Web Personalized Index based N-GRAM Extraction

Ahmed Mudassar Ali, M. Ramakrishnan

Abstract— Web mining is the analysis step of the "Knowledge Discovery in Web Databases" process which is an intersection of computer science and statistics. In this process results are produced from pattern matching and clustering which may not be relevant to the actual search. For example result for tree may be apple tree, mango tree whereas the user is searching for binary tree. N-grams are applied in applications like searching in text documents, where one must work with phrases. Eg: plagiarism detection. Thus relevancy becomes major part in searching. We can achieve relevancy using n-gram algorithm. We show an index based method to discover patterns in large data sets. It utilizes methods at the conjunction of AI, machine learning and statistics. We also induce a method of personalization where the search engine is used for indexing purposes in addition to the current n-gram techniques. A collaborative web search method is used for user's personalization in the web search engine to extract the required accurate data.

Index Terms— Web Mining, Knowledge Discovery, N-Gram, Stemming.

I. INTRODUCTION

Data mining is a technique which is used for high quality Information Retrieval (IR). N-gram methodology serves the purpose of data mining in many areas. An ***n*-gram** is a contiguous sequence of *n* items from a given sequence of text. An *n*-gram could be any combination of letters typically collected from a text. N-grams are applied in applications like searching in text documents. We get a set of n-grams by moving a floating window from the beginning to the end of the document..N-gram is a sequence of *n* terms (or generally tokens) from a document. N-grams are mainly used for the following purposes: speech recognition, OCR (optical character recognition), Intelligent Character Recognition (ICR), machine translation and similar applications, improve the retrieval performance in information retrieval systems when it is hoped to find similar "documents" given a single query document and a database of reference documents, identify the language of a text is in a document etc.

During the process of extraction we must eliminate duplicate n-grams and compute the frequency of the n-gram.

In Figure 1, the architecture of an n-gram extraction framework is described. This framework usually includes the following:

Manuscript Received on February 2015.

Ahmed Mudassar Ali, Research Scholar, Bharath University, Chennai, India,

M. Ramakrishnan Professor, School of Information Technology, Madurai Kamaraj University, Madurai, India

Document/HTML parsing: It parses terms from input document. The HTML parser reads the content of a web page into character sequences, and then marks the blocks of HTML tags and the blocks of text content. At this stage, the HTML parser uses a character encoding scheme to encode the text. Like this any given input text document can be read by the parser in a similar way.

Word net processing: Word Net is an electronic lexical database that uses word senses to determine underlying semantics. It differs from the traditional dictionary in that, it is organized by meaning. A stop words list is created and the parsed documents are looked in for the stop words and the process extracts only the lexical terms. Word Net organizes synsets of nouns and verbs as hypernyms and hyponyms. A few algorithms already exist for this purpose. At present, we use stop list algorithm which is described as follows.

Input: an arbitrary stop word dictionary T, interface 't'.

Output: resultant set of words T ′.

1. T ′ <— ϕ
2. W= set of all words in the search domain.
3. Pick a word ω from W.
4. Check the stop word constraints for ω with interface't'.
5. If no stop word is violated then T′ <— T′ U {w}.
6. Got 3 till nth word in W.
7. Return T′.

Stemming process: A stemming algorithm is a process of linguistic normalisation, in which variant forms of a word are reduced to a common form. There are several approaches to stemming. A method to perform stemming is to create a table containing index terms and their relevant stems. Consider the data given below:

Term	Stem
Engineering	engineer
engineered	engineer
engineer	engineer

Terms from queries and indexes could then be stemmed via table lookup. Using a B-tree or hash table, such lookups would be very fast. Porter Stemming Algorithm is one of the most popular stemming algorithms. It takes a list of suffixes and the criterion during which a suffix can be removed. It is simple, efficient and fast.

Word count processing: A probabilistic estimation for the frequency of words and word sequences. Often used to predict the next word when the preceding sequence is known. The stored n-grams which may be a unigram, bigram, trigram etc are looked in for the maximum occurrence in the query.

