DOCUMENT RANKING USING INFORMATION RETRIEVAL TECHNIQUES

By:

AVIL ANEJA	2017A7PS0968G
URVIL JIVANI	2017A7PS0947G
VISARG SONEJI	2017A7PS0029G
ADITYA CHOUDHARY	2017A7PS0041G
SHEJAL GUPTA	2017A7PS0122G

STUDY OF PROJECT

-*Under the guidance of Dr. Jajati Keshari Sahoo*

Submitted On: 26 November 2019.

DOCUMENT RANKING USING INFORMATION RETRIEVAL TECHNIQUES

By:

AVIL ANEJA	2017A7PS0968G
URVIL JIVANI	2017A7PS0947G
VISARG SONEJI	2017A7PS0029G
ADITYA CHOUDHARY	2017A7PS0041G
SHEJAL GUPTA	2017A7PS0122G

A report submitted in partial fulfilment of the requirements of the project.

BITS PILANI, K K BIRLA

GOA CAMPUS

ACKNOWLEDGEMENT

First of all, we are deeply grateful to the supreme almighty with whose blessings, we were able to accomplish this humble piece of work.As a special mention, we would like to express our gratitude to our mentor Dr. Jajati Keshari Sahoo under whom we were working for this project.His encouragement and constant support has lead us to a successful attempt on this project.

We would also like to thank Prof. Raghurama G, director of BITS Goa to provide us an opportunity to study in this college. His constant support and untiring work helped us towards facilitating this learning experience.

We would like to thank BITS Goa CS and Math Department for their constant support in academics due to which we were able to work on this project. Knowledge learnt from them came to be an immense help for this project.

ABSTRACT

1960s — researchers were testing web search engines on about 1.5 megabytes of text data. Fast forward to 2019, we now have billions of web pages and colossal data. Though one issue which still persists is relevance. Relevance is the core part of Information Retrieval. A relevant search result is one in which a person gets what he/she was searching for.

Naively you could go about doing a simple text search over documents and then return results. Obviously, it won't work mainly due to the fact that language can be used to express the same term in many different ways and with many different words — the problem referred to as *vocabulary mismatch problem* in IR. One other issue is to maintain a line between topical relevance (relevant to search query if it's of the same topic) and user relevance.

To address issues mentioned above regarding relevance, researchers propose retrieval models. *A retrieval model is a formal representation of the process of matching a query and a document.* It is the basis of the ranking algorithm that is used in a search engine to produce the ranked list of documents. A good retrieval model will find documents that are likely to be considered relevant by the person who submitted the query.

In this project we have focussed on some of the techniques of Information Retrieval to extract the relevant data (like clustering, NLP etc*). Next, we have used the basic google web page algorithm based on Markov chain to calculate the PageRank of each document considering it as a node and connection as edges.

TABLE OF CONTENTS

INTRODUCTION

PROBLEM STATEMENT

In an era of megabytes of text documents either it is on webpages or general documents, it became an essential to extract the documents which are most relevant to the query processed. This makes informational retrieval field to be an important field to look about.

