ISSN No. 0976-5697

Volume 8, No. 3, March – April 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Book Recommendation using Cosine Similarity

Rofeca Giri Rymmai
Department of Computer Science
Christ University
Bengaluru, India

Saleema JS
Department of Computer Science
Christ University
Bengaluru,India

Abstract: Recommendations have been a driving force for the sale of products. For years, recommendations have always been based upon the product review of customer which in a later stage, was upgraded to personalization, whereby customers having similar purchasing patterns are clubbed together and their preferences are interchanged. This paper takes on a different approach when it comes to recommendation of books. Books are usually referred based on the author, genre and book ratings. Each book has its own plot summary which is different from the description that gives only a gist of the book. These plot summaries can be used to find how two books are very much alike. The paper explains how this is achieved by the calculation of the degree of likeness between the two plots based on the terms that are used. It involved the application of text mining to find the significant terms that will help to contribute to finding the angle of closeness using the mathematical calculation of cosine similarity.

Keywords: Recommendation system, Information retrieval, TF-IDF score, Cosine similarity.

I. INTRODUCTION

Recommendations are a way of making a suggestion. A recommendation system is also termed as a recommendation engine that helps in making a suggestion to a customer based on available data, by analyzing what the user might be interested in. A recommender system helps in customer retention by helping the user to have a better experience with common methods like personalized recommendations. It ultimately leads to the boost in sales of a product(s) by endorsing other products which implicate the search product. Amazon product sales increased by 20%-30% due to personalize recommendation [1]**.** Most recommender frameworks adopt two essential strategies: collaborative filtering or content-based filtering. The first scenario, collaborative filtering is the one that depends on a model of earlier customer conduct and hence is based solely on the customer conduct. The second is based on other customers with a similar characteristic, which utilizes a mass information to shape a suggestion of like customers. Search engines can also be one type of recommendations engines that respond to a user, based on the search query [2]. In most scenarios, suggestions are based upon the deals and ratings. In Amazon dataset fields like “people who bought this also bought this” are one of the fields that aid in the recommendation. Recommendations based on search query has been made easy with the help of text mining.

As termed by Anshika Singh “Text mining is used to find knowledge in unstructured or semi-structured data”. It can also be termed as the process of extracting terms. Content mining, otherwise called content information mining or learning disclosure from printed databases, alludes toward the way of extricating intriguing and non-trifling examples or learning from content records. Text mining is the discovery by computer of new previously unknown information by automatically extracting information from different written resources [3]. Most data that is available is mostly unstructured and contains text that has a potentially high value. The main aim of text mining is to extract all the text into an expansive amount of important information. Knowledge Discovery Text (KDT), while profoundly established in NLP, draws on strategies from insights, machine learning, thinking, data extraction, information administration, and others for its revelation procedure [3]. Text mining tools are designed for working with both structured and unstructured data. The techniques that are used for performing text mining are information extraction, topic tracking, summarization, categorization, clustering, and information retrieval.

Information retrieval (IR) focuses on retrieving information based on a search input. It is based on the content of unstructured components. It is mainly used for finding documents that satisfy the request generated by the user. Information retrieval is concerned with the retrieval of information that matches a user-specific need. This research paper applies a content-based filtering method to recommend a book based on the plot summary of the book that the user has searched. The plot summary basically gives a brief description of the book. Vector space model using the cosine similarity is being used

II. LITERATURE REVIEW

A. *Recommendation System*

The first recommender system was produced by Goldberg, Nicholas, and Oki & Terry in 1992. Woven artwork was an electronic informing framework that permitted clients to either rate messages be it great or awful. Recommender system was characterized by Deshpande and G. Karypis as “a customized data filtering innovation used to either foresee whether a specific client will like a specific thing or to recognize an arrangement of n items that will hold any importance with a certain user” [4]**.** Recommended framework shape or work from particular kind of data separating framework system that attempts to suggest data things or social components that are liable to hold any importance with the client. The most common approach consists of finding the list of customers who have made a purchased as well as rated an item. The algorithm then adds items from the similar customer, takes out the items that the customer has as of now obtained or evaluated and suggests the remaining things to the client. Two mainstream adaptations of these algorithms are collaborative filtering and cluster models. Other algorithms include search based techniques and item-to-item collaborative filtering which concentrates on finding similar items and not the clients [5].