The highly overlapped terms are taken as the matched query. This follows a mathematical probability condition which says, For n-gram models,

E g for bigram model,

– $P(w_{n-1},w_n) = P(w_n \mid w_{n-1})\, P(w_{n-1})$

– $P(w_{8-1},w_8) = P(w_8 \mid w_7)\, P(w_7)$

– By the Chain Rule we can decompose a combined probability, e.g. $P(w_1,w_2,w_3)$ as given below:

– $P(w_1,w_2, \ldots,w_n) = P(w_1|w_2,w_3,\ldots,w_n)\, P(w_2|w_3, \ldots,w_n) \ldots P(w_{n-1}|w_n)\, P(w_n)$

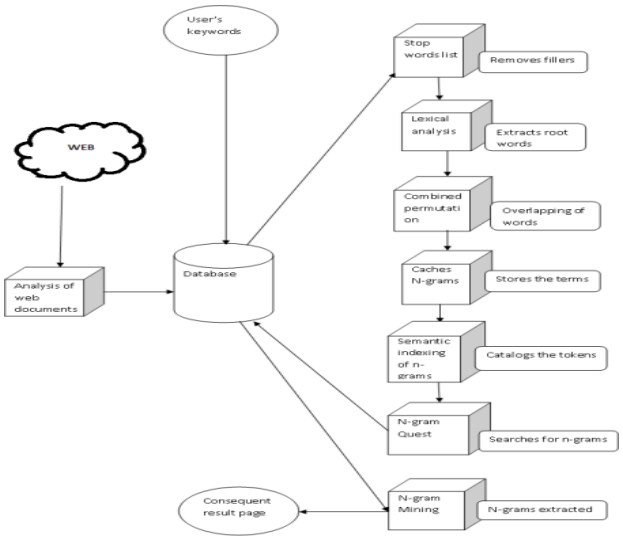
$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

Caches N-gram: As a result of the word count technique, the permutated combinations are thus stored in the database memory. Further it gets stored in all possible combination of the words which would be meaningful for user's personalized search.

Semantic analysis of n-grams: Approximate string matching is to find all the occurrences of a query string in a text database. The most important algorithm for string matching is indexing algorithms. They are classified depending on the types of the index structures they use. Indexing algorithms based on the n-gram index (simply, n gram Matching) has been widely used for approximate string matching due to the following advantage: the n-gram index is easily scalable for large databases because it is not a main memory algorithm, and thus, is not limited by the size of main memory. N-gram matching retrieves candidate results by finding documents that contain n-grams extracted from a query string, and then, performs refinement over the data. The n-grams are thus arranged, given higher priority for the overlapping terms, higher the priority in n-gram indexing. It serves the user to retrieve the exact data accurately and personally.

N-gram searching: Obviously, the next process is to locate the given query terms in the database. Already existing index based combinations of n-grams acts as the source. The purpose of our paper begins here which includes the way to use "personalized" search which deviates from our usual search, generally known as "public" search. When the user gives the keyword, the search engine searches the parsed database and retrieves the matched query data. In personalized search, this search is done relative to the field of the user.

Term Recognition: Thus n-gram extraction is relevant to the user's field. No irrelevant data is retrieved from the database. Eg. A user who belongs to medical field, registered as such here gets data relative only to that particular field. This increases the efficiency and accuracy of data retrieval. It also reduces the space and time overhead. Statistical Substring Reduction is a powerful procedure to remove redundant N-grams. And our paper induces the methodology of extracting the data in a personalized manner. The search is thus made personalized.



Consequent search page: The resultant page is thus displayed with accurate search results there by providing the most relevant data to the user. The user can also search in a different field which again provides personalization in that specific field.

Existing techniques of the n-gram extraction suppose only for public search using external sorting algorithms. These methods must handle a high number of duplicate n-grams which are removed after the frequency is computed. It results in high time and space overhead. In this paper, we show a time and space efficient method for the n-gram extraction that avoids duplication of n-grams and provides a method of personalized search. Additionally, we display a high scalability of our method; it is usable for large document collections including up-to 10^9 n-grams regardless the amount of the main memory.

In Section II, the current methods for the n-gram extraction are described. In Section III, we describe our index-based method for the n-gram extraction from large document collections. Section IV contains the result of our experiments. Finally, we outline our future work and conclude the paper.

II. RELATED WORK

The two basic approaches to repeated sequences extraction– a suffix tree based method (ST) and an inverted list based method (HT).The first algorithm (ST) makes use of a **tree data structure** known from suffix tree clustering (STC). The second method applies an inverted index, more specifically its **hash table** implementation (HT). **R.tesar, D.Fiala, F.Rousselot and K.Jezek** have inferred results yielded with some sample input data in the paper **"A comparison of two algorithms for discovering repeated word sequences"**. Their conclusion is that the Suffix T-algorithm is better with regards to time cost, whereas the HT-approach in respect of space.