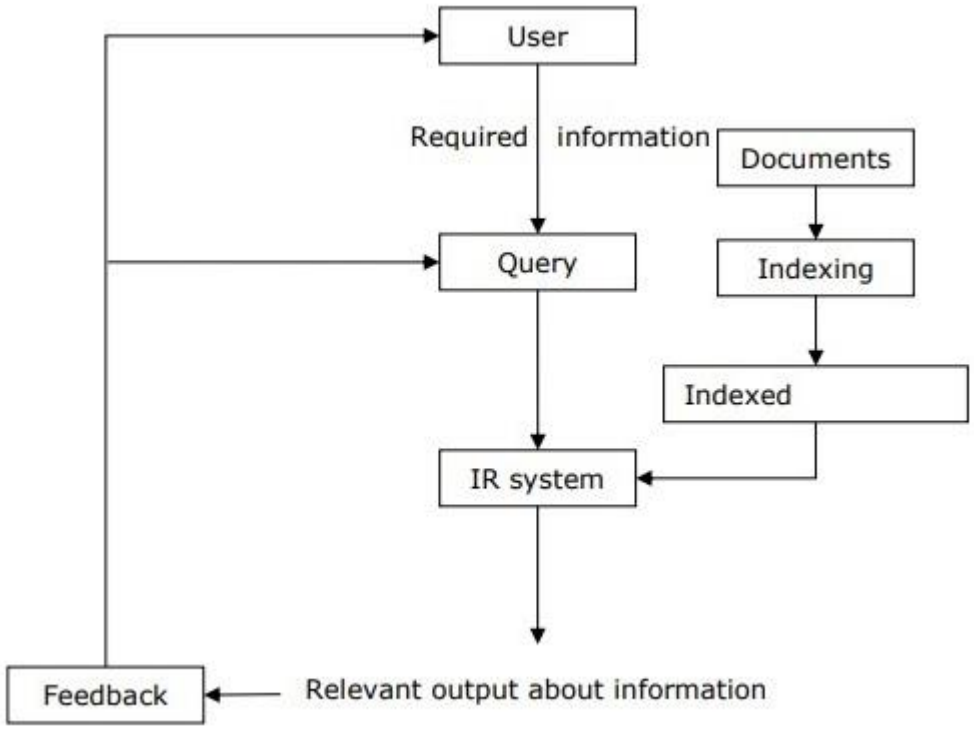
In this project we have to retrieve the relevant documents according to the query processed and apply some PageRank algorithm to rank them in order of their cosine similarity with the query processed.

WHAT IS INFORMATION RETRIEVAL?

Information retrieval (IR) may be defined as a software program that deals with the organization, storage, retrieval and evaluation of information from document repositories particularly textual information. The system assists users in finding the information they require but it does not explicitly return the answers of the questions. It informs the existence and location of documents that might consist of the required information. The documents that satisfy the user's requirement are called relevant documents. A perfect IR system will retrieve only relevant documents.

The typical approaches in information retrieval adopt probabilistic models. For example, a text document can be regarded as a bag of words, that is, a multiset of words appearing in the document. The document's language model is the probability density function that generates the bag of words in the document. The similarity between two documents can be measured by the similarity between their corresponding language models.

With the help of the following diagram, we can understand the process of information retrieval (IR) −



It is clear from the above diagram that a user who needs information will have to formulate a request in the form of query in natural language. Then the IR system will respond by retrieving the relevant output, in the form of documents, about the required information.

INTRODUCTION TO PAGERANK

PageRank (**PR**) is an algorithm used by Google Search to rank web pages in their search engine results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages. PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

PageRank is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring"

its relative importance within the set. The algorithm may be applied to any collection of entities with reciprocal quotations and references. The numerical weight that it assigns to any given element *E* is referred to as the PageRank of *E* and denoted by PR(E).

A PageRank results from a mathematical algorithm based on the webgraph, created by all World Wide Web pages as nodes and hyperlinks as edges, taking into consideration authority hubs such as cnn.com or usa.gov. The rank value indicates an importance of a particular page. A hyperlink to a page counts as a vote of support. The PageRank of a page is defined recursively and depends on the number and PageRank metric of all pages that link to it ("incoming links"). A page that is linked to by many pages with high PageRank receives a high rank itself.

In this project we will use the basic google web page algorithm. As it is based on webgraph , we will make the documents as nodes and if there is a connection between any two documents then those documents will be linked through an edge. After this going through the algorithm we will compute the PageRank of each document. Higher the value, the higher relevance to the query processed.

METHODOLOGY

TEXT PROCESSING:

1.Stop Words Removal -

In computing, **stop words** are words which are filtered out before processing of natural language data (text). Stop words are generally the most common words in a language; there is no single universal list of stop words used by all-natural language processing tools, and indeed not all tools even use such a list. Some tools avoid removing stop words to support phrase search.

