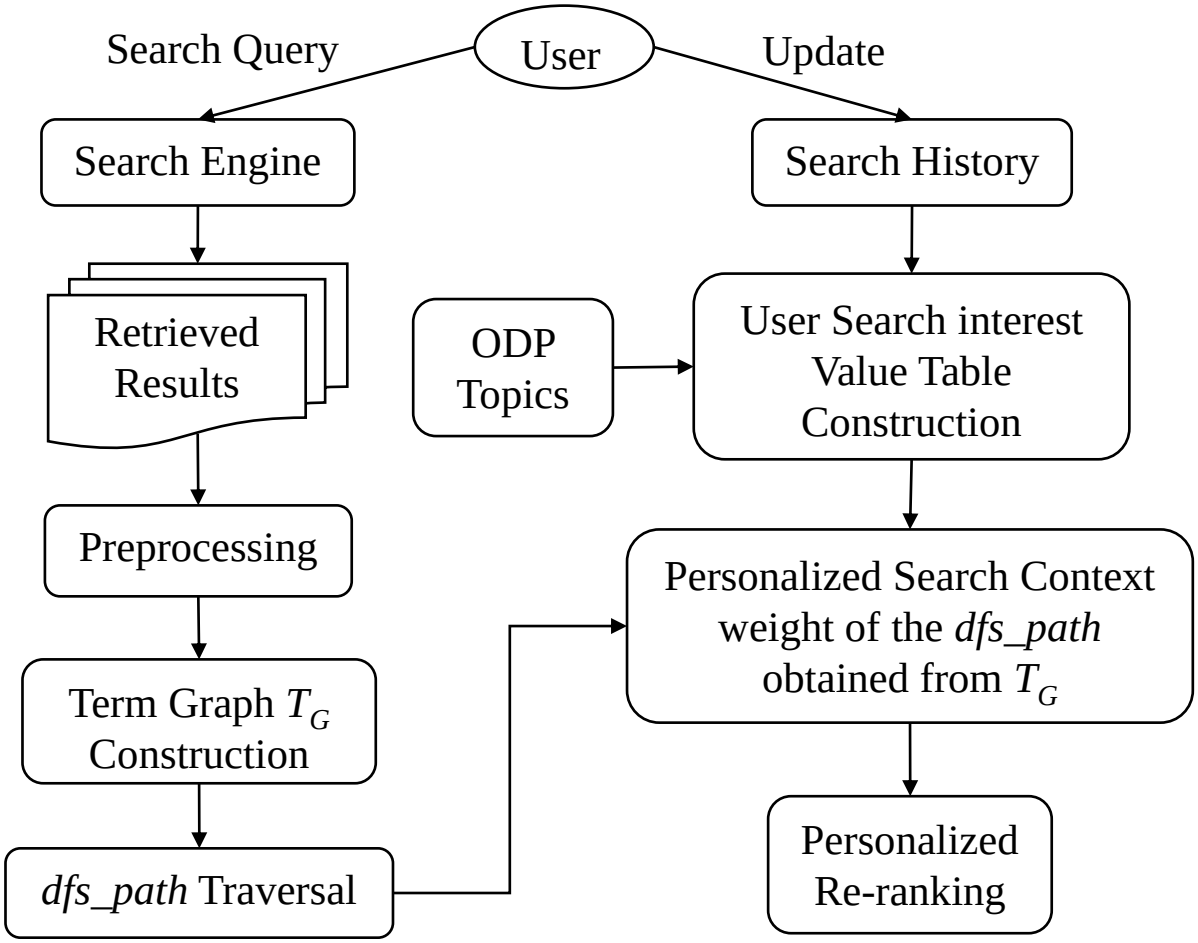


Figure 1.3. Architecture of proposed Personalization model

Module #1: Methodology [17/21]

- PTA 3: Personalized path selection
  - Document Topics
    - The user's dynamic search interests on various topics are captured from their web browser search history
    - The topics are trained on the Open Directory Project (ODP) corpus

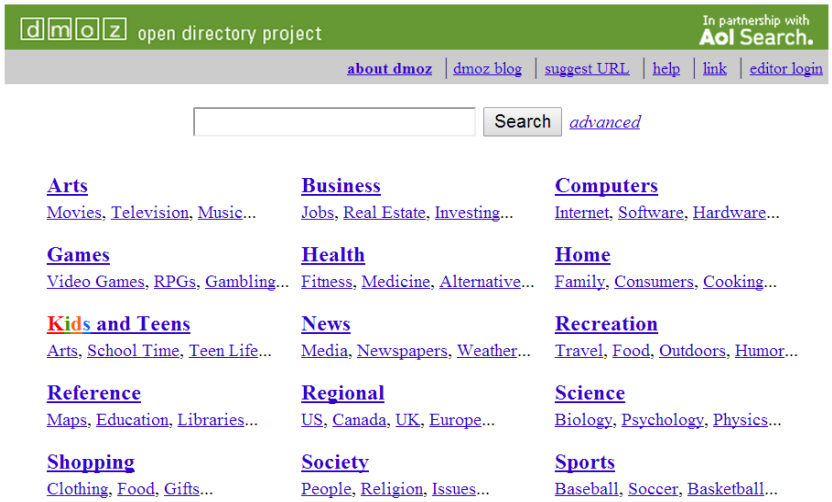


Figure 1.4. ODP main page [http://www.dmoz.org/]

- PTA 3: Personalized path selection
  - User search interest value table
    - The weight value of the user's interest topics is maintained in a Table 1.5.
    - These values have been periodically updated in order to preserve/maintain user's current search intent.

Session ID	Software	Algorithms	Healthcare	Sports	Movies	Music
S1	0.312	0.671	0.090	0.232	0.001	0.030
S2	0.134	0.245	0.322	0.301	0.023	0.010
S3	0.472	0.107	0.024	0.149	0.200	0.174
S4	0.048	0.110	0.261	0.642	0.098	0.145
S5	0.076	0.093	0.047	0.184	0.594	0.611

Table 1.5. User search interest value table

- PTA 3: Personalized path selection
  - Session Identification
    - To recognize the end of a previous session and the beginning of the current session, the Kullback-Leibler Divergence (*KLD*) has been employed
    - *KLD* compares the probability distribution of two terms in a query over the set of web pages in the collection
    - This difference measures the similarity between two queries, thus helps in separating two search sessions

$$KLD(q_1 \parallel q_2) = \sum_{t \in D_1 \cap D_2} P(t) \log \frac{P(t \in D_1)}{P(t \in D_2)} \qquad \textbf{Eq. (1.4)}$$

    - where $P(t)$ is the probability of term t in documents retrieved for query q_1 and q_2 i.e. D_1 and D_2. D_1 and D_2 is the set of documents retrieved for q_1 and q_2 respectively.

- PTA 3: Personalized path selection
  - Personalized Search Context weight (PSC_{weight})
    - Search context weight is a semantic metric used to determine the relevancy based on user preference
    - Using the identified search context, it is more likely to re-rank the *dfs_paths* according to its relevance with a user's search interest

$$PSC_{weight} = \frac{1}{|t|}\left(\left(\sum_{i=1}^{\#topics}\left(siv_i\left(\sum t \in T_i\right)\right)\right)\times\left(1-\frac{\#t \notin T}{|t|}\right)\right) \qquad \textbf{Eq. (1.5)}$$

    - where, $|t|$ is the total number of terms in path including the query term. T is the set of user interested topics. siv_i is the search interest value of the i^{th} topic of specified user. This value is taken from user search interest value table.

- PTA 3: Personalized path selection

Algorithm 1.3. Personalized_Similarity_Re-ranking (Documents, Query_term *q*)

Input: *dfs_paths*, Depth based Similarity matrix DS_{sim}

Output: Re-ordered documents

(1) siv_i=0;
(2) *count*=0;
(3) MAX=0;
(4) for each dfs_path_i
(5) for each term *t* in *dfs_path* ∈ topic *T* in ODP
(6) $siv_i = siv_i + siv_i(t)$;
(7) for each term *t* in *dfs_path* ∉ topic *T* in ODP
(8) *count* = *count*+1;
(9) PSC_{weight}=(siv_i × (1-(count/|t|)))/|t|
(10) if PSC_{weight} > MAX then
(11) MAX= PSC_{weight}
(12) Order the documents that possesses the terms in dfs_path_{MAX}

- Experimental Dataset Description

Document Corpus	Usage for Evaluation	# of documents	# of queries	Avg. doc. length	Avg. doc. Length after pre-processing
Real dataset (Results from traditional search engine)	Personalized model & Non-personalized models	Top 50 results for a query	100	34	21
Synthetic dataset (OHSUMED)	Non-personalized models	348,566	106	210	64

Table 1.5. Statistics about the Dataset

- Evaluation Setup and Metrics

Datasets used / Proposed Approaches	Real Dataset	Synthetic dataset
Personalized Scheme (PTA 3) with subjective evaluation	Variation in user search intents, Information Richness, and Average Information Richness	---
Non-personalized Schemes (TRA, PTA1, & PTA2) with objective evaluation	Accuracy in terms of Precision at various search results positions, Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG)	Accuracy in terms of precision at various search results positions, MAP, and NDCG

Table 1.6. Evaluation Design

- Subjective Evaluation
  - Diversity: Given a set of documents retrieved R_m, Diversity is defined as Div(R_m) to denote the number of different topics contained in R.
  - Information Richness: Given a document collection $D=\{d_1,...,d_n\}$, Information Richness is defined as InfoRich (d_i) to denote richness of information contained in the document d_i with respect to the entire collection D.

$$InfoRich(R_m) = \frac{1}{Div(R_m)} \sum_{k=1}^{Div(R_m)} \frac{1}{N_k} \sum_{i=1}^{N_k} InfoRich(d_k^i) \qquad \text{Eq. (1.6)}$$

where d^i_k represent one of N_k documents associated with the k^{th} topic. The average information richness is defined as information richness of a set of documents

- Objective Evaluation metrics
  - Precision (P) $$P@k = \frac{\#of\_relevant\_doc\_retrieved\_among\_k}{k}$$ **Eq. (1.7)**
  - Recall (R) $$R@k = \frac{\#of\_relevant\_doc\_retrieved\_among\_k}{total\#of\_relevant\_documents}$$ **Eq. (1.8)**
  - Mean Average Precision (MAP)

$$MAP = \frac{\sum_{q=1}^{|Q|} AP(q)}{|Q|} \qquad AP = \frac{1}{R}\sum_{k=1}^{R}((P@k).(rel(k))) \qquad \textbf{Eq. (1.9)}$$

  - Mean Reciprocal Rank (MRR) $$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$ **Eq. (1.10)**
  - Normalized Discounted Cumulative Gain (NDCG)

$$NDCG_K = \frac{DCG_K}{IDCG_K} \qquad DCG_K = \sum_{i=1}^{K}\frac{2^{r_i}-1}{\log_2(i+1)} \qquad \textbf{Eq. (1.11)}$$

- Baseline approaches[1/2]
  - K-Means algorithm

Algorithm 1.4. *K*-means (D_n)

Input: A set of Documents $\{D_{j=1,\ldots,n}\}$, Number of representatives *K*

Output: Clusters $C_{i=1,\ldots,k}$

1. randomly Select *K* documents as the initial centroids i.e. cluster centers;
2. for each document D_j do
   assign its membership to the cluster C_i that has the largest *similarity*(D_j,C_i);
1. calculate the new centroids for each cluster by finding the most centrally located document in each cluster;
2. repeat steps 2 and 3 until no reassignments of the centroids takes place;
3. return;

  - Affinity graph ranking

$$affinity(d_i, d_j) = \frac{d_i . d_j}{\|d_i\|} \qquad \textbf{Eq. (1.12)}$$

    - Each link in the graph has been assigned a weight indicating the similarity relationship between the corresponding document pair.

- Baseline approaches[2/2]
  - Term Graph Model
    - Text documents are modeled as a graph whose vertices represent words, and whose edges denote meaningful statistical (e.g. co-occurrence) or linguistic (e.g. grammatical) relationship between the words.
  - Concept Graph Model
    - Graph model has been adapted to a concept representation of documents which captures the dependencies between concepts found in document text.
  - BM25 Model

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i).\frac{tf_i.(k_1+1)}{tf_i + k_1.\left(1-b+b.\frac{|D|}{avgdl}\right)} \qquad \textbf{Eq. (1.13)}$$

Module #1: Experimental Results [1/10]

- Non-Personalized Evaluation on Real dataset - **Precision**

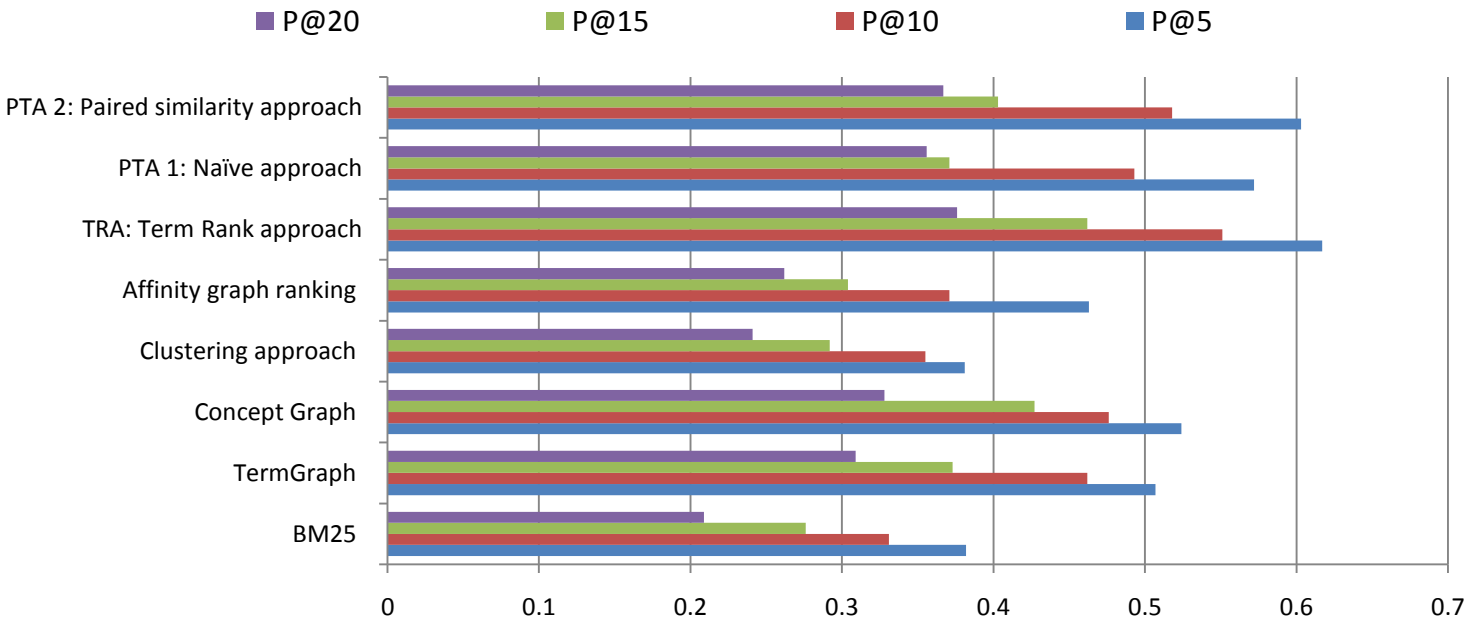


Figure 1.5. Precision at *k* search result positions for 30 queries (*k*=5, 10, 15, 20)

