International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307

A Novel Sequence Graph Representation for Searching and Retrieving Sequences of Long Text in the Domain of Information Retrieval

Soumya George[1], M. Sudheep Elayidom[2], T. Santhanakrishnan[3]

[1]Research Scholar, Department of Computer Applications, Cochin University of Science and Technology, Kochi, Kerala, India
[2]Associate Professor, Division of Computer Engineering, Cochin University of Science and Technology, Kochi, Kerala, India
[3]Scientist, Govt. of India, Ministry of Defence, Naval Physical and Oceanographic Laboratory, Kochi, Kerala, India

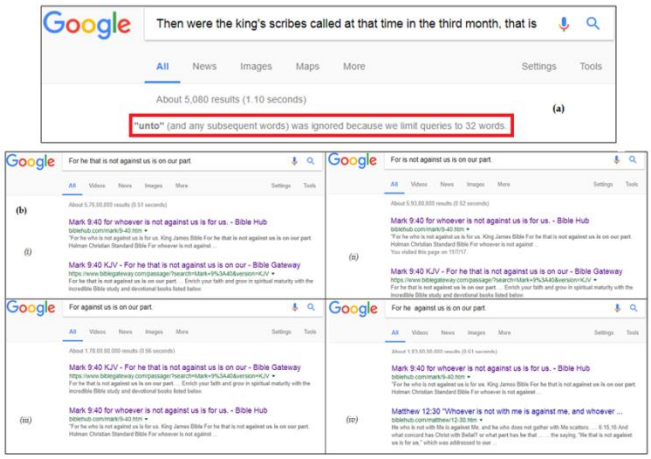
ABSTRACT

Long tail queries or keywords are becoming the norm for the user to search for what they intend and relying on keyword based SEO tactics never wins the game. A full text sequence based indexing approach for the document is needed to manage these lengthy search queries. This paper presents a highly efficient and novel graph based document representation, Word Sequence Graph model, to enhance text search and retrieval of any length including stop words by exploiting the unique features of a graph database. It is a one-for-all model where document and content information lies at the same place. This methodology is of high relevance in many real world applications that includes searching huge collection of documents. The examples are demonstrated with the help of bible texts.
Keywords : Search engine, Stop words, Graph database, Word Sequence Graph Model

I. INTRODUCTION

Big data drastically changed the search pattern of user queries and users are no longer depending on keyword based short phrases. User queries are getting longer day by day to find the exact result from the sheer amount of information available on the web. Searching and retrieving sequences of long text becomes a cumbersome task even for the most highly popular and efficient search engines. Figureure 1(a) shows that even Google limits queries to 32 words. Relying on keyword based SEO tactics to find exact matches for user queries never wins the game. The quality and reliability of a search engine depends upon the validity of search results based on user queries. A full text sequence based indexing approach for the document is needed to manage these lengthy search queries.

Normally search engines insist users to follow certain rules for phrase search, for e.g., phrases should be included in quotes or using Boolean operators like "and", "or" etc. to capture pages that contain the given phrase. Otherwise the search engine will find all pages that contain all of the keywords entered but the order of the words will not take into account which will leads to irrelevant results. Figureure 1(b) shows the problems with Google search engine by eliminating stop words in query search. Google displays the same set of results for four queries that differ even in their meaning by the presence or absence of stop words by modifying the verse, "For he that is not against us is on our part." (Mark 9:40). This totally affects the efficiency, reliability and validity of a search engine. Using a phrase list, which indexes commonly occurring phrases, is not a permanent solution because it restricts the phrase search only to list elements. Again, most of the search engines use a comprehensive list of common words called stop words to filter search in order to save disk space. This will often leads to irrelevant results which in turn affects the quality and reliability of a search engine.



Figureure 1: Google result with query limit of 32 word warning for Long-tail query search in (a) and Stop word elimination problems of Google search engine by displaying same set of results for queries that differ even in their meaning in (b)

Lots of works related to graph based index creation and representation of documents have been proposed where vertices represent unique terms and edge represents a semantic relationship between terms such as co-occurrence within a window size, synonymy etc. This will often lead to fast phrase search and retrieval [1-5].But there are certain limitations:-

- Only contents are stored, no details about the underlying document
- A restricted window size, N (typically between 2 and 10 words) is used, so that it can catch only phrases that co-occur within this sliding window size.
- Stop word removal is a default document pre-processing step in each that prevents the relevant long sequences text search and retrieval.