Any set of words can be chosen as the stop words for a given purpose. For some search engines, these are some of the most common, short function words, such as *the*, *is*, *at*, *which*, and *on*. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as "The Who", "The The", or "Take That". Other search engines remove some of the most common words—including lexical words, such as "want"—from a query in order to improve performance.

Hans Peter Luhn, one of the pioneers in information retrieval, is credited with coining the phrase and using the concept.] The phrase "stop word", which is not in Luhn's 1959 presentation, and the associated terms "stop list" and "stoplist" appear in the literature shortly afterwards.

In search engine optimization, stop words are of interest because it is not useful to bid on phrases including them.

With Stop Words	Without Stop Words
/growing-up-with-hearing-loss/	/growing-hearing-loss/
/coming-to-terms-with-hearing-loss/	/coming-terms-hearing-loss/
/the-world-of-being-hearing-impaired/	/world-being-hearing-impaired/
/echo-in-ears-from-talking-people/	/echo-ears-from-talking-people/
/listening-is-exhausting/	/listening-exhausting/
/living-with-hearing-loss/	/living-hearing-loss/
/what-is-hearing-loss/	/what-hearing-loss/

Example showing sentences with stop words and without stop words

After removing stop words we have performed two operations on text i.e Stemming and Lemmatization.The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

2. Stemming -

In linguistic morphology and information retrieval, **stemming** is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Algorithms for stemming have been studied in computer science since the 1960s. Many search engines treat words with the same stem as synonyms as a kind of query expansion, a process called conflation.

A computer program or subroutine that stems word may be called a *stemming program*, *stemming algorithms*, or *stemmer*.Porter Stemmer,Snowball Stemmer,Lancaster Stemmer are commonly used Stemmers in the industry.We have used Porter Stemmer in our project.

Example - A stemmer for English operating on the stem *cat* should identify such strings as *cats*, *catlike*, and *catty*. A stemming algorithm might also reduce the words *fishing*, *fished*,

and *fisher* to the stem *fish*. The stem need not be a word, for example the Porter algorithm reduces, *argue*, *argued*, *argues*, *arguing*, and *argus* to the stem *argu*.

3. Lemmatization -

Lemmatisation (or lemmatization) in linguistics is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form.

In computational linguistics, lemmatization is the algorithmic process of determining the lemma of a word based on its intended meaning. Unlike stemming, lemmatization depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document. As a result, developing efficient lemmatization algorithms is an open area of research.

Lemmatization is closely related to stemming. The difference is that a stemmer operates on a single word *without* knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on the part of speech. However, stemmers are typically easier to implement and run faster. The reduced "accuracy" may not matter for some applications. In fact, when used within information retrieval systems, stemming improves query recall accuracy, or true positive rate, when compared to lemmatization. Nonetheless, stemming reduces precision, or true negative rate, for such systems.

For instance:

1. The word "better" has "good" as its lemma. This link is missed by stemming, as it requires a dictionary look-up.
2. The word "walk" is the base form for the word "walking", and hence this is matched in both stemming and lemmatization.
3. The word "meeting" can be either the base form of a noun or a form of a verb ("to meet") depending on the context; e.g., "in our last meeting" or "We are meeting again tomorrow". Unlike stemming, lemmatization attempts to select the correct lemma depending on the context.

Document indexing software like Lucene can store the base stemmed format of the word without the knowledge of meaning, but only considering word formation grammar rules. The stemmed word itself might not be a valid word: 'lazy', as seen in the example below, is stemmed by many stemmers to 'lazi'. This is because the purpose of stemming is not to produce the appropriate lemma – that is a more challenging task that requires knowledge of context. The main purpose of stemming is to map different forms of a word to a single form. As a rules-based algorithm, dependent only upon the spelling of a word, it sacrifices accuracy to ensure that, for example, when 'laziness' is stemmed to 'lazi', it has the same stem as 'lazy'.

Commonly used Lemmatizers are Wordnet,Spacy,TextBlob,CLips Pattern,etc.We have used Wordnet in this project.