Module #1: Experimental Results [2/10]

- Non-Personalized Evaluation on Real dataset – **Statistical Significance Test**

Vs. TermGraph (TG)	Paired t-test (*p*-value)	**Vs. Concept Graph (CG)**	Paired t-test (*p*-value)	**Vs. Affinity Graph (AG)**	Paired t-test (*p*-value)
TRA	**0.002****	TRA	**0.047***	TRA	**0.001****
PTA1	0.619	PTA1	0.180	PTA1	**0.004****
PTA2	**0.008****	PTA2	0.825	PTA2	**0.002****

Table 1.7. Summary of Significance test results on Real dataset

Module #1: Experimental Results [3/10]

- Non-Personalized Evaluation on Real dataset – Mean Reciprocal Rank **(MRR)** & Mean Average Precision **(MAP)**

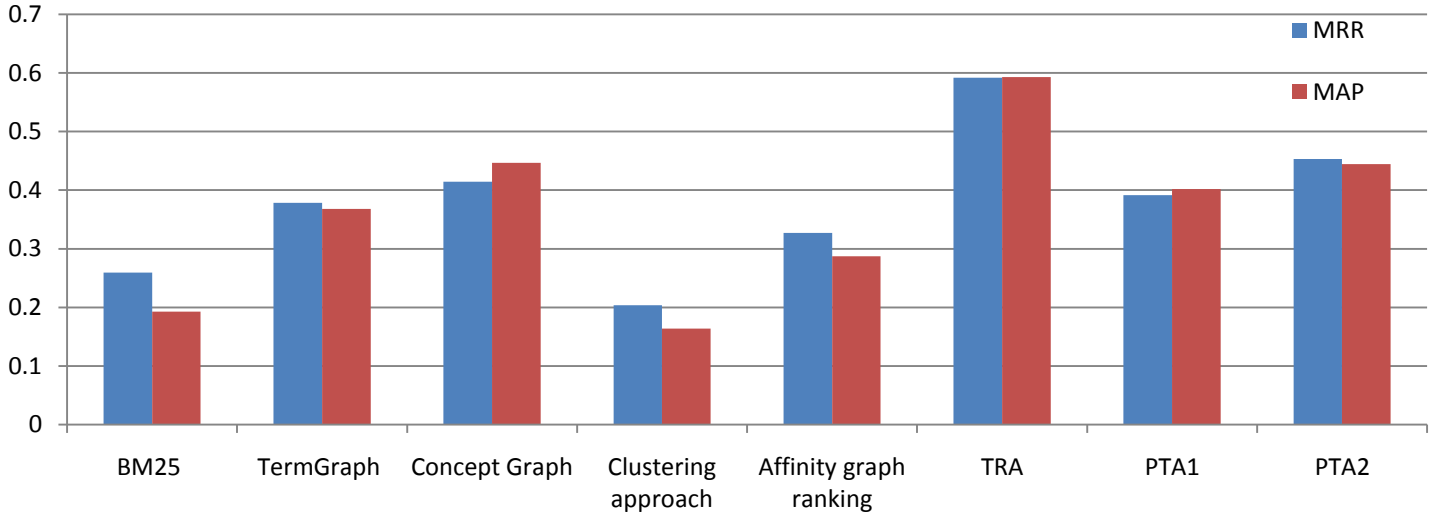


Figure 1.6. MRR for 10 queries and MAP for 10 queries at top 5 results over real dataset

Module #1: Experimental Results [4/10]

- Non-Personalized Evaluation on Real dataset – Normalized Discounted Cumulative Gain (**NDCG**)

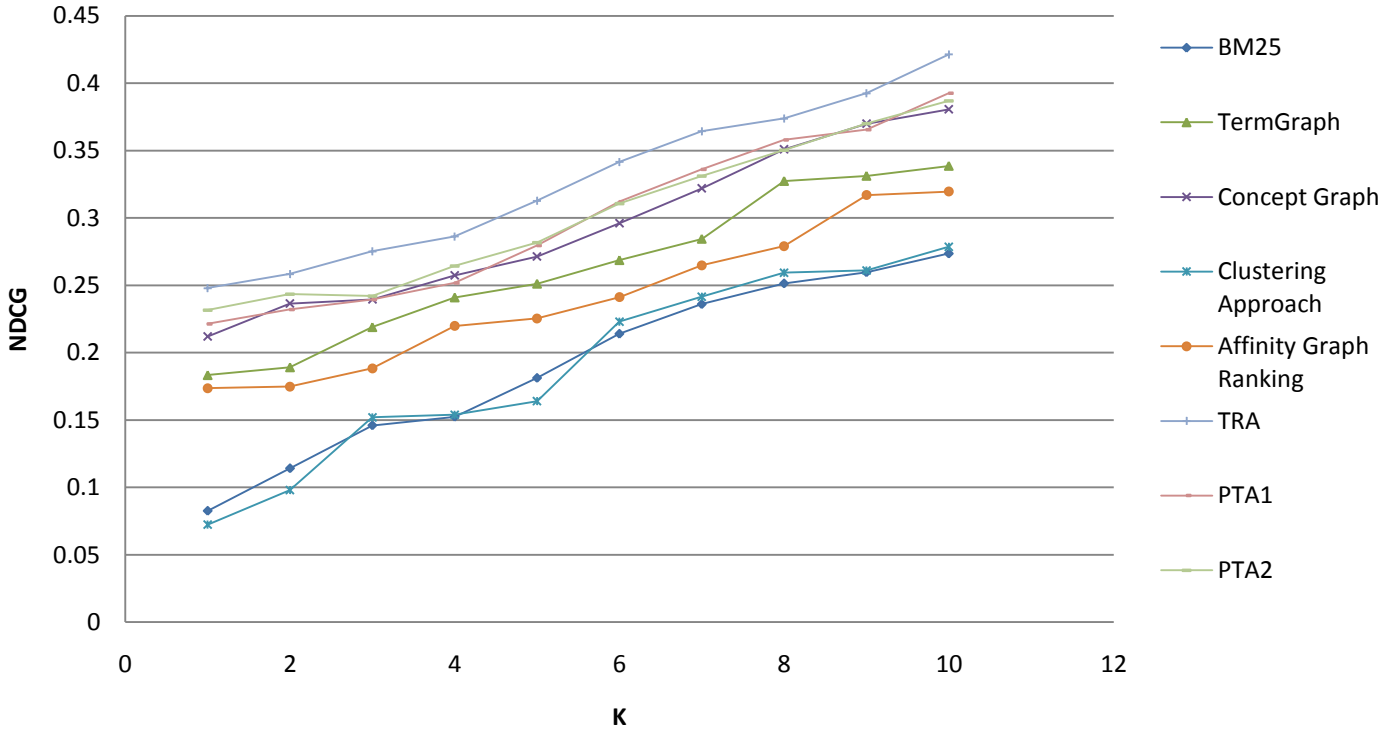


Figure 1.7. NDCG at K obtained for 20 queries

Module #1: Experimental Results [5/10]

- Non-Personalized Evaluation on Real dataset –**NDCG Improvement**

Result Positions	**NDCG Improvement over TG in %**			**NDCG Improvement over CG in %**			**NDCG Improvement over AG in %**		
	TRA	PTA1	PTA2	TRA	PTA1	PTA2	TRA	PTA1	PTA2
1	35.1690	20.6652	26.2814	16.9340	4.3868	9.2453	42.7995	27.4770	33.4101
2	36.5222	22.6216	28.6998	9.2640	-1.8613	3.0034	47.6844	32.6472	39.2224
3	25.6164	9.3151	10.5479	14.8164	-0.0835	1.0434	46.0967	27.1375	28.5714
4	18.7552	4.4398	9.6680	11.2320	-2.1764	2.7206	30.2093	14.5132	20.2457
5	24.4825	11.2261	12.1417	15.2174	2.9477	3.7951	38.7311	23.9574	24.9778
6	27.1306	16.1146	15.5936	15.3664	5.3698	4.8970	41.6252	29.3532	28.7728
7	28.0492	18.1019	16.3445	13.1718	4.3802	2.8270	37.5755	26.8882	25.0000
8	14.1417	9.2853	6.9945	6.4976	1.9664	-0.1710	33.8947	28.1978	25.5106
9	18.5085	10.3261	11.7452	6.1385	-1.1898	0.0811	23.8561	15.3045	16.7876
10	24.4241	15.9480	14.2646	10.6936	3.1529	1.6553	31.8210	22.8411	21.0576

Table 1.8. NDCG Improvement gain in %

Module #1: Experimental Results [6/10]

- Non-Personalized Evaluation on Synthetic dataset – **Precision** and **MAP**

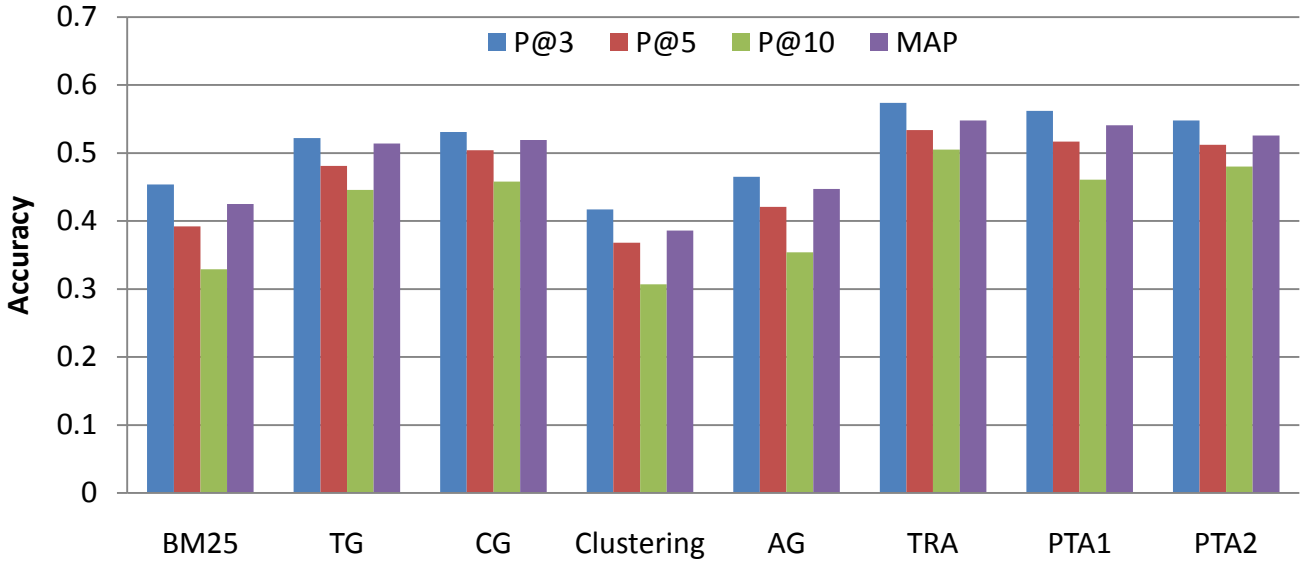


Figure 1.8. Precision and MAP obtained for 50 queries

Module #1: Experimental Results [7/10]

- Non-Personalized Evaluation on Synthetic dataset – **NDCG**

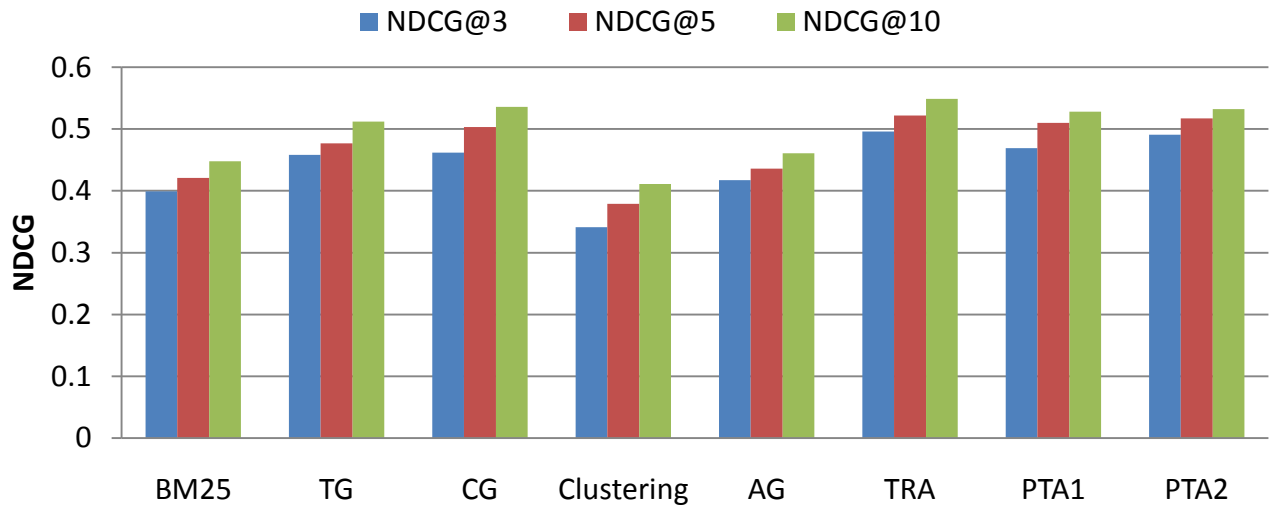


Figure 1.9. NDCG at K = 3, 5, and 10 obtained for 50 queries

Module #1: Experimental Results [8/10]

- Personalized Evaluation on Real dataset – Assessed by Information Richness (**InfoRich**)

