International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2016, pp. 401-407
e-ISSN:2278-621X

EXTRACTION-BASED SINGLE-DOCUMENT SUMMARIZATION

Clinton Kenny Fernandes[1], Amit Thapa[2], Evander Fernandes[3] and Mrs. Manimozhi R[4]

Abstract–Text summarization technique for text documents exploiting the semantic similarity between sentences to omit the redundancy from the given text. Semantic similarity scores are computed by using Random Indexing. Random Indexing, in comparison with other semantic space algorithms, presents vectorized method for dimensionality reduction. It's an efficient way to compute similarity between words, sentences and documents.

I. INTRODUCTION

Automatic Text Summarization is an important and challenging area of Natural Language Processing. The task of a text summarizer is to produce a synopsis of any document or a set of documents submitted to it. Summaries differ in several ways. A summary can be an extract i.e. certain portions (sentences or phrases) of the text is lifted and reproduced verbatim, whereas producing an abstract involves breaking down of the text into a number of different key ideas, fusion of specific ideas to get more general ones,and then generation of new sentences dealing with these new general ideas . A summary can be of a single document or multiple documents, generic (author's perspective) or query oriented (user specific), indicative (using keywords indicating the central topics) or informative (content laden). In this work we have focused on producing a generic, extractive, informative, single document summary exploiting the semantic similarity of sentences.

II. PREVIOUS WORK IN EXTRACTIVE TEXT SUMMARIZATION

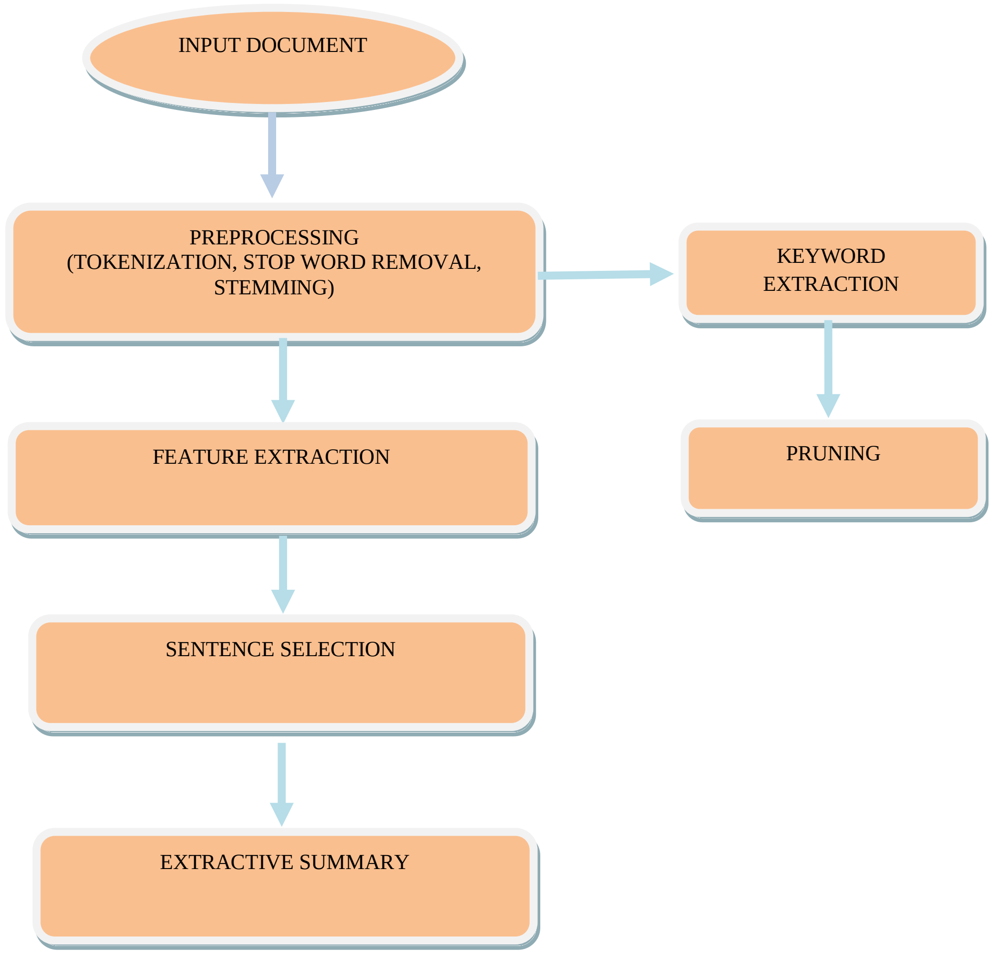
Various methods have been proposed to achieveextractive summarization. Most of them are based onscoring of the sentences. Maximal Marginal Relevance scores the sentences according to their relevance to thequery, Mutual Reinforcement Principle for Summarygeneration uses clustering of sentences to score them according to how close they are to the central theme. QR decomposition method scores the sentences using column pivoting. The sentences can also be scored bycertain predefined features.