Raw Term Frequency Model

Machine learning algorithms cannot work with raw text directly. The text should be regenerate into vectors of numbers. In tongue process, a standard technique for extracting options from text is to put all of the words that occur within the text in an exceedingly bucket. This approach is termed a bag of words model or BoW for brief. It’s named as a “bag” of words as a result of any data concerning the structure of the sentence is lost.

By exploitation the bag of words to a group, we will mechanically take away any duplicate words.

Another drawback with the bag of words approach is that it doesn’t account for noise. In different words, sure words area unit want to formulate sentences however don't add any linguistics desiring to the text. for instance, the fore most normally used word within the West Germanic is that that represents seven-membered of all words written or spoken. You

couldn’t build deduce something a few texts given the actual fact that it contains the word the.

On the opposite hand, words like smart and awing may well be wont to confirm whether or not a rating was positive or not. In tongue process, useless words area unit named as stop words. The python tongue toolkit library provides an inventory of english stop words.

1.Term Frequency (TF)

Term frequency (TF) is employed in reference to data retrieval and shows however often an expression (term, word) happens during a document. variety|the amount|the quantity} of times a word seems during a document divided by the whole number of words within the document. each document has its own term frequency. Term frequency indicates the importance of a specific term at intervals the document. Term frequency may be informative once it's set in regard to text length. this may provide you with keyword density.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

2. Inverse information Frequency (IDF)

The log of the quantity of documents divided by the quantity of documents that contain the word w. Inverse information frequency determines the burden of rare words across all documents within the corpus. It additionally live whether or not a term is common or rare in a very given document corpus. it's obtained by dividing the whole variety of documents by the quantity of documents containing the term within the corpus.

$$idf(w) = log(\frac{N}{df_t})$$

3. Tf-idf Model

tf-idf stands for Term frequency-inverse document frequency. The TF*IDF (term frequency-inverse document frequency) formula for text optimisation conjointly uses term frequency. The frequency of a keyword is viewed in regard to the document length. At an equivalent time, logarithms guarantee terms that occur a lot of often don't seem to be weighted too heavily. Moreover, words that area unit quite common within the language (such as conjunctions, prepositions, articles) area unit weighted lower. The tf-idf weight could be a weight typically utilized in info retrieval and text mining. Variations of the tf-idf weight theme area unit typically employed by the search engines in rating and ranking a document's relevance given a question. This weight could be an applied math live wont to assess however vital a word is to a document during an assortment or corpus. The importance will increase proportionately to the quantity of times a word seems within the document however is offset by the frequency of the word within the corpus (data-set).

It is merely the TF increased by Israeli Defense Force.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

4. Cosine Similarity

Cosine similarity measures similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The technique of cosine similarity is also used to measure cohesion within clusters in the field of Information Retrieval.

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q} \bullet \vec{d}}{\left|\vec{q}\right|\left|\vec{d}\right|} = \frac{\vec{q}}{\left|\vec{q}\right|} \bullet \frac{\vec{d}}{\left|\vec{d}\right|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2} \sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

1. $q_{i\ is}$ here the tf-idf score of term *i* in the query.
2. d_i is here the tf-idf score of term *i* in the document.
3. cos(*q*,*d*) is the cosine similarity of *q*and *d*… or, equivalently, the cosine of the angle between *q* and *d*.

The similarity ranges from −1 meaning exactly the opposite, to 1 meaning exactly the same, with 0 indicating orthogonality or decorrelation of documents, while in-between values indicate intermediate similarity or dissimilarity of documents.

5. Clustering

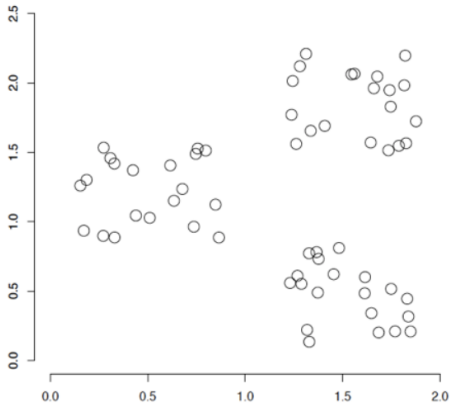
Clustering is that the method of grouping a collection of objects into categories of comparable objects: -

•Documents inside a cluster ought to be similar.

