Evaluating Information Coverage in Machine Generated Summary and Variable Length Documents

Niraj Kumar, Kannan Srinathan, Vasudeva Varma

IIIT Hyderabad, Hyderabad, India
Niraj_kumar@research.iiit.ac.in, srinathan@iiit.ac.in, vv@iiit.ac.in

Abstract

In this paper we present an automatic technique to evaluate the information coverage in machine generated summary and other variable length documents. We believe that, most of the documents (either human written summary or machine generated summary or model document) may contain more than one topic. Even the coverage and importance of topics may be different. Based on these facts, our devised system concentrates on three important issues for evaluation purpose: (1) how many topics are covered in modal document(s), (2) what are their importances, and (3) what percentage of information covered in test document(s) w.r.t. every identified topic of model document.

We introduce: (1) community detection based approach for automatic topic identification from model document, (2) a weighting scheme to identify the coverage strength of identified topics, and (3) a unique mapping based evaluation scheme, to evaluate the information coverage in test document w.r.t. given model document. We evaluate our system on (1) DUC 2005 dataset and (2) variable length documents. The experimental results show that our devised system performs better than the state-of-the-art systems of this area and also effective with variable length texts (i.e. when there is a significant variation in length of model or benchmark document and test document).

1. Introduction

Evaluation of machine generated summary and related areas have been strongly focused by TAC (Text analysis conference) and previously DUC (Document understanding conference). TAC uses two evaluation strategies to evaluate the machine generated summary, i.e. (1) manual evaluation: performed by human judges and (2) automated metrics.

Despite of the development of a lot of techniques, the human evaluation is still considered as super benchmark. The major part of this evaluation includes calculation of responsiveness.

Responsiveness: NIST assessors assigned a raw responsiveness score to each of the automatic and human summaries. The score is a coarse ranking of the summaries for each topic, according to the amount of information in the summary that helps to satisfy the information need expressed in the topic. The score was an integer between 1 and 5, with 1 being least responsive and 5 being most responsive.

1.1 Current Trends and Techniques

Human evaluation for text summarization is time consuming, costly, and prone to human variability [3]; [4]. Thus, the importance of Automatic evaluation of text summaries increases.

Current state-of-the-art techniques such as manual pyramid scores [1] or automatic ROUGE metric (considers lexical n-grams as the unit for comparing the overlap between summaries [2]) use multiple human summaries as reference. It is desirable that evaluation of similar quality can be done quickly and cheaply by using less number or single model document.

[5], [6] proposed basic elements based methods (BE), it facilitates matching of expressive variants of syntactically well-formed units called Basic Elements (BEs). The ROUGE/BE toolkit has become the standard automatic method for evaluating the content of machine-generated summaries, but the correlation of these automatic scores with human evaluation metrics has not always been consistent and tested only for fixed length human and machine generated summaries.

Donaway [12] proposed using sentence-rank-based and content-based measures for evaluating extract summaries, and compared these with recall-based evaluation measures.

International Conference on Management of Data
COMAD 2010, Nagpur, India, December 8--10, 2010

1.2 Motivating Factors

Most of the techniques of this field considers: (1) Co-occurrence of N-grams, or (2) overlapping of sequences, or (3) similarity at the level of sentences etc. These are generally tested for same length human written summary and machine generated summary.

On the contrary, we believe that (1) every word of a document has different importance, so we cannot provide equal weight to every co-occurring words or N-grams or sequence in evaluation process, similarly (2) a document may have more than one topic, (3) the importance and coverage strength of the topics of same document may be different and (4) we cannot provide equal weight to every sentence of document. Based these facts, we concentrated our attention towards the calculation of (1) roll / importance of words, (2) importance of sentences and (3) number and importance of topics in given document. We finally deploy all these facts in calculation of information coverage in test document(s) w.r.t. given model document(s).

Roll of Words: Roll of words is actually weight of words in document, generally, keyphrase extraction algorithms use such concepts, to extract terms which either have a good coverage of the document or are able to represent the document's theme. We use this concept and prepare a proper scheme to calculate the weight of word in document We deploy the features like: (1) Frequency, (2) Position of word in sentence, (3) position of sentence in which given word exist and (4) length of sentence etc. to calculate the importance of word in given model document.