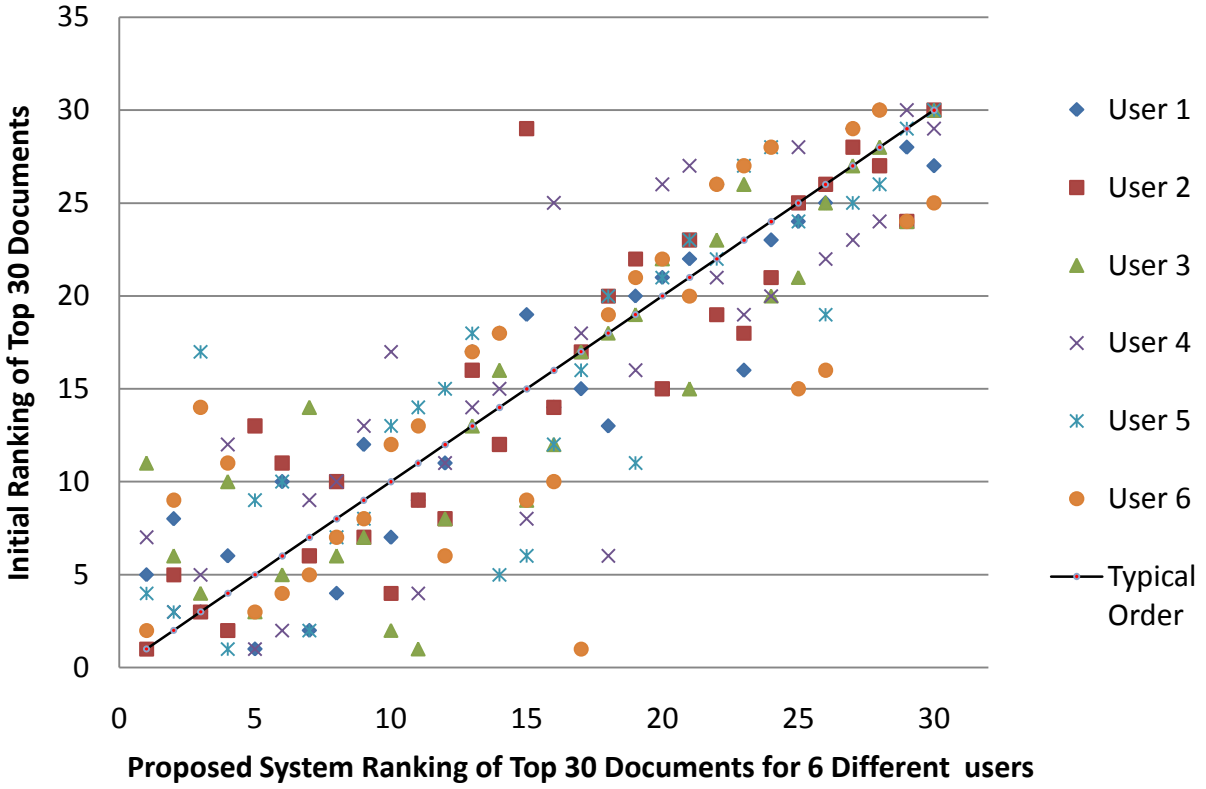


Figure 1.10. Initial ranking Vs. Personalized ranking

Module #1: Experimental Results [9/10]

- Personalized Evaluation on Real dataset – **InfoRich**

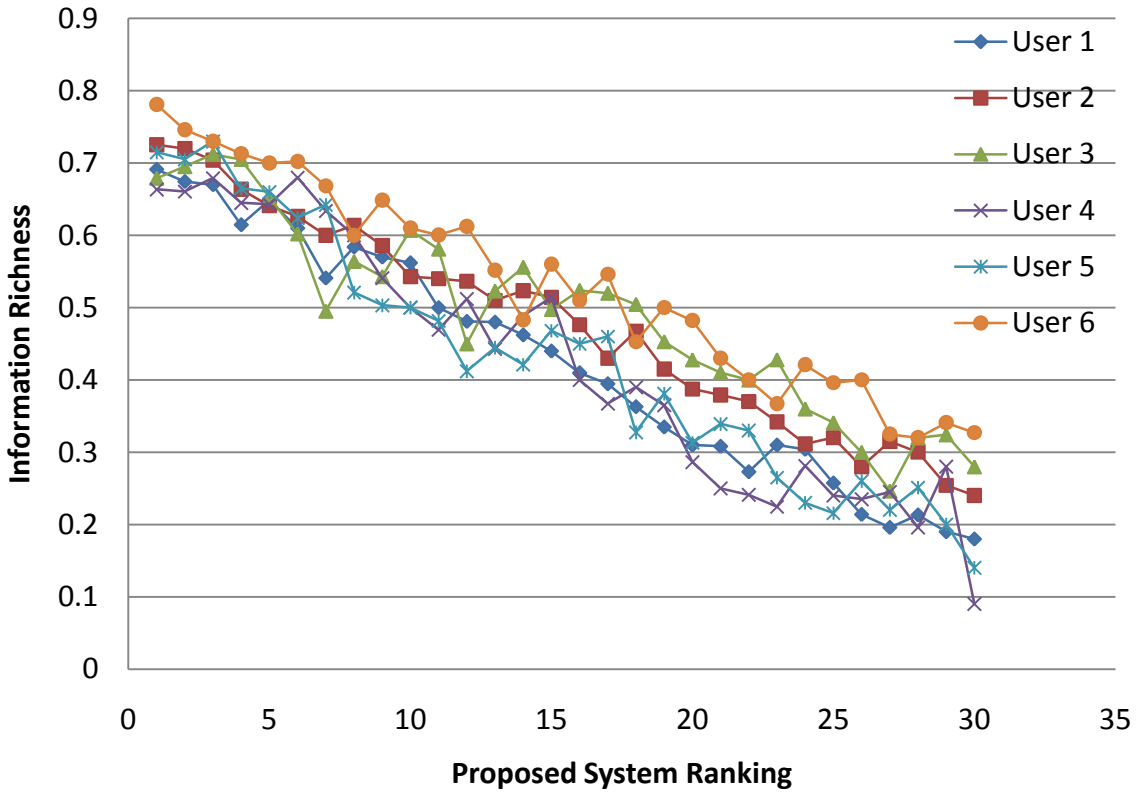


Figure 1.11. InfoRich obtained by PTA3 within top 30 search results

Module #1: Experimental Results [10/10]

- Personalized Evaluation on Real dataset – **InfoRich** improvement

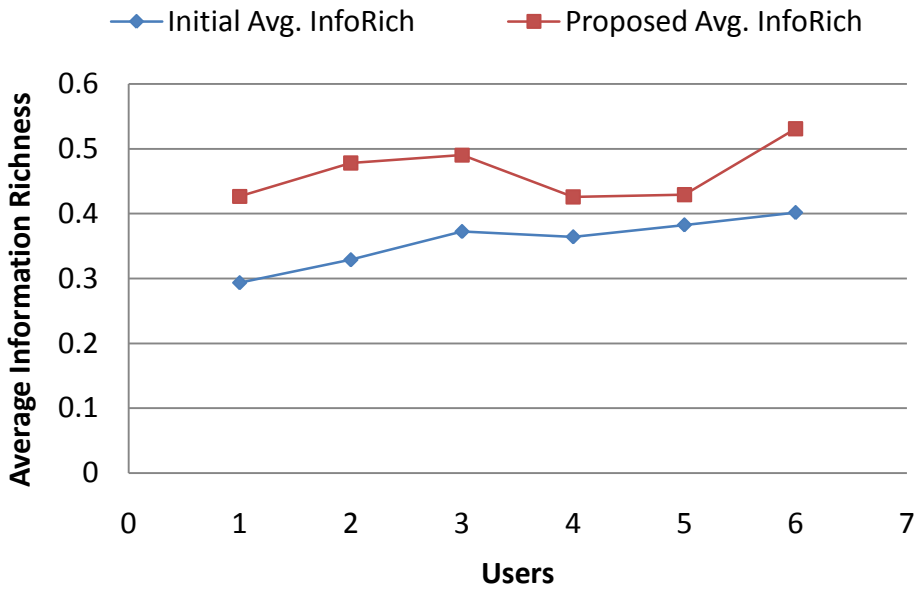


Figure 1.12. InfoRich improvement by PTA3 within top 30 search results

Module #1: Summary

- Employ term association graph model
- Suggested different methods to enhance the document retrieval task
- Captures hidden semantic association

Module #2: Introduction [1/2]

- Integration of Document topic model and User topic model for personalizing Information Retrieval
- Approach
  - Uses past Search history for user search interest modeling
  - Re-ranking (or Re-scoring) search results based on the model
- Problem statements
  - How to model and represent past search contexts?
  - How to use it for improving search results?

Module #2: Introduction [2/2]

- Methodology
  - User search context modeling
    - User profile modeling
    - Learning user interested topic
    - Finding document topic
  - Personalized Re-ranking process
    - Exploiting user interest profile model
    - Computing personalized score for document using user model
    - Generating personalized result set by re-ranking

Module #2: Methodology [1/6]

- User search context modeling
  - User profile model $\theta_u = UP_{w_i}$

$$UP_{w_i \in History(D)} = P(w_i) = \frac{tf_{w_i,D}}{\sum_{w_i \in D} tf_{w_i,D}} \quad \textbf{Eq. (2.1)}$$

  - Learning User interested topic

w_i	P(w_i)
Computer	0.012
Datastructure	0.02
Programming	0.0011
Data	0.0024
Instruction	0.001
Algorithm	0.032
Analysis	0.004

Table 2.1. Sample user profile representation

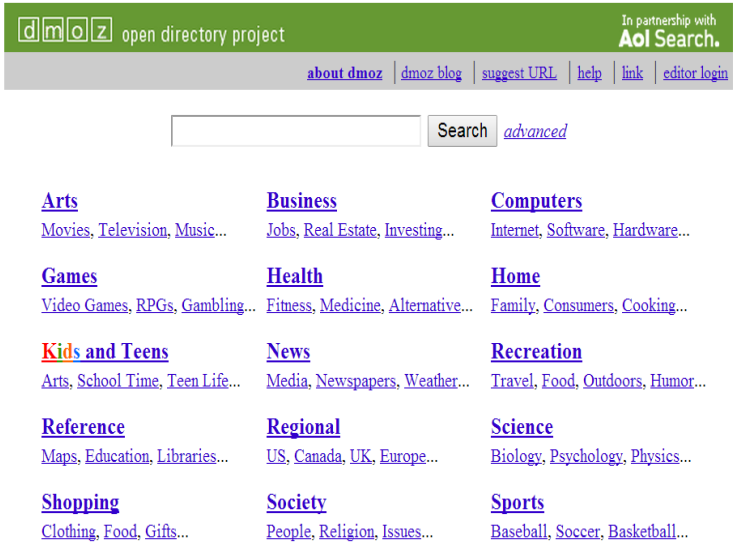


Figure 2.1. ODP main page [http://www.dmoz.org/]

Module #2: Methodology [2/6]

- User Search context modeling
  - Learning user interested topic

Algorithm 2.1. training_user_topic(q,d, θ_u)

Input: query q in search log, doc d clicked for q by user u, user_profile θ_u)

Output: Topics that are of interest for u

for each query q in user's search history

for each document d clicked by the user

for each topic T in the ODP category

compute T_u by $P(T \mid \theta_u, q) = \delta(P(T \mid q)) + (1-\delta)(P(T \mid \theta_u)P(q \mid T)$

return (T_u)

Module #2: Methodology [3/6]

- User Search context modeling
  - Finding document topic

Algorithm 2.2. finding_document_topic(q,D)

Input: initial query q, retrieved documents D

Output: Topic of the documents retrieved

for each document $d_i \epsilon D$ retrieved for the query q

for each topic T in the ODP category

compute T_d by $P(T \mid d) = P(t_i \in d \mid T)P(T)$

return (T_d)

Module #2: Methodology [4/6]

- Personalized Re-ranking process
  - Exploiting user interest profile model

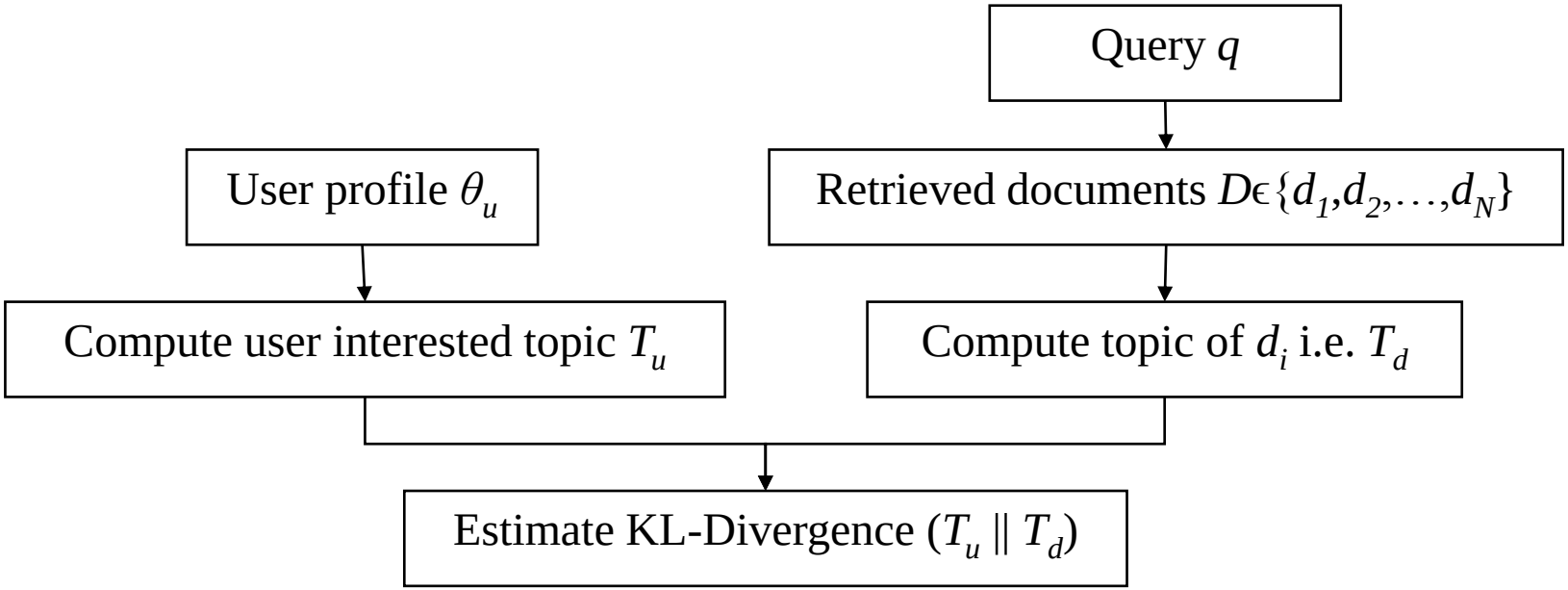


Figure 2.2. User interested topic (T_u) vs. Document topic (T_d)

Module #2: Methodology [5/6]

- Personalized Re-ranking process
  - Exploiting user interest profile model
  - A document about topic T_d is assumed relevant to a user looking for topic T_u if the following two points are met:
    - (1) $$KL - D(T_d \parallel T_u) = \sum_{t \in D \cap U} P(T_d(t)) \log \frac{P(T_d(t))}{P(T_u(t))}$$ **Eq. (2.2)**
    - (2) $$P(Q \mid T_d, T_u) = \prod_{q_i \in Q} \alpha P(q_i \mid T_d) + (1-\alpha) P(q_i \mid T_u)$$ **Eq. (2.3)**

Module #2: Methodology [6/6]