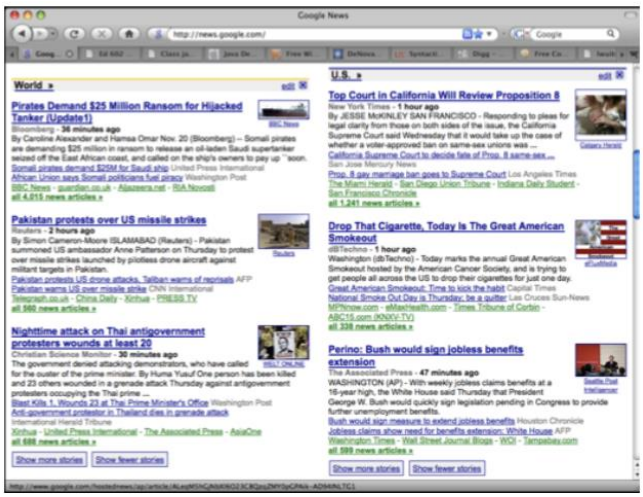
•Documents from completely different clusters ought to be dissimilar.

Cluster hypothesis - Documents within the same cluster behave equally with regard to connectedness to data wants.

A data set with clear cluster structure: -



Google News: automatic bunch offers a good news presentation figure :-



Clustering Algorithms

- •Flat algorithms
- •Usually begin with a random (partial) partitioning
- •Refine it iteratively
- •K suggests that bunch
- •(Model primarily based clustering)
- •Hierarchical algorithms
- •Bottom-up, collective
- •(Top-down, divisive)

1) K-means Clustering

Clusters supported centroids (aka the middle of gravity or mean) of points during a cluster, c:

Algorithm

Select K random docs as seeds. Until bunch converges (or different stopping criterion):
For each doc di:
Assign di to the cluster cj such dist(xi, sj) is borderline.
(Next, update the seeds to the centre of mass of every cluster) for every cluster cj
sj= m(cj)

$$\vec{\mu}(c) = \frac{1}{|c|}\sum_{\vec{x}\in c}\vec{x}$$

2. Hierarchical Clustering

- **Single Link Agglomerative Clustering -** It use most similarity of pairs and may lead to "straggly"(long and thin) clusters thanks to chaining result.

$$sim(c_i, c_j) = \max_{x\in c_i, y\in c_j} sim(x, y)$$

After merging c_i and c_j, similarity of the new cluster to another cluster, c_k, is :-

$$sim((c_i \cup c_j), c_k) = \max(sim(c_i, c_k), sim(c_j, c_k))$$

- **Complete Link -** It use minimum similarity of pairs and makes "tighter,"spherical clusters that area unit usually desirable.

$$sim(c_i, c_j) = \min_{x\in c_i, y\in c_j} sim(x, y)$$

After merging c_i and c_j , similarity of the new cluster to another cluster, c_k, is:

$$sim((c_i \cup c_j), c_k) = \min(sim(c_i, c_k), sim(c_j, c_k))$$

- **Group Average -** Here similarity of 2 clusters = average similarity of all pairs inside integrated cluster. It compromise between single and complete link.