Automatic extraction of domain-specific stop word list from a large labeled corpus is discussed by **Masoud Makrehchi Mohamed S. Kamel** in the paper "***Automatically building a stop word list for an information retrieval system***". In this, a new approach for **stop word extraction** based on the notion of backward filter level performance and sparsity measure of training data, is proposed.

According to the comparisons made, the newly proposed approach offers better results, which guarantee minimum information loss by filtering out most stop words.

Mariusz Paradowski, Halina Kwasnicka proposed in "Improved Resulted Word Counts Optimizer for Automatic Image Annotation Problem" that Automatic Image Annotation is an important research topic in pattern recognition area. The paper presents a generic approach to find correct word frequencies. Proposed method is an improved version of already presented method, called Greedy Resulted Word Counts Optimization. Optimal step selection allows to reduce the number of computations during the optimization procedure.

"Probabilistic Latent Semantic Indexing by Thomas Hofmann" depicts that the Probabilistic Latent Semantic Indexing is a novel approach to **automated document indexing** which is based on a statistical latent class model for factor analysis of count data. Text documents are generalized on the Expectation Maximization algorithm.

The HTML parser implemented in the paper "**Language Identification of Web Pages on Improved N-gram Algorithm**" proposed by **Yew Choong Chew, Yoshiki Mikami, Robin Lee Nagano** is unique in that it processes the content of the web page based on byte sequences. By using byte sequence, it eliminates the need to detect and apply character encoding scheme on the content extracted from the web page. The HTML parser parses the web page in a linear fashion. It searches the HTML tags from the beginning to the end of the page.

III. INDEX BASED N-GRAM EXTRACTION

A. The Indexing Data Structures

We model the n-gram type as a tuple (ngram, id, frequency), where ngram is the key. A common technique is that n-gram includes ids of terms instead of terms. For
the n-gram extraction we need a data structure providing the following methods:

• Insert(in ngram) – it inserts the n-gram if the n-gram type does not exist in the index, if this n-gram type exists in the index, the frequency is incremented.

• Find(in ngram, out frequency, out id) – it returns true together with frequency and id if the n-gram type is stored in the index, otherwise it returns false. (M is the number of items included in the index, in this case M = Tn):

• **B+-tree**: (h + 1) × log2 C = (logC M + 1) × log2 C = (logC Tn + 1) × log2 C, where h is the height of the tree and C is the capacity of tree's page (it means, the maximum number of items in a page). In this equation,
We do not consider the split operation of overflowed pages. Since the complexity of this is more, we can ignore it.

• **Hash table**: k is the average number of n-gram types related to one hash value. The common problem of the Hash table is that we have to precisely set the size of the Hash table and define a hash function for n-grams so we must create relatively small pages with low capacity. For example, if the number of n-gram types is 50×107, we set the Hash table size 10 × 107, it means k = 5. It is evident that, the Hash table provides the more efficient time complexity than B+-tree, however, in the case of the Hash table, we must know an approximate number of items inserted and the B+-tree provides other query operations.

B. Basic Algorithm

In List 1, we see the basic algorithm. In Lines 1 and 2, the index (B+-tree or Hash table) is created. In Line 3, we get the next n-gram from the collection. Evidently, this method includes parsing of terms and building of n-grams. In Line 4, the n-gram is inserted if it is not stored in the index; otherwise the frequency of the n-gram stored is increased. In Line 6, the index is closed. Before this step, we can, for example, save the extracted n-gram types to a file or we can save n-gram types with frequencies greater than a given value. This functionality is independent on the extraction method.

input : docs, a document collection
input : n, the length of the n-gram
output: index, an index including extracted n-gram types with frequencies
1) Index Type index;
2) index .Create();
3) while docs. Next n gram (n) do
4) index . Insert(Current N gram());
5) end
6) index. Close();

ALGORITHM: The basic algorithm of the n-gram extraction.

When a collection fits in the main memory, sorting-based algorithms provide the following time complexity O(Nn + Nn logn Nn), the index-based method provides O(Nn × (logC Tn +1) × log2 C) and O(Nn × k) for the B+-tree and Hash table, respectively, where Tn << Nn and n << C. Moreover, the index-based method provides more efficient space complexity, since it handles a lower count of n-grams
(Tn instead of Nn). The results of our experiments described in Section V confirm this theoretical model.

