A method for Automatic Text Summarization using Consensus of Multiple Similarity Measures and Ranking Techniques

Mukesh Kumar Jadon
Department of Computer Science and Engineering
The LNM Institute of Information Technology
Jaipur, India
jadonmukesh30@gmail.com

Ayush Pareek
Department of Computer Science and Engineering
The LNM Institute of Information Technology
Jaipur, India
ayush.original@gmail.com

Abstract

In the era of information overload, text summarization can be defined as the process of extracting useful information from a large space of available content using traditional filtering methods. One of the major challenges in the domain of extraction based summarization is that a single statistical measure is not sufficient to produce efficient summaries which would be close to human-made 'gold standard', since each measure suffers from individual weaknesses. We deal with this problem by proposing a text summarization model that combines various statistical measures so that the pitfalls of an individual technique could be compensated by the strengths of others. Experimental results are presented to demonstrate the effectiveness of the proposed method using the TAC 2011 Multiling pilot dataset for English language and ROUGE summary evaluation tool.

1 Introduction

What is the need of text summarization? One of the major reasons is Information explosion. A study (Cho et al., 2015) by IBM in 2013 estimated that everyday 2.5 Quintillion bytes of new information were born on the internet. The velocity with which this is increasing can be estimated from the fact that 90 percent of total data on web at that time was created in the previous two years alone. Thus there is a progressing need to effectively extract useful content from big data and use streamlined filtering methods to make it comprehensible and non- redundant.

In this paper, we have introduced a novel technique for automatic single-document extraction-based text summarization. The proposed approach uses a number of statistical models such as Pearson Correlation Coefficient, Cosine Similarity and Jaccard Similarity (Huang and Anna, 2008) that compute multiple summaries of a text and combine them, using configurable consensus methods.

Finally, we stretch a step further and use machine-learning to make the summary domain-specific or personalized. Our basic focus is on designing a technique to improve the weighing constants in the consensus step by using a genre-specific training set.

2 Related Work

Most common methods of automatic text summarization are either based on abstraction or extraction. Abstraction based methods focus on creating an internal semantic representation of source text using Natural Language Processing and generating new sentences as the summary (Dragomir and Radev, 2004; Hahn and Romacker, 2001). In contrast, extraction-based summarization is based on extracting a subset of the original text as a summary (Gupta et al., 2010). The most common way to achieve this is through sentence ranking by associating a score with each sentence and greedily choosing the highest weighting sentences for the final summary up to the required compression ratio (Nenkova et al., 2012). In this way, locally optimal solutions enable global optimization which is presented as the final summary. Another approach is based on clustering of similar sentences and successively extracting those sentences which do not represent the redundant information (Huang, 2008; Aggarwal et al., 2012).

In both of these approaches, data regarding similarity of all pairs of sentences is essential for producing the rankings. Many useful measures have been

D S Sharma, R Sangal and A K Singh. Proc. of the 13th Intl. Conference on Natural Language Processing, pages 137–143,
Varanasi, India. December 2016.

proposed like Cosine similarity, IR f-measure (Alguliev & Aliguliyev, 2007), Jaccard similarity (Mittal et. al, 2014) etc. But it has been experimentally (Alguliev & Aliguliyev, 2007) discovered that due to imperfections of any particular similarity measure, there could be significant prejudices in the scores which make the summarizer misjudge the importance of sentences. We have tried to overcome this problem by combining multiple algorithms so that erroneous rankings can be normalized.

3 Proposed Approach

Consider a document **D** containing a set of sentences $S = \{s_1, s_2, \ldots, s_m\}$ and a set of unique terms $T = \{t_1, t_2, \ldots, t_p\}$. The proposed framework for summarization of D consists of four phases as depicted in Figure 1.