There area unit 2 options:

•Averaged across all ordered pairs within the integrated cluster

•Averaged over all pairs between the 2 original clusters

$$sim(c_i, c_j) = \frac{1}{\left|c_i \cup c_j\right|\left(\left|c_i \cup c_j\right| - 1\right)} \sum_{\vec{x} \in (c_i \cup c_j)} \sum_{\vec{y} \in (c_i \cup c_j): \vec{y} \neq \vec{x}} sim(\vec{x}, \vec{y})$$

- **Ward's method -** It is supported the applied mathematics property of variance. The variance of a group of numbers measures however detached the numbers square measure.Ward's methodology tries to reduce the add of the variances of the clusters. This ends up in compact clusters with a nominal quantity of unfold around the cluster centroids. The cost, which is slightly more complicated to compute than the previous methods, is computed according to

$$COST(C_i, C_j) = \sum_{k \neq i,j} \sum_{X \in C_k} (X - \mu_{C_k}) \cdot (X - \mu_{C_k}) + \sum_{X \in C_i \cup C_j} (X - \mu_{C_i \cup C_j}) \cdot (X - \mu_{C_i \cup C_j})$$

where $C_i \cup C_j$ is the union of the instances in clusters C_i and C_j, and $\mu_{C_i \cup C_j}$ is the centroid of the cluster consisting of the instances in $C_i \cup C_j$. This cost measures what the intracluster variance would be if clusters i and j were joined.

3. K nearest neighbor clustering

It's one terribly straightforward manner of manufacturing overlapping clusters.

•K here is extremely completely different from the K in K-means cluster

• A cluster is created around each input instance.

•For input instance x, the K points that square measure nearest to x in keeping with far metric and x itself type a cluster.

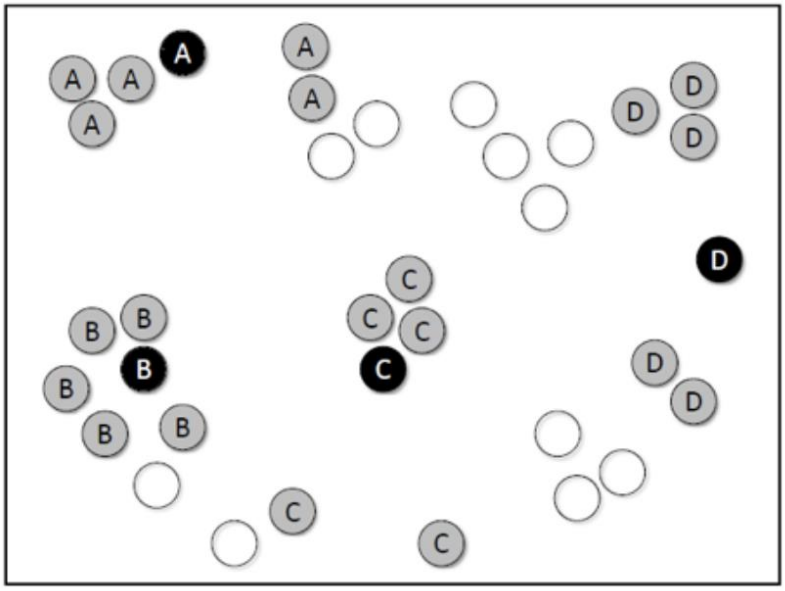
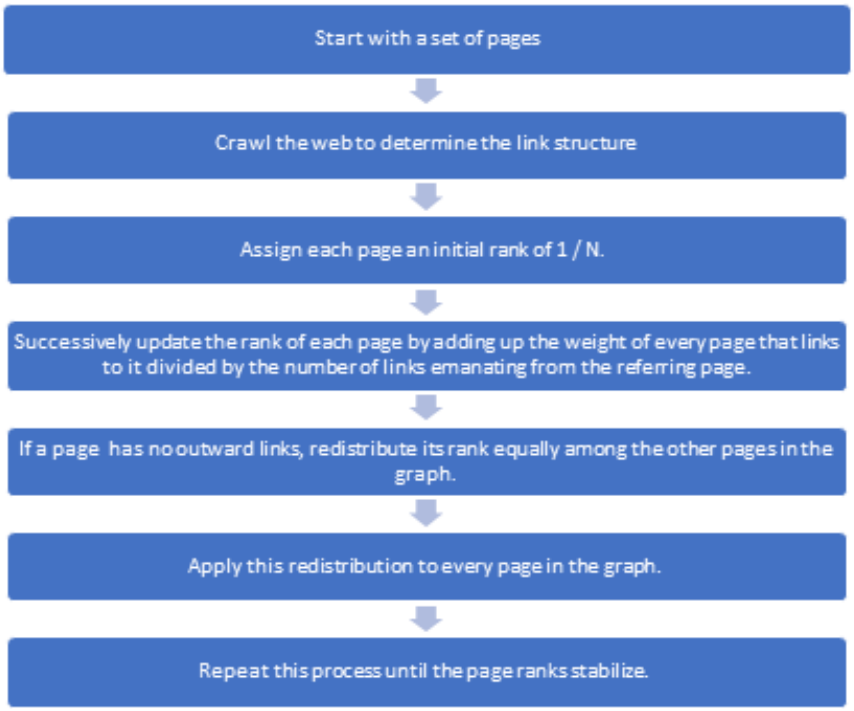


