I.J. Education and Management Engineering, **2019, 4, 11-19**
Published Online July 2019 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2019.04.02

Available online at ***http://www.mecs-press.net/ijem***

Key Term Extraction using a Sentence based Weighted TF-IDF Algorithm

T. Vetriselvi[a], N. P. Gopalan[b], G. Kumaresan[b,*]

[a,] *Department of Computer Science and Engineering, K. Ramakrishnan College of Technology, Tiruchirappalli, India*
[b] *Department of Computer Applications, National Institute of Technology, Tiruchirappalli, India*

Received: 06 November 2018; Accepted: 15 February 2019; Published: 08 July 2019

Abstract

Keyword ranking with similarity identification is an approach to find the significant Keywords in a corpus using a Variant Term Frequency Inverse Document Frequency (VTF-IDF) algorithm. Some of these may have same similarity and they get reduced to a single term when WordNet is used. The proposed approach that does not require any test or training set, assigns sentence based Weightage to the keywords(terms) and it is found to be effective. Its suitability is analyzed with several data sets using precision and recall as metrics.

Index Terms: Similarity Matrix, Term Count, WordNet.

1. Introduction

Emerging techniques in Natural Language Processing (NLP) are employed in areas like Information Retrieval, Semantic Analysis, Sentiment Analysis and Review processing systems. The underlying crux of all the above is extracting key phrases from a given document or corpus often leading to further analytics like clustering, classification etc. The keywords are set of important words which give good description about the contents of a given document. All text mining applications at the outset involve preliminary processing like stemming and stop word removal over the extracted keywords.

* Corresponding author.
E-mail address: kumareshtce@gmail.com.

The conventional TF-IDF algorithm is used for extracting term frequency in a document. Frequency of the terms in documents is counted as TF. Likewise, the occurrence of the same term in different documents is denoted by IDF [1] multiplying them.

A Variant of TF-IDF with enhanced efficiency is also used in this context. Sentences of a document play a major role while its contents are delivered. In the present work, assigning weightage to the terms based on their locations in a sentence is found to improve the efficiency of the procedure. The number of occurrences of terms may not necessarily imply that they are keywords for a document. In WordNet, a term may have several synonyms and it is possible that they may be related in some way or other. In such occasions, similarity among these words needs to be considered while extracting terms to calculate similarity measures such as Cosine, Inverse Euclidean Distance and Pearson Correlation Coefficient similarities [2]

The key term extraction is a basic requirement for all text processing techniques and there exist several methods for finding keywords and similarity. These are presented in the literature survey. The proposed method is discussed in §2 and the experimental evaluation using LT4EL.eu is presented in §3. The conclusion and the possible extension of this work are discussed in §4.

1.1. Keyword Extraction

The Keyword Extraction is done by various ways using linguistic, statistical and graphical approaches. Machine learning methods such as supervised learning and unsupervised learning are also explored under NLP processes. Further, few linguistic approaches like verb and noun phrase extraction are also used in this context but usage of statistical approaches are simple in nature. The supervised machine learning techniques involve training and testing, requiring a dataset for it. In case of unsupervised machine learning the number of clusters formed has to be mentioned at the earliest.

The graphical representations of text documents show the relationship among the terms. To name a few, semantic rank, and HITS are some of the popular methods employed in text processing. One of the most widely used statistical approaches is Traditional TF_IDF, for enumerating the relevant terms of the document.

Sungjicj Lee [16] used a two-step process for extracting keywords. In the first step, they are extracted by TF-IDF method and ranked using candidate comparison in the next step. It is evaluated with a statistical measure to bring out the importance of a term in a document [1].The following identities are used calculating TF- IDF.

$$tf_{i,j} = \frac{n_{i,j}}{\sum k \times n_{i,j}} \tag{1}$$

$$idf_i = \log \frac{|D|}{|(d_j t_j \in d_j)|} \tag{2}$$

$$tfidf_{i,j} = tf_{i,j} * idf_i \tag{3}$$

The subscripts i, j mean i^{th} term in j^{th} document.