B. *Text Mining*

Text mining is profoundly established in Natural Language Programming (NLP). It additionally draws on techniques from statistics, machine learning reasoning, and information. Some of the technologies developed are information extraction, topic tracking, summarization, categorization, clustering, concept linkage, information, visualization and question answering [6]. Text mining applications are used as a part of the following areas like distributing and media for listing, creating and improvement of data recovery, broadcast communications, energy and different administrations, information technology divisions and banks, insurance and money related markets, political foundations, pharmaceutical and think-tanks and human services. It is additionally appropriate for the learning and human asset administration for focused insight and extraction change stacking. Another application is for analyzing staffs' opinions and observing the level of fulfillment. Text mining utilized as a part of client relationship administration is implied for dealing with the substance of customer's message. The market analysis then again utilizes text mining to investigate contenders to distinguish new potential clients. Text mining can help an organization determine possibly significant business experience from text-based substance, for example, word, archives, email and posting via web-based networking media streams like Facebook, Twitter, and LinkedIn.

C. *Cosine similarity*

A collaborative filtering algorithm represents a client as an N-dimensional vector of items, N represents the quantity of distinct index items. The segments of the vector are certain for purchased or decidedly appraised items and negative for contrarily appraised items [7]. The algorithm duplicates the vector segments by the inverse frequency, making less frequently known items more relevant. It produces results depending upon the similar user. It achieves this by the measure of the cosine of the angle between the two vectors.

III. Dataset Description

The plot summaries have been extricated from Wikipedia, alongside adjusted metadata from Freebase, including book writer, title, and type. For the strategy of recommendations, the data fields mostly concentrated upon are the book title, author, book genre, plot summary. Given below is a sample of the data and metadata available.

Table I. Dataset sample

Description	Fields
1166383	Wikipedia ID
/m/04cvx9	Freebase ID
White Noise	Book title
Don Delillo	Book author
1985-01-21	Publication date
Novel, Postmodernism, Speculative, Fiction	Genres

The Plot summary consist of a paragraph of text that is used in giving the summary description of a particular book. The data has around 179 genres. Out of these genre Fiction has been picked as test data. Under fiction there are a total of 947 books out of which the plot summaries of fourteen books have been used.

IV. Methodology

The proposed recommendation system recommends a search query to the users. There are two scenarios, one based on the plot summary of the book searched is fed as the query. The system uses the plot summary of the books from the database as the documents and plot summary of the searched book as the query. The proposed system can also make recommendations based on a search query of the user. It achieves the recommendation by following the vector space model. The second scenario is based upon the book title or the searched keywords, fed as the query. The system goes through the following procedures:

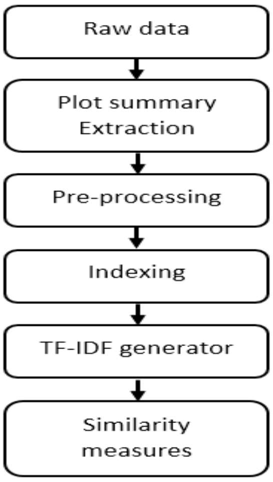

Figure 1: Flow diagram of process involve to find the cosine measures.

D. *Plot summary extraction*

The raw data contains series of metadata about the book. The plot summary of the book is extracted into a text file and all the files are stored in one folder. The filename of each of the text file will be the title of the book. When the files are fed into a corpora and later constructed into a term document matrix, the name of the file, which is the book title is set as the column header of the matrix. This helps in retrieving the other information about the book, using the book title as index.

E. *Pre-processing*

Data pre-processing refers to transforming the raw data into more refined information that can be used for a particular strategy. The data is taken in as corpora. A corpora is plural for corpus, it is a storage of text into a structured format that can later be easily used for statistical and analytical purpose. It can contain a particular metadata in the type of label or can contain archive particular metadata. The corpora utilized here is a record particular corpora where every report contains the plot rundown of the book. It helps in diminishing the intricacy of the report. Utilizing the corpora, the accompanying pre-processing systems are performed.

