RESEARCH ARTICLE OPEN ACCESS

Web Redundancy Data Handling Using IR and Data Mining Algorithm

SekharBabu.Boddu [1], Prof.RajasekharaRao.Kurra [2]
Research Scholar [2]
Dept.CSE, SCSVMV University,kanchipuram,Chennai,India.
Asst.Professor [1]
Dept.CSE, KLUniversity
Professor [2], & Director
Dept.CSE, Usharama Engg College
AP – India

ABSTRACT

The aim of web mining is to discover and retrieve useful and interesting patterns from a large dataset. There has been huge interest towards web mining. Web data contains different kinds of information, including, web documents data, web structure data, web log data, and user profiles data.Information Retrieval (IR) is the area concerned with retrieving information about a subject from a collection of data objects. IR is different from Data Retrieval, which in the context of documents consists mainly in searching which documents of the collection contain keywords of a user query. IR deals with finding information needed by the user.The WWW have distinctive properties. For example, it is extremely complex, massive in size, and highly dynamic in nature. Owing to this unique nature, the ability to search and retrieve information from the Web efficiently and effectively is a challenging task especially when the goal is to realize its full potential. With powerful workstations and parallel processing technology, efficiency is not a bottleneck.

Keywords:- *IR,URL,HTTP,webcontent*

I. INTRODUCTION

Duplicate content is content that appears on the Internet in more than one place. That “one place” is defined as a location with a unique website address (URL) - so, if the same content appears at more than one web address, you’ve got duplicate content. While not technically a penalty, duplicate content can still sometimes impact search engine rankings. When there are multiple pieces of, as Google calls it, "appreciably similar" content in more than one location on the Internet, it can be difficult for search engines to decide which version is more relevant to a given search query. Duplicate content can present three main issues for search engines:1.They don't know which version(s) to include/exclude from their indices.2.They don't know whether to direct the link metrics (trust, authority, anchor text, link equity, etc.) to one page, or keep it separated between multiple versions.3.They don't know which version(s) to rank for query results. When duplicate content is present, site owners can suffer rankings and traffic losses. These losses often stem from two main problems:

To provide the best search experience, search engines will rarely show multiple versions of the same content, and thus are forced to choose which version is most likely to be the best result. This dilutes the visibility of *each* of the duplicates. Link equity can be further diluted because other sites have to choose between the duplicates as well. instead of all inbound links pointing to one piece of content, they link to multiple pieces, spreading the link equity among the duplicates. Because inbound links are a ranking factor, this can then impact the search visibility of a piece of content.In the vast majority of cases, website owners don't *intentionally* create duplicate content. But, that doesn't mean it's not out there. In fact by some estimates, up to 29% of the web is actually duplicate content! Let's take a look at some of the most common ways duplicate content is unintentionally created:

1. URL variations.2.URL parameters, such as click tracking and some analytics code, can cause duplicate content issues. This can be a problem caused not only by the parameters themselves, but also the order in which those parameters appear in the URL itself **For example:**

www.widgets.com/blue-widgets?color=blue is a duplicateof www.widgets.com/blue-widgetswww.widgets.com/bluewidgets?color=blue &cat=3 is a duplicate of www.widgets.com/blue-widgets?cat=3&color=blue.

Similarly, session IDs are a common duplicate content creator. This occurs when each user that visits a website is assigned a different session ID that is stored in the URL.Printer-friendly versions of content can also cause duplicate content issues when multiple versions of the pages get indexed.One lesson here is that when possible, it's often beneficial to avoid adding URL parameters or alternate versions of URLs (the information those contain can usually be passed through scripts).

2. HTTP vs. HTTPS or WWW vs. non-WWW pagesIf your site has separate versions at "www.kluniversity.in" and "kluniversity.in" (with and without the "www" prefix), and the same content lives at both versions, you've effectively created duplicates of each of those pages. The same applies to sites that maintain versions at both http:// and https://. If both versions of a page are live and visible to search engines, you may run into a duplicate content issue.

3. Scraped or copied content
Content includes not only blog posts or editorial content, but also product information pages. Scrapers republishing your blog content on their own sites may be a more familiar source of duplicate content, but there's a common problem for e-commerce sites, as well: product information. If many different websites sell the same items, and they all use the manufacturer's descriptions of those items, identical content winds up in multiple locations across the web.There are some steps you can take to proactively address duplicate content issues, and ensure that visitors see the content you want them to.