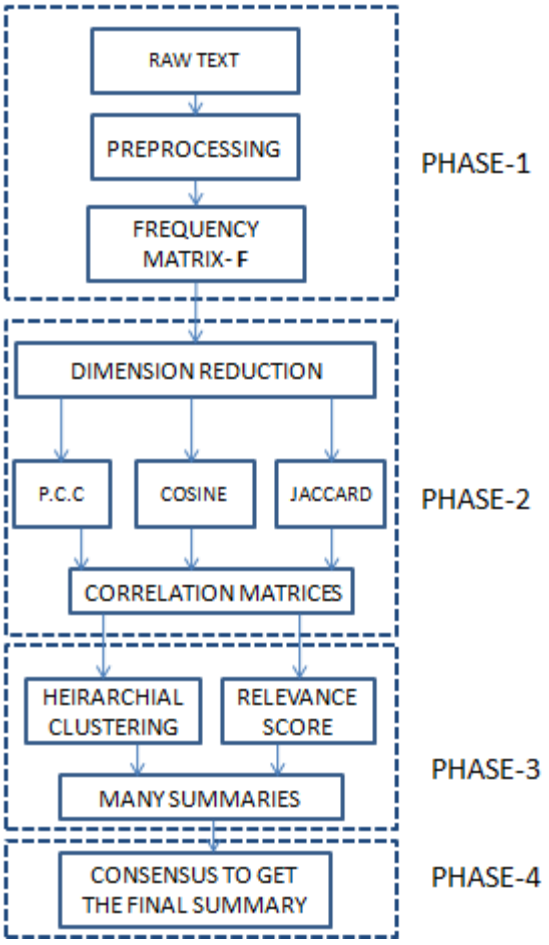

Figure 1. Framework for summarization process

3.1 Phase 1: Frequency Matrix Formation

Raw text contains noise in the form of stop words, punctuation marks, digits, special characters etc. which is not useful for our statistical analysis and increases the probability of exponential work in our process. Thus, we eliminated such terms using close-class word list (Buckley and Salton, 2013) and converted the remaining terms to lower-case. Next, we performed stemming using the Porter stemming algorithm (1980). It has been shown that this kind of preprocessing does not degrade the performance of extractive summarization (Ledeneva, 2008).

Finally, we created a frequency matrix F^{mxq} for m sentences and q distinct terms ($q<p$ since some terms have been eliminated) where element f_{ij} is the frequency of the term t_j in sentence s_i.

3.2 Phase 2: Getting Correlation and Similarity Matrices using Statistical Models

Application of text mining algorithms on matrix F^{mxq} with very high value of dimension q could be computationally costly. Hence, Zipf's law, based on Luhn's model (Luhn, 1958), could be used to eliminate terms with very high or very low cumulative frequency as shown in Figure 2. Formally, the cumulative frequency (**Cf**) for a term t_j in F^{mxq} can be represented as-

$$Cf(t_j) = \sum_{i=1}^{q} f_{ij} \quad (1)$$

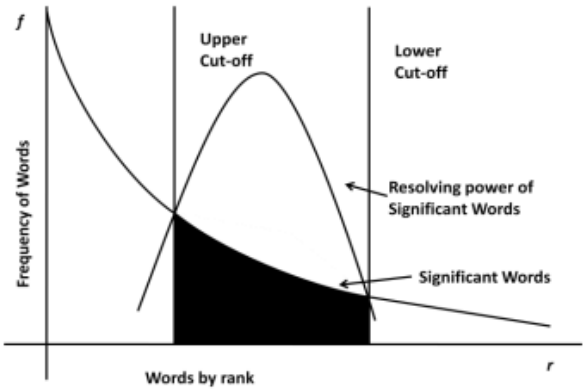

Figure 2. Zipf's Law dictates that words having rank (based on frequency) between the upper and lower cut-offs are most significant

Let n terms remain after application of Zipf's Law. We define sentence vector S_i on matrix F^{mxn} (where $n<q<p$) such that

$$S_i = [f_{i1}, f_{i2}, f_{i3}, \ldots, f_{in}] \quad (2)$$

Finally, we generated three Correlation or Similarity matrices C_1^{mxm} , C_2^{mxm} and C_3^{mxm} by using the following statistical measures (Huang, 2008) respectively-

1. Pearson Correlation Coefficient
2. Cosine Similarity
3. Jaccard Similarity

Let $c_k(i, j)$ be an element in the matrix C_k^{mxm}.

Formally,

$$c_k(i,j) = Simlarity\left(S_i, S_j\right) \quad \forall \ \mathrm{k} \in \{1,2,3\} \quad (3)$$

Figures 3, 4 and 5 show sample matrices C_1^{mxm} , C_2^{mxm} and C_3^{mxm} (with m=6) respectively.