Fig. 9.16. Example of overlapping clustering using nearest neighbor clustering with $K = 5$. The overlapping clusters for the black points (A, B, C, and D) are shown. The five nearest neighbors for each black point are shaded gray and labeled accordingly.

PAGE RANK ALGORITHM

The PageRank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on the links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process. The PageRank computations require several passes, called “iterations," through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value. Below shown is a flowchart how simple algorithm works :-



MODEL EXECUTED

IMPORTANT PYTHON FUNCTIONALITIES USED

1.NLP with NLTK

2. Basic NLP Pipeline -

- Tokenization
- StopWord Removal
- Stemming (three types -Snowball Stemmer (Multilingual) , Porter Stemmer ,LancasterStemmer)
- Lemmatization

3. NLP using spaCy

4. Count vectorizer (for ranking - from sklearn.feature_extraction.text import CountVectorizer)

5. Tf-idf Normalisation Using sklearn

6. Agglomerative clustering (from sklearn.cluster import AgglomerativeClustering)

7. K means Clustering (from sklearn.cluster import KMeans)

8. Cosine Similarity (from sklearn.metrics.pairwise import cosine_similarity)

CONCLUSIONS AND RECOMMENDATIONS

As our final result we were able to crawl through the websites related to query and provide the pagerank corresponding to the user requirement. Some of the screenshots of various intermediate steps are shown as results :

Result after applying stemming and lemmatization upto 10th level and kth level respectively :

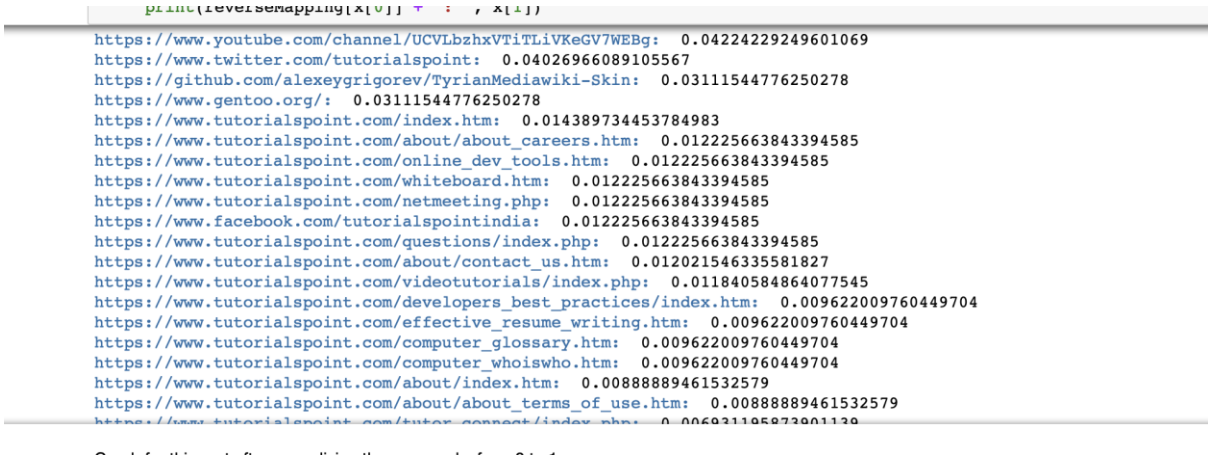
```
In [45]: calc(10)

Results - (Precision, Recall) for Level  10 :
Query  1 :  (0.9, 0.9)
Query  2 :  (0.5, 0.7142857142857143)
Query  3 :  (0.5, 0.5)
Query  4 :  (0.1, 0.14285714285714285)
Query  5 :  (0.8, 1.0)
Query  6 :  (0.5, 0.8333333333333334)
Query  7 :  (0.7, 0.875)
Query  8 :  (0.8, 0.8888888888888888)
Query  9 :  (0.3, 0.375)
Query  10 :  (0.8, 0.8)
```

```
In [28]: kthPrecisionRecall(10, ii.termList, queries, k_dict,finalDocs)

RECALL
0.3
PRECISION
0.007936507936507936
RECALL
0.2
PRECISION
0.0021436227224008574
RECALL
0.1
PRECISION
0.0012048192771084338
RECALL
0.1
PRECISION
0.0012484394506866417
RECALL
0.3
PRECISION
0.005494505494505495
RECALL
0.2
PRECISION
0.004376361487965
RECALL
0.3
PRECISION
0.0049180327868852
RECALL
0.0
PRECISION
0.0
RECALL
0.2
PRECISION
0.0029411764705882353
RECALL
0.1
PRECISION
0.002008032128514056
AVERAGE RECALL
0.18
AVERAGE PRECISION
0.003227150388207655
```