Importance of sentences: Other issue is, all the above discussed techniques do not give any weightage to the role of sentences in document or human written modal summary.

The pyramid method [1] collects the sentences in base of pyramid, which is used by maximum number of human written summary. Some time, we use a few pre defined constraints to select useful sentences for evaluation, but all these efforts have a limited scope.

In general, we cannot provide same importance to every sentence of document. The importance of sentences in same document depends upon a lot of factors (including but not limited to) (1) Information contents (i.e. weight or Importance of words exist in that sentence) and (2) order or position of sentences in document etc.

Number and importance of topics: As the number and strength of topics in any document may vary, so we introduce community detection based approach to automatically identify the sentence communities in document, which represent the topics covered in document. We extended the weighting scheme applied to calculate the weight of words and sentences to calculate the weight or information coverage of every identified topic.

Based on above discussed facts, we developed a new Automatic evaluation technique to evaluate the machine generated summary and information coverage of variable length documents. The devised system, first of all identifies the number of topics covered in given human written summary. For this it uses community detection scheme [9], [10] and identify all sentence communities in given human generated summary. Next, it calculates the importance (i.e. weightage importance) of all identified topics and then uniquely maps the most matching sentences from machine generated summary to these identified topics. Later it calculates how much information related to every identified topic exists in the most uniquely matching sentences from machine generated summary. Thus, it evaluates not only the topics covered in machine generated summary, but also able to evaluate the strength of information coverage in machine generated summary, related to each identified topics in human written summary. To evaluate the information coverage strength of variable length document, we use benchmark document in the place of human written summary and apply the above discussed process for other documents of same topic. From experimental results, it is clear that, our devised system is equally effective with variable length documents.

1.3 A simple demonstration of our Scheme

To demonstrate our system, we have taken a human written model summary and a machine generated summary, each 100 words of length. The details of document source and documents are given in Table-1.

Table-1: Baseline and Machine generated Summary

Model Summary Doc_Id: D14, DUC 2001 (100 words): Aircraft crashes between June 1988 and October 1990 destroyed planes of the United States Air Force, Navy and Marines. Individual crashes occurred on three continents, over oceans and on an aircraft carrier. No accident was caused by enemy action, but several occurred when activity was at a high level in preparation for combat. Attack aircraft, fighters, helicopters, a bomber, a transport, and a trainer were lost. Some accidents claimed no lives; one caused 13 deaths. Some aircraft fell in isolated areas; one tore through a heavily populated area.
Machine Generated Summary, Doc_Id: D14cb, DUC 2001 (100 words): US Military aircraft vary in purpose, size, speed, technology and age. They operate from a multitude of airbases around the world and aircraft carriers at sea. Flying them can be dangerous, even fatal. Aircraft crashes are not uncommon. Some are reported each year from throughout the world. Fortuitous is the crash that occurs at sea after the pilot has bailed out. No fatalities, injuries or damage other than the loss of an aircraft. More problematic is the accident in a densely populated area involving many fatalities and injuries. There is public outcry and mourning. Then the flights continue.

In Table 2, we present a simple demonstration of our scheme. For the given Model summary, our system identifies three Topics (these topics are actually sentence communities, see sec-2.3 for community detection scheme applied) and then calculate the % importance of every community in given model summary (see "Community Weight" in table 2, for detail process see sec

2.5, "Calculating Weighted Importance of Communities"). Next, the system prepares the unique mapping of the most similar sentence(s) from machine generated summary to sentence community(s) (see sec 2.5, "preparing evaluation set"). Finally, it calculates the % "weighted match" and % "score" (see sec 2.6) for every topic and then, add all such %scores.