	S_1	S_2	S_3	S_4	S_5	S_6
S_1	1	0.224	0.125	0.404	0.127	0.224
S_2	0.224	1	0.317	0.328	0.012	0.116
S_3	0.125	0.317	1	0.297	-0.092	-0.050
S_4	0.404	0.328	0.297	1	0.079	0.349
S_5	0.127	0.012	-0.092	0.079	1	-0.079
S_6	0.224	0.116	-0.050	0.349	-0.079	1

Figure 3. C_1^{6x6} using Pearson Correlation Coefficient

	S_1	S_2	S_3	S_4	S_5	S_6
S_1	1	0.452	0.029	0.172	0.035	0.221
S_2	0.452	1	0.167	0.278	0.032	0.196
S_3	0.029	0.167	1	0.284	0.002	0.010
S_4	0.172	0.278	0.284	1	0.079	0.169
S_5	0.035	0.032	0.002	0.079	1	0.093
S_6	0.221	0.196	0.010	0.169	0.093	1

Figure 4. C_2^{6x6} using Cosine similarity

	S_1	S_2	S_3	S_4	S_5	S_6
S_1	1	0.341	0.017	0.167	0.044	0.177
S_2	0.341	1	0.174	0.332	0.024	0.214
S_3	0.017	0.174	1	0.265	0.013	0.023
S_4	0.167	0.332	0.265	1	0.021	0.147
S_5	0.044	0.024	0.013	0.021	1	0.113
S_6	0.177	0.214	0.023	0.147	0.113	1

Figure 5. C_3^{6x6} using Jaccard similarity

3.3 Phase 3: Ranking Sentences using Relevance Score and Clustering Methods

The fundamental goal of this phase is to devise methods for ranking sentences using matrices C_1^{mxm} , C_2^{mxm} and C_3^{mxm} so that the summaries can be obtained using a subset of top ranked sentences. The size of the summaries would depend on user-given compression ratio.

3.3.1 Ranking through Relevance Score

We define ***Relevance Score*** or ***RScore*** of sentence s_i as the sum of its correlation magnitudes with all other sentences, except itself i.e.

$$RScore(s_i) = \sum_{j=1}^{n} c_k(i,j), \ i = 1,2,\ldots,n. \ i \neq j \quad \forall\, k \in \{1,2,3\} \quad (4)$$

In this method, we ranked sentences from highest to lowest based on decreasing order of their ***RScores.*** The sentences which have the highest ***RScores*** form a subset of S having high information content and low redundancy (Gong et al., 2001). We performed this process for all three matrices C_1^{mxm}, C_2^{mxm} and C_3^{mxm} obtained in phase 2 and generated three ranking orders of sentences - R_1, R_2 and R_3. Finally, we obtained summaries Sm_1, Sm_2 and Sm_3 by extracting the top ranking sentences from R_1, R_2 and R_3 respectively, until they satisfy the compression ratio.

3.3.2 Ranking through Hierarchical Clustering

Clustering is a form of unsupervised learning in which categorization is done based on highest similarity. The goal is to organize data into natural collections such that we obtain high intra-collection similarity and low inter-collection similarity (Huang, 2008). We have used pair wise hierarchical clustering of sentence vectors in matrices C_1^{mxm}, C_2^{mxm} and C_3^{mxm} to group together sentences which represent similar information. Detailed procedure is described as follows-

Let S_i and S_j be two sentence vectors clustered together in the matrix C_k^{mxm}. We define normalization of row vector S_i as replacing S_i with the mean of corresponding elements of S_i and S_j.

Formally,

$$N^{1xn}{}_{row}(S_i) = [\frac{c_{i,1} + c_{j,1}}{2}, \frac{c_{i,2} + c_{j,2}}{2}, \ldots \frac{c_{i,n} + c_{j,n}}{2}] \quad (5)$$

Similarly, we can define normalization of column vector S_i as-

$$N^{nx1}{}_{col}(S_i) = [\frac{c_{1,i} + c_{1,j}}{2}, \frac{c_{2,i} + c_{2,j}}{2}, \ldots \frac{c_{n,i} + c_{n,j}}{2}] \quad (6)$$

Let $R(S_i)$ denotes removing sentence vector S_i from the matrix C^{mxm} and thus reducing its dimensions to $(m-1)$ x $(m-1)$.

Also, define $c_k(x, y)$ as the maximum non-diagonal element in the matrix $C_k{}^{mxm}$ i.e.

$$c_k(x,y) = \max(c_k(i,j)) \quad \forall (i,j) \; such\; that \; i \neq j \quad (7)$$

We extract ranking orders from correlation matrices as described in Algorithm 1.