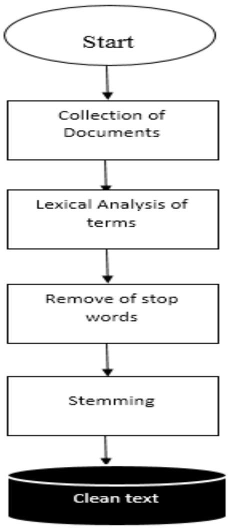

Figure 2. Process involved in preprocessing

1) Lexical analysis

It is a way towards breaking flood of characters into a surge of words. Along these lines, one if the major targets of lexical analysis is the recognizable proof of the words in the content. However, there is something else entirely to it than this. In lexical analysis there are mainly three features

- Numbers: They are generally not great records without an encompassing setting. There are scenarios that number are not of significance.
- Content transformation: It revolves around the fact that the text can be converted to lower-case.
- Stripe whitespace: They are just irrelevant space that are not suitable for mining.

2) Removal of stop-words

Stop-words ought to be removed from a content as they only clutter up space and the terms are not of importance, they do not contribute to the discovery of unknown pattern. Removing stop words lessens the dimensionality of the term space. The most widely recognized words in text records are articles, pronouns, and prepositions that do not give the significance of the reports. Stop-words are regarded immaterial for searching purpose since they occur every now and then in the dialect for which the indexing engine has been tuned. In order to spare both space and time, these words are dropped at indexing time and afterward overlooked at search time.

3) Stemming

It refers to the process of expelling prefixes and suffixes from words to diminish them to stems along these lines, wiping out tag-of-speech and other verbal or plural form. Stemming refers to the mapping of the word structure to stems or essential word frames. Word structure may vary from stems because of morphological changes essential for linguistic reasons. For instance, the plural renditions of English things are generally formed by adding an ‘s’ to the basic noun. The figure below helps in depicting the word count before and after preprocessing.

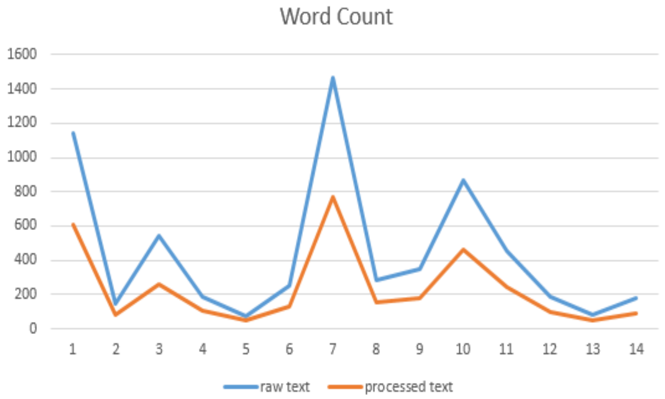

Figure 3. Word count before and after preprocessing.

The x-axis indicates the word count values and the y-axis, the document index. It can be clearly noted that preprocessing helps in removing stop words. The results shows that if a text document has a large number of words, the word count drastically decreases after preprocessing. It indicates that more the word count before preprocessing, the more is the number of stop words in the document as seen in document 1,7 and 10 (Figure 3). On the contrary, small documents contain more significant words and results in small decrease of the word count

F. Indexing

For indexing the content bearing term are extracted from the document text. It is obvious that many of the words in a document do not describe the content, for example stop words, or words like *the, is*. By utilizing report ordering, those non-significant words (function words) are removed from the document vector, so the document might be spoken to by content bearing words. This indexing can be based on term frequency, where terms that have both high and low frequency within a document are considered to be function words. Each of the term encountered are counted and mapped into the document in which they reside from. Practically speaking, term frequency has been hard to implement in automatic indexing. Instead the use of a stop list which holds common words to remove high frequency words (stop words), which makes the indexing method, language dependent. In general, 40-50% of the total number of words in a document are removed with the help of a stop list [8].

G. TF-IDF generator

The generator is a procedure of weighting of the indexed terms to enhance retrieval of the document relevant to the user. Term weighting has been clarified by controlling the comprehensivity and specificity of the search, where the thoroughly is identified with review and specificity to accuracy [9]. The term weighting for the vector space model has entirely been based on single term statistics. The following are main factor for generating the TF-IDF:

a) Term frequency factor

“ A common weighting scheme for terms within a document is to use the frequency of occurance” as stated by Luhn [10]. The term frequency is somewhat content descriptive for the documents and is generally used as the basis of a weighted document vector. In general, this factor is called collection frequency document. Term frequency is simply defined as a measure of occurrence of a term in a particular document.

The higher the occurrence the less important is the term found.

b) Normalization factor.