Table-2: Demonstrated Evaluation of our devised system

TID	**Model Summary Topics (i.e. Sentence communities Identified by our devised system) & Their % weight in entire Text.**	**Unique Mapping of the most similar sentence(s) from machine generated summary to sentence community(s) (i.e. topic(s)) and calculating Topic wise percentage Score**
1.	Aircraft crashes between June 1988 and October 1990 destroyed planes of the United States Air Force, Navy and Marines. Attack aircraft, fighters, helicopters, a bomber, a transport, and a trainer were lost. **Community Weight: 49.84%**	U.S Military aircraft vary in purpose, size, speed, technology and age. Aircraft crashes are not uncommon. Some are reported each year from throughout the world. No fatalities, injuries or damage other than the loss of an aircraft. **Percent Match: 31.82%** **SCORE: 15.89%**
2.	Answer: Individual crashes occurred on three continents, over oceans and on an aircraft carrier. No accident was caused by enemy action, but several occurred when activity was at a high level in preparation for combat. Some aircraft fell in isolated areas; one tore through a heavily populated area. **Community Weight: 43.24%**	Test: They operate from a multitude of airbases around the world and aircraft carriers at sea. Flying them can be dangerous, even fatal. Fortuitous is the crash that occurs at sea after the pilot has bailed out. More problematic is the accident in a densely populated area involving many fatalities and injuries. There is public outcry and mourning. Then the flights continue. **Percent Match: 54.17%** **SCORE: 23.82%**
3.	Answer: Some accidents claimed no lives; one caused 13 deaths. **Community Weight: 6.92%**	No matching Sentence Found. **Percent Match: 0.0** **SCORE: 0.0**
Average score generated by Human Evaluator (DUC 2001, Doc. Set ID: d14): 1 (i.e. 40.00% on % scale on range of values from 0 to 4) , **Our System Generated Score:** 39.71%		

Thus our system concentrates on three important facts: (1) how many topics are covered in model summary, (2) what are their importance, (3) what percentage of information covered in machine generated summary w.r.t. corresponding identified topic. To remove the chances of repeated evaluation of same information content by different topics, we apply a unique mapping scheme. In this scheme we uniquely map the most similar sentence(s) from machine generated summary to sentence community.

In this scheme, we do not depend only on frequency of matching words. Instead of this, we calculate the weighted importance of every word given in model summary, and then calculate the importance of every sentence. Finally, we calculate the weighted importance of every identified sentence community.

2. Framework and Algorithm

2.1 Input Cleaning and Pre-processing

Input cleaning task includes: (1) removal of unnecessary symbols, (2) stemming and (3) sentence filtration. To stem the document we have used Porter Stemmer [13].

2.2 Calculating Weight

At this phase we calculate the weight of every distinct words of given reference summary or model document. The weight calculation scheme depends upon the following features.

Frequency: It is the most widely used feature, but several times, the direct dependency on frequency can misguide us; as some noisy word may have very high frequency, or some useful word may have low frequency. So, instead of applying the direct occurrence frequency of any word, we have decided to collect the information content of that word in a given document. In order to achieve this, we apply the concept of entropy and calculate the information content of word in document. In this case the weight does not depend directly on frequency, but depend on the probability of word and thus reduce the chances of giving more weightage to highly frequent words. The scheme is given below:

$$W_1 = \frac{F}{N}\log_2\left(\frac{F}{N}\right) \qquad \text{-- (1)}$$

Where:

W_1 = Entropy of word in given document.

F = Occurrence frequency of word in document

N = Total Number of words in document.

This scheme is different from the technique used in [8], and considers all words in the given document.

Position of sentence in document in which the given distinct word exists: Here, we utilize the well known fact that the word which comes earlier is more important [11]. To convert this fact in weighting, we use the sentence index in which the given distinct word appears first and total number of sentences in given document. The calculation scheme is given as:

$$W_2 = \left(\frac{S_{total}+1}{S_f+1}\right) \qquad \text{-- (2)}$$

Where

W_2 =weight of given word, due to index position of the sentence in which the given word occurs first.

S_{total} =Total number of sentences in given document.

S_f =Sentence Index in which the given word occurs first.

The value of this ratio will be high, if the sentence index position i.e. S_f will be less.

Position of distinct word in sentence and Length of sentence: These two features are very important and affect each other. After a lot of observations, we got the following important information.