These features may includelinguistic features and statistical features, such as location, rhetorical structure,presence or absence of certain syntactic features and presence of proper names, and statistical measures of term prominence.

Rough set based extractive summarization hasbeen proposed that aims at selecting important sentences from a given text using rough sets, which has been traditionally used to discover patterns hidden in data. Methods using similarity between sentences and measures of prominence of certain semantic concepts and relationshipsto generate an extractive summary havealso been proposed.

[1] *Aloysius Institute of Management and Information Technology (AIMIT) St Aloysius College (Autonomous) Mangalore, Karnataka, India*
[2] *Aloysius Institute of Management and Information Technology (AIMIT) St Aloysius College (Autonomous) Mangalore, Karnataka, India*
[3] *Aloysius Institute of Management and Information Technology (AIMIT) St Aloysius College (Autonomous) Mangalore, Karnataka, India*
[4] *Aloysius Institute of Management and Information Technology (AIMIT) St Aloysius College (Autonomous) Mangalore, Karnataka, India*

III. PROPOSED DESIGN



A) INPUT DATA

Input file consist of raw data to be processed by the system.

B) PREPROCESSING

a) TOKENIZATION

Break down the passages into sentences and each of these sentences is further broken into a set of words or tokens. Data obtained in the form of set of words is further analyzed and stop words or most commonly occurring words are removed from the set of words by performing stop word removal.

b) STOP WORD REMOVAL

Data obtained in the form of set of words is further analyzed and stop words or most commonly occurring words like a, an, the etc are removed from the set of words. Stop word list referred is by Gerard Salton and Chris Buckley. This wordlist is 571 words in length.

c) STEMMING

The words are brought to their root form. The main objective is to assign equal importance to words having the same root. Thus, words in their different forms are considered to be the same. For e.g. the words likes 'compute', 'computed', 'computing', 'computer', 'computation', and 'computable' are brought to the root form 'comput'.

Commonly used stemming algorithm is Porter Stemmer

The following steps are followed:-

a) Get rid of plurals and –ed and -ing suffixes.

b) Turns terminal y to i when there is another vowel in the stem.
c) Maps double suffixes to single ones. –ization, -ational etc.
d) Deals with suffixes –full, -ness etc.
e) Takes off –ant, -ence etc.
f) Removes a final –e.

C)KEYWORD EXTRACTION
TF-IDF weight evaluates the importance of a word to a document in a collection.
tf–idf is calculated as

$$\text{tf-idf} = \text{tf} * \text{idf}$$

$$tf_{ij}=(n_{i,j}) / \Sigma_k n_{k,j}$$

where $n_{i,j}$ is number of occurences of term(t_i) in document d_j
$\Sigma_k n_{k,j}$ is the sum of number of occurrences of all terms in dj.

$$idf_i=\log N / n_i|$$

where N - number of documents in the collection,
ni - number of documents in which term i occurs.

For single document idf factor won't be considered because there is a single document. Therefore value of idf will be zero. So only **term frequency** will be considered.

PRUNING
A threshold for tf (term frequency) weights is defined. All terms with tf weights lesser than the threshold are pruned from the document.

D) FEATURE EXTRACTION
Various set of features is applied to the pre-processed document.