Long sequence text search and retrieval is an important and challenging application of search engine. In this paper, we present a novel graph based document representation method that can be used to convert unstructured documents into a structured format by creating a meaningful relationship between adjacent terms. The underlying technique used is Word Sequence Graph (WSG) model architecture which maintains the sentence structure by representing each sentence in each document as a graph of word model makes it an adequate model for phrase search or long text sequence search and retrieval. The proposed method can be used to index all types of documents thereby enhancing document search by document type too. The proposed model can be used for many applications including but not limited to:-

- Question Answering System
- Text summarization
- To identify discourse relationships
- Machine translation
- Cross language IR
- Document clustering & classification
- Plagiarism detection or Document similarity

II. WORD SEQUENCE GRAPH (WSG) MODEL

The Word Sequence Graph (WSG) model uses an incremental approach of processing document one at a time in a sequential order. Documents are represented in the form of a word graph model with the assumption that a document can be parsed into a set of sentences which can in turn be parsed into a set of words.

A document can be represented as a vector of sentences [1-5]:

$di = \{sij : j = 1,\ldots. pi\}$

$sij = \{tijk : k = 1,\ldots. lij\}$ Where,

di: is document i,

sij: is sentence j in document i,

pi: is the no: of sentences in document i,

tijk: is term k of sentence sij,

lij: is the length of sentence sij

A Word Sequence Graph (WSG) is directed graph, G = (V, E) where:

V: set of nodes $\{v_1,v_2,\ldots.v_n\}$ which can again be partitioned into two sets, V_D : $\{v_{D1},v_{D2},\ldots.v_{Dx}\}$ and V_T:{ $v_{T1},v_{T2},\ldots. v_{Ty}$} where V_D represents document node and V_T represent word nodes respectively with 'x' unique document nodes and 'y' unique non-stop words in the entire document pool such that x+y=n.

E: set of forward edges that connect each document node V_{Di} to the first non-stop word V_{Tj} of each of its sentence and between adjacent non-stop words of a sentence in word order.

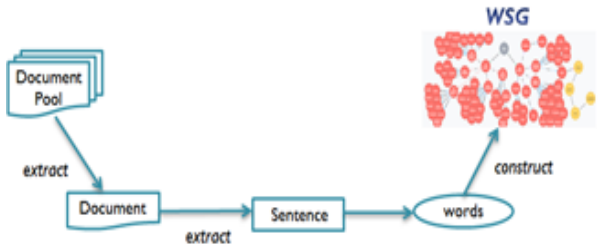


Figure 2 : Word Sequence Graph Model

The Word Sequence Graph (WSG) model consists of two types of nodes – document nodes and term nodes for the entire document pool. A document node will be created for each document when it is processed. Document node stores the properties of each unique document in the collection including name, URL, type, author name etc. Term nodes represent each unique term other than stop words appeared in any of the document in the entire document pool by parsing each document into sentences which in turn split into discrete words. An edge will be created between each document node and first non-stop word of each of its sentence with type "contents" and also between adjacent non-stop word terms with type "next_seq" in the direction of sentence order there by creating a chain of words for each sentence. All stop words, if any between two adjacent non-stop words will be identified with the help of stop word list including the punctuation marks and concatenated with a delimiter(space," ") into one string, "stp" and store in the edge in between as a relationship property. Again special care is taken in not to add space before punctuation marks for accurate text retrieval. In order to form unique term nodes for entire document collection, case of the each words will be determined either as upper case, sentence case and lower case or no case represented as 'U' , 'S' or 'N' respectively and all words in upper case or sentence case will be converted to lower case and others will be stored as such. For all words converted to lowercase, case will be stored in their respective incoming relationship edge to smoothen the sequence text retrieval. Other edge properties include sequence id, the unique document node id in which the succeeding word appears, sentence no etc. The various properties that can be stored depend on the application, for e.g.: for a book to be indexed, we need to store chapter no, name, sentence no or verse no etc. as edge properties and for a song data set, this stores the chord pattern or rhythmic information etc. A simple WSG model design is given in **Figure**. 2. The important challenge of this representation is:

Sentence terminating with stop words / Sentence containing only stop words: By including punctuation marks too into the stop word list leads to many sentences that terminates with a stop word string, "stp" resulting in a case with no end node or sentences containing only stop words. To tackle this situation, a unique node named "ter_node" will be considered as end node. Even for a sentence that terminates only with punctuation marks with no other stop word strings, "ter_node" will be considered as end node, except for punctuation mark "."(Period). This is needed for the accurate retrieval of text in order in correct format.

Figure.3 illustrates the incremental approach of constructing Word Sequence Graph model. **Figure**. 3(a) represents the graph representation of a Bible verse given in **Figure**. 3(b). The various nodes and relationship properties are given in **Figure**. 3(c). Document node includes properties like name of author, node type, book name and era and term node include name and node type. Edge properties include sequence id: seqid, chapter no, verse no as common properties and optional parameters include case and concatenated stop word string, stp. As the verse ends with only ".", no need to add "ter_node" as end node. **Figure**. 3(d) shows the incremental approach of indexing 3 verses from 3 books of Holy Bible, "Jude", "Job" and "Psalms" given in **Figure**. 3(e), by creating unique node for each non-stop word term. An end node "ter_node" is added for the first verse, Job 25:2, as it ends in a concatenated stop word string, "in his high places.". This way of representing documents in a graph database including stop words makes it easy for long sequence text search and retrieval including stop words thereby enhancing the reliability and validity of search results.

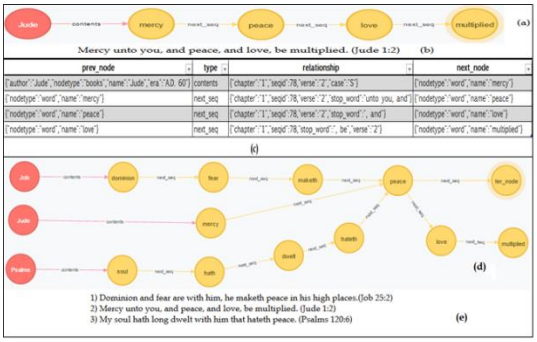


Figure 3: Word Sequence Graph Construction using incremental approach

III. LONG SEQUENCE TEXT SEARCH AND RETRIEVAL

The main aim of the proposed model is long sequence text search and retrieval without eliminating stop words. Graph search includes path traversal of visiting connected set of nodes and edges by following only out-edges. Long text search initially needs to be parsed into discrete words and searched in WSG by conducting a path traversal. The list of words will be partitioned into two lists, stop_list which is the list of concatenated stop word strings between two non-stop words in order and key_list, list of non-stop words in order and compare with list of stop_words and list of node names in the path respectively to find an appropriate match.

Long text retrieval is to retrieve a sequence of text by finding a matching path according to the search pattern and listing the connected set of node names and "stop_word" property of edges in order of path by checking conditions. As stop words are concatenated by the delimiter, space, no additional formatting is needed to recover stop words rather than just retrieve node names and stop word string, "stp" in order of matching path. Regular expressions can be used to match the citation pattern for a book.

IV. EXPERIMENTAL ANALYSIS

In order to evaluate the efficiency of proposed architecture, we constructed a graph based index of a set of documents using the html version of King James Version of Holy Bible as the data set. Bible is used as the corpus or data set in many related works [6-9]. Neo4j graph database is used as the storage back-end using java programming language [10]. The 'book of books' model of Holy Bible is a best example to demonstrate how different documents can be stored in a graph based inverted index by considering each chapter of each individual book as a document. One of the important features of holy books like Bible is that, the main content of text includes mostly stop words which make it an adequate one to show the efficiency of the model.For the current application, long text search is to find the details of a given verse text and long text retrieval will be searching for the text of a given verse. The common format to search biblical text retrieval is:
Book chapter for a chapter(John 3);
Book chapter1–chapter2 for a range of chapters (John 1–3);
Book chapter: verse1–verse2 for a range of verses (John 3:16–17);
Book chapter: verse for a single verse (John 3:16);

V. RESULTS

The important advantage of WSG Model is to search and retrieve text of any length inclusive of stop words. Figure 4(a) represents the results of searching the longest verse of bible, Esther 8:9. Figure. 4(b) represents an example of long text retrieval by retrieving a full book of bible, 2 John.