Position related Strength: It is well known fact that a word, which comes earlier in sentence i.e. near to subject position, contains relatively more information, similarly, word which comes at the object / "near-to-end" position is also effective, but not as effective as words that come at the subject/"at-starting" position. After a lot of observation, we formulated the position related strength of words by following way:

Here, we calculate position related strength of candidate Words which depends upon the index position of a given distinct word in the sentence. We use the following condition to calculate the position related strength of every distinct word that exists in the sentences of a given document.

Let,

$I(K)$= Index position of Candidate Word 'K' in given sentence 'S'.

$L(S)$= Length of sentence 'S' in which the candidate Word 'K' is present. This can be calculated by finding count of the number of words in 'S'.

Position related strength of given distinct word K in sentence 'S' can be represented by:

$$P(K)=\begin{cases} I(K) & if\left(I(K)<(L(S)/2)\right) \\ 2\times(L(S)-I(K)) & otherwise \end{cases} \quad \text{-- (3)}$$

Where:

$P(K)$= Position related strength of given distinct word 'K' in sentence 'S'.

This means that if the given distinct word's index position lies in the first half of the sentence length then we consider its index position as its position related strength. Otherwise, we calculate its importance by using the condition given in above Eq.

Strength due to combined effect of length of sentence and position related strength: Length of sentence is also a deciding factor. For example: if a distinct word exists at same index position at two different sentences and length of both sentences varies, than their importance in both sentences will also vary. Generally with the increase in length of sentence the importance of given distinct word which, exist at subject or object position increases. So we combine the both features in calculation of weight of word.

Here, we calculate the ratio of length of the sentence and position related strength of given distinct word and then calculate the sum of all such ratios for the given distinct word in a given model document or human written summary. This is an important feature and also takes into account the length of sentences in position related strength of given distinct word to make the calculation. The overall calculation scheme is given below:

$$W_3=\log_2\left(\Sigma\left(\frac{L(S)+1}{P(K)+1}\right)\right) \quad \text{-- (4)}$$

Where:

W_3 = Weight value of given distinct word, calculated by using position related strength of word in sentence and length of sentence in which it exist.

$L(S)$= Length of sentence 'S' in which the given distinct word 'K' is present. This can be calculated by finding count of the number of words in 'S'.

$P(K)$= Position related strength of given distinct word 'K' in sentence 'S'.

Description: The value of this scheme depends on the ratio $((L(S)+1)/(P(K)+1))$ i.e. depends on both, length of sentence and index position related strength of the given distinct word in that sentence. It achieves our motivation behind using this scheme because

(1) If the length of sentence increases w.r.t. the index position related strength of given distinct word, then its importance increases.

(2) If a given distinct word comes early in the sentence and hence, its index position related strength is less, then its importance increases.

(3) Similarly, if a given distinct word comes close to the end of the sentence, then according to the scheme given in above equation, the value of its position related strength will be less and hence, its importance increases, and so on.

Final Weight calculation scheme: To calculate the weight of every distinct word in given document, we deploy all the above discussed features, i.e. (1) Frequency, (2) Position of sentence in document in which the given distinct word exists and (3) Combined effect of position of given distinct word in sentence and Length of sentence. The overall scheme to calculate the weight is:

$$W(K)=(W_1\times W_2\times W_3) \quad \text{-- (5)}$$

Where

$W(K)$= weight of distinct word 'K' in given document.

For W_1, W_2 and W_3 refer to equation (1), (2) and (4) respectively.

2.3 Sentence Community Detection

We take pre-processed document and filter the sentences. Next, we prepare the graph of sentences by treating every sentence as node of the graph. An undirected graph of sentences is created, in which the connection between any

two nodes or sentences, depends upon the occurrence of common words between them. The weight of edge of this graph is calculated using the following scheme:

$$W(E_{(S1,S2)}) = \frac{1}{count\_common\_word(S1,S2)} \quad \text{-- (6)}$$

Where

$W(E_{(S1,S2)})$=weight of edges between sentences S1 and S2

$count\_common\_word(S1,S2)$=count of common words between sentences, S1 and S2.

