An Improved Web Explorer using Explicit Semantic Similarity with ontology and TF-IDF Approach

Amit R. Rajeshwarkar, Meghana Nagori

[1]Department of CSE, Government Engineering College, Aurangabad, INDIA

Abstract—*The Improved Web Explorer aims at extraction and selection of the best possible hyperlinks and retrieving more accurate search results for the entered search query. The hyperlinks that are more preferable to the entered search query are evaluated by taking into account weighted values of frequencies of words in search string that are present in anchor texts and plain texts available in title and body tags of various hyperlink pages respectively to retrieve relevant hyperlinks from all available links. Then the concept of ontology is used to gain insights of words in search string by finding their hypernyms, hyponyms and synsets to reach to the source and context of the words in search string. The Explicit Semantic Similarity analysis along with Naïve Bayes method is used to find the semantic similarity between lexically different terms using Wikipedia and Google as explicit semantic analysis tools and calculating the probabilities of occurrence of words in anchor and body texts .Vector Space Model is being used to calculate Term frequency and Inverse document frequency values, and then calculate cosine similarities between the entered Search query and extracted relevant hyperlinks to get the most appropriate relevance wise ranked search results to the entered search string.*

Keywords—Web-Explorer, TF-IDF, Lexical, Ontology, Explicit Semantic similarity analysis.

I. INTRODUCTION

Web Explorer is nothing but a web spider that traverses various links available on the world wide web. The widely spread web represents voluminous data from different contexts, different geographical locations and different intents. However retrieval of precise data based on the intent of the search is of utmost important to maintain the interests in web traversal. The task of prime importance is to traverse the hyperlinks on the web and find out the seed links that might be of interest to the search. The selection of the hyperlink is essential to ascertain the relevance of web pages with the entered search string. The ascertained links are classified into two types: the seed URLs from the Internet and the updated URLs from the unvisited list [1]. The seed URLs that are related to a search string can be extracted from the gathered search results, which are retrieved by appending google provided api to gather search results i.e. http://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=YourSearchStringHere.The first task is to find file type of pages linked with the urls. Each Page however has title text anchor text and body text. These types of text can be used efficiently by assigning weights and considering frequencies of words in search string to calculate relevance of linked page with the search string.

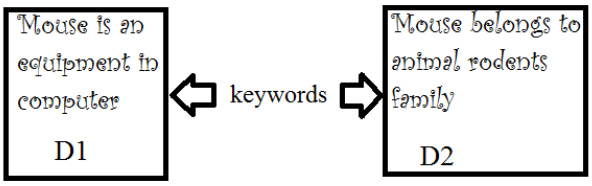
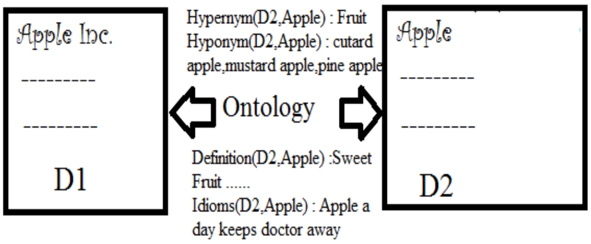

Mouse is an equipment in computer
D1
keywords
Mouse belongs to animal rodents family
D2

Fig.1: VSM, Mouse Keyword matches the two documents but the context is different.

Vector Space model comes in use to calculate correlation between search string and pages by calculating search vector and page vectors of traversed links. If Search string contains few terms that are also present in explored web pages, then the Term frequencies and Inverse document frequencies can be calculated easily, but if both have no common terms then Vector space model fails to calculate tf-idf because tf will be 0. Also, two lex-ically different terms does not mean that they must be representing two different things as they may be semantically similar [2].


Apple Inc.