Use 301s: If you've restructured your site, use 301 redirects ("RedirectPermanent") in your .htaccess file to smartly redirect users, Googlebot, and other spiders. (In Apache, you can do this with an .htaccess file; in IIS, you can do this through the administrative console.)

Be consistent: Try to keep your internal linking consistent. For example, don't link to http://www.example.com/page/ and http://www.example.com/page and http://www.example.com/page/index.htm.

Use top-level domains: To help us serve the most appropriate version of a document, use top-level domains whenever possible to handle country-specific content. We're more likely to know that http://www.example.de contains Germany-focused content, for instance, than http://www.example.com/de or http://de.example.com.

Syndicate carefully: If you syndicate your content on other sites, Google will always show the version we think is most appropriate for users in each given search, which may or may not be the version you'd prefer. However, it is helpful to ensure that each site on which your content is syndicated includes a link back to your original article. You can also ask those who use your syndicated material to use the noindex meta tag to prevent search engines from indexing their version of the content.

Use Search Console to tell us how you prefer your site to be indexed: You can tell Googleyour preferreddomain (forexample, http://www.example.com or http://example.com).

Minimize boilerplate repetition: For instance, instead of including lengthy copyright text on the bottom of every page, include a very brief summary and then link to a page with more details. In addition, you can use the Parameter Handling tool to specify how you would like Google to treat URL parameters.

Avoid publishing stubs: Users don't like seeing "empty" pages, so avoid placeholders where possible. For example, don't publish pages for which you don't yet have real content. If you do create placeholder pages, use the noindex meta tag to block these pages from being indexed.

Undrstand your content management system: Make sure you're familiar with how content is displayed on your web site. Blogs, forums, and related systems often show the same content in multiple formats. For example, a blog entry may appear on the home page of a blog, in an archive page, and in a page of other entries with the same label.

Minimize similar content: If you have many pages that are similar, consider expanding each page or consolidating the pages into one. For instance, if you have a travel site with separate pages for two cities, but the same information on both pages, you could either merge the pages into one page about both cities or you could expand each page to contain unique content about each city.

Google does not recommend blocking crawler access to duplicate content on your website, whether with a robots.txt file or other methods. If search engines can't crawl pages with duplicate content, they can't automatically detect that these URLs point to the same content and will therefore effectively have to treat them as separate, unique pages. A better solution is to allow search engines to crawl these URLs, but mark them as duplicates by using the rel="canonical" link element, the URL parameter handling tool, or 301 redirects. In cases where duplicate content leads to us crawling too much of your website, you can also adjust the crawl rate setting in Search Console.Duplicate content on a site is not grounds for action on that site unless it appears that the intent of the duplicate content is to be deceptive and manipulate search engine results. If your site suffers from duplicate content issues, and you don't follow the advice listed above, we do a good job of choosing a version of the content to show in our search results.However, if our review indicated that you engaged in deceptive practices and your site has been removed from our search results, review your site carefully. If your site has been removed from our search results, review our Webmaster Guidelines for more information. Once you've made your changes and are confident that your site no longer violates our guidelines, submit your site for reconsideration.

