

Re-Ranking Web Data per Knowledge Domain

Grace Zhao ^{*}, Xiaowen Zhang [†]

Abstract

We propose a re-ranking algorithm to effectively assay and re-sequence the web data crawled by some credited web search engines, to meet the user needs in a domain space. The algorithm studies the structure and semantics of the domain ontology graph and constructs computational relations among nodes. After examining matching terms between ontology dictionary and the textual content (text, metadata) of the retrieved documents, we calculate three-dimensional information scores – *distance*, *direction*, and *relationship* of each document in the top *k* search result set. We further explore the *directional relation* with three information degrees: *granularity*, *diversity* and *generality*, and subsequently re-rank the retrieved documents. ¹

Keywords: document re-ranking, information retrieval, knowledge management, ontology learning

1 Introduction

To build a domain-specific knowledge base of web resources, we have to face the challenge of having the computers gather domain-relevant knowledge feeds. For it, a web resource re-ranking algorithm that provides data feed to the system becomes necessary.

We dissect the relationship among nodes on the ontology graph and create a mapping schema that classify the *distance*, *direction*, and *attributive relationship* among nodes. We also introduce the concept of *domain characteristic classifier* (DCC) to augment the query search, which is instrumental in bringing domain-relevant content into initial search result set, as a means of reducing noises and ensuring quality data for later filtering.

To implement the re-ranking algorithm, we first collect a set of documents that are already highly related to the knowledge domain, by adding a DCC to the query string. We then evaluate the *domain information richness* (DIR) of each document in reference

^{*} Farmingdale State College, State University of New York, New York, USA

[†] College of Staten Island, City University of New York, New York, USA

¹An earlier short version of the paper, entitled “A Domain-Specific Web Document Re-Ranking Algorithm,” was published In Proceedings of the 8th International Conference on E-Service and Knowledge (ESKM). July 9-13, 2017, Hamamatsu, Japan.

to the domain ontology to possibly re-rank and improve the search results.

The three information scores we calculate between two nodes in an ontology graph are: *distance* – the least number of hops from one node to another; *direction* – the directional relation between two nodes; and *relationship*, represented by edge labels. The *directional relation* is subsequently broken down into three semantic degrees: *granularity*, *diversity*, and *generality*, which we will elaborate in the later section.

The re-ranking process is only necessary if we use general-purpose search engines to perform the initial crawling. If using domain-specific search engines, a re-ranking is unlikely needed. However, domain-specific search engines are not necessarily available for all knowledge domains, or may be limited in data source. Additionally, the general-purpose search engines may have already indexed the data the domain-search engines are hosting. We therefore focus our research on the data set retrieved by authoritative general-purpose search engines.

Though our re-ranker and our prototype system can be applied to both textual and multimedia web documents, we focus our experiment on textual web documents, since textual documents tend to contain more searchable data (text and metadata) compared to multimedia web resources (mostly metadata only).

The terms “web document,” “web resource,” and “web page” are interchangeable throughout this paper. The paper uses dictionary *term* and domain *concept* interchangeably.

2 Related Work

Many researchers have explored web document re-ranking methodologies in the context of a knowledge domain. Researchers studied link text and structure [1, 2], document comparison [3, 4], machine learning [5, 6], and recommendation systems [7], in order to improve search results via re-ranking. Ontologies have been playing an important role in context-aware search [8], personalized search [9, 10], document ranking [11, 12] and re-ranking. Baziz et al. [13] used ontology structure for pair-wise document comparison, document indexing and query evaluation. Hawalah and Fasli [14] proposed a hybrid reranking algorithm that combines different information sources such as reference ontology, user profile and original search engine’s ranking.

Our algorithm employed the notions of “generality,” “granularity,” and “diversity,” which are the theoretical building blocks of our proposed methodology. Those concepts have been well used and studied in prior literature, however, with varied connotations.

A concept-based computational algorithm [15] was developed to estimate the “semantic granularity” of documents with reference to domain ontology, along with “similarity” and “popularity” measures. The paper introduces an ontology-based computational model that estimates document and query granularity by analyzing the semantic contents captured in a document. The authors defined document *granularity* as the levels of “semantic generality or specificity conveyed by documents.” The statement seems to give a contradictory assertion to two seemingly antonymous terms: granularity and specificity.

The proposal of *subtopical retrieval* [16] was an automatic granularity-based document ranking algorithm. It argued that, in cases such as a literature survey, documents that cover as many different subtopics of a general topic as possible need to be found.

The term “diversity” appeared in [4], in which Liu et al. introduced a re-ranking algorithm, called Affinity Rank. The rank was based on two aspects: the topic coverage of a group of documents and the amount of information contained in a document. The former was the diversity aspect, measuring how many different topics covered in a document group while the latter was represented as the *information richness score* that uses Vector Space Model and cosine similarity to calculate the similarity between documents. The scores then are penalized by the diversity factor and the documents are re-ranked based on the final scores.

Huang and Hu [5] promoted information diversity in their re-ranking algorithm in the biomedicine domain. The algorithm computes the maximum probability of its hidden properties corresponding to each retrieved passage iteratively until all subsets achieve stability, and then these passages are re-ranked from different subsets.