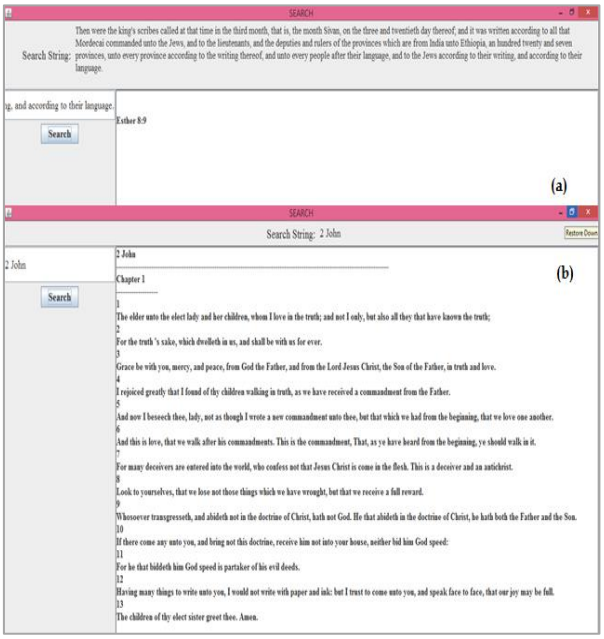
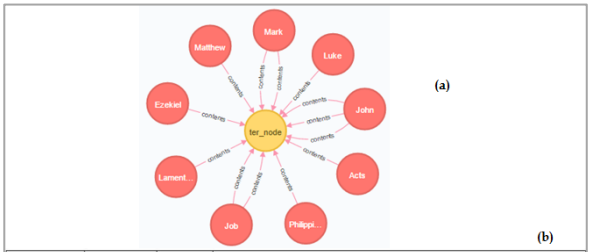


Figure 4: Long sequence text search in (a) & retrieval in (b)

KJV Holy Bible consists of 13 bible verses containing only stop words. These verses with their graph model and tabular representation are represented in Figure. 5(a) and 5(b) respectively.The model can be used to find or retrieve any of these verses. Figure.5(c) represents the screenshot of searching for the verse, Philippians 2:4 containing only stop words.



book	chapter	verse_no	verse
Philippians	2	4	Look not every man on his own things, but every man also on the things of others.
Acts	19	7	And all the men were about twelve.
John	9	12	Then said they unto him, Where is he? He said, I know not.
John	9	9	Some said, This is he: others said, He is like him: but he said, I am he.
John	1	3	All things were made by him; and without him was not any thing made that was made.
Luke	19	32	And they that were sent went their way, and found even as he had said unto them.
Mark	10	31	But many that are first shall be last; and the last first.
Mark	9	40	For he that is not against us is on our part.
Matthew	19	30	But many that are first shall be last; and the last shall be first.
Ezekiel	1	17	When they went, they went upon their four sides: and they turned not when they went.
Lamentations	3	35	To turn aside the right of a man before the face of the most High,
Job	34	27	Because they turned back from him, and would not consider any of his ways:
Job	16	22	When a few years are come, then I shall go the way whence I shall not return.

