International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 02 | Feb 2020 www.irjet.net p-ISSN: 2395-0072

DOCUMENT COMPARISON BASED ON TF-IDF METRIC

Dr.M. Umadevi[1]

[1]*Faculty, Department of CSE, Universal College of Engineering a**nd Technology**, Dokiparru, Guntur*

Abstract - Rapid progress in digital data led to huge volume of data. More than 80 percent of data is composed of unstructured or semi-structured data. The discovery of appropriate patterns and trends to analysis the text documents from massive volume of data is a big issue. Extraction of valuable information from a corpus of different document is a tedious and tiresome task. Text mining is a process of extracting interesting and nontrivial patterns and significant patterns from huge amount of text documents. The selection of appropriate technique for mining text reduces the time and effort to find the relevant patterns for analysis and decision making. In the proposed work documents are represented vector of terms. Each term is given weight based on it frequency and its significance in over all document. The metric called term frequency- and inverse document frequency (TF-IDF) is used, which is significant within the document and not relevant in other document. TF-IDF as feature vector is used to comparison of document. Cosine similarity metric is used to compare the document similarity.

Key Words: Term frequency, Inverse Document frequency, cosine similarity.

1. INTRODUCTION

The size of data is increasing at exponential rates day by day. Almost all type of institutions, organizations, and business industries are storing their data electronically. A huge amount of text is flowing over the internet in the form of digital libraries, repositories, and other textual information such as blogs, social media network and emails. It is challenging task to determine appropriate patterns and trends to extract valuable knowledge from this large volume of data. Traditional data mining tools are incapable to handle textual data since it requires time and effort to extract information. Text mining is a process to extract interesting and significant patterns to explore knowledge from textual data sources. Text mining is a multi-disciplinary field based on information retrieval, data mining, machine learning, statistics, and computational linguistics. Text mining is the concept it has interaction with other techniques like summarization, classification, clustering etc., can be applied to extract knowledge.

Text mining deals with natural language text which is stored in semi-structured and unstructured format. Collecting unstructured data from different sources available in different file formats such as plain text, web pages, pdf files etc., Extraction of valuable information from a corpus of different document is a tedious and tiresome task. The selection of appropriate technique for mining text reduces the time and effort to find the relevant patterns for analysis and decision making. The objective of this paper is to comparison of document using tf-idf as feature.

Different text mining techniques are available that are applied for analyzing the text patterns and their mining process. Document classification (text classification, document standardization), information retrieval(keyword search / querying and indexing), document clustering (phrase clustering), natural language processing (spelling correction, lemmatization, grammatical parsing, and word sense disambiguation), information extraction (relationship extraction / link analysis), and web mining.

2. LITERATURE SURVEY

Dan MUNTEANU et al(2007) presented vector space model for document representation with Boolean and term weighted models , ranking methods based on cosine factor. This paper proposed methodology for ranking matched documents according to their relevance.

Mazid et al (2009) proposed that the comparison gives the detailed study of association ruled based mining model. In which the rule based mining (which may be performed through either supervised learning or unsupervised learning techniques) are compared with recent research proposals using predefined test sets. In terms of accuracy and computational complexity.

Hinrich et al (2011) proposed that if two documents describe similar topics, employing nearly the same keywords, these texts are similar and their similarity measure should be high. Usually dot product represents similarity of the documents. To normalize the dot product, it can be divided it by the Euclidean distances of the two documents. This ratio defines the cosine angle between the vectors, with values between the „0" and „1" this is called cosine similarity

3. DOCUMENT BASED TEXT MINING

Document based text mining is used to analyze the concept at the document level. The concept based term frequency tf, the number of the occurrences of a concept c in the original document is calculated. The tf is a local measure on the document level. Use feature-vector to represent documents, that is, take one document as a set of Term Sequences, including term t and term weight w.

3.1. Representation of Document in vector space model

In text mining each document is represented as a vector. The elements in the vector reflect the frequency of terms in documents, and each word is a dimension and documents are vectors. Each word in a document has weights. These weights can be of two types: local and global weights. If local weights are used, then term weights are normally expressed as Term Frequencies (TF). If global weights are used, Inverse Document Frequency (IDF), IDF values, gives the weight of a term. It is possible to do better term weighing by multiplying „TF" values with „IDF values, by considering local and global information. Therefore total weight of a „term=TF*IDF". This is commonly referred to as, "TF*IDF" weighting.

