The DF-ICF Algorithm- Modified TF-IDF

Puneet Goswami, PhD
Associate Professor
Galaxy Global Group of Institutions Dinarpur
Ambala, Haryana, India

Vidya Kamath
P.G Scholar
Galaxy Global Imperial Technical Campus
Dinarpur, Ambala, Haryana, India

ABSTRACT
The tf-idf is an algorithm which is generally used where massive data processing is done. Tf-idf is the weight given to a particular term within a document and it is proportional to the importance of the term. This paper aims to use the idea behind the tf-idf algorithm to design the df-icf algorithm which finds the importance of a particular document within the given corpus.

General Terms
DF-ICF algorithm, TF-IDF algorithm

Keywords
DF-ICF, TF-IDF, Document frequency, Term frequency, Corpus, Page Ranking

1. INTRODUCTION
Tf-idf (Term Frequency-Inverse Document Frequency) is an algorithm which is used often in search engines, text similarity computations; web data mining etc. i.e. it is used in applications which require massive data processing. Therefore, calculating the tf-idf algorithm quickly and efficiently is very important. Tf-idf is a weight which is a statistical measure used to evaluate how important a word is to the document in the collection. The documents in the collection are collectively called as corpus. The algorithm is based on the idea that the more the word appears in a particular document the more is its importance; but the importance is offset by the frequency of the word in the corpus [2, 3].

Imagine a situation where a person wants to do online shopping. Hence the item he wishes to purchase will be the term whose importance has to be found. Suppose he wants to buy a laptop. The term of importance will be the name of the laptop (viz hp pavilion, dell inspiron and so on) and the document will be the page about the particular item in any particular site. Now it is obvious that if you are visiting a page on dell then the occurrence of the term ('dell') within the document (page) will be very high. So if we are using the tf-idf to find importance of the word then we will get a positive result. Note that the importance of the word is completely dependent on the document. What if the document is not so important? If you could have found the importance of the document first, you can arrive at a better conclusion.

Df-icf (Document Frequency Inverse Corpus Frequency) has been proposed in this paper. This paper is an effort to apply the idea behind the tf-idf algorithm to documents instead of terms. The main advantage of this is that knowing the importance of a document is equally important as knowing the importance of a term. A term which has a high level of importance in a low weighted document obviously loses its importance. So the importance of a word not just depends on the number of times it appears, but also on the document in which it appears.

This paper aims to give an insight of the tf-idf algorithm and its importance. Then an effort has been made to extend the same idea to try find out importance of a document using the df-icf. A way to use this df-icf algorithm efficiently has also been proposed.

2. TF-IDF ALGORITHM
The tf-idf depends on two factors. The term frequency and the inverse document frequency.

A. The term count in the document is the number of times the word appears in the document. For the term t_i in a particular document d_j, its term frequency is defined as follows:

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

In the formula, $n_{i,j}$ is the number of occurrences of the considered term t_i in the document d_j. The denominator is the sum of the number of occurrences of all the terms in document d_j.

B. The inverse document frequency is a measure of the general importance of the term. The formula is defined as follows:

$$idf_i = log \frac{|D|}{|\{j: t_i \epsilon d_j\}|}$$

In the formula, $|D|$ is the total number of documents in the corpus; $|\{j: t_i \epsilon d_j\}|$ is the number of documents where the term t_i appears; i.e. $n_{i,j} \neq 0$.

C. The tf-idf weight of the term t_i in a particular document d_j is the product of tf and idf. The formula is as follows [2]:

$$(tf - idf)_{i,j} = tf_{i,j} \times idf_i$$

3. PROPOSAL OF DF-ICF ALGORITHM
The df-icf algorithm is similar to the tf-idf algorithm. But here the document is considered as the central focus. Instead of finding the importance of a term within a document, the importance of the document within the corpus is found out using the same idea. The corpus here for example can be a particular site like yahoo, facebook, twitter etc. The count here is not number of occurrences but the number of times the particular document is viewed. The df-icf is based on two factors; the document frequency and inverse corpus frequency. In short the df-icf is as follows:

A. The document count is the number of views of a particular document in the corpus. The corpus can be the internet as a

whole considering the upper bound or it can be a part of the internet like the data in a particular site or center. The document frequency within a given corpus is as follows:

$$df_{x,y} = \frac{n_{x,y}}{\sum_k n_{k,y}}$$

In the formula, $n_{x,y}$ is the number of times the document d_x is viewed from corpus c_y. The denominator is the sum of the number of views of all the documents in corpus c_y.