Just as “granularity” and “diversity” that were used by researchers in different context and defined scopes, “generality” can be interpreted differently when it comes to web re-ranking algorithms. Allen and Wu [17] measured document generality based on the mean generality of domain concepts contained in a document. The notion “generality” was defined as “general things or concepts,” which are easier for individuals to understand than are jargon-filled technical documents. They stated that their measurement based on WordNet ontology can detect the more general term in a hierarchy structure. For instance, the term “animal” was computed as a more “general” term compared to the term “dog.”

Yan et al. [18] defined overall document generality as “broad enough to cover as many different aspects of a certain topic as possible.” Their re-ranking algorithm computed a combined score of similarity and generality of the document related to the query. The authors tried to improve the query performance of domain-specific (biomedical literature in this paper) information retrieval by re-ranking retrieved documents on generality. A conceptual marking tree was suggested to help define a generality score.

The final generality score is calculated based on the *document scope* – how broad or vague a document is for describing a certain topic and *document cohesion* – the state or quality that the elements of a text “hang together.”

$$generalityScore(d_i) = \frac{Scope(d_i)}{Cohesion(d_i) + 1},$$

The document scope $Scope(d_i)$, normalized within the range of 0 and 1, is given by

$$Scope(d_i) = e^{-\frac{\sum_{i=1}^n depth(c_i)}{n}},$$

where n is the total number of concepts of both the ontology concepts and general concepts. $depth(c_i)$ is the distance between concept i and the root of the ontology tree.

$$Cohesion(d_i) = \frac{\sum_{i,j=1}^n Sim(c_i, c_j)}{NumberofAssociations}, (n > 1, i < j).$$

$$Sim(c_i, c_j) = -\log \frac{len(c_i, c_j)}{2D},$$

where D is the ontology tree maximum depth, $len(c_i, c_j)$ is the shortest path between concepts i and j in the ontology tree.

$$NumberofAssociations = \frac{n(n-1)}{2},$$

where n is the total number of ontology concepts in a document d_j .

In our experiments, since the above generality algorithm shares many similarities of our algorithm, we use it as our base line algorithm.

Another instrumental concept we use in our algorithm is augmented query.

ARCH [12] enhances a users initial query based on the users interaction with a modular concept hierarchy. The domain-specific concept hierarchy requires the augmented query to integrate the apriori knowledge in the problem domain which may be unfamiliar to the user.

The IntelliZap system [1] utilizes the semantic network to extract keywords from the context surrounding the user-selected text. These keywords are added to the text to form an augmented query, leading to context-guided information retrieval.

The process of extracting keywords was rather complex and lengthy. The authors first performed 100 iterations of the K-means algorithm and vector-based semantic to build an adjacency matrix A , where $A(i, j)$ contained the number of iterations when words i and j were assigned to the same cluster. Then the matrix was altered based on the distances between words estimated by the WordNet-based metric. Finally, the authors identified the strongly connected components from matrix A to form the keyword cluster.

The question here is not the strenuous process of building the keyword cluster, but the augmented query string's impact on the search result. Since every single word (except for stop words) in the query string takes into account weighting the retrieved documents, if too many keywords are added to the query string, they might subjugate the original semantic meaning of the query.

Though we did not base our algorithm on any of the aforementioned researches in particular, we shared the conceptual understanding of semantic re-ranking with the authors on many levels. As we see that most of the studies focused on one or two aspects of the semantic features of the ontology structure, our proposed algorithm on the other hand takes all possible aspects into consideration to bring about a robust solution.

3 Semantic Hypothesis

3.1 The DCC

To score a quality re-ranking, the initial search result set should be in a clean slate that is with as few noisy data as possible. To make it possible, we select one word or a short

phrase that characteristically represents the domain knowledge being discussed, and add it to the seed query string. In doing so, we semantically narrow down the search scope and bring the search results within, or close to the intended knowledge domain. This step acts as a preparatory pruning process, to assure a set of quality data to start with and to be conducive to the later re-ranking process. We call this chosen auxiliary term the *domain characteristic classifier (DCC)*.

This is the simplest and for that reason the most effective way to roughly bring a search result within a knowledge domain, if we pick an appropriate DCC.

Since a DCC plays an instrumental role in an augmented query search, a rigorous procedure for the selection of such a term becomes necessary. We conclude a few core principles for DCC selection:

1. The DCC should be non-ambiguously representing the topical domain, and should be representing the domain knowledge as a whole, but not partially. One of the qualified DCC candidates should be the domain name itself. Another possible route of picking such a DCC is to scan the keywords from the definition of the domain knowledge.
2. The DCC should be terse, with the least amount of words, ideally one word only. This concern is to address the adherence and degree of loyalty to the intended meaning of the original query string. In other words, to keep DCC to its minimum presence, it will not heavily influence the search result, deviating the semantics from the initial search query.

What if we can not find one DCC that represents a domain to its totality. Can we use more than one DCCs in an augmented query? We experimented using two or more DCCs, and found that using more auxiliary keywords increases the degree of overpowering the original query terms, hence causing the search results to be less likely relevant to the original query search results. Therefore, we recommend to use only one DCC as the augmented query.

3. A jargon or a rare word is not preferred and should be avoided from being chosen as a DCC, since we rely on one or a few general-purpose search engines to perform our initial search, a common and familiar word may induce a better search result. For example, in our prototyping system, we built a platform in the domain of gerontology. “Gerontology” can be categorized as a “jargon,” which is not suitable for a DCC. After studying the definition of gerontology: “The scientific study of old age, the process of aging, and the particular problems of old people ²,” we consider “old age” to be representative in the gerontology domain and is a daily use phrase, therefore qualified to be a DCC candidate.