INPUT: $Matrices \; C_k{}^{mxm} \forall k \in \{1,2,3\}$

OUTPUT: Ranking orders of sentences - $R_k \forall k \in \{1,2,3\}$

```
1.  F(S_t): Returns the
      position of S_t in source text
2.  Do for t = 1 to k
3.    Do for l = m to 1
4.      If (m = 1) Then
5.        R_t.append(F(S_1))
6.      Else
7.        Find c_k(x, y)          //eq. 7
8.        Do N_row(S_z) and N_col(S_z),
            where z = max(x, y)
9.        Do R(S_w) where w = min(x, y)
10.       R_t.append(F(S_w))
11.     EndIf
12.   EndDo
13. EndDo
```

Algorithm 1. Pseudo-code for getting ranking orders from Correlation matrices

Summaries Sm_4, Sm_5 and Sm_6 are derived by extracting the top ranking sentences from R_1, R_2 and R_3 until they satisfy the compression ratio.

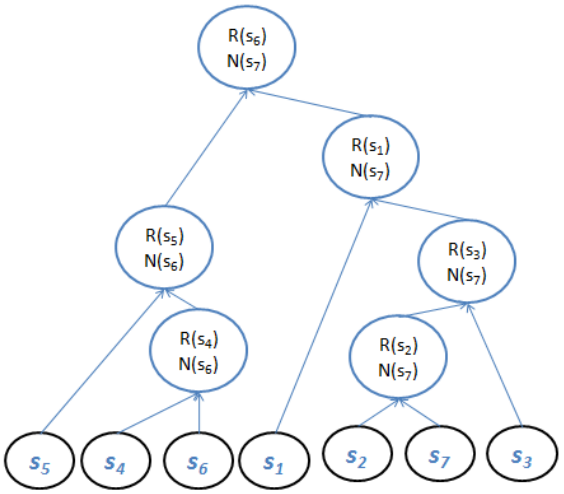

Figure 6. One of the possible clustering patterns by application of the ranking algorithm on seven sentences.

.

By forming clusters, we are essentially grouping together two sentences with similar information and using the sentence which occurs first, as its representation in the ranking order. We call the sentence vector which is removed from the cluster as 'representative sentence vector' of the cluster. The other sentence vector is normalized using eq.5 and eq.6. This is performed to improve the accuracy of similarity measure magnitudes for non-representative sentence vector by averaging its coefficients with those of the representative sentence vector, since they are found out to be the most similar among the sentence vectors that are present in the Correlation matrix.

3.4 Phase 4: Consensus Methods to get the final summary

At the end of Phase 3, we have six ranking orders , namely- R_1, R_2 , R_3 R_4, R_5 and R_6 .Given a compression ratio **r**, we can obtain six summaries from these rankings, namely-Sm_1, Sm_2 , Sm_3 ,Sm_4, Sm_5 and Sm_6 ,by extracting the top ranked sentences till r is satisfied and then sort in the order they appeared in the original text.

3.4.1 *Generic Summaries*

In this section, we will describe a method for getting a generic summary by giving equal importance to all the summaries obtained. Let us generalize the number of summaries we obtained from phase 3 as ***k.***

Let weight W_i where $i \in \{1,2,..,k\}$ represents the importance given to the i^{th} summary in deriving the final summary.
For generic summary,

$$W_i = \frac{1}{k} \forall\, i \in \{1,2,\dots,k\} \tag{7}$$

For all sentences S_i for which ∃ at least one summary $\mathbf{Sm_j}$ ($j \in \{1, 2, \dots, k\}$) such that $\mathbf{S_i} \in \mathbf{Sm_k}$, we define the Final-Score as-

$$FScore(S_i) = \sum_{j=1}^{k} W_j B_j \tag{8}$$

where, $B_j = 1$ if the sentence S_i is present in the Summary j and $B_j = 0$, otherwise. To get the final summary, we arranged all sentences in summaries Sm_j ($j \in \{1, 2, \dots, k\}$) in decreasing order of ***FScore*** and extracted from the top till the compression ratio was satisfied. Then, the sentences were sorted in the original order of source text and finally presented as a generic-summary.