B. The inverse corpus frequency is a measure of the general importance of the document. The formula is defined as follows:

$$icf_x = log \frac{|C|}{|\{y: d_x \epsilon c_y\}|}$$

In the formula, $|D|$ is the total number of corpus (which is 1 if the entire internet is taken to be the corpus.) in the considered collection; for example, if you are interested in online shopping, you can consider a collection of sites like Amazon, eBay etc. in which case your document might be a page on the item you want to purchase, the corpus are the sites which you have taken in your collection. $|\{y: d_x \epsilon c_y\}|$ is the number of corpus where the document d_x appears; i.e. $n_{x,y} \neq 0$.

C. The df-icf weight of the document d_x in a particular corpus c_y is the product of df and icf. The formula is as follows:

$$(df - icf)_{x,y} = df_{x,y} \times icf_x$$

4. USING DF-ICF

The df-icf is the weight of the document. Now taking our previous example, if you want to find importance of the term t_i- 'dell' in the page about dell d_j from a site c_k(e.g. Amazon) then you have to first find out the df-icf of the page about dell by considering the page has the document, the site Amazon as the corpus. The collection can be a set of sites the user wants to look in. After finding the df-icf, the tf-idf should also be found using the tf-idf algorithm. To find the final weight (importance) of the term you can use the following formula:

Weight of $t_i = (tf - idf)_{i,j} \times (df - icf)_{j,k}$

Now let us take an example in order to find out whether the df-icf calculations are worth enough. Looking straight it is obvious to think that it is time consuming to find df-icf, tf-idf and then find the weight. But if you observe closer then you can see that calculating the df-icf reduces the number of times the tf-idf has to be calculated. This is because of the fact that the finding out the importance of the term in a less worthy document is useless. So you only have to calculate df-icf and filter the documents with some boundary value. Further computations are done only with the documents which have passed this filter. This assures that the result we arrive at using the tf-idf is worth enough. The complete procedure is as shown in Fig 1.