The technique looks similar to query expansion, however, with different purpose and implementation procedure. Query expansion concentrates on lexical meaning of the seed query string and reformulates the query string to include synonyms, stem-word-based morphological expressions, or simply rephrases the query. In fact, most of the general search engines have already implemented query expansion in their search algorithm. We on the other hand focus on bringing the search scope down to a domain space. To avoid confusion, we use query augmentation but not query expansion to illustrate our DCC concept.

²The Oxford Dictionary of Difficult Words

3.2 The Three-Dimensional Semantic Degrees

To further probe the amount of information contained in a document in relation to the query string and the domain of interest, we have identified three semantic degrees (weights): *gran-ularity*, *diversity*, and *generality*.

Information *granularity*, sometimes referred to as *sub-concepts* of the query term, is considered to contain information specialty and depth in a domain locality. Information *generality* provides a more overall view than the query string does in reference to the topic and the domain. *Granularity* and *generality* can be considered to be in close quarters in terms of information *relevance* per the query string. In our algorithm, we distribute more weight to *granularity* over *generality*, due to a domain-specific context, with the consideration that a domain expert tends to know more in-depth knowledge. Information *diversity* refers to the amount of information that is related to the query term, but it is neither more *general* nor more *granular* in the domain structure. We assign the *diversity* weight midway between *generality* and *granularity*. The postulation is that information *diversity* may speak with a more cohesive quality of the domain space than its *generality* counterpart, but contain less information comprehensiveness than the *granularity* property.

We call the total weight of the three semantic degrees the *Domain Information Richness (DIR)*.

4 The Re-Ranking Algorithm

4.1 Mathematical Notations

In order to articulate the informational and semantic degrees among nodes, we stipulate a domain ontology as a top-down hierarchical taxonomy structure, where the root node is the domain name (root class) and the inner nodes and the leaf nodes are the sub-classes or the instances.

Definition of domain ontology: Let $G = (V, E, L)$ be a rooted, directed tree (the ontology graph), where V is the set of nodes/vertices in the tree. Each node $v \in V$ represents a core concept in the domain ontology, v is composed with a name (term), or multiple names (synonyms) and a description. V is the dictionary of the domain. E is a collection of all edges (arcs) in G . All edges in the graph are labeled. L is a set of labels in G , which describe the taxonomic relations between nodes. If the query string is mapped to a node i in G , we call node i the *reference node*.

Definition of knowledge: We interpret G as a knowledge or a knowledge domain. The domain name is rN . A concept, or a dictionary term, is a node in G , the relationship between two adjacent nodes is defined by the edge label between them. The edge is a directed edge. The edge direction is always pointing to the direction away from rN .

Definition of augmented query: Let Γ be a set of domains, and some generic query q . A domain $\gamma \in \Gamma$, its *domain characteristic classifier* is τ_γ , implying $\tau_\gamma \in \gamma$. The *augmented query* is $q_\gamma = q \cdot \tau_\gamma$, where \cdot denotes string concatenation, and hence, $q_\gamma \in \gamma$.

Definition of initial rank order: The initial rank order $R_{init(i)}$ per an augmented query q_γ is a document collection $R_{init(i)} = \{d_m | m = \{1, 2, \dots, N\}\}$, where *reference node* i in the ontology graph of domain γ is corresponding to q . N is the number of retrieved documents. Document d_1 ranks the highest, and d_N the lowest.

Link Relations

Link relations among all nodes in G can be represented by a *link relation matrix*, an adjacency $|V| \times |V|$ matrix M . Each entry in M represents the weight of a link between two nodes, i and j . Let \hat{M} be the normalized M , where the sum of each row is 1. Please note: $(i, j) = 0$, if $i = j$. Thus in the matrix \hat{M} , all diagonal entries are zeros.

We demarcate the link relations into three major categories between two distinct nodes i and j in G : *distance relation*, *directional relation*, and *attributive relation*.

Let $Dep(G)$ be the height of the ontology graph, which is the longest distance between the root node and a leaf node in G . The distance between nodes i and j is $d(i, j)$, a non-negative integer, which is the number of edges between the two nodes along the shortest path. In turn $d(i, j)$ defines the *distance relation* between the nodes i and j .

The *directional relation* is comprised of the three information degrees: *generality*, *granularity*, and *diversity*. The three degrees are represented in an ontology graph as *parent/ancestor* nodes, *child/descendant* nodes, and *sibling/remote relative* nodes, respectively:

1. *Parent/ancestor*: Node j is said to be an *ancestor* of node i , if j is on the (shortest) path from root to i in G , and $j \neq i$. Node j is the *parent* if it is adjacent to node i . The relationship denotes the *generality* degree of node j in reference to node i .
2. *Child/descendant*: Node j is said to be a *descendant* of node i , if i is on the (shortest) path from root to j in G , and $j \neq i$. Node j is a *child* of node i if it is adjacent to i . The relationship denotes the *granularity* degree of the node j in reference to node i .
3. *Sibling/remote relative*: Nodes i and j are said to be *siblings* if they have a common *parent* in G , and $j \neq i$. The relationship implies *diversity* degree of node j in reference to node i .