The purpose of a web crawler used by a search engine is to provide local access to the most recent versions of possibly all web pages. This means that the Web should be crawled regularly and the collection of pages updated accordingly. Having in mind the huge capacity of the text repository, the need for regular updates poses another challenge for the web crawler designers. The problem is the high cost of updating indices. A common solution is to append the new versions of web pages without deleting the old ones. This increases the storage requirements but also allows the crawler repository to be used for archival purposes. In fact, there are crawlers that are used just for the purposes of archiving the web. Crawling of the Web also involves interaction of web page developers. As Brin and Page mention in a paper about their search engine Google, they were getting e-mail from people who noticed that somebody (or something) visited their pages. To facilitate this interaction there are standards that allow web servers and crawlers to exchange information. One of them is the robot exclusion protocol. A file named robots.txt that lists all path prefixes of pages that crawlers should not fetch is placed in the http root directory of the server and read by the crawlers before crawling of the server tree. Usually, crawling precedes the phase of web page evaluation and ranking, as the latter comes after indexing and retrieval of web documents. However, web pages can be evaluated while being crawled. Thus, we get some type of enhanced crawling that uses page ranking methods to achieve focusing on interesting parts of the Web and avoiding fetching irrelevant or uninteresting pages.
The web user information needs are represented by keyword queries, and thus document relevance is defined in terms of how close a query is to documents found by the search engine. Because web search queries are usually incomplete and ambiguous, many of the documents returned may not be relevant to the query. However, once a relevant document is found, a larger collection of possibly relevant documents may be found by retrieving documents similar to the relevant document. This process, called *similarity search*, is implemented in some search engines (e.g.,Google) as an option to find pages similar or related to a given page. The intuition behind similarity search is the *cluster hypothesis* in IR, stating that documents similar to relevant documents are also likely to be relevant. In this section we discuss mostly approaches to similarity search based on the content of the web documents. Document similarity can also be considered in the context of the web link structure.

The *vector space model* defines documents as vectors (or points) in a multidimensional Euclidean space where the axes (dimensions) are represented by terms. Depending on the type of vector components (coordinates), there are three basic versions of this representation. Boolean, term frequency (TF), and term frequency–inverse document frequency (TFIDF).The Boolean representation is simple, easy to compute and works well for document classification and clustering. However, it is not suitable for keyword search because it does not allow document ranking. Therefore, we focus here on the TFIDF representation.
In the *term frequency* (TF) approach, the coordinates of the document vector _ *dj* are represented as a function of the term counts, usually normalized with the document length. For each term t_i and each document d_j , the TF (t_i , d_j) measure is computed.
This can be done in different ways; for example:
Using the sum of term counts over all terms (the total number of terms in the document):
$TF(t_i,d_{j)}=\int^0 n_{ij}/\Sigma^m{}_{k=1}\, n_{kj}$ if n=0 or n>0
Using the maximum of the term count over all terms in the document:
$TF(t_i,d_{j)}=\int^0 n_{ij}/Max_k\, n_{kj}$ if n=0 or n>0

In the Boolean and TF representations, each coordinate of a document vector is computed locally, taking into account only the particular term and document. This means that all axes are considered to be equally important. However, terms that occur frequently in documents may not be related to the content of the document. This is the case with the term *program* in our department example. Too many vectors have 1's (in the Boolean case) or large values (in TF) along this axis. This in turn increases the size of the resulting set and makes document ranking difficult if this term is used in the query. The same effect is caused by stop words such as *a*, *an*, *the*, *on*, *in*, and *at* and is one reason to eliminate them from the corpus. The basic idea of the *inverse document frequency* (IDF) approach is to scale down the coordinates for some axes, corresponding to terms that occur in many documents. For each term *ti* the IDF measure is computed as a proportion of documents where *ti* occurs with respect to the total number of documents in the collection. Let D $=U^n{}_1$ d_j be the document collection and D_{ti} the set of documents where term t_i occurs. That is, $D_{ij} == \{dj \mid n_{ij} > 0\}$. As with TF, there are a variety ofways to compute IDF; some take a simple fraction $|D|/|D_{ij}|$

IDF(*ti*) = log(|*D*||*Dti* |)

Cosine Similarity

The query-to-document similarity which we have explored so far is based on the vector space model. Both queries and documents are represented by TFIDF vectors, and similarity is computed using the metric properties of vector space. It is also straightforward to compute similarity between documents. Moreover, we may expect that document-to-document similarity would be more accurate than query-to-document similarity because queries are usually too short, so their vectors are extremely sparse in the highly dimensional vector space. In similarity search we are not concerned with a small query vector, so we are free to use more (or all) dimensions of the vector space to represent our documents. In this respect it will be interesting to investigate how the dimensionality of vector space affects the similarity search results.
Generally, several options can be explored:
Using all terms from the corpus. This is the easiest option but may cause problems if the corpus is too large (such as the document repository of a web search engine).
Selecting terms with high TF scores (usually based on the entire corpus). This approach prefers terms that occur frequently in many documents and thus makes documents look more similar. However, this similarity is not indicative of document content.
Selecting terms with high IDF scores, this approach prefers more document specific terms and thus better differentiates documents in vector space. However, it results in extremely sparse document vectors, so that similarity search is too restricted to closely related documents. Combining the TF and IDF criteria, For example, a preference may be given to terms that maximize the product of their TF (on the entire corpus) and TDF scores. As this is the same type of measure used in the vector coordinates (the difference is that the TF score in the vector is taken on the particular document), the vectors will be better populated with nonzero coordinates.

