Textual Information Retrieval using Inverted Index

Khadija Zavery
Information Science and Engineering Dept.
R.V College of Engineering
Bangalore, Karnataka
khadijiatzavery@gmail.com

Haripriya Ramesh
Information Science and Engineering Dept.
R.V College of Engineering
Bangalore, Karnataka
haripriya.ramesh1996@gmail.com

Vaishnavi Singh
Information Science and Engineering Dept.
R.V College of Engineering
Bangalore, Karnataka
vaishnavi.singh2703@gmail.com

Dr. B.M Sagar
Information Science and Engineering Dept.
R.V College of Engineering
Bangalore, Karnataka

***Abstract*—Information Retrieval (IR) is the field that deals with retrieval of unorganised data, such as textual documents, responses to a query entered by the user in the unstructured or structured form. With technological advancement, there is an increasing need for effective techniques of IR. It has become possible to store large amounts of information and therefore, finding relevant information from such collections has become a necessity.**

This paper talks about the different processing techniques for implementing information retrieval and why some techniques are more efficient than others. This paper also includes a brief description of how natural language processing has made the process of information retrieval very convenient.

Keywords—Information retrieval; Information Retrieval System(IRS); query; Natural Language Processing(NLP); term document matrix; indexing inverted index; posting list; 0/1 vectors; stemming; stopwords; chunking; compound words

1. INTRODUCTION

Information retrieval is generally considered as a subfield of computer science. Representation, storage, and access of information are all part of an information retrieval system. To achieve this, IR systems use large database collections. The main goal of information retrieval system (IRS) is to find relevant information or a document that satisfies users information needs. The 3 main processes an information retrieval system has to support are: representing content of all the documents, representing the user's query, comparing the two representations. Figure 1 gives an overview of the main tasks of an IR system.

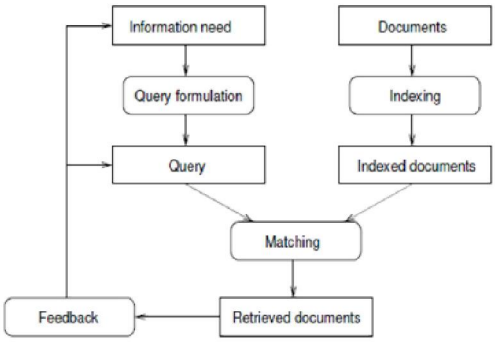

Fig. 1. Common Architecture of an Information Retrieval System

In the figure, the rounded boxes show processes and squared boxes show data. A user's information needs are usually given in natural language and therefore it is only natural to perform textual information retrieval using natural language processing.

In response to a user's query an IR System performs the following processes:

· Creates an inverted index for the documents : In this phase, a Python dictionary is used to store each word from the document as "key" and the document numbers that it appears in as its "postings values"

· Transform the query: When the query is received from a client, the query is analyzed by the system and it is transformed using NLP techniques such as stemming, stopwords removal and a few other techniques, which are discussed below hoping that this representation matches the way in which the document is structured[9].

· Comparison: A comparison is made between the document and the query. Based on the processed query words the system returns the document which contains the description closest to the query.

· Documents are then listed according to relevance

There are two basic methods for assessing the quality of the retrieved documents: precision and recall. Precision is the fraction of retrieved documents that are relevant to the user's query. Recall is the fraction of relevant documents in the collection that were retrieved[1].

2. Implementation of Information Retrieval

2.1 Common Data structures used

Data structures help us to store and organize information in a computer so that it can be retrieved and used most productively. When dealing with the task of information retrieval, the presence of data structures is absolutely necessary. For this implementation, the two data structures considered are- Term Document Index and Inverted Index, which seemed very apt for the task of information retrieval. Comparison is then done to determine the more efficient one between the two.

2.1.1 Term-Document Index matrix

In this two-dimensional matrix the rows of the matrix are words or "terms". The columns of the matrix are the documents. Each entry in the matrix is either zero or one and these are called 0/1 vectors[10]. The queries are answered using vectors which are made for the terms in the query and then combined using boolean functions like AND,OR NOT.

This model uses the Boolean retrieval system and is known as the boolean retrieval model. In this model, a query is taken in the form of a Boolean expression of terms. Terms are combined using operators and,not and or. This model sees each document as a collection of words or as in this case, terms.