Node j in G is said to be a *remote relative* of node i , if node j is not a *parent/ancestor*, or a *child/descendant*, or a *sibling* of node i . The relationship also implies the *diversity* degree of node j in reference to node i .

The directional relation of the aforementioned three semantic degrees per a reference node is illustrated in Figure 1 and Figure 2 using our gerontology ontology, generated via OBO-Edit.³

The *attributive relation* concerns the link label properties between nodes m and n . If there is an edge $(m, n) \in E$, the label $l(m, n) \in L$ defines the *attributive relation* between m and n . There are two major labels in our prototyping system: *is-a* (a subclass) and *part-of* (an instance, an individual, a leaf node).

³OBO-Edit is an open source ontology editor written in Java.

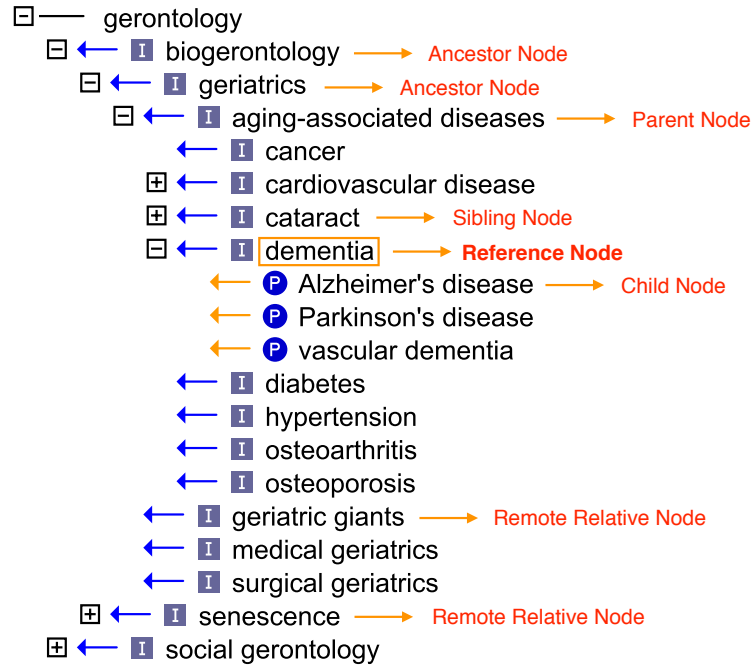


Figure 1: Partial View of Gerontology Ontology (Tree View)

An instance of a concept is said to be more specific than a subclass of a concept. Therefore, we assign more weight to meronymy *part-of*-labeled nodes than *is-a*-labeled nodes.

4.2 The Proposed Algorithm

The re-ranking algorithm is simply carried out to a candidate web page d_m in three major steps:

1. Computation of the *base score* θ based on d_m 's original ranking position.
2. Computation of the *Domain Information Richness* (DIR) score ω of d_m . More specifically, how much *distance*, *direction*, and *attributive* relation between reference node i and any other node j in G does d_m hold?
3. Computation of the new rank R' of d_m .

4.2.1 Base Score

In order to make the Base Score more computation-friendly and with more judicious semantic sense, we bend the uniformly distributed linear line of the initial rank order $R_{init(i)}$ into a decreasing curve with non-uniformed distribution – giving the top k ranked items special concentration, in concave shape, and the rest in convex shape (See Figure 3 for illustration). Such operation aligns with *search engine persuasion*⁴ moral.

⁴Search engine persuasion refers to the phenomenon that there may be millions of sites pertaining in some manner to broad-topic queries, but most users will only browse through the first k (e.g.: 10) results returned by the search engine.

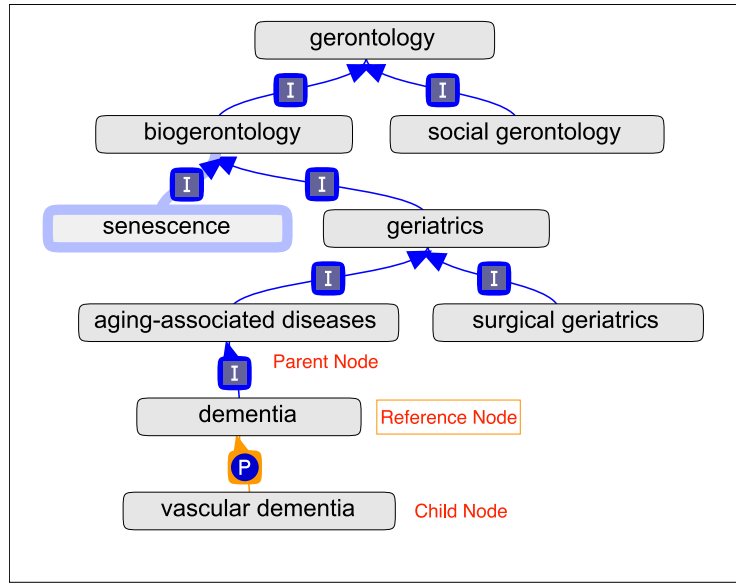


Figure 2: Partial View of Gerontology Ontology (Graph View)