- Personalized Re-ranking process
  - Computing personalized score for document using user model

$$P(D \mid Q, \theta_u) = \frac{P(D \mid \theta_u) P(Q \mid D, \theta_u)}{P(Q \mid \theta_u)} \qquad \textbf{Eq. (2.4)}$$

$$P(Q \mid D, \theta_u) = P(Q \mid T_d, T_u) + \prod_{q_i \in Q} (\beta P(q_i \mid \theta_u) + (1-\beta) P(q_i \mid D)) \qquad \textbf{Eq. (2.5)}$$

  - Generating personalized result set by re-ranking
    - The documents are then scored based on the probability $P(Q \mid D, \theta_u)$
    - Result set is arranged based on descending order of the personalized score

Module #2: Experimental Evaluation [1/7]

- Experimental Design

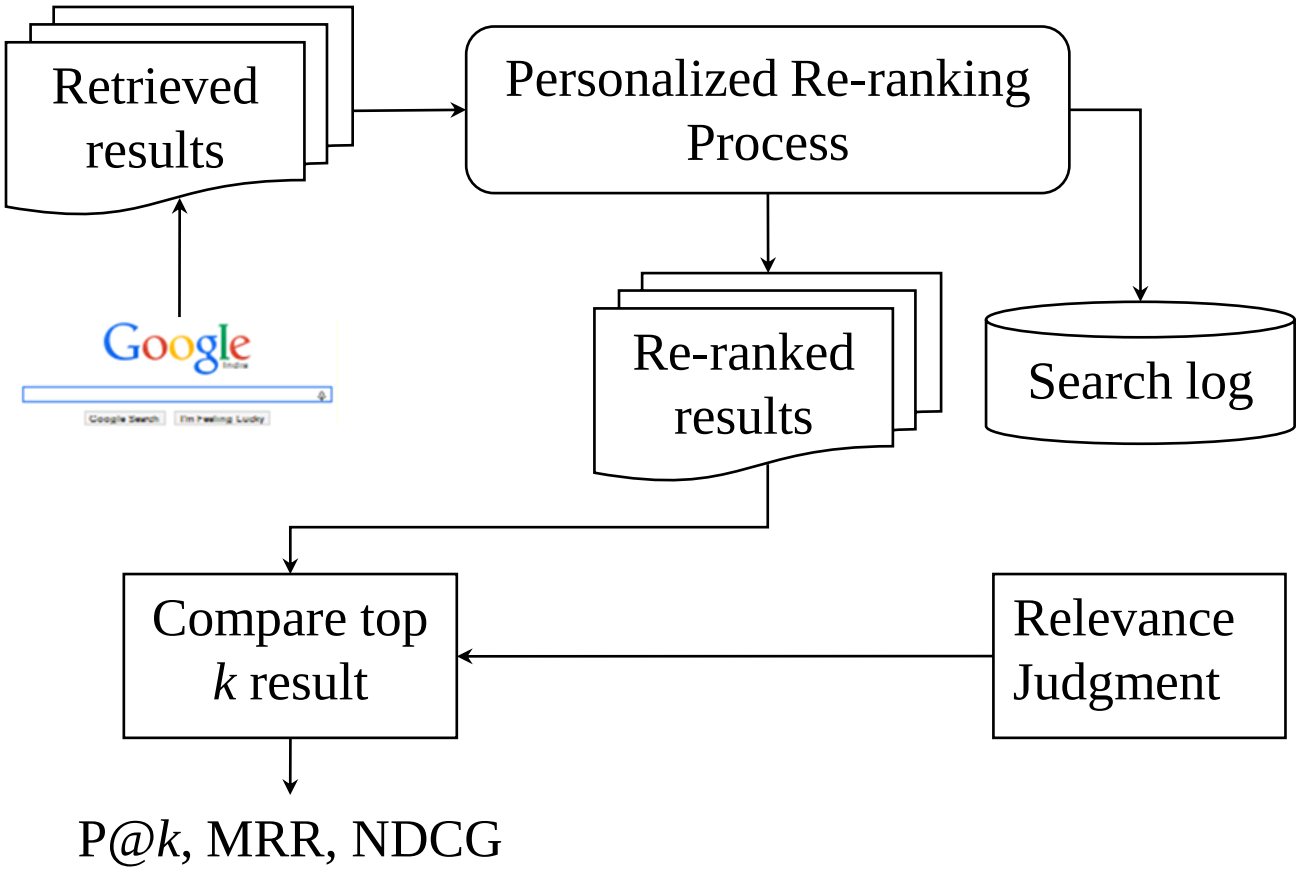


Figure 2.3. Experimental Setup for Evaluation

- Baseline approaches
  - Best Matching (BM25)

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i).\frac{tf_i.(k_1+1)}{tf_i + k_1.\left(1-b+b.\frac{|D|}{avgdl}\right)} \qquad \textbf{Eq. (2.6)}$$

$$IDF(q_i) = \log \frac{N - df_i + 0.5}{df_i + 0.5}$$

  - Rocchio Algorithm

$$Score(D,Q) = \sum_{w \in Q} \left(\frac{tf_{w,Q}}{|Q|} + \frac{tf_{w,UP}}{|UP|} \right).\frac{tf_{w,D}}{|D|} \qquad \textbf{Eq. (2.7)}$$

- Baseline approaches
  - Language Model (LM) approach

$$P(Q \mid D, UP) = \prod_{q_i \in Q} \alpha P(q_i \mid UP) + (1-\alpha) P(q_i \mid D) \qquad \text{Eq. (2.8)}$$

  - Query Language Model approach

$$P(q_i \mid UP) = \beta P(q_i \mid UP) + (1-\beta) P(q_i \mid QueryLog) \qquad \text{Eq. (2.9)}$$

- Evaluation metrics
  - Precision (P)

$$P@k = \frac{\#of\_relevant\_doc\_retrieved\_among\_k}{k} \quad \textbf{Eq. (2.10)}$$

  - Mean Reciprocal Rank (MRR)

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \quad \textbf{Eq. (2.11)}$$

  - Normalized Discounted Cumulative Gain (NDCG)

$$DCG_K = \sum_{i=1}^{K}\frac{2^{r_i}-1}{\log_2(i+1)} \qquad NDCG_K = \frac{DCG_K}{IDCG_K} \quad \textbf{Eq. (2.12)}$$

- Experimental Dataset Description
  - AOL Search query log

Number of lines of data	36,389,567
Number of instances of new queries (with or without click-through data)	21,011,340
Number of requests for "next page" of results	7,887,022
Number of user click-through data	19,442,629
Number of queries without user click-through data	16,946,938
Number of unique queries	10,154,742
Number of users log	657,426

Table 2.2. Statistics about AOL Search log

- Experimental Dataset Description
  - Real Dataset

Users	# of Queries	Total # of relevant documents	Average # of relevant documents
User 1	43	225	5.23
User 2	39	125	3.21
User 3	63	295	4.68
User 4	62	188	3.03
User 5	37	190	5.14
User 6	28	91	3.25
User 7	45	173	3.84
User 8	31	96	3.10
User 9	51	240	4.71
User 10	39	128	3.28

Table 2.3. Sample Real dataset Statistics of 10 Users

Module #2: Experimental Evaluation [7/7]

- Parameter Tuning
  - Learning α and β parameters

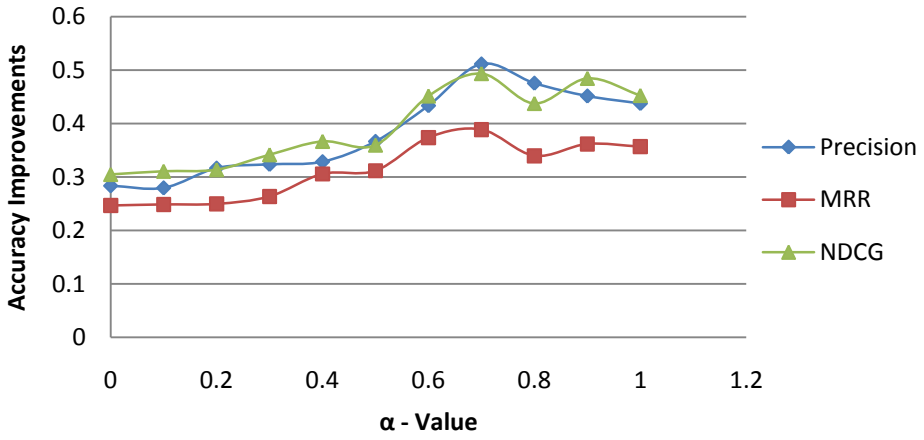


Figure 2.4.*α* Parameter tuning on Eq(2.3) for 5 queries at top 10 search results

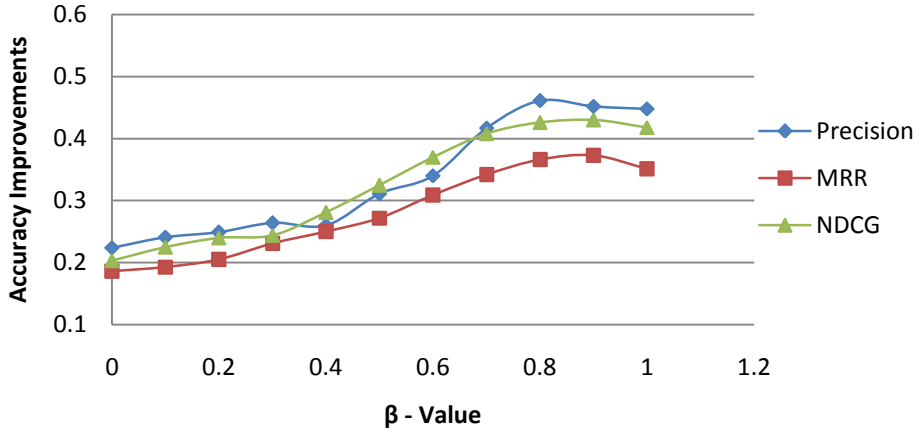


Figure 2.5. *β* Parameter tuning on Eq(2.5) for 5 queries at top 10 search results

Module #2: Experimental Results [1/3]

- Evaluation on Real Dataset
  - MRR and Precision

Method	MRR@5	P@5	P@10
Best Matching (BM25)	0.239	0.3607	0.2914
Rocchio algorithm	0.305	0.4322	0.3783
Document Language Model approach (LMD)	0.332	0.473	0.4145
Query Language Model approach (LMQ)	0.371	0.5118	0.447
Proposed integrated topic model and user model approach (TU)	0.428	0.5605	0.4926

Table 2.4 MRR and Precision at top-*k* results for 10 queries

Module #2: Experimental Results [2/3]

- Evaluation on Real Dataset
  - Precision Recall and NDCG

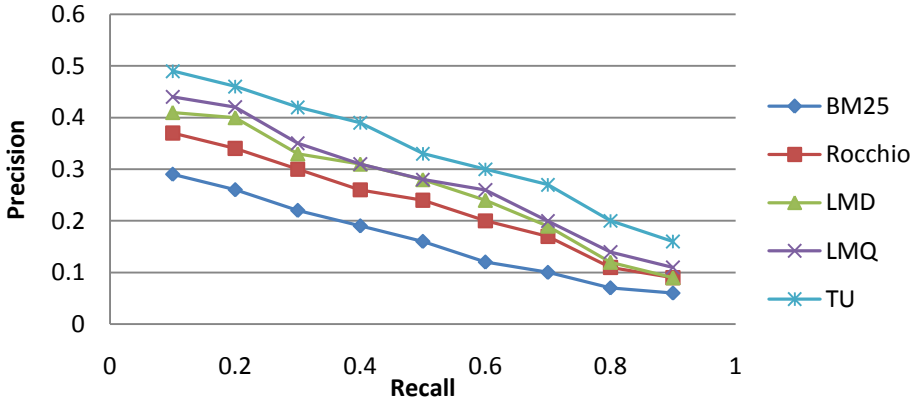


Figure 2.6 Precision vs. Recall obtained for 10 queries

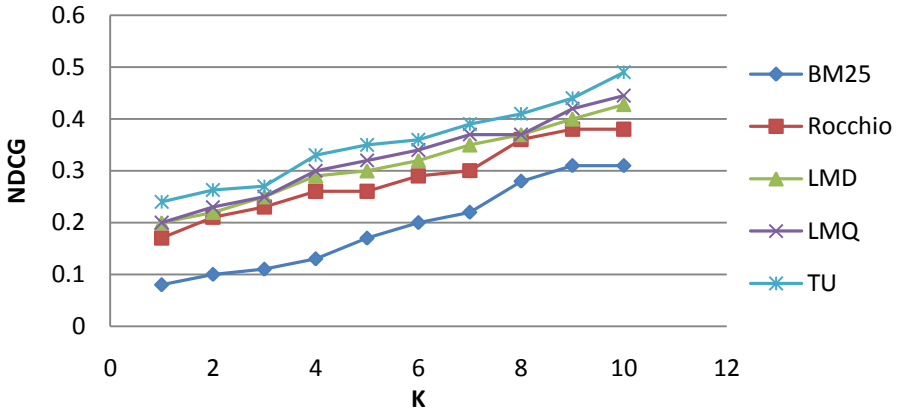


Figure 2.7 NDCG at *K* obtained for 10 queries

Module #2: Experimental Results [3/3]

- Evaluation on AOL Dataset
  - Original Rank Vs. Personalized Rank

User Id : 1038			
Query	**Web search engine results**	**Original Rank**	**Personalized Rank**
Shane mcfaul	http://www.uefa.com	14	1
	http://archives.tcm.ie	2	2
	http://www.nottscountyfc.premiumtv.co.uk	4	3
Brochure	http://www.acsm.net	10	1
	http://www.createchange.org	6	2
	http://www.brochure-design.com	2	3
Joe afful	http://aupanthers.collegesports.com	9	1
	http://seattletimes.nwsource.com	2	2
	http://www.ajc.com	8	3

Table 2.5 The original top 3 Results from a web search engine for some queries and Re-ranked results using proposed model

Module #2: Summary

- Personalization has been performed at client side
- User topical interest modeling has been exploited in personalizing IR system especially for web data
- To produce re-ranked list of documents based on user information need in the order of relevance
- Not all the queries would require personalization to be performed

Module #3: Introduction [1/3]

- Computational Intelligence for Document Re-ranking
- Approach
  - To identify the documents that might contain the desired information using computational intelligence technique named Genetic Algorithm (GA).
- Problem statements
  - How to represent documents as chromosomes?
  - How to evaluate fitness of search results?

Module #3: Introduction [2/3]

- Why GA based IR?
  - The document search space represents a high dimensional space i.e. the size of the document corpus is multitude in IR
    - GA is one of the powerful searching mechanisms known for its quick search capabilities
  - The traditional relevance feedback mechanisms are not efficient when no relevant documents are retrieved with the initial query.
    - The probabilistic exploration induced by GA allows the exploration of new areas in the document space.