D1
Hypernym(D2,Apple) : Fruit
Hyponym(D2,Apple) : cutard apple,mustard apple,pine apple
Ontology
Definition(D2,Apple) :Sweet Fruit ......
Idioms(D2,Apple) : Apple a day keeps doctor away
Apple

D2

Fig.2: Ontology, Different ontologies of apple are considered but Apple and Apple Inc cannot be contextually differentiated.

Thus ontology is used to find out the synonyms, hypernyms and hyponyms to be able to gain more insight of the context of the search query. For e.g. apple has hyponyms as pine apple, custard apple, etc. and its hypernym is fruit, apple may have a synonym as its biological name. The definition of apple may contain some words that represent some other fruit with same properties that search might be interested in such as sweet fruit.

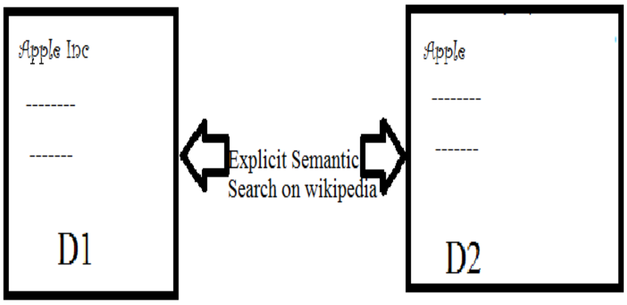


Fig.3: Explicit Semantic Search, Wikipedia can help us distinguish between Apple Fruit and Apple Company

Some of the technical terms or names of companies cannot be found in dictionaries nor have any relations like hypernyms, hyponyms and synonyms. In such case Semantic analysis from an external source can be considered such as Wikipedia or Google. Semantic Similarity can be calculated by again taking into consideration different contexts of the same keyword by calculating naïve Bayesian probabilities for links in Wikipedia or Google representing different contexts.

Thus the proposed system is the combination of Vector Space Model, Ontology based Semantic Similarity and Explicit Semantic analysis of the entered search string.

Our model is helpful in determining more accurate search result links by traversing its contents checking ontologies, determining semantic similarity from external sources such as Wikipedia then computing Tf-Idf values and finally aggregating the weighted average to obtain most relevant resulting links.

II. RELATED WORK

Our model extracts the most relevant hyperlinks to an entered search query. This can be achieved by aggregating results of Vector Space Model, Semantic Similarity model using ontologies and External Source and computing the cosine similarity between traversed links and the search query.

TF-IDF Approach: Term Frequency Inverse Document Frequency approach sequentially first of all calculates keywords by removing stop words from documents such as a, an, the, is, for, etc. Then for remaining terms which are said to be keywords form the query are the term frequency from each document for each keyword is found out.

$$\text{RelScore}_d = \sum_{t \in q \cap d} w_{t_d}, w_{t_d} \equiv TF_{t,d} \cdot IDF_t$$

Where,

Term frequency TF(t,d) = Frequency of occurance of term t in document D .

The term frequency is normalized in huge datasets by dividing the frequency count with total count of the term t in all documents .

And Inverse Document Frequency (IDF) of term t is given by

IDF(t) = Log (f(t)/N)

Where f(t) is frequency of occurance of term t and N is total number of documents.

RelScore(d) is the TFIDF value of document which can also be called as the document vector.

Now the same procedure is followed to calculate query vector for entered search string where term frequency of absent terms is taken as 0 and multiplying it with IDF values previously calculated.

Now the similarity between Document vectors and Query vector is calculated by vector space model to compute cosine similarity using

$$Sim(q, d) - \frac{\sum_i q_i d_i}{\sqrt{\sum_i q_i^2}\sqrt{\sum_i d_i^2}},$$

Where q is the query vector and d is document vector

Semantic Similarity with Ontology:

Semantic similarity model works in three steps

The weight qi of each query term i is adjusted based on its relationships with other semantically similar terms j within the same vector

$$q'_i - q_i + \sum_{\text{sim}(i,j)>t}^{j \neq i} q_j \text{sim}(i, j),$$

Where t is a user defined threshold (t= 0:8 in this work). Multiple related terms in the same query reinforce each other (e.g., “railway”, “train”, “metro”). The weights of non-similar terms remain unchanged (e.g., “train”, “house”). For short queries specifying only a few terms the weights are initialized to 1 and are adjusted according to the above formula.[3]