Final PageRank calculated:



For increasing the accuracy, we can further implement more advanced PageRank algorithms like Rank net. In our project we basically focused on the Information retrieval part and tried to extract that web pages which are related to that. Further improvements can be gained by removing named entity disambiguation at the start or by using cache of the user for searches.

FUTURE SCOPE

Generally, Page Rank of a web page is computed using frequency of the keywords appearing in it which is sometimes exploited towards giving undue promotions to websites by including the keywords repetitively at many times, therefore, falsely raising its PageRank.Thus instead of putting emphasis on the frequency of the keywords appearing in the web page alone, it is suggested that the content of the web pages in terms of the semantics should also be considered towards raising the rank of the document in question. This work can also be extended to incorporate the integration of semantic information drawn from a web application's domain knowledge into all phases of the web usage mining process (pre-processing, pattern discovery, and recommendation/prediction) with the goal to have an intelligent semantics-aware web usage mining framework.

Since artificial intelligence field is growing dramatically, another important prediction about the future of SEO is that the search results will be completely monitored and handled by artificial intelligence. As we get advanced with technology and the machines began to understand the content better with time, the results will get more accurate.

We can also optimize our content by creating what are target audience wants to see. However this will deflect from the query run by user but combining both things will result in better search results.

REFERENCES

https://en.wikipedia.org/wiki/Tf–idf

https://www.geeksforgeeks.org/tf-idf-model-for-page-ranking/

https://www.machinelearningplus.com/nlp/cosine-similarity/

https://www.machinelearningplus.com/nlp/cosine-similarity/

https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Chang X, Spitkovsky I. , Christopher D. Manning, Eneko Agirre . A comparison of Named-Entity Disambiguation and Word Sense Disambiguation . Computer Science Department, Stanford University, Stanford, CA, USA

Roberts Eric , The Google PageRank Algorithm , Stanford University

https://www.geeksforgeeks.org/page-rank-algorithm-implementation/

Rogers Ian, The Google Pagerank Algorithm and How It Works. IPR Computing Ltd.

Antoine Bordes, et al. "Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing." AISTATS(2012)

Mikolov, et al. "Distributed representations of words and phrases and their compositionality." ANIPS(2013): 3111-3119(word2vec)

Sutskever, et al. "Sequence to sequence learning with neural networks." ANIPS(2014)

Learning to Rank for Information Retrieval and Natural Language Processing - Hang Li