2.1.2 Inverted Index

The main data structure used in IR systems today is the inverted index. This data structure is seen in all modern systems comprising laptops and also search engines[4]. This data structure overcomes the sparsity problem that was present in term document matrix. In this method, each document is identified by a 'docID' which is the serial number of the document. As we have seen, the document is split into terms and each term is associated with the list of documents that it occurs in. It is tempting to implement the inverted index using fixed size arrays, however, arrays prove to be very inefficient. This is because, some words could appear in multiple documents whereas others might not. Moreover, appending document numbers to a fixed size array is tedious. The next best choice that comes to mind is the variable size list which is what we use in our implementation. These lists are commonly called the 'postings list' and they are generally stored on disks in a contiguous manner. We do not use this storage technique as postings lists for search engines are very large, instead they are represented as linked lists in memory. In the implementation the Inverted Index is represented as a dictionary in Python. The terms are used as the keys of the dictionary and the postings list is analogous to the values of the dictionary. Each value corresponding to the key is called a posting. For the inverted index construction we first go through some pre-processing steps shown in Figure 2:

· Tokenizer - converts the documents into a set of word tokens which is the elementary unit of indexing[3].

· Linguistic Modules - We often don't index the exact words that are present in the documents,instead the tokens are passed to linguistic modules that modify the tokens based on various rules.

· Indexer - the modified tokens are fed to the indexer to make the inverted index.

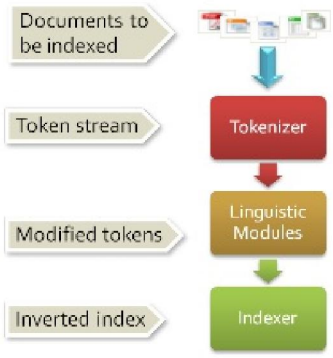

Fig. 2. The preprocessing steps required for the construction of an Inverted Index

2.1.3 Comparison between Term-Document Index and Inverted Index

Term-Document matrix is not the most efficient data structure to be used. This is because the matrix is extremely sparse, that is ,it has very few non-zero entries. A much better way of implementing is to record only the things that do occur, that is, the 1 positions only. Inverted Index on the other hand is a data structure that exploits the 'sparsity' of the Term Document matrix and allows for very efficient retrieval and this is why this data structure is what we have chosen for our implementation.

2.2 Statistical processing of natural language

This is the primary model of information retrieval systems. In this model, a set of keywords called terms are extracted from each documnet.

In this processing method, the words contained in the document are taken as indexed terms. The method of statistical natural-language processing uses probabilistic, stochastic, and statistical methods to resolve a lot of difficulties, especially those which are caused by the presence of longer sentences and ambiguity, which results in multiple answers [10].

These statistical models are then used to match the document's words with that of the words present query. Its accuracy and simplicity has made it the most popular model in textual IR systems. This method of processing uses common pre-processing techniques like elimination of stopwords, stemming, tagging, etc, as discussed below.

2.2.1 Stopwords

Removing stopwords is an integral part of an information processing system. Stopwords are high frequency words, function words and low-content words. Since these words are not directly relevant to the content of the documents or the query itself, their removal helps in increasing the performance [2]. In our project,the documents as well as the query is first processed using nltk before making the inverted index as it reduces the amount of time taken to read each document. Stopwords are removed using the stopwords package provided by nltk.

from nltk.corpus include stopwords : Using this package,the stopwords like the, a, etc are removed from both, the documents and the query, so that the true content of the data is not lost amongst irrelevant words. Moreover, stopwords are not required for making the inverted index so their removal increases the performance of the system.

2.2.2 Stemming

Stemming is the process of mapping words to some base form. The two main methods are:

i. Dictionary-based stemming which has higher stemming accuracy but also higher implementation costs

ii.Porter-style stemming has lower accuracy but also lower implementation costs and is sufficient for IR

Stemming[2], by mapping several terms to their base forms, increases similarities between documents and queries because they have more common terms after stemming as compared to before stemming. Stemming slightly increases performance and efficiency. Stemming (especially Porter-Style Stemming) has a relatively low processing cost. It also reduces the index size. This property makes it very attractive for use in IR. We have used the PorterStemmer in the implementation of an information retrieval system. It is efficient and easy to implement. This stemmer helps in removing the suffix from a word and obtaining the root word so as to take into account any small variations between the query and the document information. Also,obtaining the root word reduces the number of words that need to be processed for inverted index as all words with the same root are mapped to the same index instead of creating a separate value for each. This further increases the efficiency of the IR system.

2.2.3 Part-of-Speech Tagging

Part-of-Speech tagging refers to the process of assigning a syntactic category to each word present in a text. This is done to resolve any ambiguities in the text.