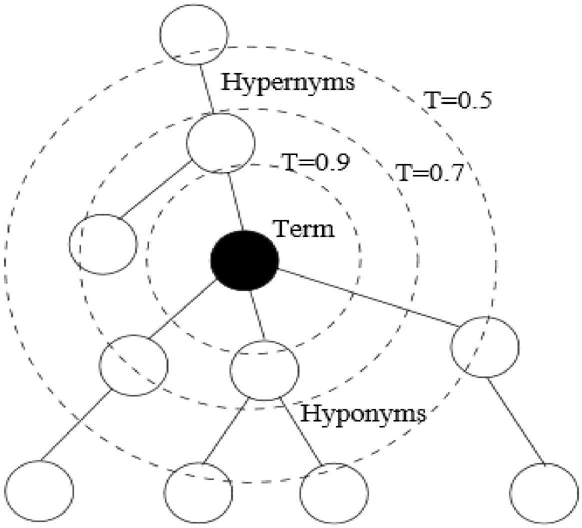


Fig.4: Weights for assigning qi in above formula depending on relationship whether Hypernym, Hyponym or synonym.

Hypernym (pigeon)=bird, a generalized term which may act as source word for contextual analysis.

Hyponym(pigeon)=sparrow, crow, etc. other elements of under same class.

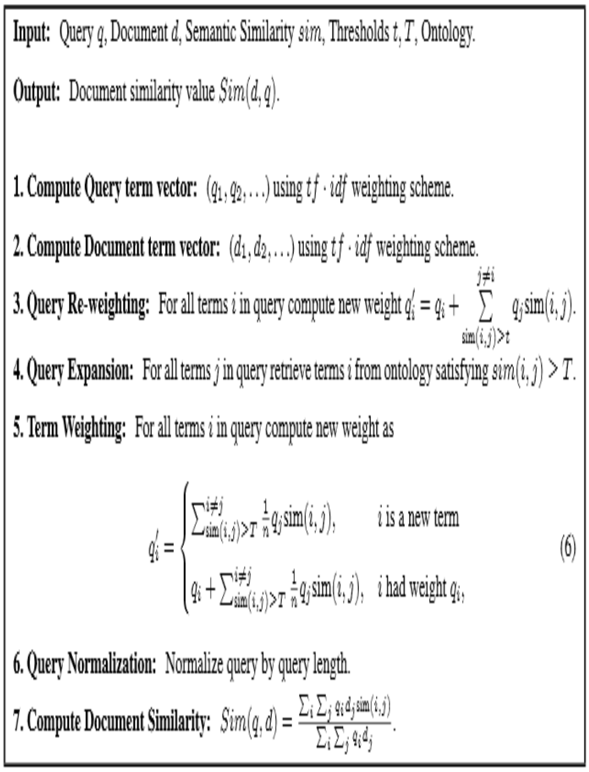
Synonym are alternate words that can replace a given word.[4]

Definition words = The words in the definition of a term may determine semantic similarity of the words for example in the words Induction and Deduction definition they are antonyms.

Co-occurring terms = Words that are joint e.g. Railway station Phrases=Incandescent light where synset of incandescent includes light.

Explicit Semantic Analysis:

Wikipedia is the largest online data library with different contexts related to the same term which can also retrieve appropriate results for non-dictionary terms or names that cannot have ontologies or word background to retrieve more about its related context. So Wikipedia along with naïve bayes can help us retrieve probabilities of terms in search query and thus semantic relatedness of search string to the Wikipedia pages so far retrieved.[5]



Input: Query q, Document d, Semantic Similarity sim, Thresholds t, T, Ontology.

Output: Document similarity value $Sim(d, q)$.