The second weighting factor is a document length normalization factor. It is of importance due to the reason that the documents of the plot summary can be both large and small. Long documents can more likely have a terms with higher term frequency than that of a smaller document. The term can be equally important on both but can be of more occurring on the large document. Hence the term needs to be normalized depending upon the size of occurrence and the size of the document. A simple way to normalize is to divide the term frequency by the total number of terms. Thus, the term frequency is often divided by the document length (the total number of terms in the document) as a way of normalization:

$$TF\ (t) = N_t / T$$

where,

N_t- Number of times t appear in a document.

T- Total number of terms in the document.

c) Inverse document frequency

The inverse document frequency helps in finding the relevant terms. It finds the rarity of a term and helps to show how important a word is. Term frequency considers all the terms as equally important. If the stop words are not removed, they would have a higher weight than most terms, which are not relevant to the search query. Certain terms, such as "is", "of", and "that", may show up a great deal of times but however have little significance. In this manner we have to weigh the successive terms while scale up the uncommon ones by figuring the following:

$$IDF\ (t)= \log_e(D/D_t)$$

where,

D- Total number of documents.

D_t- Number of documents with term t in it.

d) TF-IDF

TF-IDF stands for term frequency-inverse document frequency. The TF-IDF is used for evaluating the importance of a term is to a document that is in the corpus. The significance expands relatively to the number of times a word shows up in the document, but however, is balanced by the recurrence of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance, given a user query. One of the least complex positioning capacities is processed by summing the TF-IDF for every question term; numerous more refined positioning capacities are variations of this basic model. TF-IDF can be effectively utilized for stop-words separating in different subject fields including text summarization and classification.

H. Similarity Measures

The cosine similarity between two vectors (or two documents on the Vector Space) is a measure that calculates the cosine of the angle between them. This metric is a measurement of orientation and not magnitude, it can be seen as a comparison between documents on a normalized space because not only the magnitude of each word count (TF-IDF) is taken into consideration of each document, but the angle between the documents. In order to build the cosine similarity equation is to solve the equation of the dot product for the following

$$\cos\theta = \frac{\vec{a}.\vec{b}}{||\vec{a}|| * ||\vec{b}||}$$

where,

$$\vec{a}.\vec{b} = ||a||\ ||b|| \cos\theta$$

This is the cosine similarity formula. Cosine Similarity will generate a metric that says how related are two documents by looking at the angle instead of magnitude. The cosine when broken down into simpler terms can be equated as follows:

$$Cosine\ Similarity(Query, Document1) = \frac{Dot\ Product(Query, Document1)}{||Query|| * |Document1|}$$

V. RESULTS

The recommendations are based on two scenarios. The first scenario is that of recommending based on a search query. The second recommendation of a book is based on the plot summary of the searched book. Both the scenarios have a common procedure where the TF-IDF of the all the book plot is generated and ultimately the cosine similarity is found based on the query or the searched book plot. A total number of 14 text documents was used, collected into a single corpora and preprocessing is done. Next, indexing was done, which helped to indicate the existence of a term in a particular document. All the terms occurring in all the documents were listed, with the count of the term occurring in each document is calculated. There are a total number of 1457 terms occurring in the overall document after preprocessing. The procedure for finding the TF-IDF for both the overall documents and the query document follows the same. The results are shown based on the two scenarios as follows:

I. Search query

In the first scenario, a total number of 14 text documents was used. The query term for the test data was “Children of Earth”. The 14 files were collected into a single corpora, before text mining was done and the raw text had to be pre-processed first. Next, indexing was done, which helped to indicate the existence of a term in a particular document. Out of the 14 files, five files have been displayed in the table below. The term document matrix containing 14 documents had 1457 terms, after cleaning of data.

Table II. Term document matrix of all the documents.

Children of Zion	***Children of Tomorrow***	***Children of this Earth***	***Children of Magic Moon***	***Children of God***	***Term/ Documents***
0	0		0	0	Young
5	1	1	2	3	Children
2	0	1	0	0	Also
1	1	1	0	1	Becom
1	0	1	1	0	World
0	2	1	0	1	John
0	1	1	1	4	Find
0	1	1	0	2	Earth

The term “children” and “earth" was found in the three documents, “Children of God”, “Children of this Earth” and “Children of tomorrow”. The next step is the normalization of the terms, which helps to ensure that a term frequency and the size of the document does not affect the term weightage.