Part-of-Speech tagging is another feature provided by NLTK that we have used in our implementation of the IR system. POS tagging helps in tagging each word depending on the type of word,i.e,whether it is a noun or a verb and so on. This type of tagging not only reduces the amount of time it would take to compare the information but it also helps in understanding the relevance or the context in which the word has been used [2], [3]. This property helps us rank the documents according to the relevance of the word by seeing how it has been used in the query.

2.2.4 Compounds and Statistical Phrases

Compounds and statistical phrases indexes multitoken units instead of single tokens. In this method, we take pairs of adjacent words which are not stopwords and then use them if their frequency crosses some threshold value [2]. A mixture of single tokens and multitokens is used because only single tokens shouldn't get matched and only multitokens puts restrictions on even slight variations.

This method increases precision. The use of this method in our project is done so as to obtain the context of the words to increase the efficiency of the matching. By processing multiple tokens at a time, we can obtain the most relevant document which contains the entire query phrase and not just the documents which contain any one word. This helps in improving the performance of our system.

2.2.5 Compound Word Splitting

A compound word is formed when we join two or more words. In machine learning and natural language processing writing the compound words as separate words is necessary and makes the processing more efficient [2],[9].

The compound words may contain filler letters. Therefore, we first split the compound word into known words and then perform further splits.

2.2.6 Chunking and Parsing

A chunk basically contains non overlapping regions in a text. In chunking, we divide the text into parts or segments and give them labels or tags such as the ones given by a POS tagger. This method is highly beneficial for entity recognition. Parsing means performing syntax analysis based on a grammar defined [7]. Chunking and parsing are generally used together for performing natural language processing. An example for chunking is Noun-Phrase chunking or NP chunking where, chunks of text are made with noun phrases.

2.2.7 Head-Modifier pairs

Head modifier pairs are used in natural language information retrieval.The pair consists of a head and a modifier and they have a meaningful relation. A head can be paired with more than one modifier. Usually nouns are heads and its attributes are modifiers. It is represented using a binary tree [8],[9].

2.3 Semantic processing of natural language

This technique focuses on processing of natural language using linguistic knowledge. It is sometimes also referred to as semantic processing.

In this method, text is tagged based on grammatical theories. After this, syntax analysis and retrieval is performed on the tagged text. Meaning is extracted from the structure of the text which is composed of phrases and sentences.

There exists a lexical database in english known as the WordNet which includes synonyms referred to as synsets[3], in this context. The WordNet is available for free download as it is very useful for linguistics and natural language processing.

Linguistic Processing is performed when a fast response is needed as semantic processing is a taxing and comparatively a longer process when it comes to information retrieval as a number steps of pre-processing are required.

3. Packages Available

3.1 PyLucene

PyLucene is an extension of Python. It is used for accessing Java Lucene which is basically an open source Java library for indexing and searching from within large collections of documents. Pylucene is not purely python and binary packages are required. It is a wrapper around Java Lucene. It aims to allow you to use Lucene's text indexing and searching capabilities from Python.

3.2 Whoosh

Whoosh is another package which can used for textual information retrieval. This package is fast, provides efficient full-text indexing and searching library implemented in pure Python. It is used by programmers to easily add the search functionality to their applications. Every part of how Whoosh works can be modified to meet the needs of the user.

Some of Whoosh's features include:

i.Fast indexing and fast retrieval

ii.Purely a python package- i.e, no binary packages are required

iii.Python API

iv.Powerful query language

4.Result

In our implementation of IR using inverted index, a database of text documents can be used as the information source. These text documents can be related to any subject. For simplicity, we have made our own corpus and included in it text documents which contains documents about seasons, animals, environment, earth, etc. The implementation involves performing the preprocessing techniques on the query as well as the documents and creating an inverted index for the documents. As an example, the query: "What are aquatic mammals?" gave us the result as shown in Figure 3. This result displays the document number that the information is present in and also displays the entire content of the document. After running our program through several different queries, the Inverted Index method of Information Retrieval was verified