Module #3: Introduction [2/3]

- Methodology
  - Applying GA with an adaptation of probabilistic model
  - The probabilistic similarity function has been used for fitness evaluation which leads to a better retrieval performance than that obtained by using a simple matching function such as Cosine similarity
  - The documents are assigned a score based on the probability of relevance. This probability has been used to rank the documents.
  - Probability associated with each documents are sought using GA approach in order to optimize the search process i.e. finding of relevant document not by assessing the entire corpus or collection of documents

Module #3: Introduction [3/3]

- Steps in Genetic Algorithm for Information Retrieval

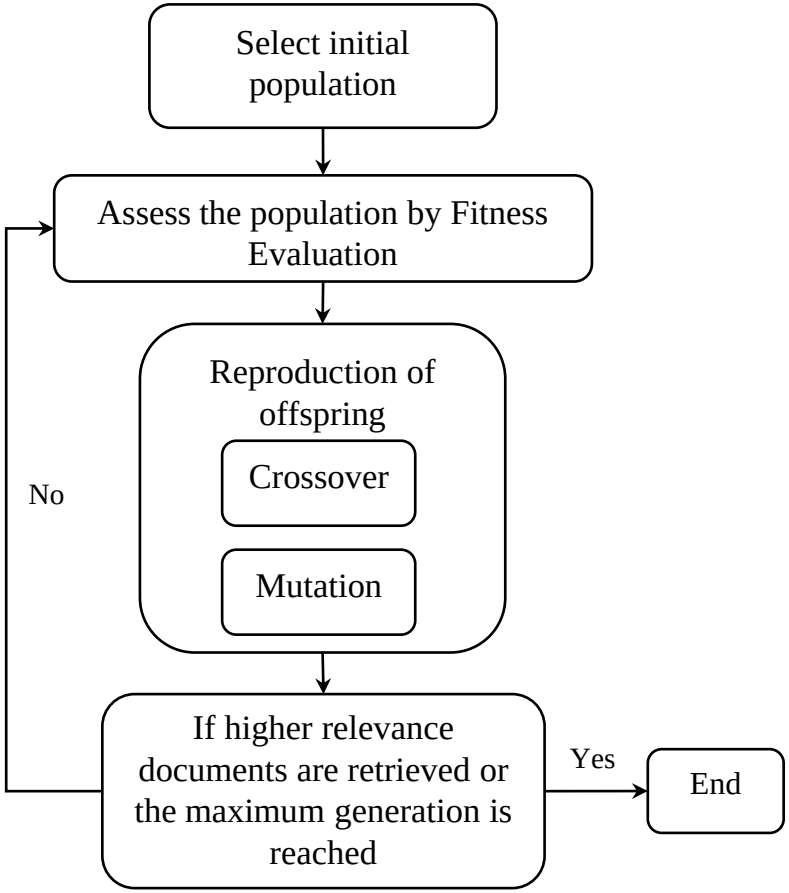


Figure 3.1 Steps in GA for IR

Module #3: Methodology [1/4]

- Probabilistic model for Fitness Evaluation
  - Representation of Chromosomes

$$\begin{array}{c} \\ 000000000000 \\ 000000000001 \\ \vdots \\ \vdots \\ 111111111111 \end{array} \begin{bmatrix} & T_1 & T_2 & T_3 & \cdots & T_t \\ D_1 & w_{11} & w_{12} & w_{13} & \cdots & w_{1t} \\ D_2 & w_{21} & w_{22} & w_{23} & \cdots & w_{2t} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{n1} & w_{n2} & w_{n3} & \cdots & w_{nt} \end{bmatrix}$$

  - Fitness functions

$$P(q \mid d) = \sum_{w \in d} (P(q \mid w) P(w \mid d)) \qquad \textbf{Eq. (3.1)}$$

$$P(q \mid d) = \alpha P(q \mid C) + (1 - \alpha) \sum_{w \in d} P(q \mid w) P(w \mid d) \qquad \textbf{Eq. (3.2)}$$

$$P(q \mid C) = \frac{Count(q\_in\_C)}{|C|} \qquad P(w \mid d) = \frac{Count(w\_in\_d)}{|d|}$$

Module #3: Methodology [2/4]

- Probabilistic model for Fitness Evaluation
  - Chromosome Selection
    - Roulette-wheel selection

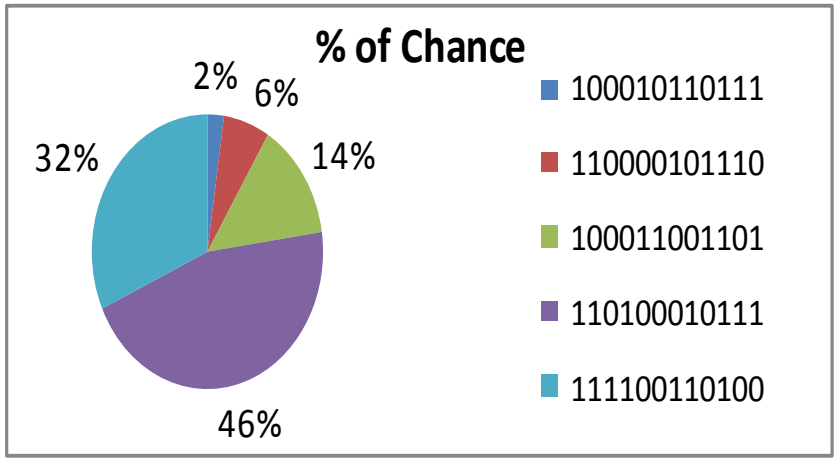


Figure 3.2 Roulette-wheel selection

Chromosomes	Fitness value	% of chance to be chosen
100010110111	0.021	2%
110000101110	0.059	6%
100011001101	0.127	14%
110100010111	0.415	32%
111100110100	0.29	46%

Table 3.1 Chromosomes and its fitness value

  - Elitist selection
    - The fit members of each generation are guaranteed to be selected. Fitness value of a chromosome which is above 0.7 is retained for further generations.

Module #3: Methodology [3/4]

- Probabilistic model for Fitness Evaluation
  - Reproduction Operators
    - Crossover
      - The probability of crossover (P_c) is the probability that the crossover will occur at a particular mating i.e. not all mating must be reproduced by crossover.

Mask:	0110011000110	
Parents:	101000111010 0	0011010010001
Offspring:	0011001010101	1010010110000

    - Mutation
      - Mutation operator is performed to introduce diversity in the population.
      - The probability of mutation (P_m) of a bit is assumed as $1/l$ where l is the length of the chromosome i.e. 1 out of 12 bits is chosen at random and modified.

Parent:	101000 1 110100
Offspring:	101000 0 110100

Module #3: Methodology [4/4]

- Personalized model for Fitness Evaluation
  - Fitness functions

$$P(Q \mid D, \theta_u) = P(Q \mid T_d, T_u) + \prod_{q_i \in Q} (\beta P(q_i \mid \theta_u) + (1-\beta) P(q_i \mid D)) \quad \text{Eq. (3.3)}$$

$$P(Q \mid T_u, T_d) = \prod_{q_i \in Q} (\alpha P(q_i \mid T_u) + (1-\alpha) P(q_i \mid T_d)) \quad \text{Eq. (3.4)}$$

- Experimental Dataset Description

Test Corpus	# of docs	# of queries	Avg. doc. length	Avg. doc. length after pre-processing
CACM (articles published in the Communications of the ACM)	3204	52	40	18
CISI (articles about information sciences)	1460	35	104	46
Real dataset (Results from search engine for a query)	Top 50 results	100	34	21

Table 3.2 Characteristics of Dataset

- Baseline approaches
  - Classical IR (Okapi-BM25) based on GA

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i).\frac{tf_i.(k_1+1)}{tf_i + k_1.\left(1-b+b.\frac{|D|}{avgdl}\right)} \quad \textbf{Eq. (3.5)}$$

$$IDF(q_i) = \log \frac{N - df_i + 0.5}{df_i + 0.5}$$

- Evaluation metrics
  - Interpolated Precision

$$P_{Interpolated}(r) = \max_{r' \geq r} P(r') \quad \text{Eq. (3.6)}$$

  - MAP

$$MAP = \frac{\sum_{q=1}^{Q} Avg\_precision(q)}{Q} \quad \text{Eq. (3.7)}$$

$$Avg\_precision = \frac{1}{|\#rel\_doc|} \sum_{k} (P@k).(rel(k))$$

  - NDCG

$$DCG_K = \sum_{i=1}^{K} \frac{2^{r_i} - 1}{\log_2(i+1)} \quad \text{Eq. (3.8)}$$

$$NDCG_K = \frac{DCG_K}{IDCG_K}$$

Module #3: Experimental Evaluation [4/6]

- Parameter Tuning
  - Experiments were carried out for 30 different queries on the retrieval models based on GA
  - It is observed that the proposed fitness functions performs well on both benchmark and real dataset with the crossover probability (P_c) of 0.5 and mutation probability (P_m) of 0.08.
  - It is verified that this setting yields reasonable improvement in Precision, NDCG, and MAP

Module #3: Experimental Evaluation [5/6]

- Parameter Tuning on Crossover probability

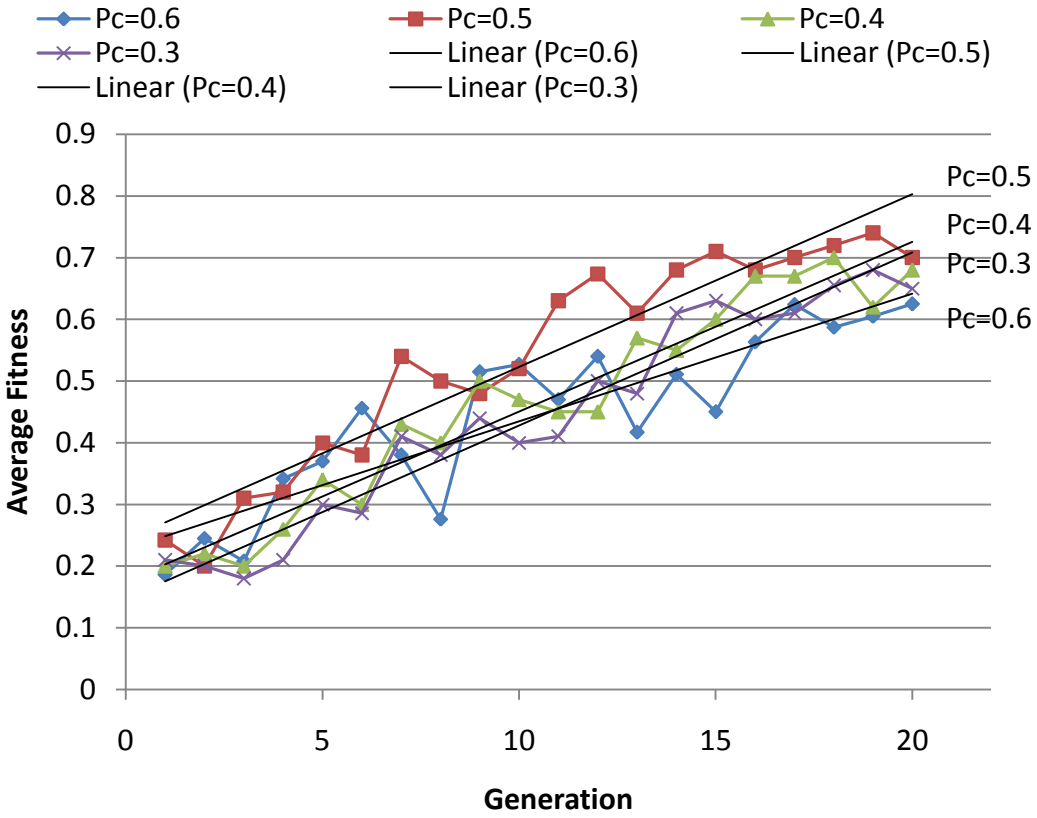


Figure 3.3 linear improvements on average fitness vs. generation for varying P_c

Module #3: Experimental Evaluation [6/6]

- Parameter Tuning on Mutation probability

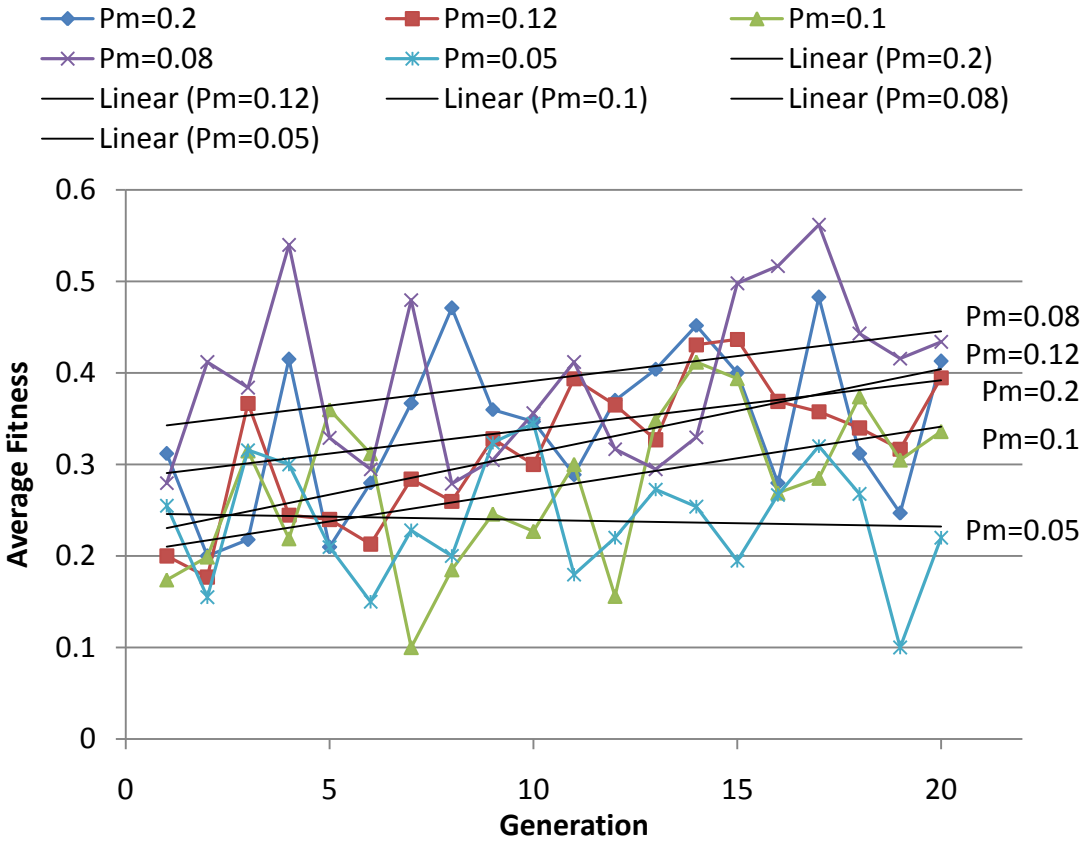


Figure 3.4 linear improvements on average fitness vs. generation for varying P_m

Module #3: Experimental Results [1/4]

- Evaluation on Benchmark Dataset
  - Precision at *k* on CACM and CISI collection for 30 queries