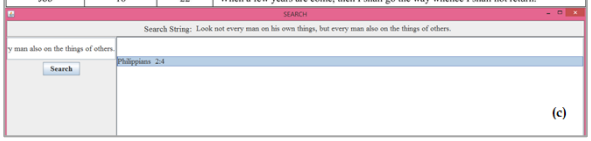


Figure 5: Bible verses containing only stop words with graph model in (a) and tabular representation in (b) and search result in (c)

VI. ADVANTAGES AND LIMITATIONS

From the results, it is shown that the Word Sequence Graph model, WSG is an efficient model to store, search and retrieve all types of documents. The main advantages and limitations of the model can be summarized as follows:-

Advantages

- One-for-all model where document details and data are stored at one place.
- Fast search and retrieval by exploiting the index free adjacency feature of a graph database by speedy path traversals.
- A distinguished efficient model that can be used to retrieve text of long sequences, even a full document, or a full book or chapters of book in the same format of source file with punctuation marks and stop words without additional burden of constructing a phrase list.
- Storing case of each word makes it possible for case sensitive search, displaying valid results for abbreviations or acronyms.
- WSG is a flexible model that can be converted easily to normal indexing approach by simply deleting concatenated stop word and symbols property from all the edges or it is also possible to search query by ignoring stop words, comparing only sequence of node names in a path.
- Efficiency, reliability and validity of search results of WSG Model always remains high without depending on the presence or absence of stop words.

Limitations

- Only text data can be searched and retrieved. Images or other forms of data are not incorporated into the model.
- Text font, color, size or other formatting options are not stored and hence cannot be retrieved.
- Word stemming or semantic integration is not applied due to the need of sequence text retrieval.

VII. CONCLUSIONS AND FUTURE SCOPE

In this paper, we introduced a novel graph based document representation model, Word Sequence Graph (WSG), to enable fast and efficient search and retrieval of lengthy queries or long-tail queries including stop words. The underlying method is to convert unstructured documents into a structured format by maintaining the sentence structure using term dependence relationship. For a search engine application alone where text search is only needed, word stemming and semantic integration of word using Wordnet can be applied. The model can be used for many applications including document clustering, classification, text summarization, question answering system, plagiarism detection etc

VIII. ACKNOWLEDGEMENTS

I sincerely thank UJRF Fellowship provided by Cochin University of Science & Technology, India for supporting my research in the form of fellowship.

IX. REFERENCES

[1]. Rao, B., & Mishra, S. N. (2017). An Approach to Text Documents Clustering with {n, n-1,….., 1}-Word (s) Appearance Using Graph Mining Techniques. IJSEAT, 4(12), 756-762.

[2]. Ravinuthala,M. K. V.& Ch, S. R (2016). Thematic Text Graph: A Text Representation Technique for Keyword Weighting in Extractive Summarization System. International Journal of Information Engineering and Electronic Business(IJIEEB), 8(4), 18.

[3]. Murtaza Munawar Fazal and Muhammad Rafi (2014). Clustering textual documents by extracting sequence from word-of-graph. Journal of Independent Studies and Research – Computing Volume 12 Issue 1

[4]. S. S. Sonawane, and Dr. P.A. Kulkarni (2014). Graph based Representation and Analysis of Text Document : A Survey of Techniques. vol. 96, no. 19, pp. 1–8.

[5]. Hammouda, K. M., & Kamel, M. S. (2004). Efficient phrase-based document indexing for web document clustering. IEEE Transactions on knowledge and data engineering, 16(10), 1279-1296.

[6]. Pfaffe, P., Tillmann, M., Lutteropp, S., Scheirle, B., & Zerr, K. (2016). Parallel String Matching.

[7]. Rolston, L., & Kirchhoff, K. (2016). Collection of Bilingual Data for Lexicon Transfer Learning.

[8]. Hewitt, J., Post, M., & Yarowsky, D. (2016). Automatic Construction of Morphologically Motivated Translation Models for Highly Inflected, Low-Resource Languages. AMTA 2016, Vol., 177.

[9]. Wolf, L., Hanani, Y., Bar, K., & Dershowitz, N. (2014). Joint word2vec networks for bilingual semantic representations. International Journal of Computational Linguistics and Applications, 5(1), 27-44.

[10]. Rani, A., Goyal, N., & Gadia, S. K. (2016, October). Efficient Multi-depth Querying on Provenance of Relational Queries Using Graph Database. In Proceedings of the 9th Annual ACM India Conference (pp. 11-20). ACM