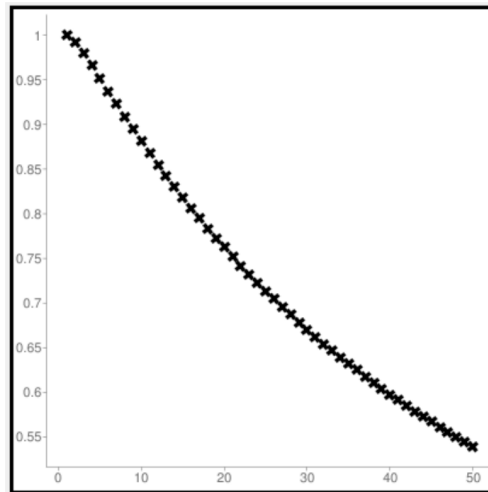


Figure 3: Scatter Plot of Base Scores (N=50, b=10)

The *base score* $\theta(m)$ is computed as such:

$$\theta(m) = \begin{cases} 1, & \text{if } m = 1, \\ \frac{N+2*\log_b(m+1)}{m+N}, & \text{otherwise,} \end{cases} \quad (1)$$

where N is the total number of retrieved documents, $1 \leq m \leq N$ is the original rank position, and b denotes the base of the logarithm. Presumably, b can be considered the top b articles of high quality in authority and relevance in ranking.

The above formula is one of the theoretical contributions of our research.

4.2.2 Domain Information Richness (DIR) Score

We now compute the distance, direction, and relationship scores of document d_m .

Distance

The distance score can be considered as the similarity measure between reference node i and some node j in G , where $i \neq j$. We adopted the popular Leacock Chodorow Similarity algorithm [19], a logistic function measuring the shortest path between two concepts, as the basis of our distance score formula. The basic intuition of the Leacock-Chodorow function is that the semantic similarity between two concepts is estimated based on the conceptual links (i.e., the distance) between these concepts in ontology.

The distance (similarity) score between reference node i and some node j is defined as:

$$sim(i, j) = -\log_2 \frac{d(i, j)}{2Dep(G)}, \quad (2)$$

where $d(i, j)$ is the distance between nodes i and j .

This is a monotonically decreasing function with respect to the increasing neighborhood distance between *reference node* i and any other node j allotted in G .

Direction

Score

We assign weights to the three sub-degrees of *directional relation* - *granularity*, *diver-sity*, and *generality*, based on our hypothesis stated in Section 3.2.

Let v_{gr}, v_{di}, v_{ge} be the weights assigned to *granularity*, *diversity*, and *generality*, respectively, where $v_{gr} \geq v_{di} \geq v_{ge}$. For simplicity, let $v_i(j)$ denote one of the aforesaid directional weights of node j in reference to node i .

In our experiment, we set and tuned the three weights as follows:

$$v_i(j) = \begin{cases} v_{gr} = 0.15, & \text{if } j \text{ is a } child/descendant \text{ of node } i, \\ v_{di} = 0.13, & \text{if } j \text{ is a } sibling/remote \text{ relative of node } i, \\ v_{ge} = 0.11, & \text{if } j \text{ is a } parent/ancestor \text{ of node } i. \end{cases}$$

We set the upper bound of the combined weight rates to 40% (0.4), in order to not overpower the original rank order. The total of the above coefficients is $0.15 + 0.13 + 0.11 = 0.39$.

The direction score $drn_i(j)$ between *reference node* i and some node j is computed as:

$$drn_i(j) = v_i(j) \times sim(i, j). \quad (3)$$

Relationship Score

The relationship weight, v_{re} , should be assigned a fraction of $v_i(j)$, since the value is additive to any of the directional weights. We set $v_{re} = 0.02$.

The relationship score, $attr(j)$, is defined as:

$$attr(j) = \begin{cases} v_{re} \times sim(i, j), & \text{if } j \text{ is an instance node,} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Link Relation Matrix

With the distant score and the directional coefficients in place, we can compute the square matrix M with the link relation weights.

$$M(i, j) = \begin{cases} 0, & \text{if } i = j, \\ drn_i(j) + attr(j), & \text{otherwise.} \end{cases} \quad (5)$$

Let $\|M_i\|_1 = 1$, where $\|M_i\|_1$ denotes the L1 norm of M_i , the i th row in M , then we get a normalized matrix \hat{M} .

Note that this matrix (see Algorithm 1) can be pre-calculated and cached before any re-ranking computation takes place.

Algorithm 1: How to compute link relation matrix

Data: Nodes and (labeled) Edges in G

Result: linkRelationMatrix

[Declaration];

HashMap linkRelationMatrix;

[Initialization];

Node nodes[V] \leftarrow [input from database];

$Dep(G)$ = [height of ontology tree];

$v_{gr}, v_{di}, v_{ge}, v_{at}$ \leftarrow [weight constants];

while (loop V : index i) **do**

 Array row[V];

while (loop V : index j) **do**

 compute $v_i(j)$;

 compute $d(i, j)$;

 compute $attr(j)$;

 row[j] = $v_i(j) * (-\log_2 \frac{d(i, j)}{2 * Dep(G)}) + attr(j)$;

 normalize(row);

 linkRelationMatrix[i] = row;

Domain Information Richness (DIR) Score

We use DIR score to measure how much information a single document contains in its topic domain locality. The DIR score of d_m per reference node i is the sum of all direction and relationship scores mapped to d_m .

Let v_{i, d_m} be a vector composed with 0s and 1s, $|v_{i, d_m}| = |V|$. If d_m contains node j , the corresponding j th element in v_{i, d_m} will be set to 1, 0 otherwise. Let g be the number of 1s in v_{i, d_m} .