- Position of sentence: - Position of the sentence in the text, decides its importance. This feature can involve several items such as the position of a sentence in the document, section and paragraph etc. Suppose we consider the first five sentences in the paragraph.
  F1 (S) = 5/5 for 1st,
  4/5 for 2nd,
  3/5 for 3rd,
  2/5 for 4th,
  1/5 for 5th,
  0/5 for other sentences

- Proper nouns: - Weights will be assigned to sentences containing named entities (Proper Nouns), since named entities usually contain key information.

$$F2(S) = \frac{\text{number of proper nouns in the sentence}}{\text{sentence length}}$$

- Title feature: - This feature gives the measure of the similarity between the title sentence and every other sentence of the document. This is determined by counting the number of matches between the content words in a sentence and the words in the title.

$$F3(S) = \frac{\text{number of title words in the sentence}}{\text{number of words in the title}}$$

- Sentence length: - A longer sentence will tend to contain more information while a very short one may contain no information at all.

$$F4(S) = \frac{\text{length of the sentence s}}{\text{length of the longest sentence in a document}}$$

- Numerical data: - If at all, any numerical data is available in the document, they are important. Hence, a weight of one is assigned to the sentences having numerical values, zero otherwise.

$$F5(S) = \begin{cases} 1 & \text{if sentence has numerical data} \\ 0 & \text{otherwise} \end{cases}$$

- Sentence to sentence similarity: - This feature is a similarity between sentences. Each sentence S, the similarity between S and each other sentence is calculated by the cosine similarity measure with a resulting value between 0 and 1. Vectors are represented by the term weight $w_i$and w_j of t to n term in sentence Si and Sj. The similarity of each sentence pair is calculated based on similarity formula

$$Sim(S_i, S_j) = \frac{\sum_{t=1}^{n} w_{it} \ x \ w_{jt}}{\sqrt{\sum_{t=1}^{n} w_{it}^2} \ x \ \sqrt{\sum_{t=1}^{n} w_{jt}^2}}$$

The score of this feature for a sentence S is obtained by computing the ratio of the summation of sentence similarity ofsentence S with each other sentence over the maximum of summation

$$S_{FS(s)} = \frac{\sum Sim(S_i, S_j)}{Max(\sum Sim(S_i, S_j))}$$

The above $S_{FS(S)}$ value is normalized by diving it with maximum similarity.

- Keyword weight: - Keywords occurring a sentence may be of great importance.

This feature is calculated by

$$F8(S) = \frac{\text{number of keywords}}{\text{length of the sentence}}$$

E) SENTENCE SELECTION

All sentences in a document are ranked in descending order based on their score. Select Top n sentencesbased on extent of summarization. Finally the sentences in the summary are arranged in the order they occur in original document.

15
13
10
6

Suppose
15 - 4 position in original doc
13 - 1 position in original doc
10 - 3 position in original doc
6 - 2 position in original doc
Define extent of summarization: say suppose 50 %

(50 / 100) × Total number of sentences
We have in this case total number of sentences=4
In this case value will be 2

So select top 3 sentences.

15 - 4 position in original doc
13 - 1 position in original doc

Therefore display of sentences will be (Sentences will be displayed by looking at the position in the original doc)

13
15

IV. EXPERIMENT AND RESULT

Generation of rules
Consider the following sentences with following feature values
Say F1= Sentence position
Say F2= Title word
Say F3= Numerical value
Say F4= Keyword weight
Say F5= Proper noun
Say F6= Sentence to sentence similarity
Say F7= Sentence length
(After rounding up the values to 3 decimal points)
Sentence 1: (F1=1, F2=0.667, F3=0.05, F4=0.25, F5= 0.4,F6=0.373, F7=0.741)
Sentence 2: (F1= 1, F2=0.5, F3=0.08, F4=0.16, F5=0.04, F6=0.45, F7=0.926)
Sentence 3: (F1=1, F2=0.167, F3=0.071, F4=0.214, F5= 0.071, F6=0.356, F7=0.519)
Sentence 4: (F1=1, F2=0.5, F3=0, F4=0.238, F5= 0.286, F6=0.419, F7=0.778)
Sentence 5: (F1=1, F2=0.333, F3=0, F4=0.25, F5= 0.083, F6=0.532, F7=0.444)
Sentence 6: (F1=0.5, F2=0,F3=0, F4=0.167, F5= 0, F6=0.257, F7=0.222)
and so on……….