II. JACCARD SIMILARITY

There is an alternative to cosine similarity, which appears to be more popular in the context of similarity search (we discuss the reason for this later). It takes all terms that occur in the documents but uses the simpler Boolean document representation. The idea is to consider only the nonzero coordinates (i.e., those that are 1) of the Boolean vectors. The approach uses the *Jaccard coefficient*, which is generally defined (not only for Boolean vectors) as the percentage of nonzero coordinates that are different in the two vectors.

sim (*d*1, *d*2) = |*T* (*d*1) ∩ *T* (*d*2)| |*T* (*d*1) ∪ *T* (*d*2)|

Two simple heuristics may drastically reduce the number of candidate pairs:

1. Frequent terms that occur in many documents (say, more than 50% of the
Collection) are eliminated because they cause even loosely related documents
to look similar.

2. Only documents that share at least one term are used to form a pair

The Jaccard coefficient on two sets by representing them as smaller sets called *sketches*, which are then used instead of the original documents to compute the Jaccard coefficient. Sketches are created by choosing a random permutation, which is used to generate a sample for each document. Most important, sketches have a fixed size for all documents. In a large document collection each document can be represented by its sketch, thus substantially reducing the storage requirements as well as the running time for precomputing similarity between document pairs. The method was evaluated by large-scale experiments with clustering of all documents on the Web [8]. Used originally in a clustering framework, the method also suits very well the similarity search setting.So far we have discussed two approaches to document modelling: the TFIDF vector and set representations. Both approaches try to capture document semantics using the terms that documents contain as descriptive features and ignoring any information related to term positions, ordering, or structure. The only relevant information used for this purpose is whether or not a particular term occurs in the documents (the *set-of-words approach*) and the frequency of its occurrence (the *bag-of-words approach*). For example, the documents “Mahesh loves Jhanaki” and “Jhanaki loves Mahesh” are identical, because they include the same words with the same counts, although they have different meanings. The idea behind this representation is that *content* is identified with *topic* or *area* but not with *meaning* (that is why these approaches are also called *syntactic*). Thus, we can say that the topic of both documents is people and love, which is the meaning of the terms that occur in the documents.There is a technique that extends the set-of-words approach to sequences of words. The idea is to consider the document as a sequence of words (terms) and extract from this sequence short sub sequences of fixed length called ***n-grams* or *shingles***. The document is then represented as a set of such *n*-grams. For example, the document “Mahesh loves Jhanaki” can be represented by the set of 2-grams {[Mahesh, loves], [loves, Jhanaki]} and "Jhanaki loves Mahesh" by {[Jhanaki, loves], [loves, Mahesh]}. Now these four 2-grams are the features that represent our documents. In this representation the documents do not have any overlap. We have already mentioned *n*-grams as a technique for approximate string matching but they are also popular in many other areas where the task is detecting sub sequences such as spelling correction, speech recognition, and character recognition. Shingled document representation can be used for estimating document ***resemblance***. Let us denote the set of shingles of size *w* contained in document *d* as *S*(*d*,*w*). That is, the set *S*(*d*,*w*) contains all *w*-grams obtained from document *d*. Note that *T* (*d*) = *S*(*d*,1), because terms are in fact 1-grams. Also, *S*(*d*,|*d*|) = *d* (i.e., the document itself is a*w*-gram, wherev*w* is equal to the size of the document). The *resemblance* between documents *d*1 and *d*2 is defined by the Jaccard coefficient computed with shingled documents: Although the number of shingles needed to represent each document is roughly the same as the number of terms needed for this purpose, the storage requirements for shingled document representation increase substantially. A straightforward representation of w-word shingles as integers with a fixed number of bits results in a w-fold increase in storage. we can also eliminate those that are too similar in terms of resemblance [with large values of $r_w(d1,d2)$]. In this way, duplicates or near duplicates can be eliminated from the similarity search results.