The initial DIR score of d_m per reference node i is defined as:

$$\omega_i^0(m) = \langle \hat{M}_i, v_{i, d_m} \rangle, \quad (6)$$

where \hat{M}_i is the i th row of \hat{M} , $\langle \hat{M}_i, v_{i, d_m} \rangle$ is the inner product of vectors \hat{M}_i and v_{i, d_m} .

We further define the DIR score of d_m per i as:

$$\omega_i(m) = \begin{cases} 0, & \text{if } g = 0, \\ \omega_i^0(m)/\log_b(g+1) \times 10, & \text{if } g \geq 1, \end{cases} \quad (7)$$

where b is the base of the logarithm. The $g + 1$ is to avoid the divisor being 0. Due to normalization, the DIR value tends to be far smaller than the *base* score θ , so we increase it tenfold.

We use an inverted index as the coefficient here to penalize a high number of occurrences of matched ontology terms in a document. This could happen to a large document (e.g., a 60-page research article) or a tutorial that may match 20+ ontology terms in its textual body.

4.2.3 Re-Ranking Score

The new ranking score of d_m in domain γ with reference node i (in correlation with the query string) is a linear combination of *base score* θ and the DIR score ω .

$$R'_i(m) = \alpha \times \theta(m) + (1 - \alpha) \times \omega_i(m), \alpha \in [0, 1], \quad (8)$$

where α is the adjusting parameter (currently set to 0.85, similar to the formulation of PageRank [20, 21]).

5 Experiment, Methods and Results

5.1 Dataset and Queries

We built an experiment system in the domain of gerontology and created the initial prototyping ontology, using OBO-Edit software. The ontology language therefore was life science preferred (gene-annotation-initiated) late ontology language: OBO, which stands for Open Biomedical Ontologies. There are 121 terms (concepts) in the graph and the graph depth is 7.

Restricted by our limited resources, we chose three query strings – “fall prevention,” “calorie restriction,” and “dexterity” – three nodes from the ontology graph, and 30 human evaluators for the experiments. We used “old age” as the DCC at the time of the experiments.

We chose Google as the generic search engine and collected top ten documents of each augmented query string. We gave the three sets of top ten documents (Table 1) that Google returned (we retrieved the first two queries on June 1, 2016 and the third query on Feb 20, 2017) to the 30 evaluators, and asked them to hand pick the top five ranked articles per topic. The selections were based on relevance, comprehensiveness, and helpfulness in the domain of gerontology. We chose to have a small number of documents to be evaluated in an attempt to minimize human errors (false positives or false negatives).

With the 30 graders’ top five ranked articles, we were able to obtain the graded ranking order of the ten articles per each augmented query.

Table 1: Web Documents Used in Our Experiments

RN	URL
f-p	http://www.who.int/ageing/publications/Falls_prevention7March.pdf http://www.who.int/ageing/projects/falls_prevention_older_age/en/ http://www.cdc.gov/homeandrecreationalafety/falls/ http://www.mayoclinic.org/healthy-lifestyle/healthy-aging/in-depth/fall-prevention/art-20047358 http://nihseniorhealth.gov/falls/aboutfalls/01.html https://www.ncoa.org/healthy-aging/falls-prevention/preventing-falls-tips-for-older-adults-and-caregivers/6-steps-to-protect-your-older-loved-one-from-a-fall/ https://www.ncoa.org/healthy-aging/falls-prevention/preventing-falls-tips-for-older-adults-and-caregivers/debunking-the-myths-of-older-adult-falls/ http://orthoinfo.aaos.org/topic.cfm%3Ftopic%3Da00135 http://ageing.oxfordjournals.org/content/35/suppl_2/ii37.short%3Frss%3D1%26source%3Dmfc (no longer available) https://www.nia.nih.gov/health/publication/falls-and-fractures
c-r	http://ajcn.nutrition.org/content/78/3/361.full https://www.fightaging.org/archives/2002/11/calorie-restriction-explained/ http://www.nature.com/articles/ncomms4557 https://www.nia.nih.gov/newsroom/topics/calorie-restriction http://www.ncbi.nlm.nih.gov/pubmed/24691430 http://www.nih.gov/news-events/news-releases/nih-study-finds-calorie-restriction-lowers-some-risk-factors-age-related-diseases https://en.wikipedia.org/wiki/Calorie_restriction http://www.lifeextension.com/protocols/lifestyle-longevity/caloric-restriction/Page-01 http://www.livescience.com/2666-live-longer-anti-aging-trick-works.html http://www.crsociety.org/
dx	http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4331509 http://biomedgerontology.oxfordjournals.org/content/58/2/M146.full.pdf http://apk.hhp.ufl.edu/wp-content%5Cuploads/Kornatz.pdf http://etheses.bham.ac.uk/447/1/Martin09PhD.pdf https://en.wikipedia.org/wiki/Aging_movement_control https://en.wikipedia.org/wiki/Fine_motor_skill http://www.inclusivedesign toolkit.com/betterdesign2/UCdex/dex.html http://www.dictionary.com/browse/dexterity http://themotorstory.org/building-strength-and.../finger-development-for-dexterity/ http://jap.physiology.org/content/94/1/259

RN: reference node. f-p: fall prevention; c-r: calorie reduction; dx: dexterity.