IV. EXPERIMENTAL RESULTS

Generally, a session for web search is a series of successive queries to satisfy a single information need and some clicked search results. In this paper, we focus on inferring user search goals for an exacting query. Therefore, the single session containing only one query is introduced, which distinguishes from the conservative session. Meanwhile, the feedback session in this paper is based on a single session, although it can be extended to the full session. The proposed n-gram algorithm consists of both clicked and unclicked URLs and ends with the last URL that was clicked in a particular session. It is motivated that before the last click, all the URLs have been evaluated and scanned by users. Therefore, besides the clicked URLs, the unclicked ones before the last click should be a part of the user feedbacks

Search results	Click sequence	Binary vector
www.thesun.co.uk/	0	0
www.nineplanets.org/sol.html	1	1
www.solarviews.com/eng/sun.htm	2	1
en.wikipedia.org/wiki/Sun	0	0
www.thesunmagazine.org/	0	0
www.space.com/sun/	0	0
en.wikipedia.org/wiki/The_Sun_(newspaper)	3	1

Fig 4 Example for Click Sequence Count

V. CONCLUSION

Our paper provides a new way to approach web search, known as "personalized search". To do this, we ensure a login page where the user registers his field and then surfs for information relatively. At this juncture we have worked upon three components of our proposed framework. Firstly, the web documents and text documents are parsed and stored in our database created. Thus the database contains the source for retrieval. Secondly, it undergoes a word net process where a stop word list is used to filter all the non-lexical words. Now the resultant document contains only meaningful words for the user's field. Followed by stemming those words, to extract only the root words that is more efficient for the user's personal search.

REFERENCES

1. "Language Identification of Web Pages on Improved N-gram Algorithm"- Yew Choong Chew, Yoshiki Mikami, Robin Lee Nagano , IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011, ISSN (Online): 1694-0814.
2. "*Automatically building a stop word list for an information retrieval system*"- Masoud Makrehchi **Mohamed S. Kamel, ECIR 2008, LNCS 4956, pp. 222-233, 2008.**
3. "Advanced database indexing"- Y.Manolopoulos , Y.Theodoeidis and V.J.Tsotras, Springer 2000.
4. "Probabilistic Latent Semantic Indexing"- Thomas Hofmann, SIGIR '99 Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Pages 50-57,1999.
5. SRCluster: Web Clustering Engine based on Wikipedia : Yuvarani Meiyappan, N. Ch. S. Narayana Iyengar and A. Kannan , International Journal of Advanced Science and Technology Vol. 39, February, 2012.
6. XML with Cluster Feature Extraction For Efficient Search : D. Divya, Dr. A. Muthukumaravel, Dr. P. Mayilvahanan , International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 8, August 2013).
7. A Model for XML Schema Integration : Kalpdrum Passi, Louise Lane, Sanjay Madria, Bipin C. Sakamuri, Mukesh Mohania, and Sourav Bhowmick , EC-Web 2002, LNCS 2455, pp. 193–202, 2002.
8. Learning to Cluster Web Search Results : Hua-Jun Zeng Qi-Cai He Zheng Chen Wei-Ying Ma Jinwen Ma, SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.
9. WEB SEARCH RESULT CLUSTERING- A REVIEW : Kishwar Sadaf1 and Mansaf Alam2, International Journal of Computer Science & Engineering Survey (IJCSES) Vol.3, No.4, August 2012.
10. Web Usage Mining Based on Complex Structure of XML for Web IDS : Marjan Eshaghi, S.Z. Gawali, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-5, April 2013.
11. Beyond Single-Page Web Search Results : Ramakrishna Varadarajan, Vagelis Hristidis, and Tao Li , IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 20, NO. 3, MARCH 2008
12. Extracting Knowledge from Web Search Engine Results : Andreas Kanavos, Evangelos Theodoridis, Athanasios Tsakalidis , IEEE 24th International Conference on Tools with Artificial Intelligence, 2012.
13. A New Search Results Clustering Algorithm based on Formal Concept Analysis : Yun Zhang, Boqin Feng, Yewei Xue, *School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China*, Fifth International Conference on Fuzzy Systems and Knowledge Discovery., FSKD'08, IEEE, pp. 356-360, 2008.

AUTHORS PROFILE

Ahmed Mudassar Ali received a Master Degree in Computer Science and Engineering from Rajalakshmi Engineering College, Anna University, Chennai. He is working as Associate Professor in Department of Information Technology, S.A. Engineering College, Chennai. He has started his research in Web Mining while pursuing his Ph.D and he presented more than 6 papers in various National and International Conferences. He is member of various societies like CSI, ISTE etc.