It is further classifieds into two types - a Feature based TF-IDF and a variant TF-IDF. In the former, the location of the term and its relevant information and in the later Basic Term Frequency (BTF), two Normalized Term Frequencies NTF1, and NTF2 are used for extraction. In the above normalization, the repeated and synonymous terms are considered to be identical and ranked using domain filtering.

1.2. Semantic Graphs

George et al [4] constructed a weighted semantic graph using words as nodes and the relation between them as edges. The length of the path between words in the semantic net is weight of the edge connecting them in the semantic graph. It is called weight assignment in distributional measure. Weights are also calculated with omiotis [3] in thesaurus based method.

1.3. Key Phrase Ranking

A keyword ranking is to find the highly relevant terms in one or more document(s) and syntactic, semantic, and statistical approaches may be employed to accomplish it. In statistical method the weight based arrangement of keywords are helpful to identify the rank of the keywords.

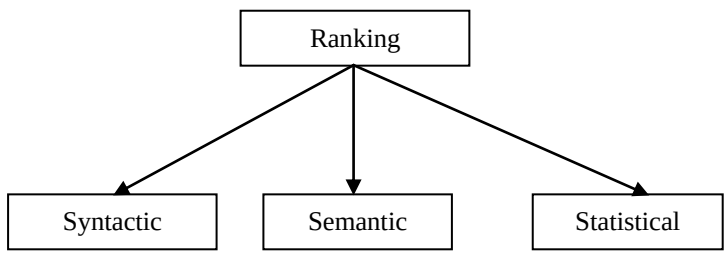

Fig. 1. Various Ranking Methods

The high ranking term is a word which is connected with a maximal number of words in a semantic graph. In syntactic method the noun and verb phrases are counted and ranked which is shown in Fig. 1.

2. Related Work

The description of a document can be inferred from a set of highly significant keywords in it and their extraction is crucial task. Such keywords can be extracted by identifying frequent items in the document in which duplication is removed by a suitable stemming algorithm. Then, VTF-IDF method is used to measure the frequency of words. Semantic similarity among words can be obtained from WordNet measures and subsequently a Similarity matrix is formed for further analysis [5].

2.1 Initial Pre-Processing

Tokenization, stemming and stop word removal are set of activities in pre-processing. Tokenization spilt the sentences in to token i.e. small lexical units. Stemming is the process of identifying the root word. The supporting words {is, and, the, of, for,} have to be removed, hence those are removed by the process of stop word removal.

2.2 TF-IDF variants

After pre-processing, each term was assigned by certain weight using several variants in TF-IDF.
Term Count (tc):
The tc represents frequency of the term that is how often a given term occurs in a document collection.

$$term\,count_i = \sum_{j=1}^{|D|} n_{i,j}$$

Where n is the total number of terms in the document.

2.3 Normalized term frequency (ntf1 and ntf2)

The first bias of Term Count (tc) that we need to be removed is a bias towards tf value is even larger than idf value. To remove this bias, we use the first variant that normalized by the Maximum TC in a given document collection. The second bias of tc is that the words appeared in a long documents may have larger frequency and may be regarded as more important words. Hence, we want to reduce such weight of document's length, which results in the second normalized tc, ntf1. The ntf2 is given as input for IDF variant.

First normalized form:

$$ntf1 = \frac{tc1}{MAX(tc1, tc2, tc3|T|)} \tag{4}$$

Second normalized form:

$$ntf2 = \sum_{j=1}^{|D|} \frac{n_{i,j}}{\sum_{k-1}^{T_j} n_{k,j}} \tag{5}$$

ntf2 value for the terms are multiplied by idf, hence it form ntfidf. The value of ntfidf supports to arrange the key terms in descending order based on the weight calculated. Those keywords are stored in SimMat, to find similarity between terms. SimMat is nothing but a vector Space between the ({t1,t2}) terms. Terms are act as row as well as columns. The function Synset in NLTK tool kit is used to find the values of the SimMat [13].

The normalized similarity values always fall in between 0-1.The major role played here is more similar terms are removed and rest of the terms stored in Term Set. From the Term Set. One constraint to remove the term from the term set is that the similarity value more than the threshold.