We first assigned some linearly-increasing weights to the top five ranks respectively (see Table 2). Then we computed the total score of each article among the ten returned articles by adding up the products of the number of votes and the corresponding linear weight when the articles were chosen for any place in the top five. In detail, we computed the total score of each document as:

$$w_j = \sum_{i=1}^5 (\text{No. of votes}) \times \text{weight}_i, j \in [1, 10].$$

Table 2: Linear Rank Weights

Rank No.	1	2	3	4	5
Weight	0.5	0.4	0.3	0.2	0.1

We sorted the total scores of the ten documents in descending order to obtain the evaluators' graded ranking order.

5.2 Evaluation Methodology and Experiment Results

The 30 graders returned a total of 35 answer sheets for the three queries. Among the 35 submissions, there were ten valid entries for "calorie restriction," 14 for "fall prevention," and eight for "dexterity," while seven entries were invalid. Table 3 shows the computation results of our re-ranking algorithm for the top ten retrieved documents of each query. Table 4 displays the new ranking results from the evaluators and our re-ranking algorithm.

We used the Document Generality (DG) re-ranking algorithm [18] as our experiment baseline. The algorithm follows a similar trail as ours, in that it extracts matching ontology terms from text, and uses ontology structure as the the oracle to spin out a more domain-relevant rank order. The result is presented in Table 5.

To compare the ranking results, we adopted popular ranking evaluation methodologies such as Pairwise Error Probability (e.g.: Kendall Tau Distance), Root Mean Square Error/Cosine Similarity, Discounted Cumulative Gain (DCG), and Mean Squared Error (MSE).

The Normalized Discounted Cumulative Gain is calculated as the following:

$$\sum_{i=1}^p \frac{2^{rel_i} - 1}{\log(i+1)},$$

where rel_i is the user graded relevance of the result at position i . The logarithmic reduction is to penalize the lower rank in the research result.

For easy calculation and maintain the result into a small range, we used the inverted rank as the rank order before plugging it into the formula. For example, for rank order $\{1, 2, 3, 4, 5\}$, we use $\{1, 1/2, 1/3, 1/4, 1/5\}$ instead.

Table 3: Re-Ranking Scores

InitRank	f-p	c-r	dx
1	1.11185282741	0.97949153524	0.982258798424
2	0.832905887174	1.00597922231	0.832905887174
3	0.821198150665	0.94581726568	0.861589664529
4	0.947279042998	0.879452495662	0.977483471286
5	0.796779220466	0.796779220466	0.841246303145
6	0.883024207382	0.784583333358	0.803428056725
7	0.933640309711	0.937930424779	0.848699847134
8	0	0.873122550855	0.778999540533
9	0.843317600717	0.813930326999	0.820624694494
10	0.839634438398	0.771910780312	0.774130886433

initRank: Google rank order; f-p: fall prevention; c-r: calorie reduction; dx: dexterity.

Note: When we conducted the survey, document number 8 with “fall prevention” was no longer available online. We crossed it out and did not count it for evaluation.

Table 4: Ranking Results

f-p	GradedRank	4	5	10	1	3	7	6	9	2	[8]
	ReRank	1	4	7	6	9	10	2	3	5	[8]
	DGRank	1	7	10	4	9	6	2	3	5	[8]
c-r	GradedRank	1	2	6	8	7	9	3	10	5	4
	ReRank	2	1	3	7	4	8	9	5	6	10
	DGRank	7	2	1	8	3	9	4	10	5	6
dx	GradedRank	1	4	6	10	9	5	7	3	8	2
	ReRank	1	4	3	7	5	2	9	6	8	10
	DGRank	4	1	5	3	7	10	9	6	8	2

f-p: fall prevention; c-r: calorie reduction; dx: dexterity.

GradedRank: Rank order by evaluators.

ReRank: Rank order by our re-ranking algorithm.

DGRank: Rank order by Document Generality re-ranking algorithm.

Table 5: Evaluation Results
Numbers in bold are the best performed in the list

		DCG	COSINE	KT	MSE
f-p	ReRank	1.2521	0.1099	0.4444	1.0298
	DGRank	1.0975	0.0899	0.4722	1.0380
	*Google	1.0897	0.0879	0.4722	1.0392
c-r	ReRank	1.4007	0.1228	0.3555	1.0147
	DGRank	1.1520	0.0950	0.3111	1.0292
	*Google	1.6147	0.1488	0.4667	1.0262
dx	ReRank	1.5545	0.1568	0.4222	1.0228
	DGRank	1.4087	0.1355	0.2888	1.0161
	*Google	1.4683	0.1415	0.6222	1.0305

f-p: fall prevention; c-r: calorie reduction; dx: dexterity.

DGRank: Document Generality re-ranking algorithm.

ReRank: Our re-ranking algorithm. *Google: Google base rank.

5.3 Experiment Result Analysis

From the results of our experiments (Table 5), we could see that DG algorithm only outperformed our re-ranking algorithm in two (f-p:KT and dx:MSE) out of the twelve experimental categories (DCG, COSINE, KT, MSE for f-p, c-r, and dx, respectively); our algorithm was superior to or equally good for the rest. It showed a promising performance of our re-ranking algorithm.

Another observation was made through comparing our new rank with Google rank order. Though we did not treat Google initial rank as one of our baselines, we did calculate its performance as a reference point. Our re-ranker did better on “Fall prevention” and “dexterity” among the three queries over Google, which nonetheless purports levels of viability of our re-ranking algorithm.