1. Compute Query term vector: $(q_1, q_2, \ldots)$ using $tf \cdot idf$ weighting scheme.

2. Compute Document term vector: $(d_1, d_2, \ldots)$ using $tf \cdot idf$ weighting scheme.

3. Query Re-weighting: For all terms i in query compute new weight $q_i' = q_i + \sum_{\mathrm{sim}(i,j)>t}^{j \neq i} q_j \mathrm{sim}(i, j)$.

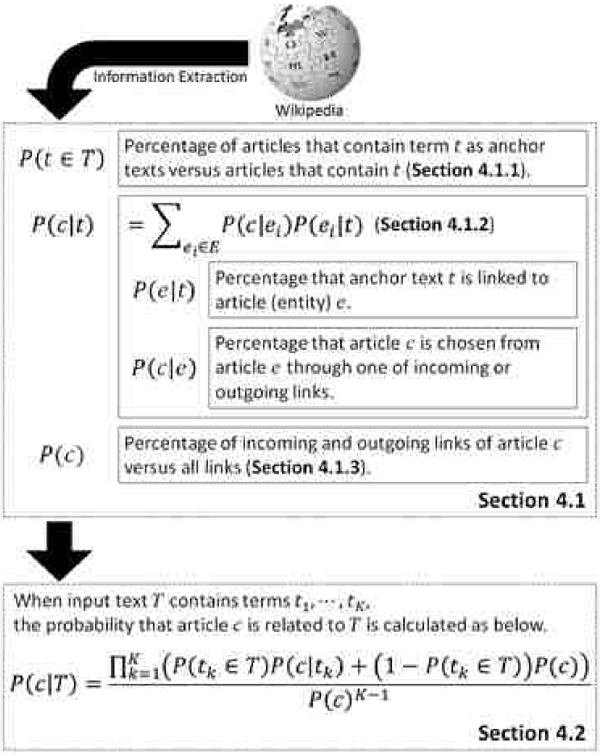
4. Query Expansion: For all terms j in query retrieve terms i from ontology satisfying $sim(i, j) > T$.

5. Term Weighting: For all terms i in query compute new weight as

$$q_i' = \begin{cases} \sum_{\mathrm{sim}(i,j)>T}^{i \neq j} \frac{1}{n} q_j \mathrm{sim}(i, j), & i \text{ is a new term} \\ q_i + \sum_{\mathrm{sim}(i,j)>T}^{i \neq j} \frac{1}{n} q_j \mathrm{sim}(i, j), & i \text{ had weight } q_i, \end{cases} \quad (6)$$

6. Query Normalization: Normalize query by query length.

7. Compute Document Similarity: $Sim(q, d) = \frac{\sum_i \sum_j q_i d_j \mathrm{sim}(i,j)}{\sum_i \sum_j q_i d_j}$.



$P(t \in T)$ Percentage of articles that contain term t as anchor texts versus articles that contain t (**Section 4.1.1**).

$P(c|t)$ $= \sum_{e_i \in E} P(c|e_i)P(e_i|t)$ (**Section 4.1.2**)

$P(e|t)$ Percentage that anchor text t is linked to article (entity) e.

$P(c|e)$ Percentage that article c is chosen from article e through one of incoming or outgoing links.

$P(c)$ Percentage of incoming and outgoing links of article c versus all links (**Section 4.1.3**).

Section 4.1

When input text T contains terms $t_1, \cdots, t_K$, the probability that article c is related to T is calculated as below.

$$P(c|T) = \frac{\prod_{k=1}^{K}\left(P(t_k \in T)P(c|t_k) + \left(1 - P(t_k \in T)\right)P(c)\right)}{P(c)^{K-1}}$$

Section 4.2

Fig.5: Summary of Existing SSRM Infrastructure

III. SYSTEM DEVELOPMENT

The search string is appended to the Google search api to retrieve the seed urls of entered query. The preference of links can be computed using the comparison of Anchor texts, title texts and body texts of the links in the seed url. Along with that the Ontology based on weighted relationships of synonyms, hyponyms and hypernyms etc. is calculated. Further the explicit semantic similarity between search string and Wikipedia pages related to different contexts of query terms are computed and the tf-idf approach is used to compute query and document vectors. The average of all the methods provides us with a score that decides the relevance of the entered query with the hyperlinks based on all calculations. The greater is the value of score for a link ,the more is the document relevance, semantically and contextually to the link.