Calculate low and high value for each feature considering all sentences.
Now for this 6 sentences

Low= $\frac{min+max}{2}$
High= all values higher than mean value

For feature F1
Low = $\frac{1+0.5}{2}$
= 0.75
= 0 to 0.75
High = >0.75 to 1

For feature F2
Low =$\frac{1+0.167}{2}$
= 0.084
= 0 to 0.084
High = > 0.084 to 1

For feature F3
Low = $\frac{0+0.08}{2}$

= 0.04
= 0 to 0.04
High = >0.04 to 1

For feature F4
Low = $\frac{0.16+0.25}{2}$
= 0.205
= 0 to 0.205
High = >0.205 to 1

For feature F5
Low = $\frac{0+0.286}{2}$
= 0.143
= 0 to 0.143
High = >0.143 to 1

For feature F6
Low = $\frac{0.257+0.532}{2}$
= 0.395
= 0 to 0.395
High = >0.395 to 1

For feature F7
Low = $\frac{0.222+0.926}{2}$
= 0.574
= 0 to 0.574
High = >0.574 to 1

For all features LOW will be represented as 0 and HIGH is represented as 1.

Only this Single Rule will be written
If (F1= 1, F2=1, F3=1,F4=1,F5=1,F6=0, F7=1) then sentence is important.
All features will take a value 1 only sentence to sentence similarity will take a value 0 because we want less similar sentences as the output.
Now for each sentence map F1, F2,…F7 value and check if they fall in Low or High range.
Sentence 1: (F1=1, F2=1, F3=1, F4=1, F5= 1, F6=0, F7=1)
Sentence 2: (F1= 1, F2=1, F3=1, F4=0, F5=0, F6=1, F7=1)
Sentence 3: (F1=1, F2=1, F3=1, F4=1, F5= 0, F6=0, F7=0)
Sentence 4: (F1=1, F2=1, F3=0, F4=1, F5= 1, F6=1, F7=1)
Sentence 5: (F1=1, F2=1, F3=0, F4=1, F5= 0, F6=1, F7=0)
Sentence 6: (F1=0, F2=0, F3=0, F4=0, F5= 0, F6=0, F7=0)
In this way we pass all the sentences through that **SINGLE RULE.** Now we check each feature value with the value of the rule.
If there is a mismatch we write 1 and if there is a match we write 0.
Sentence 1: (0, 0, 0, 0, 0, 0, 0)
Sentence 2: (0, 0, 0, 1, 1, 1, 0)
Sentence 3: (0, 0, 0, 0, 1, 0, 1)
Sentence 4: (0, 0, 1, 0, 0, 1, 0)
Sentence 5: (0, 0, 1, 0, 1, 1, 1)
Sentence 6: (1, 1, 1, 1, 1, 0, 1)
Count the number of 1's. (These are mismatching features)
Sentence 1= 0
Sentence 2 = 3
Sentence 3= 2

Sentence 4= 2
Sentence 5= 4
Sentence 6 =7

V. CONCLUSION

In the previous approach sentence similarity was considered and similar sentences were selected, but then selected sentences would be similar and may not take a better coverage and in order to overcome this problem we modified this technique i.e., to overcome the similarity problems and to get more diversified results.

REFERENCES

[1] Inderjeet Mani, “Advances in Automatic Text Summarization”, MIT Press, Cambridge, MA, USA, 1999.
[2] Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell, “Summarizing text documents:sentence selection and evaluation metrics”, ACM SIGIR, 1999, pp 121–128.
[3] E.H. Hovy and C.Y. Lin, “Automated Text Summarization in SUMMARIST”, Proceedings of the Workshop on Intelligent Text Summarization, ACL/EACL-97. Madrid, Spain, 1997.
[4] J. Carbonell and J. Goldstein, “The use of MMR, diversity based re-ranking for reordering documents and producing summaries,” ACM SIGIR, 1998, pp. 335–336.
[5] ZhaHongyuan, “Generic Summarization and Key phrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering”, ACM, 2002.
[6] Svore, K., Vanderwende, L., and Burges, C. (2007). Enhancing single-document summarization by combining RankNet and third-party sources. In Proceedings of the EMNLP-CoNLL, pages 448–457. [7, 8]
[7] SiyaSadashivNaik,ManishaNaikGaonkar “survey of extractive based automatic text summarization techniques” In International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 22 Issue 2 – MAY 2016.
[8] Dipanjan Das,Andre F.T. Martins A Survey on Automatic Text Summarization “”,Language Technologies Institute Carnegie Mellon University,2007.