Finally, we apply the shortest path betweenness strategy, as applied in [9]; [10] to calculate the sentence community. We use the faster version of community detection algorithm [9] which is optimized for large networks. This algorithm iteratively removes edges from the network to split it into communities. The edges removed being identified using graph theoretic measure of edge betweenness. The edge betweenness can be defined as the number of shortest paths between vertex pairs that go along an edge. To estimate the goodness of the certain graph partition, the authors of [9] propose the notion of modularity. Modularity [9] is a network‟ s property which refers to a specific proposed division of that network into communities. It measures whether the division is a good one, in the sense that there are many edges within communities and only a few between them.

2.4 Calculating Weighted Importance of Communities

After step 2.3, we have sentence communities for given reference summary or model document. These sentence communities are referred as topics covered in document. Now the issue is to calculate the weighted importance of every identified Topic.

To calculate the weighted importance of any topic or sentence community we depend on the Sum of weighted importance of all words in the given sentence community. The calculation of weighted importance of any community can be given as:

$$W(C) = \sum W_{wd} \quad \text{-- (7)}$$

Where

$W(C)$= weight of given community ‘C’

$\sum W_{wd}$ =weight of all words in given community. (see sec 2.2 for calculation of weight of words).

Next, we calculate the percentage of weighted information of every community in all identified community. The percentage weighted importance of any identified sentence community can be calculated as:

$$\%W(C) = \left(\frac{W(C)}{\sum W(C)} \times 100\right) \quad \text{-- (8)}$$

Where:

$\%W(C)$=percentage weight of given community ‘C’.

$\sum W(C)$=sum of weighted importance of all identified communities.

$W(C)$= weight of given community ‘C’

2.5 Preparing Evaluation Sets

At this stage, we uniquely map the sentences from machine generated summary to sentence communities. For this, we consider only the most matching sentences from machine generated summary. This mapping may be one-to-one, one-to-many. This depends upon the topics covered in the machine generated summary and human written summary. Thus finally each evaluation set contains an identified topic (i.e. sentence community) and uniquely mapped set of sentences from machine generated summary).

For a basic demonstrative example of evaluation set, see Table-2, TID=1. This topic contains two sentences from model summary and four uniquely mapped sentences from machine generated summary (an example of “one-to-many” mapping, i.e. for one sentence community more than one uniquely mapped sentences exists).

From section 2.4 we already have percentage weighted importance of every identified community. Thus evaluation set created at this step, with all these information, help us in identifying number and strength of topics covered in machine generated summary w.r.t. model summary.

2.6 Evaluation Scheme

The main aim of this evaluation set is to calculate the strength of information coverage by machine generated summary w.r.t. corresponding identified topics. We apply this scheme to convert this coverage strength into score.

At this step, we take every evaluation set one by one and check, if it contains uniquely mapped sentence(s) from machine generated summary then we calculate the matching score for every such set. For this, first of all we calculate the weighted score for matching words in both i.e. sentence community of model summary and uniquely mapped sentences from machine generated summary. For non-matching words, we check if any non matched word of mapped sentences is synonyms of any existing word in given sentence community (or topic). If such match occurs then we consider this also as matching entry. To check for synonym match, we depend upon oxford dictionary synonym list. Now we apply following formula to calculate the weighted score in any given evaluation set S_i.

$$Score(S_i) = \left(\frac{\sum Count_{match}(word)}{\sum Count(word)} \times 100\right) \times \left(\frac{\%W(C)}{100}\right)$$

This implies:

$$Score(S_i) = \left(\frac{\sum Count_{match}(word)}{\sum Count(word)} \right) \times (\%W(C)) \quad \text{-- (9)}$$

Where:

$Score(S_i)$ = Evaluation score obtained at set S_i. This is a percentage score.

$\sum Count_{match}(word)$=count of all such words in sentence community, which co-occur in both i.e. Sentence community (topic) and uniquely mapped sentence(s) from machine generated summary. As described earlier, we use synonym list to broaden our vision of matching entries.

$\sum Count(word)$ = Count of all words in given sentence community.

Note: In any given evaluation set, if there does not exist any mapped sentences for given sentence community, then we set the evaluation score of that set to zero. i.e.

$$Score(S_i) = 0; \quad \text{-- (10)}$$

Calculating Final Score: For this we just add the score of all evaluation sets. This can be given as:

$$Final\_Score = \sum_{i=1}^{k} Score(S_i) \quad \text{-- (11)}$$

Where:

$Final\_Score$=sum of percentage scores obtained from all evaluation sets.