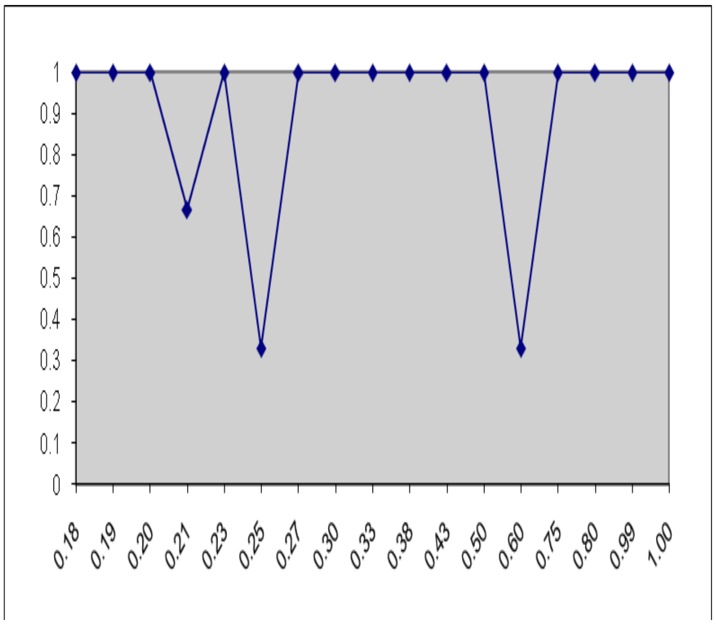
$r_w(d1,d2) = |S(d1,w) \cap S(d2,w)|\ |S(d1,w) \cup S(d2,w)|$

$r_e(d1,d2) = |L(d) \cap L(d2)|\ |L(d1) \cup L(d2)|$

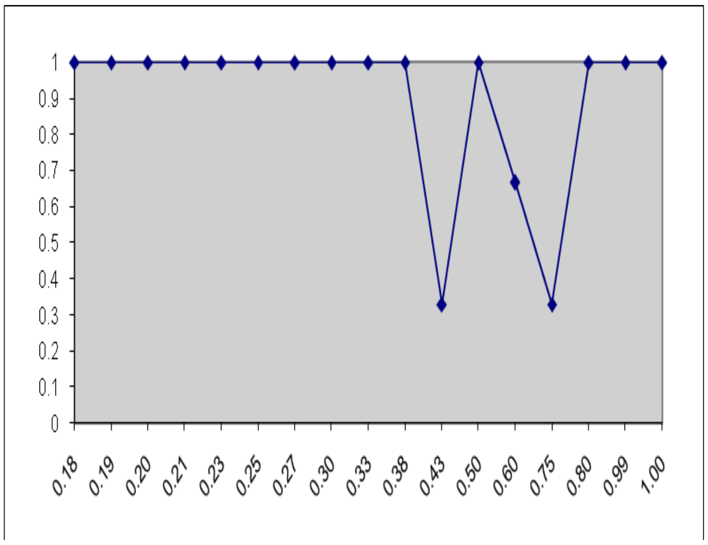
L(d) is a smaller set of shingles called a sketch of document d.

Result:

Existing system precision average is=0.35



Proposed IR System precsisoin average is=0.59



III. CONCLUSION

The web data conatins redundacny data , by using IR and RDF Techniques, we can handle the web redundant data with indexed keywords and query optimization concepts through search engine interface. Finally , after done TF& IDF ,cosine and jacarrd similarities and Document resemblance of their precision value is improved than the previsous one .it means similarity data handle handled postitively.

REFERENCES

[1]. Hassan F. Eldirdiery, A. H. Ahmed," Detecting and Removing Noisy Data on Web document using Text Density Approach ",International Journal of Computer Applications (0975 – 8887) Volume 112 – No. 5, February 2015.

[2]. Ms. Shalaka B. Patil, Prof. Rushali A. Deshmukh," , Enhancing Content Extraction from Multiple Web Pages by Noise Reduction", International Journal of Scientific & Engineering Research, Volume 6, Issue 7, July-2015 ISSN 2229-5518

[3]. Rajni Sharma, Max Bhatia," Eliminating the Noise from Web Pages using Page Replacement Algorithm, International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3066-3068.