Table III. Normalization of the term document matrix of all the documents.

Children of Zion	***Children of Tomorrow***	***Children of this Earth***	***Children of Magic Moon***	***Children of God***	***Term/ Document s***
0	0	0.0208333	0	0	Young
0.0581395	0.0196078	0.0104166	0.0238095	0.0049668	Children
0.0232558	0	0.0104166	0	0	Also
0.0116279	0.019607	0.0104166	0	0.0016556	Becom
0.0116279	0	0.0104166	0.0119047	0	World
0	0.0392156	0.0104166	0	0.0016556	John
0	0.0196078	0.0104166	0.0119047	0.0066225	Find
0	0.0196078	0.0104166	0	0.0033112	Earth

The next step that follows is the finding of the inverse document frequency. It helps in determining how truly unique a term is. The term “children” occurs frequently in the 15 documents, hence is not considered of high importance. The term “earth” has a higher value and is seen to occur less frequently in the over documents. IDF helps in determining the terms that relate to a document.

Table IV. IDF of all the terms.

IDF	***Term/ Documents***
2.540445	Young
1	Children
1.847298	Also
1.693147	Becom
1.559616	World
2.540445	John
1.336472	Find
2.540445	Earth

The last step is calculating the TF-IDF. It is a simple multiplication of each normalized term and the respective idf value. It helps in identifying if a term is important, by measuring the frequency of the term in all the documents and hence evaluating the weight of it in a particular document.

Table V. TF-IDF of all the terms found in all the documents.

Children of Zion	***Children of Tomorrow***	***Children of this Earth***	***Children of Magic Moon***	***Children of God***	***Term/ Documents***
0	0	0.052925	0	0	Young
0.058139	0.019607	0.010416	0.023809	0.004966	Children
0.05814	0.019608	0.010417	0.02381	0.004967	Also
0.04296	0	0.019243	0	0	Becom
0.019688	0.033199	0.017637	0	0.002803	World
0.018135	0	0.016246	0.018567	0	John
0	0.099625	0.026463	0	0.004206	Find
0	0.026205	0.013922	0.01591	0.008851	Earth

In order to calculate the cosine similarity between the documents and the query, the query is passed and stored as a document. The query document will undergo the same steps as that of all the documents. Hence the following results will be obtained:

The query “Children of earth” will contain only the two terms “children” and “earth” after preprocessing, the preposition “of” is not considered.

Table VI . Normalization of the term document matrix of the searched query

Term Frequency	***Query Terms***
0.5	Children
0.5	Earth

Once the normalization of the term is calculated, the intersection of the terms from the union of the query term and the documents are extracted, in order to get the inverse document frequency (IDF) of the terms. The TF-IDF is calculated once the IDF of the query term are found.

Table VII. TF-IDF of the terms of the searched query

TF-IDF	***Query Terms***
0.5	Children
1.270223	Earth

The final step after the TF-IDF of both the documents and the query is calculated, is the calculation of the cosine similarity. It helps in measuring how two documents are closely related. The results indicate that two documents contain the term “children” and “earth”. Hence these two books can be recommended when a search query like “Children of earth” is made. When a query was made by a user, the document with cosine similarity closest to 1 is being recommended (Table VIII)

Table VII. The Cosine similarity measures between the documents and the searched query

Cosine similarity scores	***Documents***
0.987486981	Children of God.txt
0.366276752	Children of Magic Moon.txt
1	Children of This Earth.txt
1	Children of Tomorrow.txt
0.366276752	Children of Zion

J. Book Searched

Similarly, instead of a query, the plot of a searched book is passed as a query. In this case, the TF-IDF of all the documents is calculated as in the previous scenario. Each document is fed as a search query one by one and the cosine similarity is found. This will result in a square matrix of the cosine similarity. The cosine similarity of the first five occurring documents is showed below.