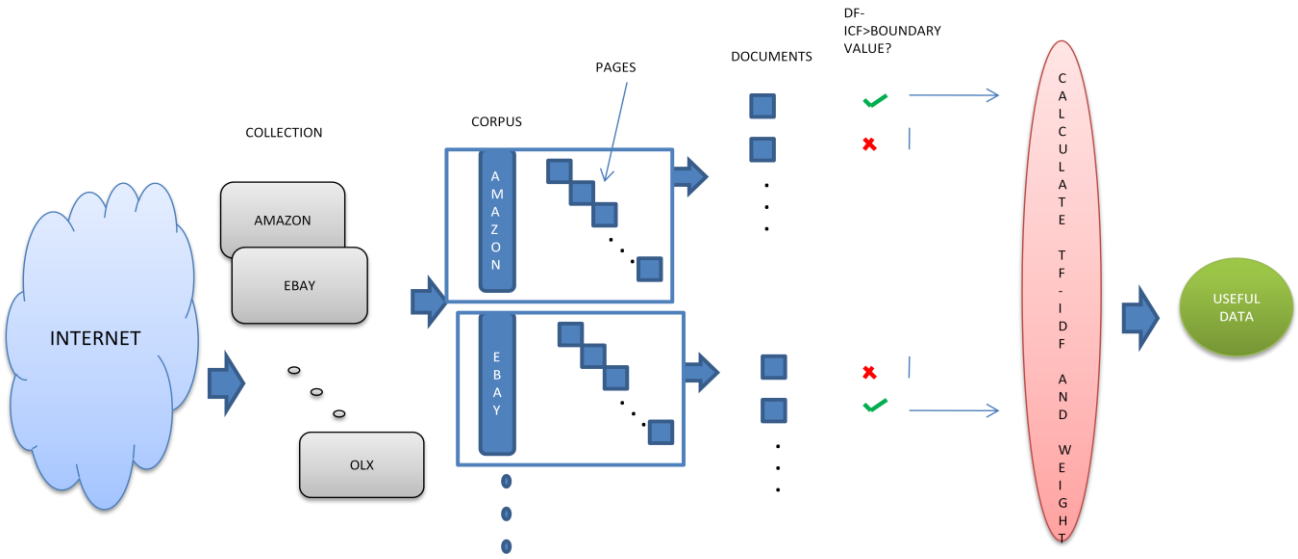

Fig 1: An example of finding useful data using df-icf and tf-idf

5. CONCLUSION AND FUTURE THOUGHTS

In this paper an effort has been made to propose an algorithm which can be used to find the importance of the document. The assumption here is that the importance of a document increases with the number of times it is viewed similar to the way the importance of a term increases with its number of occurrences. The tf-idf algorithm has been used and idea has been extended to design the df-icf algorithm.

Calculating tf-idf is very important since it is used in situations where massive data processing is done. So the tf-idf has to be done quicker and faster. If you find the df-icf first, you can filter out the documents which are of less importance and avoid the calculations of tf-idf within them. This not only saves your time but also makes your result more reliable.

If you can efficiently apply the df-icf algorithm to large data sets then you can save your effort, time and resources. Practical implementation of this algorithm can be a thought of in the future considering smaller data sets first and then moving towards the upper bound. Although implementing the algorithm to the entire internet which will be the upper bound of the algorithm is not so practical, you can still make use of the algorithm in many large and massive applications.

6. REFERENCES

[1] SALTON G, BUCKLEY C. Term-weighting approaches in automatic text retrieval [J]. Information Processing and Management, 1988, PP513 - 523.

[2] SALTON G, CLEMENT T Y. On the construction of effective vocabularies for information retrieval[C].Proceedings of the 1973

[3] Bin Li, Yuan Guoyong-" Improvement of tf-idf for Hadoop Framework" The 2nd International Conference on Computer Application and System Modeling (2012)

[4] LiThomas H Davenport, Jill Dyche- "Big Data in Big Companies" International Institute for Analytics, may 2013.

[5] Moty Fania, John David Miller- White paper- "Mining Big Data in the Enterprise for Better Business Intelligence", Intel july 2012

[6] Puneet Goswami, Vidya Kamath-"Big Data- Driving force for innovation and Value Receation", IJARCSSE volume 4, issue 3-march 2014.

[7] Dr. A.K Sharma, Puneet Goswami "Information Retrieval Tools: A Review". Published at the proceedings of national conference on Research and Practices in current areas of IT at SLIET Longowal Punjab on March 26-27 2004. Page No. : 71-74

[8] Puneet Goswami paper titled "Security in Cloud Reference Models and Secure Identity Management Mechanism",1st International IBM Cloud Academy Conference ICA CON 2012,April 19-20, 2012 at the IBM Employee Activity and Fitness Center Building 400, Cornwallis Drive Research Triangle Park, North Carolina Organized by : The IBM Cloud Academy and IBM Centers for Advanced Studies (RTP, Chicago, Heritage Corridor, Tucson, Florida).

[9] Puneet Goswami, Varun Kumar, Anuj Sharma "A Framework for Intelligent Meta Search Engine". Published in Voyager - The Journal of Computer Science and Information Technology ISSN 0973-4872, Vol. 4, No.1 (2006) Institute of Technology & Management".

[10] "Making data Analytics Work- three Key Challenges. McKinsey and Company. IDC Digital Universe Study, sponsored by EMC, June 2011

[11] Stamatis Karnouskos-" Big data analytics for Smart Grid Cities" . Eurescom mess@ge 1- 2013.

[12] Anastasius Gavras-" Big data – Overview on a much-hyped concept". Eurescom mess@ge 1- 2013.