Here the threshold was fixed as 0.6.SimMat(t1,t1)will be one for all terms hence it is diagonal, SimMat(t1,t2)will be from 0.1 to 0.9,The SimMat values above o.6 are removed from the Term set. Hence the Term Set contains the reduced number of keywords which are most relevant. The weight is still improved by our proposed algorithm and is a way to improve the quality of keyword extraction. Sentences are ranked only by the keyword ranking. In This method length of the sentences are analyzed for extracting keyword. All the scientific articles are simple in nature. The length of the sentence varies from 6 to 21. Our model gives Weightage to the keywords based on the length of the sentences.

It comprise of three steps.

Step 1: Find the length of the sentences

S = {s1,s2,s3,s4,...sn}

SL = {sl1,sl2,sl3,sl4.sl5....sln}

Step 2: Find the Average sentence length

$$ASL=\sum_{n}^{i=0} SL \div N$$

Step 3: Ranked keywords are updated by its weight.

Step 4: Re-ranking

2.4 Sentence Processing Algorithm

A New Approach:

Keyword Weightage by sentence processing.

/*can Extract Keyword from any kind of Corpus */

Objective: Give Priority to the terms occur in short sentences by manipulating the term weight
Input: Set of Documents D={d1,d2,d3,d4….dn}
Set of Sentences in a Document S={s1,s2,s3,s4…sn}
Set of Terms in a Sentence={t1,t2,t3….tn}
Calculate the term weight for each term by TF-IDF
Output: The Descriptive terms that describes the entire document

Step 0: Reduce the terms based on similarity value
Step 1:
/*Find the length of the sentence by counting the terms.*/
Sentence Length (SL) =Number of terms in a sentence
Step 2:
/*Find the Average Sentence Length (ASL) from the Array of sentence length
Step 3:
Check the condition that the Sentence is Greater than or Equal to or Less then the ASL
If (SLi<ASL)
TC(ti) = TC(ti)*2
/*improve the priority of the term which is in Small sentence*/
elif(SLi==ASL)
TC(ti)==TC(ti)

/*for the sentences having same length as ASL, the weight of the terms will not change, remain same*/

Elif(SLi>ASL)
TC(ti)=TC(ti)*0.5

/*reduce the priority of the term which is in small sentence*/

Step4: Arrange the terms based on their weight, write in descending order

Step5: The resultant set contains the Descriptive key terms

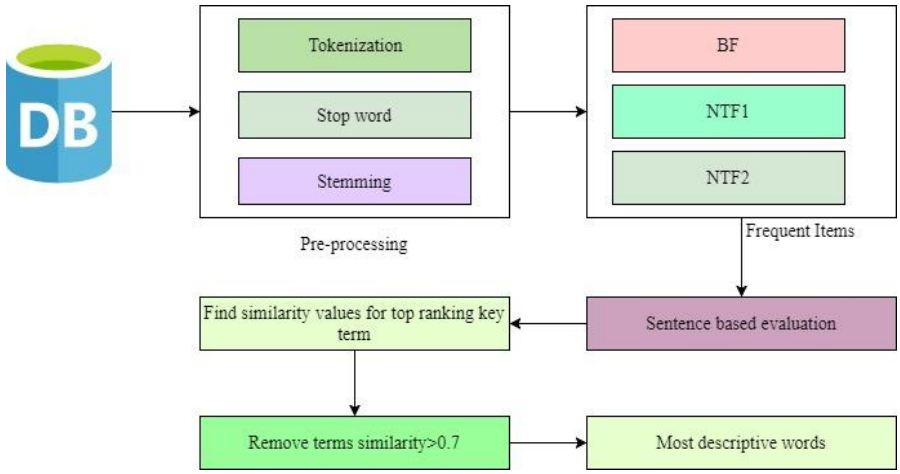

Fig. 2. Proposed Approach

3. Experimental Setup

NLTK tools to identify the similarities between words and Python are used to develop SimMat for keyword extraction. The data set are taken from Learning technology for E-Learning (LT4EL) [15] for English. It consists of 45 documents and the number of keywords extracted by traditional method is 1174. Fig. 2. gives the pictorial view of the proposed model.