Top *k* results	Precision on CACM			Precision on CISI		
	BM_25-GA	Prob_IR-GA_1	Prob_IR-GA_2	BM_25-GA	Prob_IR-GA_1	Prob_IR-GA_2
5	0.544	0.6805	0.721	0.573	0.716	0.742
10	0.443	0.625	0.66	0.43	0.592	0.708
15	0.415	0.58	0.625	0.39	0.55	0.635
20	0.365	0.51	0.572	0.325	0.493	0.58

Table 3.3 Precision at top *k* results on CACM and CISI collection for 30 queries

  - MAP at *k* on CACM and CISI collection for 30 queries

Strategy	MAP @ 5		MAP @ 10	
	CACM	CISI	CACM	CISI
BM25-GA	0.47	0.452	0.445	0.452
Prob_IR-GA_1	0.624	0.615	0.58	0.563
Prob_IR-GA_2	0.649	0.653	0.662	0.647

Table 3.4 MAP at top *k* results on CACM and CISI collection for 30 queries

Module #3: Experimental Results [2/4]

- Evaluation on Benchmark Dataset
  - Precision Recall Graph on CACM and CISI collection

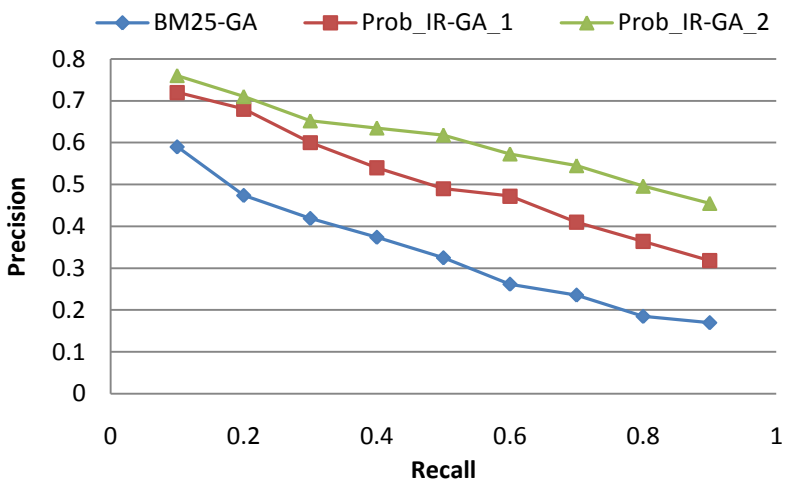


Figure 3.5 Precision Recall Curve on CACM dataset

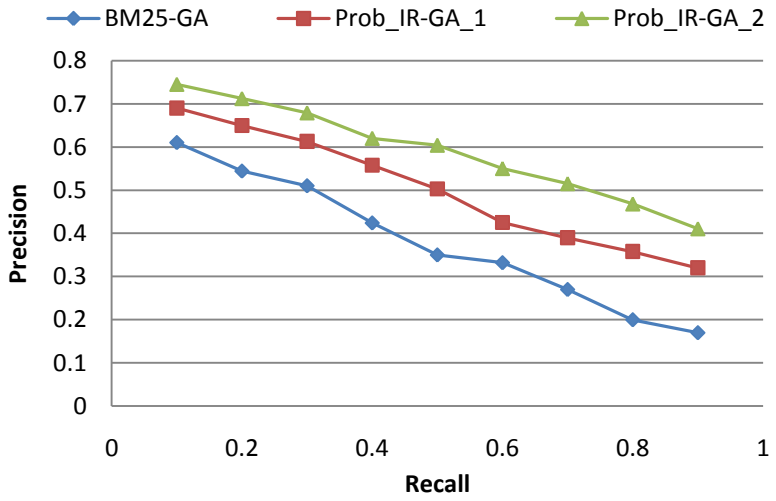


Figure 3.6 Precision Recall Curve on CISI dataset

Module #3: Experimental Results [3/4]

- Evaluation on Benchmark Dataset
  - NDCG on CACM and CISI collection

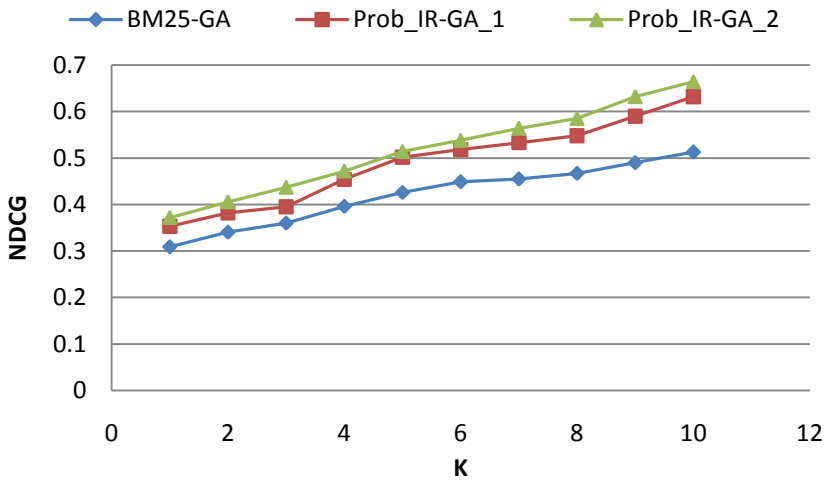


Figure 3.7 NDCG on CACM dataset for varying K

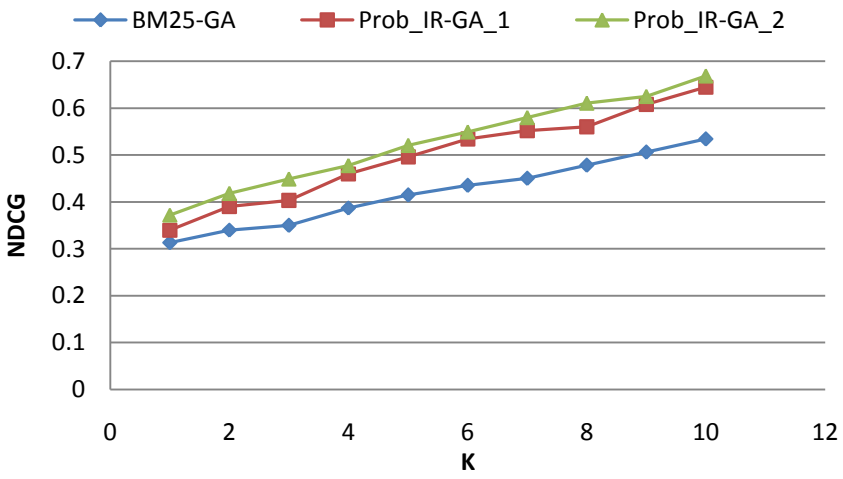


Figure 3.8 NDCG on CISI dataset for varying K

Module #3: Experimental Results [4/4]

- Evaluation on Real Dataset
  - Precision Recall Graph & NDCG

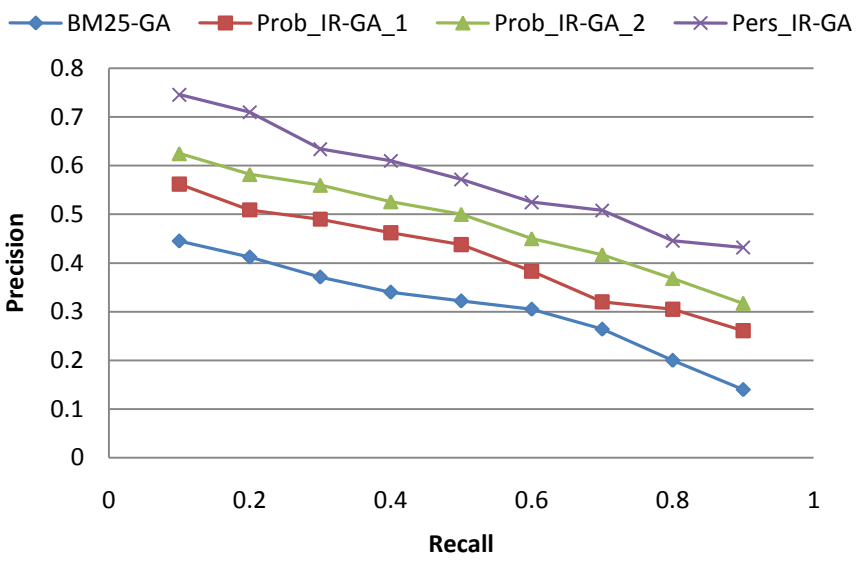


Figure 3.9 Precision Recall Curve

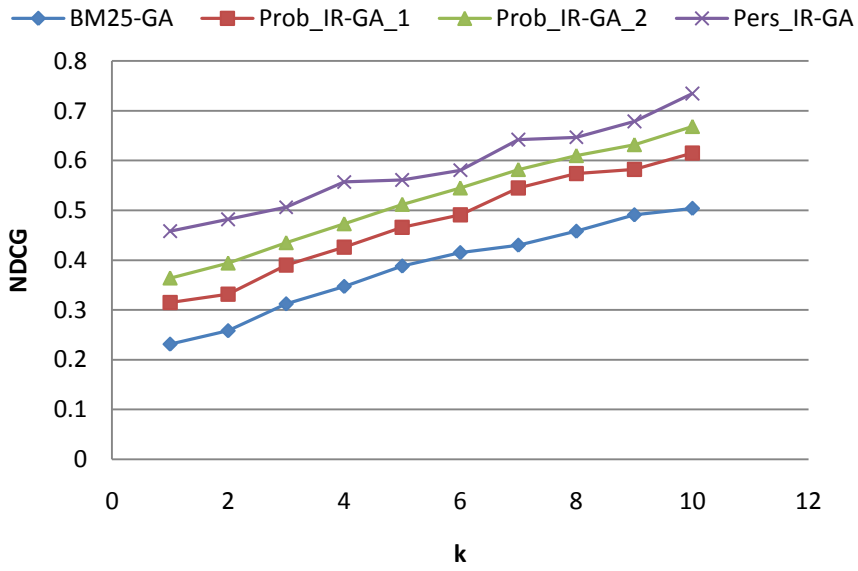


Figure 3.10 NDCG for varying *k*

Module #3: Summary

- Explored the utility of incorporating Genetic Algorithm to improve search ranking.
- The GA terminates when either a maximum number of generations are reached or an acceptable fitness level has been achieved for the population of candidate solutions.
- If the algorithm gets terminated due to maximum number of generations, satisfactory solution may or may not have been achieved.
- Adaptation of personalization in retrieval using GA provides more desirable results.
- It is verified that the proposed fitness functions found to be efficient within a specific domain as it retrieves more relevant results.

Module #4: Introduction [1/2]

- Computational Intelligence for Search Query Reformulation
- Approach
  - Typically, users often modify previously issued search query with an expectation of retrieving better results. These modifications are called query reformulations or query refinements.
  - Query refinement is an iterative process of query modification that can be used to narrow down the scope of search results.
- Problem statements
  - How to address vocabulary mismatch problem in IR?
  - How to change the original query to form a new query that would find better relevant documents?

Module #4: Introduction [2/2]

- Methodology
  - Exploiting Ant Colony Optimization (ACO) approach to suggest related key words
  - The short queries normally formed with not more than two or three words which may not avoid the inherent ambiguity of search query
  - ACO is explored to build adaptive knowledge structures for query reformulation in order to provide interactive search.
  - Idea is that the self-organizing principles which allow the highly coordinated behavior of real ants are exploited to coordinate populations of artificial agents that collaborate to solve computational problems

Module #4: Methodology [1/13]

- Terminologies
  - Artificial Ant
  - Pheromone
  - Typical Ant System
    - The first ant finds the food source (*F*), via any path (*a*), then returns to the nest (*N*), leaving behind a trail pheromone (*b*)
    - Ants indiscriminately follow different possible paths. This essentially strengthens the path by making it more attractive as the shortest route.
    - Ants take the shortest route to travel further and leave the other paths which essentially lose their pheromone trails.

Module #4: Methodology [2/13]

- Block Diagram

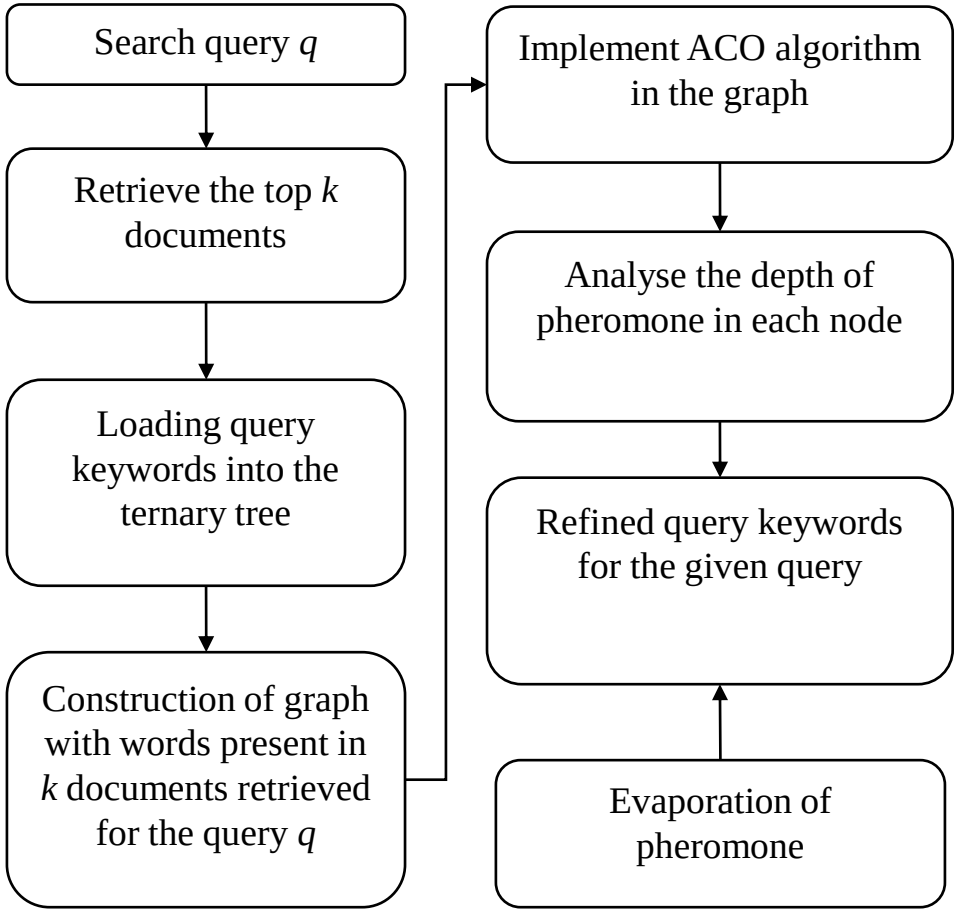


Figure 4.1 Block Diagram of ACO Approach

Module #4: Methodology [3/13]

- Ant System enabled IR
  - The first ant finds the related keywords (rk_j) via any path (p) in the graph, then returns to the query keyword (q), leaving behind a pheromone (p_w) trails i.e. weight or importance of the path.
  - Ants arbitrarily follow various paths in the term graph. This in turn strengthens the path and makes it more attractive as an important route.
  - Ants take the important route to find the related keyword which retrieve relevant documents and leaves the other paths.