Then the document will be made up of the pairs of <t, w>. t1, t2, t3…. tn represents the features that is expressed the document content. And also treat them as an N-dimension coordinate. W1, W2, W3…, Wn represent the value relevant to coordinate. So every document(d) is mapped to the target space as a feature-vector V(d)=(t1,w1,t2,w2,t3,w3…. tn, wn). The most important of data pre-processing is to deal with the data resource and also build up the feature vectors. Also use the weight as the criterion of feature selection. The values of the vector elements wi for a document d are calculated as a combination of statistics TF (t, d) and DF(t). The term frequency TF (t, d) is the number of times word t occurs in document d. The document frequency DF (t) refers to the number of document in which the word t occurs at least once. The inverse document frequency IDF (t) can be calculated from the document frequency. Log (||)/ DFtDIDFt|D| |D| is the whole number of documents. The document frequency of inverse of a word is low if it occurs in many documents and is highest if the word occurs in only one. The values wi of features ti for document d is then calculated as the product (,) ( )(2) iii W=TF td. IDF t wi is called the weight of the word ti in document d. The heuristically says the word weighting a word ti is an important indexing term for document d if it occurs in frequently in it. A word which occurs in many documents is rated less important indexing terms due to their low inverse document frequency and also used to find t. The inverse document frequency IDF (t) can be calculated from the document frequency.

Tf(t)=(Number of times term t appears in a document)/(Total number of terms in the document)

IDF(t)= log_e(Total number of documents/Number of documents with term in it)

W(t)=tf*idf

3.2 .Creating Corpus

The first prerequisites that install the R on machine from their respective websites. Start R and install the packages tm, Pdftools, ggplot2 and word cloud etc. First, let us start by loading the data, which can be any text or collection of texts (commonly referred to as a corpus) from which we want to extract useful information. First load the tm package then use the function Corpus () to create a collection of documents. This loads our text documents residing in a particular folder into a corpus object. Corpus is the set of n documents. Each of this document defined as a set of m terms (radicals, words or a set of words)

3.3. Preprocessing

Preprocessing the text means removing punctuations, numbers, strip white spaces, convert the text into lower case, remove stop words, stemming etc…These characters do not convey much information and are hard to process. For example English stop words like "the", "is", "myself", "about" ,"I" etc . Do not tell you much information about the content of the text. This step is performed to clean the data to increase the efficiency and robustness of our results by removing words that don't necessary.

- Convert to lower case
- Remove punctuation
- Remove numbers
- Stemming
- Stop words

3.4. Text feature extraction TF-IDF

The **term-frequency is used** to represent textual information in the vector space. However, the main problem with the term-frequency approach is that it scales up frequent terms and scales down rare terms which are empirically more informative than the high frequency terms. The basic intuition is that a term that occurs frequently in many documents is not a good discriminator, and really makes sense (at least in many experimental tests); the important question here is: why would you, in a classification problem for instance, emphasize a term which is almost present in the entire corpus of documents.

The tf-idf weight comes to solve this problem. What tf-idf gives is how important is a word to a document in a collection, and that‟s why tf-idf incorporates local and global parameters, because it takes in consideration not only the isolated term but also the term within the document collection. What tf-idf then does to solve that problem, is to scale down the frequent terms while scaling up the rare terms; a term that occurs 10 times more than another isn‟t 10 times more important than it, that‟s why tf-idf uses the logarithmic scale to do that.

In information retrieval or text mining, the term frequency – inverse document frequency (also called **tf-idf**), is a well know method to evaluate how important is a word in a document. tf-idf are is a very interesting way to convert the textual representation of information into a Vector Space Model (VSM), which is actually the term count of the term in the document . The use of this simple term frequency could lead us to problems like keyword spamming, which is when we have a repeated term in a document with the purpose of improving its ranking on an IR (Information Retrieval) system or even create a bias towards long documents, making them look more important than they are just because of the high frequency of the term in the document.

3.5. The Cosine Similarity

Cosine Similarity will generate a metric that says how related are two documents by looking at the angle instead of magnitude, like in the examples below:

Let‟s begin with the definition of the dot product for two vectors: a=(a1,a2,...) and b=(b1,b2,...), where a_n and b_n are the components of the vector (features of the document, or TF-IDF values for each word of the document) and the n is the dimension of the vectors. 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 we are not taking into the consideration only the magnitude of each word count (tf-idf) of each document, but the angle between the documents. What we have to do to build the cosine similarity equation is to solve the equation of the dot product for the :

And that is it, 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, like in the examples below:a a.b=|a|.|b| cosθ