3.4.2 *Query-based Summaries*

Since we have k distinct summaries, their compatibility with user-given keywords or title of the text can be measured by calculating a query score based on the distribution of query terms and can be added to its Final-Score. To simplify this process, we propose using the 'cumulative frequency of keywords in a summary' as the primary metric for its relevance and calculate ***FScore*** using this hypothesis. Moreover, sophisticated metrics giving weight age to keyword distribution can also be used in further research.
By defining F_j as the cumulative frequency of all keywords in summary j, we have-

$$W_j \propto F_j \tag{9}$$

Equation (8) in this case becomes,

$$FScore(S_i) = \sum_{j=1}^{k} F_j B_j \tag{10}$$

where, F_j= total frequency of keywords in j^{th} summary. As described in section 3.4.1, we arranged all sentences in Summaries Sm_i ($i \in \{1, 2, \dots, k\}$) in decreasing order of Final-Score and extract from the top till the compression ratio is satisfied. Then the sentences were sorted in original order of source text and presented as final summary.

3.4.3 *Domain-specific and Personalized Summaries*

A Machine Learning based method for generating summaries which improve themselves using a certain training set is described as follows.

The ***F-measure*** represents the accuracy of test summary with respect to the ideal summary (Powers, 2011).

$$F = 2\frac{precision.recall}{precision + recall} \tag{11}$$

The fundamental idea is to utilize a domain-specific training set with v documents to adjust the weights- $W_i \; \forall \; i \in \{1, 2, \dots, k\}$ assigned to each summary, based on its ***F-measure*** with respect to the ideal summary. Then, the final summary is obtained using ***FScore***s (Equation 8). Hence, the performance of summarizer can be significantly improved when another document of similar domain is given as input. Let f_{js} ($s \in \{1, 2, \dots, k\}$ be the ***F-measure*** of summary Sm_j ($j \in \{1, 2, \dots, k\}$ obtained from document s using proposed approach. Let A_j be the algorithm used to derive Sm_j. We define the mean-***F-measure*** for all summaries obtained using A_j
as-

$$F_{mean_j} = \frac{\sum_{s=1}^{v} f_s}{v} \tag{12}$$

For summarizing a document after training the summarizer, we followed the same approach till phase 3 but modified Equation (8) in section 3.4.1 as follows-

$$FScore(S_i) = \sum_{j=1}^{k} F_{mean_j} B_j \quad (13)$$

Rest of the approach is same as described in Section 3.4.1. Thus, we are using supervised learning to measure the mean performance of different algorithms on a domain-specific training set and using this knowledge to assign weights to the summaries derived by these algorithms. The final summary of a new document of the same domain will represent a consensus of these algorithms in proportion to their performance on the testing data.

For personalized summaries, a user profile is required, containing the keywords which will represent the interest of user. These keywords may be retrieved from online social media like LinkedIn or facebook, blogs etc. or they could be explicitly provided by the user. In this case, we will have **m** summaries namely $\mathbf{Sm_j}$ ($\forall\ j \in \{1, 2, \ldots, k\}$) by following the same approach till phase 3. We define Summary Score as follows-

$$SumScore_j = M_j \quad \forall\ j \in \{1,2,\ldots,k\} \quad (14)$$

where, $\mathbf{M_j}$ = metric to determine the amount of relevance a particular summary has with respect to the input keywords. To simplify this procedure, we could take the frequency measure of the keywords in the particular summary as a metric but much better metrics which take into account the distribution of these keywords can also be applied. We simply output the summary having the maximum ***SumScore*** as the final summary.

Another approach requires the user to train the summarizer by identifying ideal summaries with respect to his requirement, using data sets and their summaries which satisfy his specifications.
Hence, the summarizer is trained using personalized supervised learning to adjust its weights and adapt to the exposed configuration. This approach essentially converts a particular user's personalized configuration as a new domain and then follows the domain-specific approach discussed previously in Section 3.4.3.