We set the new rank linear combination coefficient to 0.85 due to a conservative attitude because we want to respect the initial rank order. However, Table 6 shows a strong correlation between documents and the ontology dictionary terms. One of the documents with query “Fall Prevention” contained 27 matched dictionary terms in its text. Among the total thirty documents, apart from one document that could not be retrieved at the time of experiment, there were only two documents that contained zero matched ontology term. The coefficient therefore could be tuned to a level that is more in favor of the re-ranker.

Table 6: Number of Matched (Ontology)Terms Found in Each Retrieved Document (Table 1)

initRank	1	2	3	4	5	6	7	8	9	10
f-p	27	0	5	13	12	5	11	NULL	6	12
c-r	9	13	7	4	5	13	15	9	6	2
dx	9	0	1	15	5	2	4	1	3	5

initRank: Google rank order.

f-p: fall prevention; c-r: calorie reduction; dx: dexterity.

Note: When we conducted the survey, document number 8 with “fall prevention” was no longer available online.

6 Conclusion

Our re-ranking algorithm attempts to explore the implied semantic meanings of a document through seeking the innate relations between the retrieved document and the domain ontology structure. By using DCCs, the algorithm starts off straightly with a domain-specific search.

The algorithm can be used not only to provide data feed to a knowledge base, but also, with some minor adjustment, to detect a document’s topical domain. In the latter case, there is no need to involve DCCs. The document is to be checked against different domain ontologies to compute a DIR score. The ontology with the highest DIR score may indicate the most likely domain to which the document belongs.

Though our experiments have shown a promising result of our re-ranking algorithm, we need to collect more data to continue evaluating and improving the algorithm. Applications of the algorithm are necessary to prove its validity and practicality.

References

- [1] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, “Placing search in context: The concept revisited,” in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 406–414.
- [2] V. W. Tam and J. Shepherd, “Webpage relationships for information retrieval within a structured domain,” in *Proceedings of the 21st ACM conference on Hypertext and hypermedia*. ACM, 2010, pp. 307–308.
- [3] E. Krikon, O. Kurland, and M. Bendersky, “Utilizing inter-passage and inter-document similarities for reranking search results,” *ACM Transactions on Information Systems (TOIS)*, vol. 29, no. 1, p. 3, 2010.

- [4] Y. Liu, B. Zhang, Z. Chen, M. R. Lyu, and W.-Y. Ma, "Affinity rank: a new scheme for efficient web search," in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*. ACM, 2004, pp. 338–339.
- [5] X. Huang and Q. Hu, "A bayesian learning approach to promoting diversity in ranking for biomedical information retrieval," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 307–314.
- [6] C. Kang, X. Wang, J. Chen, C. Liao, Y. Chang, B. Tseng, and Z. Zheng, "Learning to re-rank web search results with multiple pairwise features," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 735–744.
- [7] W. Li, "Domain-specific information retrieval using rcommenders," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 1327–1328.
- [8] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li, "Context-aware ranking in web search," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 451–458.
- [9] A. Sieg, B. Mobasher, and R. Burke, "Web search personalization with ontological user profiles," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007, pp. 525–534.
- [10] F. Qiu and J. Cho, "Automatic identification of user interest for personalized search," in *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006, pp. 727–736.
- [11] M. Daoud, L. Tamine, and M. Boughanem, "A personalized graph-based document ranking model using a semantic user profile," in *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 2010, pp. 171–182.
- [12] L. Tamine-Lechani, M. Boughanem, and N. Zemirli, "Personalized document ranking: Exploiting evidence from multiple user interests for profiling and retrieval." *JDIM*, vol. 6, no. 5, pp. 354–365, 2008.
- [13] M. Baziz, M. Boughanem, G. Pasi, and H. Prade, "An information retrieval driven by ontology from query to document expansion," in *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2007, pp. 301–313.
- [14] A. Hawalah and M. Fasli, "A hybrid re-ranking algorithm based on ontological user profiles," in *2011 3rd Computer Science and Electronic Engineering Conference (CEECE)*. IEEE, 2011, pp. 50–55.
- [15] X. Yan, R. Y. Lau, D. Song, X. Li, and J. Ma, "Toward a semantic granularity model for domain-specific information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 29, no. 3, p. 15, 2011.

- [16] C. X. Zhai, W. W. Cohen, and J. Lafferty, “Beyond independent relevance: methods and evaluation metrics for subtopic retrieval,” in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2003, pp. 10–17.
- [17] R. B. Allen and Y. Wu, “Generality of texts,” in *International Conference on Asian Digital Libraries*. Springer, 2002, pp. 111–116.
- [18] X. Yan, X. Li, and D. Song, “Document generality: its computation for ranking,” in *Proceedings of the 17th Australasian Database Conference-Volume 49*. Australian Computer Society, Inc., 2006, pp. 109–118.
- [19] C. Leacock and M. Chodorow, “Combining local context and wordnet similarity for word sense identification,” *WordNet: An electronic lexical database*, vol. 49, no. 2, pp. 265–283, 1998.
- [20] L. Page, S. Brin, R. Motwani *et al.*, “The pagerank citation ranking: Bring order to the web,” Stanford University, Tech Rep, 1997: 0072, Tech. Rep.
- [21] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.” Stanford InfoLab, Tech. Rep., 1999.