Module #4: Methodology [4/13]

- Ant System enabled IR
  - Graph Representation phase
    - Ternary tree construction
    - Term graph construction
  - Query Reformulation phase
    - Generation of candidate suggestions
    - Preparation of Query suggestions

Module #4: Methodology [5/13]

- Graph Representation phase
  - Ternary tree construction

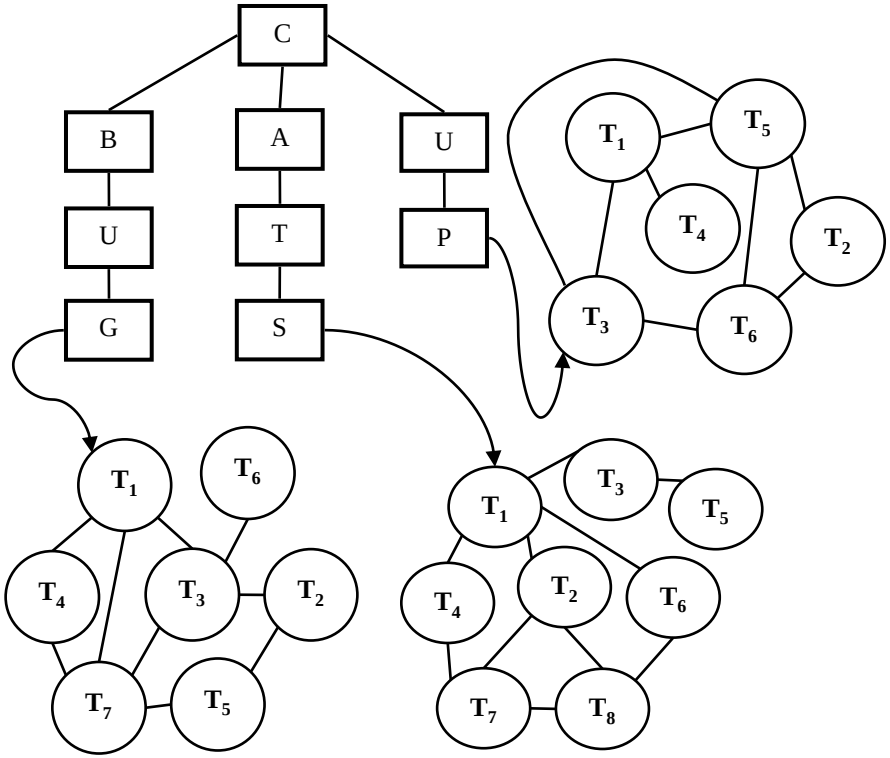


Figure 4.2 Ternary tree nodes and its associated Term Graph

- Graph Representation phase
  - Term graph construction

Algorithm 4.1 Term_Graph_Construction (q_i, D_n)

Input: query term q_i, terms in retrieved documents d_t

Output: Graph structure G_d

for each document $d_i \epsilon D$ retrieved for the query q

extract terms ($t_{1,\ldots,n}$) present in d_i and create node for each term and create edges between these nodes to be complete graph;

for each term d_t

if a term d_t present in more than one document then

create an edge connecting d_t with terms present in documents

Module #4: Methodology [7/13]

- Query Reformulation phase
  - Generation of candidate suggestions
    - The data set contains computer science related keywords and the keywords rating
    - Attempts to select the best few suggestions by ACO heuristics.
  - Preparation of Query suggestions
    - The model built with ACO takes the form of a graph structure where root nodes are query terms and edges point to are possible query refinements.
    - The weight on the edges encodes the importance of the association between the nodes i.e. query terms.

Module #4: Methodology [8/13]

- Possible ACO implementations
  - Steps of Non-personalized query reformulation using ACO
    - Select a vertex $r \in V[G]$ to be a root vertex that is query node
    - Compute a minimum spanning tree T for G from root r using Prim's algorithm
    - Let L be the list of vertices visited in a pre-order tree walk of T
    - Return the Hamiltonian cycle H that visits the vertices in the order L
  - Steps of Personalized query reformulation using ACO
    - Select a vertex $r \in V[G]$ to be a root vertex that is query node
    - Traverse the graph according to the search navigation history of different user
    - Choose the list of vertices visited by most of the users from the query node
    - Return the top few words visited as suggestions

Module #4: Methodology [9/13]

- Work Flow

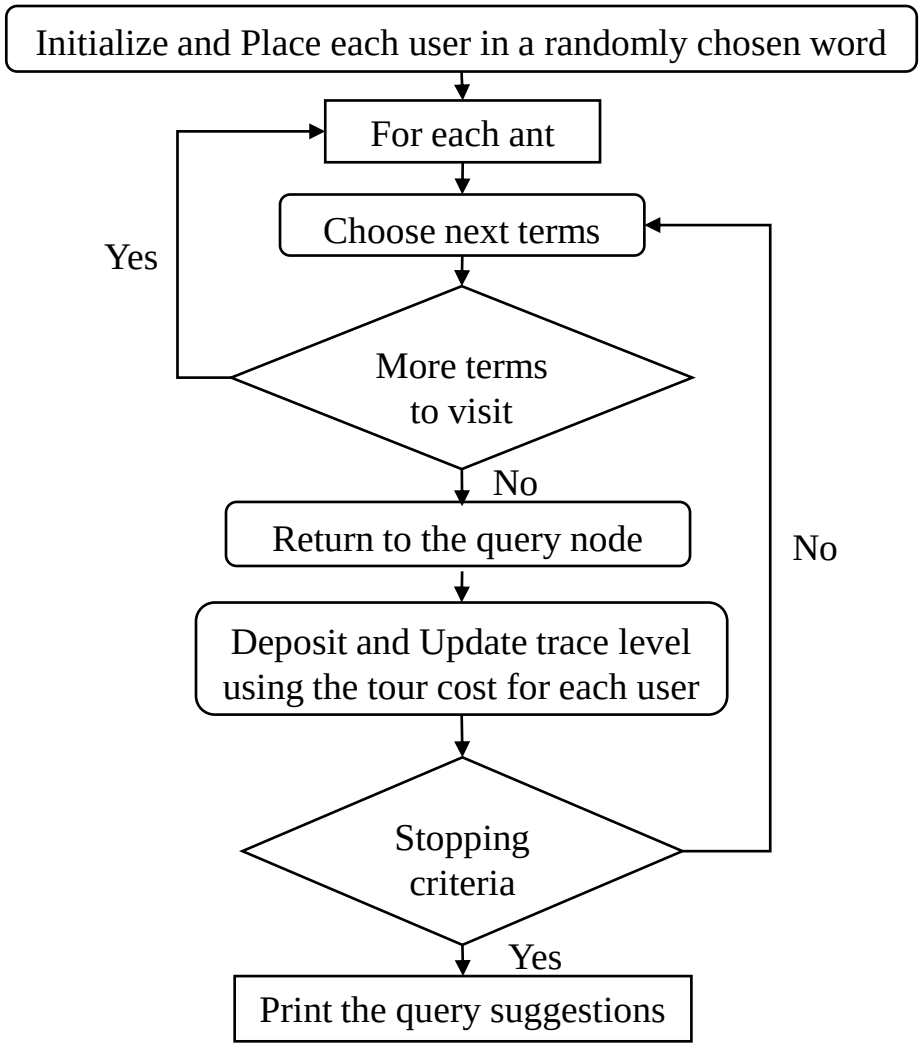


Figure 4.3 Flow of proposed ACO implementation

Module #4: Methodology [10/13]

- Notion of autocatalytic behavior
- If the query q_i is issued by 12 different users then 12 numbers of ants are assumed to find related suggestions.
- Each ant is a simple agent with the following characteristics:
  - It chooses the query term to go to with a transition probability that is a function of the similarity and of the amount of trail present on the connecting edge between terms;
  - User navigation over web pages retrieved for a query is treated as ant movement over graph.
  - To force the user to make legal transition, navigation to already visited terms are not allowed until a tour is complete.
  - When the user completes a tour, a substance called trail or trace or pheromone is laid on each edge (i,j) visited.

Module #4: Methodology [11/13]

- Transition Probability
  - In a Graph $G(N,E)$ where N is the number of terms (nodes) and E is the edges and d_{ij} is the edge weight i.e. similarity between i and j. Ant moves from a node i to the next j with the transition probability.

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{k \in allowed_k}[\tau_{ik}(t)]^{\alpha}[\eta_{ik}]^{\beta}} & \text{if } k \in \text{allowed}_k \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{Eq. (4.1)}$$

  - Each edge is associated a static value based on the similarity weight $\eta_{ij} = 1/d_{ij}$. Each edge of the graph is augmented with a trace τ_{ij} deposited by ants (users). Initially it is 0. Trace is dynamic and it is learned at run-time.

- Trail Deposition and Train value Update
  - The similarity between any two node in the graph $\tau_{ij}(t)$ denote the intensity of trail on edge (i,j) at time t

$$\tau_{ij}(t+1) = p \times \tau_{ij}(t) + \Delta\tau_{ij} \qquad \text{Eq. (4.2)}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}$$

  - where p is the rate of trail decay per time interval i.e. pheromone evaporation factor & m is the number of users i.e. ants.

Module #4: Methodology [13/13]

- Trail Deposition and Train value Update

$$\Delta\tau_{i,j}^{k} = \begin{cases} \dfrac{c}{L_k} & if\ (i,j) \in \text{High Similarity} \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{Eq. (4.3)}$$

  - The trail intensity is updated subsequently upon the completion of each algorithm cycle.
  - Each ant subsequently deposits trail of quantity $1/L_k$ on every edge (i,j) visited in its individual tour.
  - The sum of all newly deposited trails is denoted by $\Delta\tau_{ij}$.
  - L_k is defined as the length of the ant i.e. the similarity between the terms visited by the user.
  - c is a constant defined as the average similarity weight of all edges in the graph.

Module #4: Experimental Evaluation [1/2]

- Experimental Dataset Description
  - The dataset used in this work is AOL search log which possesses implicit feedback in the form of click-through data collected by a search engine.
  - The AOL Search Data is a collection of real query log data that is based on real users.
  - For evaluating the proposed approach,
    - Only the queries issued by at least 10 users were employed and the pre-processed top 20 documents retrieved for that query were used to construct graph.
    - For example, if the query is issued by 10 different users then 10 numbers of ants are assumed to suggest related queries. 270 single and two word queries issued by different users from AOL search log are taken.

Module #4: Experimental Evaluation [2/2]

- Baseline Approaches
  - Association Rule based approach
  - SimRank Approach
  - Backward Random Walk approach (BRW)
  - Forward Random Walk approach (FRW)
  - Traditional ACO based approach
- Parameter Tuning
  - Depth: This refers to the number of hop between the nodes in order to recommend queries. Depth was set as 5 i.e. top ranked 5 related queries
  - Evaporation factor (ρ): This was set to 0.5
  - Pheromone updating schemes

Module #4: Experimental Results [1/8]

- MRR Evaluation

Approaches	MRR
Association rule based approach	0.387
SimRank	0.362
Backward Random Walk (BRW)	0.314
Forward Random Walk (FRW)	0.321
Traditional ACO based approach	0.411
Proposed query reformulation using ACO	0.527

Table 4.1 MRR obtained for 10 queries

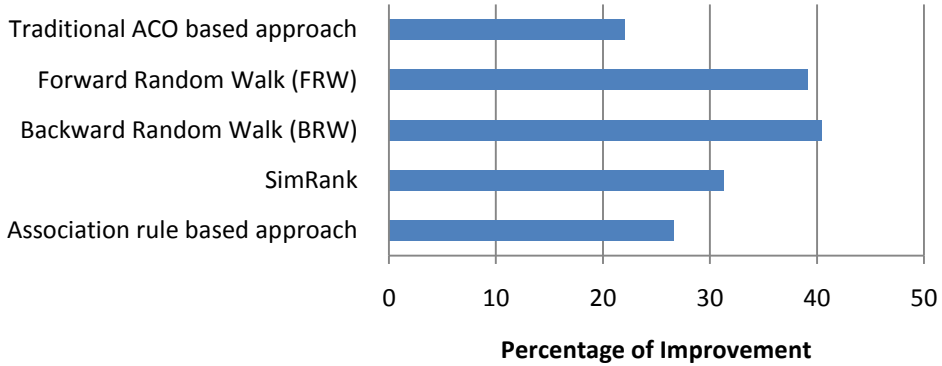


Figure 4.4 Percentage of MRR improvement

Module #4: Experimental Results [2/8]

- MRR Evaluation

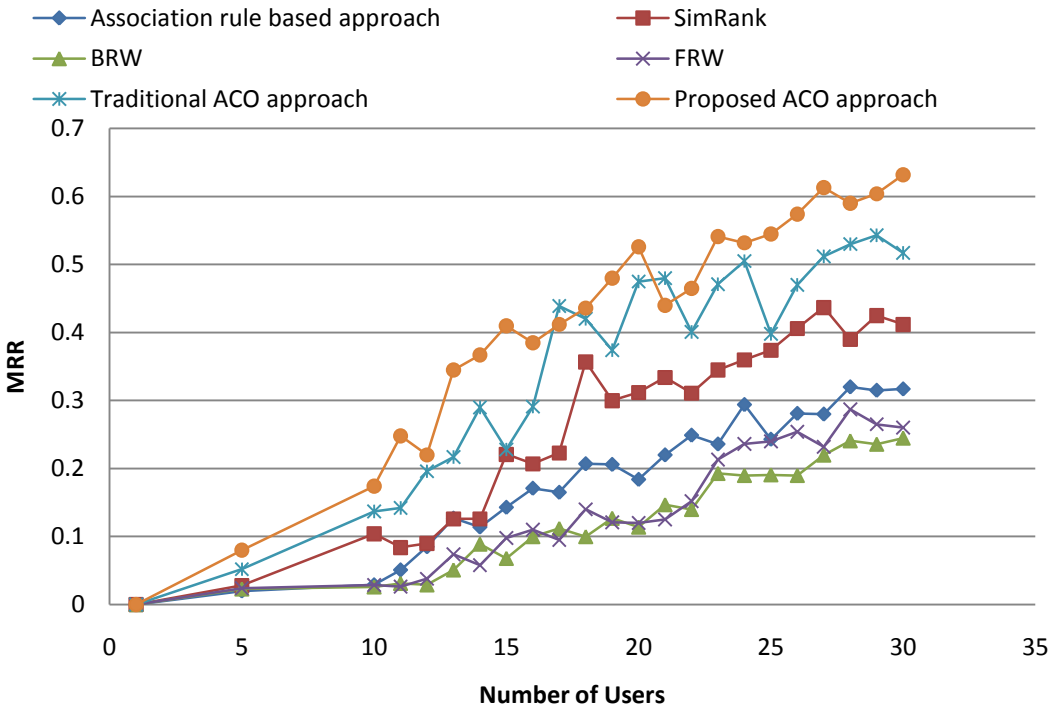


Figure 4.5 MRR Vs. Number of Users

Module #4: Experimental Results [3/8]