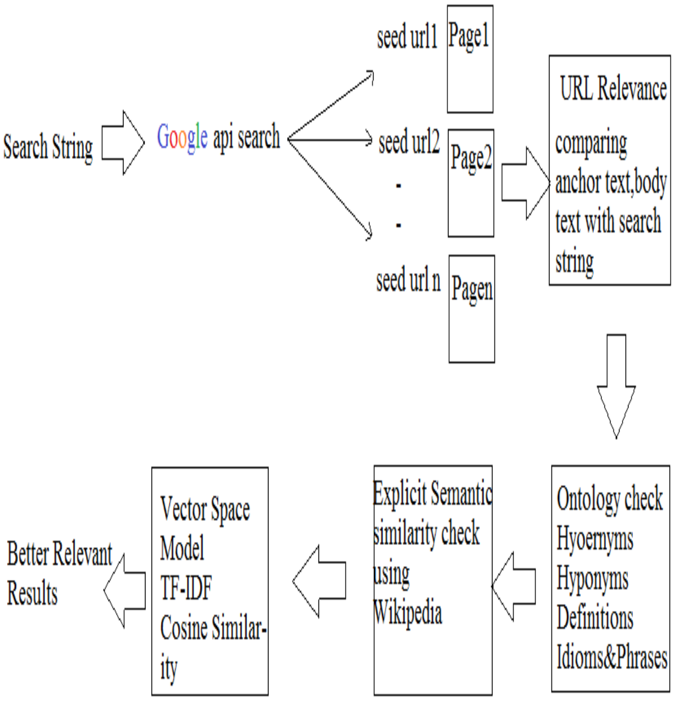


Fig.6: Proposed system for better result retrieval using combination of VSM ,Semantic Similarity Model using ontologies from wordnet ,Explicit Semantic Similarity using Wikipedia.

IV. CONCLUSION

This method is proposed to improve the efficiency of Web explorer by combining the Vector Space Model that computes similarity between two objects containing common terms, Semantic similarity method with ontologies that provides the related and alternate terms for queries there by increasing the scope of search, Explicit Semantic Similarity that helps retrieval of information about terms that are not possible with ontologies within different contexts. Thus it is now possible to create a web explorer that can remain focused and explores a greater scope to retrieve more contextually relevant links.

V. ACKNOWLEDGEMENT

We thank the anonymous reviewers for their constructive suggestions and comments, which helped us to improve the content, quality, and presentation of this article.

REFERENCES

[1] H.X. Zhang, J. Lu, “an online semi -supervised clustering approach to topical web crawlers”, Appl. Soft Computing, 490–495, 2010.

[2] Y.J. Du, Q.Q. Pen, Zhaoqiu Gao, “A topic -specific crawling strategy based on semantics similarity”, Data Knowl. Eng. 88,75–93,2013.

[3] A. Hliaoutakis, G. Varelas, et al.,” Information retrieval by semantic similarity”, Int. J. Semant. Web Inf. Syst. 3 (3),55–73,2006.

[4] Liu, S., Liu, F., Yu, C., Meng, W., “An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases.” In: ACM SIGIR’04, Sheffied, Yorkshire, UK,266–272,2004

[5] M. Shirakawa, K. Nakayama, T. Hara, S. Nishio., “Wikipedia-based Semantic Similarity Measurements for Noisy Short Texts Using Extended Naive Bayes”,IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING,2168-6750,2015