From the data set a sample document is taken for evaluation. And variant TF-IDF (formula 1)is applied. The parameters count (Tc), NTF1 and NTF2 are calculated and it gives still reduced number of terms for a given document.

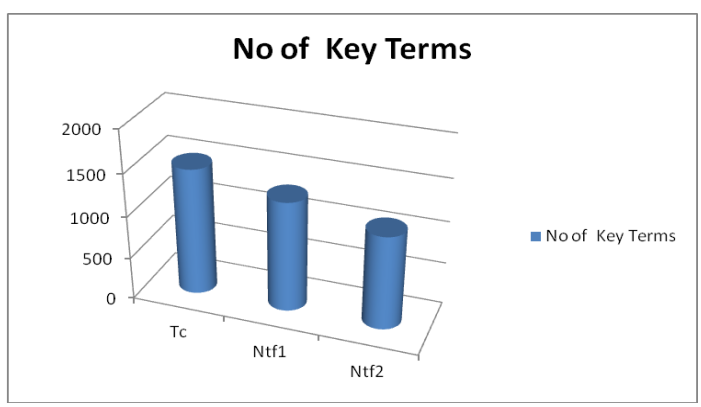

Fig. 3. Number of Terms variant

The above Fig. 3. clearly depicts the terms related to frequency from the E-Learning material provided in the website LT4eL.SimMat created based on the available terms in Term Set.

Synset in NLTK help to find similarity between the terms. The terms which are not available in WordNet can be assumed as 0 value .Here it proceeds to do the process of reduction, it is a kind of second filtering .The similarity values between the terms are fall in between 0-1, ultimately all diagonal values are 1, so make it as

zero first. Then remove the terms which has more than 0.7 as similarity value. In the analysis five different documents of various sizes are taken for experimentation.

Table 1. Comparison with other ideas

Methods	Recall	Precision	F-Measure
TF-IDF	0.48	0.26	0.32
RIDF	0.33	0.18	0.22
ADRIDF	0.47	0.28	0.32
VTF-IDF	0.48	0.28	0.33
STF-IDF	0.51	0.29	0.3695

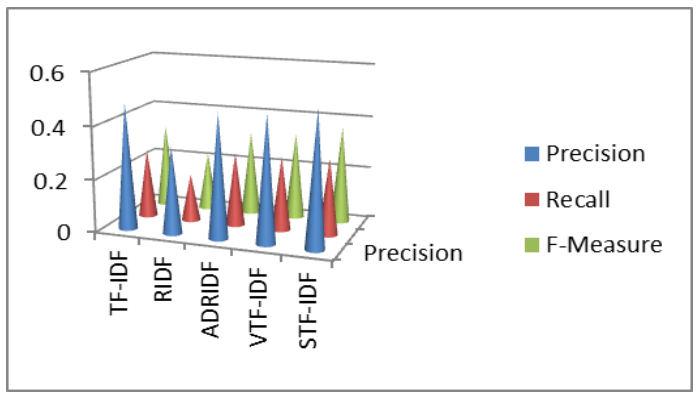

Fig. 4. Comparison of Recall and Precision

The set of all 45 documents is taken for the present analysis based on STF-IDF. This produced 1095 keywords which is more by 2 % in comparison with other conventional methods and consequently from Fig. 4. It can be seen that there is an improvement in precision and recall metrics. Also, from Table 1 the present method can be observed to result in a better harmonic mean in comparison with earlier studies and the term set is most descriptive of the documents taken.

4. Conclusions

This System creates a useful TermSet containing the most descriptive and highly related terms for any the given document. The shortcoming in traditional TF-IDF algorithm has been overcome with the usage of attributes tc, ntf1 and ntf2. The newly employed sentence based processing on keywords helps to improve the precision and recall. Further, this may also improve WordNet supported word similarity values from those in E-Learning corpus. Hence, the present approach proves to be efficient in comparison with the earlier traditional key term extraction methods.