[4]. Hui Xiong, Member, IEEE, Gaurav Pandey, Michael Steinbach, Member, IEEE,and Vipin Kumar, Fellow, IEEE," Enhancing Data Analysis with Noise Removal", IEEE Transactions On Knowledge And Data Engineering.

[5]. Rekha Garhwal," Improving Privacy in Web Mining by eliminating Noisy data & Sessionization", International Journal of Latest Trends in Engineering and Technology (IJLTET).

[6].Erdinc,uzun,et.al," A hybrid approach for extracting informative content from web pages", Information Processing and Management: an International Journal archive Volume 49 Issue 4, July, 2013,Pages 928-944.

[7].Shine N. Das,et.al." Eliminating Noisy Information in Web Pages using featured DOM tree, International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 2– No.2, May 2012.
[8]. Xin Qi and JianPeng Sun, Eliminating Noisy Information in Webpage through Heuristic Rules, 2011 International Conference on Computer Science and Information Technology (ICCSIT 2011*)*

[9]. Thanda Htwe," Cleaning Various Noise Patterns in Web Pages for Web Data Extraction", Iinternational Journal of Network and Mobile Technologies Issn 1832-6758 Electronic Version Vol 1 / Issue 2 / November 2010.

[10]. Byeong Ho Kang and Yang Sok Kim," Noise Elimination from the Web Documents by Using URL paths and Information Redundancy**".**

[11]. Thanda Htwe, Nan Saing Moon Kham," Extracting Data Region in Web Page by Removing Noise using DOM and Neural Network", 2011 3rd International Conference on Information and Financial Engineering.

[12] .Tieli Sun, Zhiying Li, Yanji Liu, Zhenghong Liu, "Algorithm Research for the Noise of Information Extraction Based Vision and DOM Tree", International Symposium on Intelligent Ubiquitous Computing and Education, pp 81-84, May 2009.

[13].Jinbeom Kang, Joongmin Choi, "Block classification of a web page by using a combination of multiple classifiers", Fourth International Conference on Networked Computing and Advanced Information Management, pp 290 -295, September 2008.

[14]. Lan Yi," Eliminating Noisy Information in Web Pages for Data Mining".2003.

[15]. Bassma, S., Alsulami, Maysoon, F., Abulkhair and Fathy E. Eassa, "Near Duplicate Document Detection Survey", International Journal of Computer Science & Communication Networks, Vol. 2, No. 2, pp. 147-151, 2011.

[16].Huda Yasin and Mohsin Mohammad Yasin, "Automated Multiple Related Documents Summarization via Jaccard's Coefficient", International Journal of Computer Applications, Vol. 13, No. 3, pp. 12-15, 2011.

[17]. Syed Mudhasir, Y., Deepika, J., Sendhilkumar, S., Mahalakshmi, G. S, "Near-Duplicates Detection and Elimination Based on Web Provenance for Effective Web Search", (IJIDCS) International Journal on Internet and Distributed Computing Systems, Vol. 1, No. 1-5, 2011.

[18]. Kanhaiya Lal & N.C.Mahanti ,"A Novel Data Mining Algorithm for Semantic Web Based Data Cloud" International Journal of Computer Science and Security (IJCSS), Volume (4): Issue (2),Pg.160-175,2010.

[19].Ranjna Gupta, Neelam Duhan, Sharma, A. K. and Neha Aggarwal, "Query Based Duplicate Data Detection on WWW", International Journal on Computer Science and Engineering Vol. 02, No. 04, pp. 1395-1400, 2010.

[20] .Alpuente, M. and Romero, D. "A Tool for Computing the Visual imilarity of Web pages", Applications and the Internet (SAINT), pp. 45-51, 2010.

[21].PrasannaKumar, J. and Govindarajulu, P. "Duplicate and Near Duplicate Documents Detection: A Review", European Journal of Scientific Research, Vol. 32 No. 4, pp. 514-527, 2009.

[22]. Poonkuzhali, G., Thiagarajan, G. and Sarukesi, K. "Elimination of Redundant Links in Web Pages - Mathematical Approach", World Academy of Science, Engineering and Technology, No. 52, p. 562,2009.