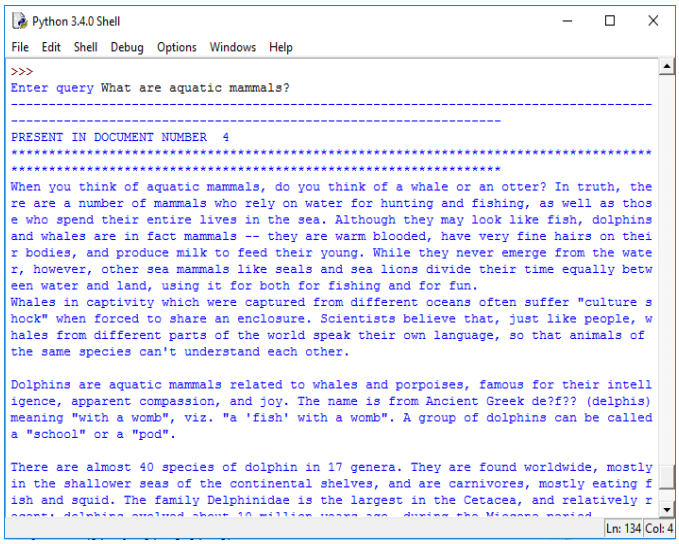

Fig. 3. Screenshot of result of an example query

5.Applications

IR research was initially done on a very small scale. Previously, the applications were limited to bibliographic databases which contained comparatively less amount of data. The processing and IR systems have now developed include different IR models, query processing, weighting of the terms and relevance of the data.

IR researchers have found that retrieval techniques are playing a major role in providing information services. These services are being used more frequently in several applications in order to process data and queries. There are several applications of IR, both general and domain specific. For example, It is used in creating digital libraries [1]. It is also used in filtering information or to create recommender systems.

A major application of IR is media search [1]. IR can be used to perform operations like Blog search, Image Retrieval, music retrieval, speech/video retrieval, etc. Information Retrieval systems are most often used to make search engines which can be of any type- site search, desktop search, mobile search and others. Some other applications which make use of information retrieval include spam filtering, document classification, question-answering systems, Auto-summarization, Cross-lingual retrieval systems and compound term processing.

Information retrieval can also be used in applications specific to the domain. Some domain specific applications include- Information retrieval for chemical structures, Expert search finding, Genomic information retrieval and Geographic information retrieval.

These new applications of information retrieval systems have now made them an invaluable tool in data processing.

6. CONCLUSION

Information Retrieval is a process of searching and retrieving the knowledge based information from a collection of documents. In this paper, we deal with the basics of information retrieval and go on to explain the data structures we used in our implementation and why we preferred one over the other. We also talk about the different NLP techniques that we use in order to achieve this task of Information retrieval. We have described the different methods of processing text in statistical methods of information retrieval such as stop word removal, stemming, POS tagging, processing related to compound words, chunking parsing and using header-modifier pairs. As we have seen, there are several differences between the two methods-statistical and semantic processing. Both these approaches are beneficial but in most applications of NLP, they are used together. This would give a better and accurate result as a number of processing steps are involved along with the usage of linguistic knowledge.

In the field of information retrieval, among the processing methods that we have discussed, the statistical processing techniques are mostly applied in commercial domains. However, it is hard to conclude the efficiency of different processing methods in NLP without taking into account what they are used for or to which domain the technique is being applied in.

Some of the packages used have also been mentioned i.e, PyLucene and Whoosh along with brief description of their origin and usage.

This paper also includes the area of IR applications, where brief details of evolution of IR systems, domains that use IR system and a few other applications have also been mentioned.

REFERENCES

[1] Akram Roshdi and Akram Roohparvar," Review: Information Retrieval Techniques and Applications", Department of Engineering, Khoy branch, Islamic Azad University, Khoy, IRAN, VOL. 3, NO. 9, SEPTEMBER 2015, 373–377

[2] Thorsten Brants, "Natural Language Processing in Information Retrieval", Google Inc. in CLIN 2004

[3] Tulika Narang, "Natural Language Processing Techniques Applied in Information Retrieval-Analysis and Implementation in Python", International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 5, Issue 4 April 2016

[4] Ajit Kumar Mahapatra , Sitanath Biswas,"Inverted Indexes: Types and Techniques", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011

[5] Ellen M. Voorhees and Donna K. Harman (editors), "TREC: Experiment and Evaluation in Information Retrieval" ,National Institute of Standards and Technology) Cambridge, MA: The MIT Press (Digital libraries and electronic publishing series, edited by William Y. Arms), 2005

[6] Steven Bird, Ewan Klein, and Edward Loper , "Natural Language Processing with Python",Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

[7] Giuseppe Attardi and Felice Dell'Orletta, "Chunking and Dependency Parsing" ,

[8] Cornelis H. A. Koster, "Head/Modifier Frames for Information Retrieval" , Computing Science Institute, University of Nijmegan

[9] Tornek Strzalkowski and Barbara Vauthay, "Information retrieval using robust NLP", Courant Institute of Mathematical Sciences.

[10] Ambesh Negi, Mayur Bhirud, Dr.Suresh Jain and Mr.Amit Mittal, "Index Based Information Retrieval System", International Journal of Modern Engineering Research(IJMER), Vol.2,Issue 3,May-June 2012