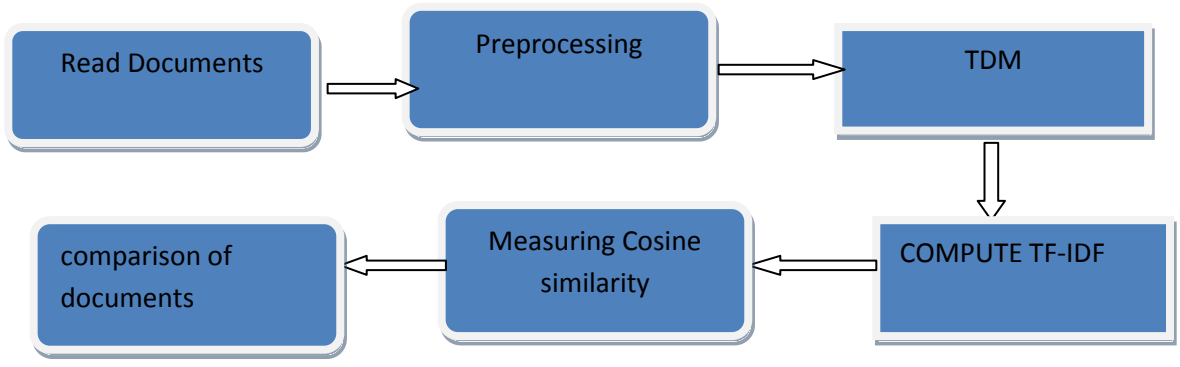

Fig 1: Process of Document Comparison based on TF_IDF

3.5 Document Comparison based on Cosine similarity

In the proposed work 5 different documents are read using pdf reader and preprocessed the document to form a corpus. Term document matrix prepared from corpus . The term frequency TF and IDF is calculated based on TF_IDF cosine similarity calculated for the five document. These five documents are taken from different domains like Agriculture, Economy, Polity, Entertainment and Science and Technology. Here cosine similarity is used to compare the document. The result are tabulated as follows in Table 1. If the two documents are similar cosine metric is 1 and it is represented in the form degree as zero i.e the

two documents are represented by same vector. If the two documents are entirely different the cosine metric is 0 and the two vectors are separated by some degree. In this paper document 2 is different from other four documents. Cosine similarity metric of Document 4 and document 5 is more compared to other documents. Hence document 2 is entirely different from ther document. The remaining 4 document are showing some similarity with varying metric. Hence TF_IDF and other similarity metrics used for further clustering of document for making search process simple.

Tabel 1: Cosine Similarity Metric and Degree for Comparison of Documents

	Document1 Agriculture	Document2 Economy	Document3 Entertainment	Document4 Polity	Document5 Science&Techn
Document1 Agriculture	cosine: 1	0	0.2212955	0.2390312	0.2896331
	Degree: 0.0000008537736	90	77.21487	76.17063	73.16401
Document2 Economy	0	1	0	0	0
	90	0	90	90	90
Document3 Entertainment	0.2212955	0	1	0.290161	0.311088
	77.21487	90	Nan	73.1324	71.87519
Document4 Polity	0.2390312	0	0.290161	1	0.4101863
	76.17063	90	73.1324	NaN	65.78346
Document5 Science&Techn	0.2896331	0	0.311088	0.4101863	1
	73.16401	90	71.87519	65.78346	Nan

4. FUTURE EXPANSION

In the proposed work the documents represented as features and compared using cosine similarity measure. The further expansion of the work includes comparison of documents based on other similarity measures like Jaccard similarity, Euclidean distance etc. As TF-IDF is does not capture position in text semantics co-occurrences in different documents etc. Hence TF-IDF is only useful as a lexical level feature.

5. BIBLIOGRAPHY:

[1] Feinerer , "An Introduction to Text Mining in R " VOLUME 8, ISSN 1609-3631
[2] October 2008.
[3] Dan Munteanu, "Vector Space Model for Document Representation in Information Retrieval", University of Galati Fascicle ,2007
[4] Hinrich Schutze, " Introduction to Information Retrieval", Institute for Natural Language Processing, University of Stuttgart, 2011
[5] Mazid, Mohammad, " A comparison between rule based and Association rule mining algorithm", Third International Conference on Network and System Security, NSS 2009, Gold Coast, Queensland, Australia, October 19-21, 2009.
[6] www.tfidf.com
[7] Feinerer. tm: Text Mining Package, 2008. URL http: //CRAN.R-project.org/package=tm. R package version 0.3-1.
[8] Anna Huang "Similarity Measures for Text Document Clustering", NZCSRSC 2008, April 2008, Christchurch, New Zealand
[9] Stephen Robertson, (2004) "Understanding inverse document frequency: on theoretical arguments for IDF", Journal of Documentation, Vol. 60 Issue: 5, pp.503-520, https://doi.org/10.1108/00220410410560582
[10] D.L. Lee ; Huei Chuang ; K. Seamons, "Document ranking and vector spacemodel", IEEE Software Volume: 14 , Issue: 2 , Mar/Apr 1997
[11] Yung-Shen Lin, Jung-Yi Jiang, and Shie-Jue Lee, "A Similarity Measure for Text Classification and Clustering", IEEE Transactions on Knowledge and Data Engineering, 2013.
[12] Shady Shehata,Fakhri Karray, Mohamed S. Kamel, "An Efficient Concept-Based Mining Model for Enhancing Text Clustering", IEEE Transactions on Knowledge and Data Engineering, , 2010.
[13] K. Aas and L. Eikvil, "Text Categorisation: A Survey," Technical Report 941, Norwegian Computing Center, June 1999.
[14] G. Salton and M.J. McGill, Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
[15] Jake Teo "text mining social network documentation", Technical report

[16] Riya, Namita. G, "Text mining of criminal document", International Journal of Advances in Electronics and Computer Science, ISSN: 2393-2835 , volume-3, issue 9, sep 2016
[17] Chauhan SR, Desai A. A Review on Knowledge Discovery using Text Classification Techniques in Text Mining". 2015; 111:6.
[18] Mining Text Data, Charu C. Aggarwal, ChengXiang Zhai, SPRINGER,2012