- Personalized suggestions for the initial queries

Testing Queries	Suggestions				
	Top 1	**Top 2**	**Top 3**	**Top 4**	**Top 5**
Game	Game aoe	Game stores	Game car	Game theory	Game online
Graph	Graph theory	Graph function	Graph software	Graph types	Graph plotter
Algorithm	Algorithm analysis	Algorithm ACO	Algorithm programming	Algorithm computer	Algorithm flowchart
Java	Java oracle	Java download	Java updates	Java JVM	Java tutorials
Database	Database relational	Database DBMS	Database examples	Database applications	Database types
Apple	Apple store	Apple iphone	Apple computer	Apple ios	Apple updates
Marathon	Marathon running	Marathon athletic	Marathon fitness	Marathon race	Marathon oil

Table 4.2 Top *5* suggestions generated by the proposed ACO method

Module #4: Experimental Results [4/8]

- Manual Evaluation
  - Users were asked to rate the query suggestion results. The rating score ranges from 0 to 3 (3 - highly relevant, 2 - relevant, 1 - hard to tell, 0 - irrelevant)

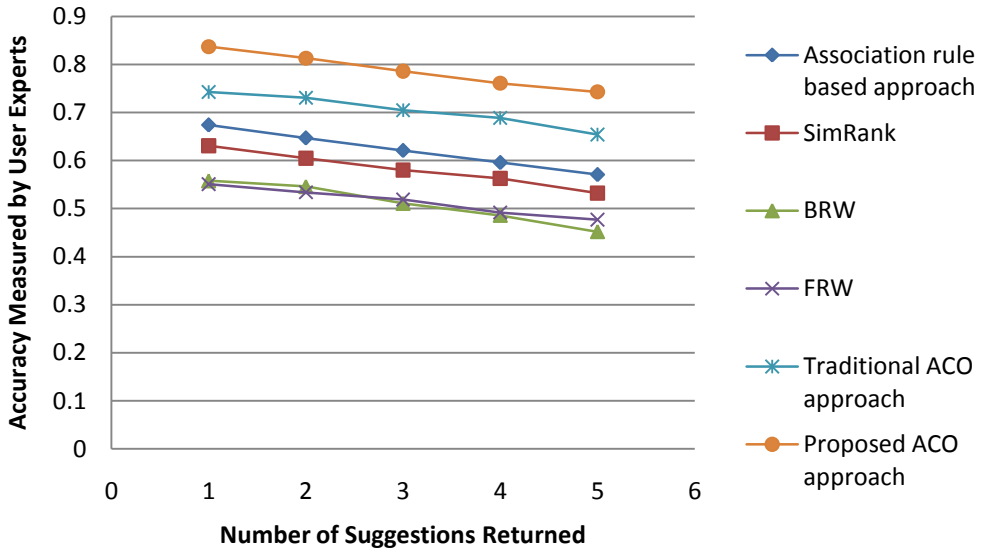


Figure 4.6 Accuracy comparisons measured by users

  - Increases the accuracy for about 21, 26.1, 35.2, 34.7, and 10.6 percent

Module #4: Experimental Results [5/8]

- Benchmark Evaluation
  - The ODP dataset described by has been utilized for automatic evaluation
  - The similarity between two categories C_p and C_p' is defined as the length of their Longest Common Prefix divided by the length of the longest path between C_p and C_p'.

$$sim(C_p, C_p') = \frac{|LCP(C_p, C_p'|}{MAX\{|C_p|, |C_p'|\}} \qquad \textbf{Eq. (4.4)}$$

  - Example: The length of the two query terms 'Algorithms' and 'Pseudo code' is computed as 2/3=0.667

Module #4: Experimental Results [6/8]

- Benchmark Evaluation

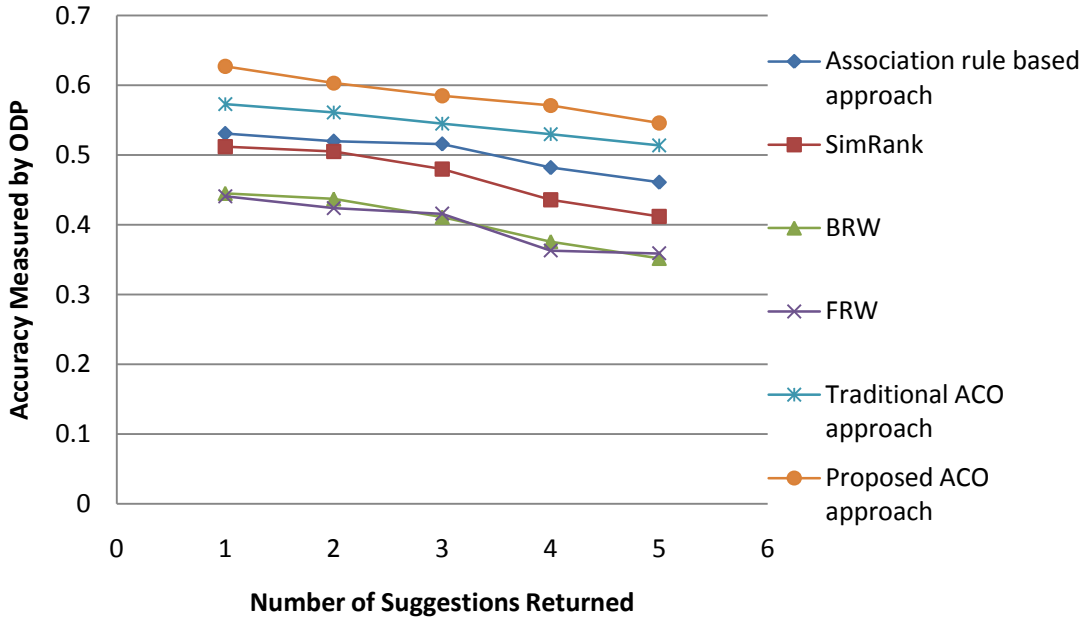


Figure 4.7 Accuracy comparisons measured by ODP

- Increases the suggestion accuracy for about 14.4, 20, 31.1, 31.7, and 7.1 percent

Module #4: Experimental Results [7/8]

- Evaluation on Accuracy of Personalized Query suggestion
  - In order to evaluate the quality of the ACO based personalized query recommendation approach, 10 groups namely $G_1, G_2, \ldots, G_{10}$ have been created
  - Randomly select 5 users from the user list for each group. Totally 50 users are considered
  - For each of these users, ants start the transition with the submitted user query as source. After preparing the results, query suggestions were rated by the expert users.
  - The range of rating score has been defined from 0 to 3 (3 - highly relevant, 2 - relevant, 1 - hard to tell, 0 – irrelevant)

Module #4: Experimental Results [8/8]

- Evaluation on Accuracy of Personalized Query suggestion

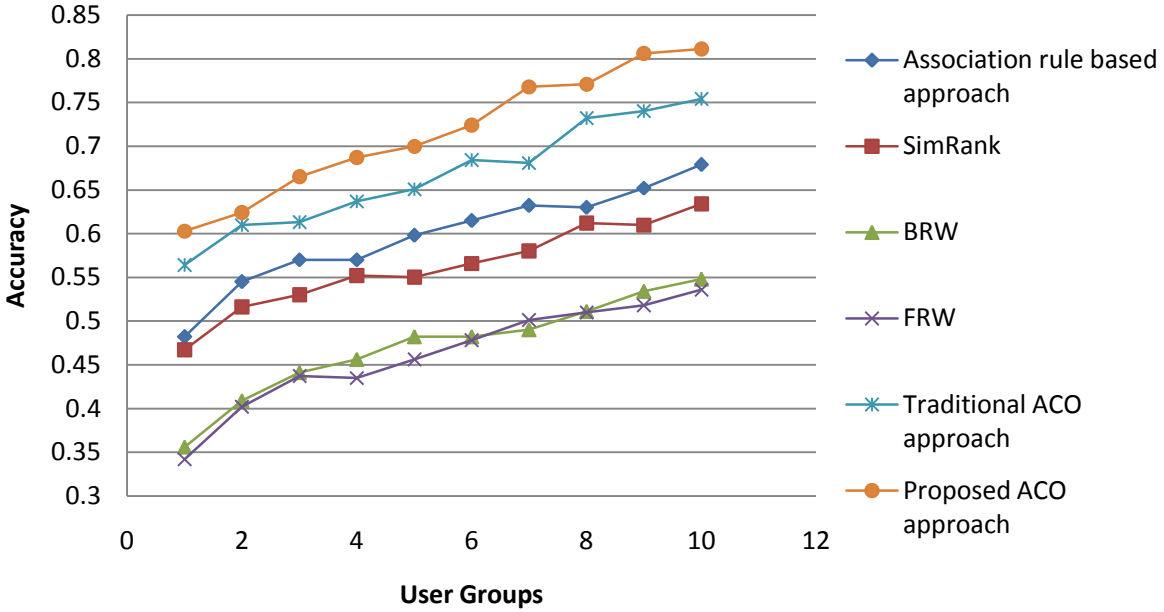


Figure 4.8 Accuracy of personalized query recommendations

- Increases the suggestion accuracy for about 16.56, 21.53, 34.22, 35.53, and 6.88 percent

Module #4: Summary

- Algorithm for search query reformulation on large scale term graph using ACO principle has been presented
- The generated query suggestions are semantically related to the initial query
- The challenge of determining how to suggest the more appropriate query reformulation suggestions for an ambiguous query from a number of possible candidates has been addressed

Conclusion

- MODULE #1
- MODULE #2
- MODULE #3
- MODULE #4

Journal Publications

[1] Veningston. K, Shanmugalakshmi. R. *Computational Intelligence for Information Retrieval using Genetic Algorithm*. INFORMATION-An International Interdisciplinary Journal, Vol.17, No.8, pp. 3825-3832, August 2014. ISSN 1343-4500(print), ISSN 1344-8994(electronic), Published by International Information Institute, Japan.

[2] Veningston. K, Shanmugalakshmi. R. *Combining User Interested Topic and Document Topic for Personalized Information Retrieval*. S. Srinivasa and S. Mehta (Eds.): Big Data Analytics (BDA), Lecture Notes in Computer Science (LNCS), vol. 8883, pp. 60–79, Springer International Publishing Switzerland, 2014.

[3] Veningston. K, Shanmugalakshmi. R. *Efficient Implementation of Web Search Query reformulation using Ant Colony Optimization*. S. Srinivasa and S. Mehta (Eds.): Big Data Analytics (BDA), Lecture Notes in Computer Science (LNCS), vol. 8883, pp. 80–94, Springer International Publishing Switzerland, 2014.

Conference Publications

[1] Veningston. K, Shanmugalakshmi. R. *Combining User Interested Topic and Document Topic for Personalized Information Retrieval*. In Proc. International Conference on Big Data Analytics (BDA-2014), Indian Institute of Technology, Delhi, INDIA, December 20-23, 2014. [**To be Held**]

[2] Veningston. K, Shanmugalakshmi. R. *Efficient Implementation of Web Search Query reformulation using Ant Colony Optimization*. In Proc. International Conference on Big Data Analytics (BDA-2014), Indian Institute of Technology, Delhi, INDIA, December 20-23, 2014. [**To be Held**]

[3] Veningston. K, Shanmugalakshmi. R, *Information Retrieval by Document Re-ranking using Term Association Graph*. In Proc. International Conference on Interdisciplinary Advances in Applied Computing (ICONIAAC 2014), Amrita University, Coimbatore, INDIA, October 10-11, 2014. [**Best Paper Award**] Published in ACM Digital Library] DOI: http://dx.doi.org/10.1145/2660859.2660927

[4] Veningston. K, Shanmugalakshmi. R. *Personalized Grouping of User Search Histories for Efficient Web Search*. In Proc. 13th WSEAS International Conference on Applied Computer and Applied Computational Science (ACACOS 2014), Universiti Teknologi Malaysia, Kuala Lumpur, MALAYSIA, pp. 164-172, April 23-25, 2014. Published with ISBN: 978-960-474-368-1 ISSN: 1790-5109

[5] Veningston. K, Shanmugalakshmi. R. *Statistical language modeling for personalizing Information Retrieval*. In Proc. IEEE International Conference on Advanced Computing and Communication Systems (ICACCS 2013), pp. 1- 6, December 19-21, 2013.
[**Best Paper Award**] [Published in IEEE Xplore] DOI: http://dx.doi.org/10.1109/ICACCS.2013.6938717

[6[Veningston. K, Shanmugalakshmi. R, Ruksana. N. *Context aware Personalization for Web Information Retrieval: A Large scale probabilistic approach*. In Proc. International Conference on Cloud and Big Data Analytics (ICCBDA 2013), PSG College of Technology, Coimbatore, INDIA, February 8-9, 2013.

[7] Veningston. K, Shanmugalakshmi. R. *Enhancing personalized web search Re-ranking algorithm by incorporating user profile*. In Proc. 3rd IEEE International Conference on Computing, Communication and Networking Technologies (ICCCNT 2012), pp. 1-6, July 26-28, 2012. [Published in IEEE Xplore] DOI: http://dx.doi.org/10.1109/ICCCNT.2012.6396036

Book References

[1] Salton. G, and McGill. M. J. **Introduction to modern information retrieval.** McGraw-Hill. New York (1986)

[2] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, "**Modern Information Retrieval**", Addison Wesley (1999)

[3] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, "**Introduction to Information Retrieval**", Cambridge University Press (2008)

[4] D. E. Goldberg, "**Genetic Algorithms in Search, Optimization, and Machine Learning**", Addison-Wesley, (1989).

References

[1]Nicolaas Matthijs, Filip Radlinski, "*Personalizing Web Search using Long Term Browsing History*", ACM- WSDM, February 2011.

[2] Eugene Agichtein, Eric Brill, Susan Dumais, "*Improving Web Search Ranking by Incorporating user behavior information*", ACM - SIGIR, August 2006.

[3] Ponte, Jay M., and W. Bruce Croft, "*A language modeling approach to information retrieval*", In Proc. ACM SIGIR, pp. 275-281, 1998.

[4] Lafferty, John, and Chengxiang Zhai, "*Document language models, query models, and risk minimization for information retrieval*", In Proc. ACM SIGIR, pp. 111-119, 2001.

[5] Kushchu, I, "*Web-Based Evolutionary and Adaptive Information Retrieval*". IEEE Transactions on Evolutionary Computation, Vol. 9, No. 2, pp. 117 – 125, 2005.

[6]Kenneth Wai-Ting Leung, and Dik Lun Lee, "*Deriving Concept-based User profiles from Search Engine Logs*". IEEE Trans. Knowledge and Data Engineering, Vol. 22, No. 7, pp. 969-982, July 2010.

[7] Roi Blanco, and Christina Lioma, "*Graph-based term weighting for information retrieval*". Springer Information Retrieval, Volume 15, Issue 1, pp 54-92, February 2012.

Thanking You