References

[1] S.Akter, AS.Asa and MP.Uddin, MD Hossain"An extractive text summarization technique for Bengali document (s) using K-means clustering algorithm "on IEEE International Conference Imaging, Vision & Pattern Recognition (icIVPR), pp 1-6 , 2017.

[2] R.Silveira, V.Furtado, and V.Pinheiro " Ranking Keyphrases from Semantic and Syntactic Features of Textual Terms", Brazilian Conference on Intelligent Systems (BRACIS), pp 134-139, , 2015

[3] M.Litvak and M.Last "Graph based keyword extraction for single –document summarization" on MMIES '08 Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization pp:17-24,2008.

[4] P.Alireza and K.Mohadesh,"A Probabilistic Relational Model for Keyword Extraction" International Conference on Statistics in Science, Business and Engineering (ICSSBE) ,pp 1-5,2012.

[5] Sneha .S Desai, and Dr.J.A.Laxmonarayana "WordNet and Semantic Similarity based Approach for Document Clustering"International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), pp 312-317 ,2016

[6] A .Guo, and T .Yang "Research And Improvement Of Feature Words Weight Based On Tfidf Algorithm," Information Technology, Networking, Electronic and Automation Control Conference, IEEE 2016 ,pp 415-419,2016

[7] C.Clifton, R.Cooley and J.Rennie "Topcat: Data Mining For Topic Identification In A Text Corpus" IEEE Transactions on Knowledge and Data Engineering Vol 16, pp 949-964,Issue: 8, Aug. 2004

[8] L.Suanmali and N.Salim" Fuzzy Genetic Semantic Based Text Summarization.IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), pp 1184-1191,2014

[9] A. Kiani, and MR. Akbarzadeh Automatic Text Summarization Using: Hybrid Fuzzy GA-GP "IEEE International Conference on Fuzzy Systems Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada ,pp 977-983,2006

[10] P.Arora and O.Vikas " Semantic Searching and Ranking of Documents using Hybrid Learning System and WordNet" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3,pp 113-120,2011

[11] L. Lemnitzer and P. Monache" Extraction and evaluation of keywords from Learning Objects – a multilingual approachs" Language Resources and Evaluation Conference LREC, pp 112-120,2008

[12] YA.Jaradat and AT.Al-Taani "Hybrid-based Arabic Single-Document Text Summarization Approach Using Genatic Algorithm "7th International Conference on Information and Communication Systems (ICICS), pp 85-91 ,2016

[13] Porter M.F., "An Algorithm for Suffix Stripping", MCB UP Ltd Program, Vol. 14, no. 3, pp. 130-137, 1980.

[14] http://www.nltk.org/howto/wordnet.html

[15] http://www.lt4el.eu/review_luxembourg.php

Authors' Profiles

T. Vetriselvi: Part-Time Research Scholar at Department of Computer Applications, National Institute of Technology Tiruchirappalli. She is currently an assistant professor at the K. Ramakrishnan College of Technology, Tiruchirappalli. She has 11 years of teaching experience and 3 years of industry experience. She is the author of 4 journal scientific papers, 2 books and 3 proceedings. Her areas of interest include Data Mining and Programming Languages.

N. P. Gopalan: Professor of Computer Applications Department, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India. He obtained his PhD. from Indian Institute of Science, Bangalore. Interested in Data Mining, Cryptography, Distributed Computing and Theoretical Computer Science.

G. Kumaresan: Research Scholar at Department of Computer Applications, National Institute of Technology Tiruchirappalli. He received MCA from Thiagarajar College of Engineering, Madurai, India. M.Tech from Bharathidasan University, Tiruchirappalli, India and M.Phil. from St. Joseph College, Tiruchirappalli, India. His areas of interest include Public Key Cryptography, Cellular Automata and Cloud Security and Programming Languages.

How to cite this paper: T. Vetriselvi, N. P. Gopalan, G. Kumaresan,"Key Term Extraction using a Sentence based Weighted TF-IDF Algorithm", International Journal of Education and Management Engineering(IJEME), Vol.9, No.4, pp.11-19, 2019.DOI: 10.5815/ijeme.2019.04.02