4 Experiments

For testing our approach, we have used the following two datasets: (1) TAC 2011 Multiling pilot summarization task dataset (Giannakopoulos et al., 2011) which is derived from publicly available WikiNews (http://www.wikinews.org/). It is divided into ten collections with each collection containing ten documents. A single collection has all documents of the same topic. We have only tested on the English version of the documents. The dataset contains at least one "golden-summary" for comparison. (2) Six English News document collections retrieved for Summarization-research purpose in 2012 (http://dbdmg.polito.it/wordpress/research/document-summarization/). Each collection is made up of ten news documents of the same topic each. This data set was retrieved from Google News for research related to TAC2011 MultiLing Pilot Overview (Giannakopoulos et al., 2011).
We divided the ten documents in each collection into a training set and a testing test randomly with each set consisting of 5 documents. The upper limit of the summary size was kept to be 250 words. The widely used summary evaluation tool 'ROUGE' (Lin, 2004) was used to compare the results with several other summarizers. Our competitors consist of summarization methods that competed in the TAC 2011 conference (UBSummarizer and UoEssex), a widely used summarizer which is integrated with Microsoft-Word (autosummarize) and the recently proposed Association Mixture Text Summarization (AMTS) (Gross et al., 2014). Test results the Recall, Precision and F-measure using ROUGE-2 and ROUGE-SU4 summary evaluation techniques that is shown in Table1 and Table 2 respectively.

Summarizer	ROUGE-2		
	Recall	Precision	F1
Our Summarizer	**0.0740**	0.1616	**0.1015**
UBSummarizer	0.0466	0.0952	0.0625
AMTS	0.0705	**0.1633**	0.0984
autosummarize	0.0425	0.0824	0.0560
U_oEssex	0.0712	0.1617	0.0988

Table 1: Test Results using ROUGE-2

Summarizer	ROUGE-SU4		
	Recall	Precision	F1
Our Summa-rizer	0.0838	**0.2080**	**0.1194**
UBSummarizer	0.0711	0.1652	0.0994
AMTS	**0.0843**	0.1994	0.1185
autosummarize	0.0684	0.1567	0.0952
U_oEssex	0.0839	0.1976	0.1177

Table 2: Test Results using ROUGE-SU4

5 Conclusion and Future Work

Based on different similarity measures and ranking procedures, we presented a model for performing consensus of multiple summarization algorithms that perform extraction-based text summarization. Experimental results reveal that our approach has significantly outperformed some of the widely used techniques. We have argued that this is due to the strength of combining various similarity measures to get rid of their individual weakness and also due to the 'domain-adaptive' nature of our summarizer.

For further research, we will add more similarity measures and ranking techniques in the model to make it more accurate.

References

Anastasios Tombros and Mark Sanderson. Advantages of Query Biased Summaries in Information Retrieval. In Research and Development in Information Retrieval, pages 2–10, 1998.

Anna Huang. "Similarity measures for text document clustering."Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand. 2008.

Charu C. Aggarwal, and ChengXiang Zhai. Mining text data. Springer Science & Business Media, 2012.

Chin-Yew Lin, "Rouge: A package for automatic evaluation of summaries. " Text summarization branches out: Proceedings of the ACL-04 workshop. Vol. 8. 2004.

Chris Buckley et al. "Automatic query expansion using SMART: TREC 3."NIST special publication sp (1995): 69-69.

David Martin Powers. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." (2011).

G Erkan and Dragomir R. Radev, "LexRank: Graph-based Centrality as Salience in Text Summarization", Journal of Artificial Intelligence Research, Re-search, Vol. 22, pp. 457-479 2004.

George Giannakopoulos, Mahmoud El-Haj, Benoit Favre, Marina Litvak, Josef Steinberger, & Vasudeva Varma. (2011). TAC2011 MultiLing Pilot Overview. TAC 2011 Workshop. Presented at the TAC 2011, Gaithersburg, MD, U.S.A.

H. Luhn, "The automatic creation of literature abstracts. IBM Journal of Research and Development", 2(2), 139–145. The article is also included in H. P. Luhn: Pioneer ofInformation Science, 1958.

John Makhoul et al. "Performance measures for information extraction."Proceedings of DARPA broadcast news workshop. 1999.

Namita Mittal, et al. "Extractive Text Summarization." (2014).

R. Alguliev, and R. Aliguliyev. "Experimental investigating the F- measure as similarity measure for automatic text summarization." Applied and Computational Mathematics 6.2 (2007): 278-287.

Vishal Gupta, and Gurpreet Singh Lehal. "A survey of text summarization extractive techniques." Journal of emerging technologies in web intelligence2.3 (2010): 258-268.

Yihong Gong and Xin Liu. "Generic text summarization using relevance measure and latent semantic analysis." Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001.

Yongwon Conrad Cho and Sunwhie Hwang. "Future Trends in Spatial Information Management: Suggestion to New Generation (Internet of Free-Open)." International Journal of Signal Processing Systems 3.1 (2015): 75-81.