Table IX. The cosine similarity between the plot summary of all the books

Children of Scarabaeus	***Children of Paranoia***	***Children of Orpheus***	***Children of Magic Moon***	***Children of God***	***Term/ Documents***
0.099011	0.1332137	0.1665818	0.0880752	0.9997589	***Children of God***
0.146655	0.1491062	0.2994236	0.9930862	0.2711478	***Children of Magic Moon***
0.068841	0.1063945	0.9988898	0.1226489	0.2891741	***Children of Orpheus***
0.094037	1	0.2312964	0.1697167	0.3371702	***Children of Paranoia***
0.987627	0.151322	0.2555516	0.2302606	0.3428928	***Children of Scarabaeus***

The diagonal values are either 1 or approximately equivalent to 1. It indicates that they are highly similar or of the same. The row indicates the query document. The lower triangle and the upper triangle does not result in same value. The reason this happens can be explained using an example. Suppose there are 5 terms in the query book "Children of God" that is common within the whole document. When the cosine similarity between the book and say the other book "Children of Magic Moon" is found, it may be different from when "Children of Magic Moon" is used as query against the document "Children of God". Cosine similarity help in finding the degree of closeness between two vectors. The vectors in this scenario are the two documents that are being compared, which is the book plot of the two books.

To find the highest value of each row, we first make the diagonal values equivalent to 0. The diagonal elements indicate the query book itself. Since the row is the query document, the maximum cosine similarity value which is the highest related query to a set of documents is extracted along with the name of the query book and the document book name. The result is represented by the following figure

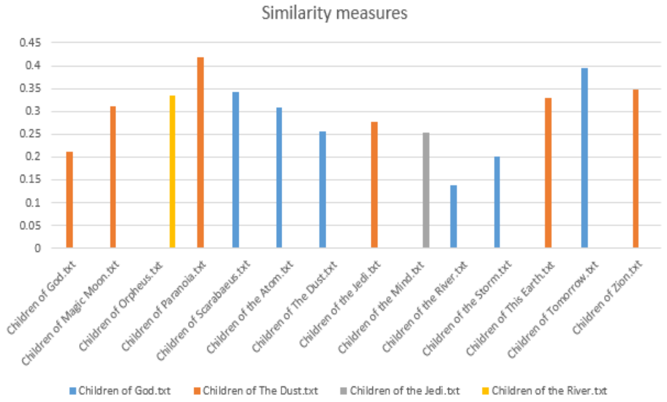

Figure 4. Cosine Similarity of all the books is seen to be closely related to four out of the fourteen books.

It can be seen that when each document is given as a query four documents were found to be closely related. For instance when "Children of God" is used as a search query then "Children of the Dust" is recommended. Using the 14 book plots, it is found that the books are more closely related to four books which are "Children of God", "Children of the Dust", "Children of the Jedi" and "Children of the River". Hence when a book is used as a search query, the book with the highest cosine similarity found is recommended to the user.

VI. REFERENCES

[1] Ashwin Ravi Ittoo, Yiyang Zhang, Jianxin Jiao, "A text mining-based recommendation system for Customer Decision Making in Online Product Customization" Management of Innovation and Technology, 2006 IEEE International Conference on

[2] P.N.Vijaya Kumar, Dr. V. Raghunatha Reddy "A survey on recommender system (RSS) and Its Application" International Journal of Innovative Research in Computer and Communication Engineering *(An ISO 3297: 2007 Certified Organization)* Vol. 2, Issue 8, August 2014

[3] Anshika Singh ,Dr. Udayan Ghosh "Text Mining: A Burgeoning technology for knowledge extraction" International Journal of Scientific Research Engineering & Technology (IJSRET), Vol1 Issue12

[4] Mukta kohar,Chhavi Rana, "Survey Paper On Recommendation System" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2) , 2012,3460-3462 3460

[5] Greg Linden, Brent Smith, and Jeremy York "Amazon Recommendation: Item to Item Collaborative filtering"

[6] Vishal Gupta, Gurpreet S. Lehal, "A survey of Text Mining Techniques and Application" Journal of Emerging Technologies in Web Intelligence, vol. 1, no. 1, august 2009

[7] "Introduction: Vector Space Model" [online] Available: http://cogsys.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html

[8] M. Jalali, H. Gholizadeh, and S. A. H. Golpayegani, "An improved hybrid recommender system based on collaborative filtering, content based, and demographic filtering," *International Journal of Academic Research*, vol. 6, no. 6, pp. 22–28, 2014

[9] "Tf-idf: A single-page Tutorial - information retrieval and text mining,". [Online]. Available: http://www.tfidf.com/.

[10] Gerard Salton, Chris Buckley "Term Weighting Approaches in Automatic Text Retreival"