K = Denote the total number of evaluation sets.

3. Pseudocode

The pseudo code for entire system can be given as:

Input: CASE 1: (1) human written / model summary, (2) machine generated summary, both in ASCII format.

CASE2: To check the information coverage and flow of information in variable length documents, we use "Wikipedia" document of given topic as model or baseline document and documents obtained from other source as test document.

Output: %score

Algorithm:

1. Apply pre-processing and input cleaning for both i.e. model summary and machine generated summary.
2. Calculate the weight of every word of model summary. (see step 2.2).
3. Identify the sentence community(s) in model document (also addressed as topic(s); see sec-2.3).
4. Calculate the weighted importance of every identified sentence community (see sec-2.4).
5. Prepare separate evaluation set for every identified sentence community of model summary by uniquely mapping the sentences from machine generated summary (see sec 2.5).
6. Use all Evaluation sets and apply evaluation scheme to generate the final score (see sec-2.6).

Evaluation of Variable Length Documents: to evaluate the variable length documents we apply the same algorithm as discussed above. The only difference is: We use Wikipedia documents as model documents in place of model summary and document from other source in the place of machine generated summary. The output of this system is %score, which represent the information coverage and flow of information w.r.t. Wikipedia document.

4. Experiments with DUC dataset

4.1 Dataset Used

To check the effectiveness of our measures, we use DUC 2005 dataset (which is one of the mostly used dataset in the experiments related to "automatic evaluation of summaries"). This dataset contains summary for 50 DUC 2005 topics, written by Ten NIST assessors. It contains, 30 of the topics each has 4 human summaries; the remaining 20 topics each has either 9 or 10 human summaries.

4.2 Evaluation criteria

Similar to NIST evaluation criteria, we used the two step evaluation process. This evaluation process includes calculation of correlation with (1) average of scaled responsiveness score generated by human evaluator and (2) macro-averaged scores of ROUGE2 and ROUGE-SU4 recall; computed by NIST.

Now, to compute the correlation with these scores, we compute the Spearman coefficient and Pearson coefficient between our devised system generated scores and average scaled responsiveness scores (see above). Spearman coefficient is a nonparametric test, which assesses the strength of the associations between two variables. Higher Spearman coefficient suggests higher correlation. Pearson coefficient is a parametric test, which measures the tendency of both variables to increase or decrease together. Higher Pearson coefficient indicates higher linear correlation.

Since ROUGE scores depend on the number of human summaries, so we used the macro-average for ROUGE scores (i.e. ROUGE-2 recall and ROUGE-SU4 recall). Similar to baseline applied in DUC 2005 stemming option is used but no stopwords are removed.

4.3 Results Generated by our system

As our system does not depend on number of human summaries, because, it uses single human written summary for evaluation of machine generated summary. So we evaluate the machine generated summary by using every human written summary and calculate the average score. Our system generates score in percentage (%). We use this score and calculate the correlation with average of scaled responsiveness score generated by human

evaluator and (2) macro-averaged scores of ROUGE2 and ROUGE-SU4 recall; computed by NIST. For this, we compute the Spearman coefficient and Pearson coefficient between our devised system's generated scores and average scaled responsiveness scores discussed above.

4.4 Correlation with Human Evaluations

Table 3 shows the Spearman correlation and Pearson correlation of ROUGE our devised systems score vs. human judgments for the DUC 2005 multi-document summarization tasks. Higher scores are represented as bold font. From the results given in table 3, it is clear that our system shows higher correlation with human judgements and comparable with ROUGE scores.

Table 3: Spearman correlation and Pearson correlation of ROUGE and our devised system vs. human judgments for the DUC 2005 multi-document summarization tasks

Metric	Spearman coefficient	Pearson coefficient
ROUGE-2	0.901	0.928
ROUGE-SU4	0.872	0.919
OUR SYSTEM	**0.921**	**0.934**

4.5 Correlation with ROUGE

Table-4 shows the Spearman correlation and Pearson correlation of our devised system's score vs. macro-averaged ROUGE for the DUC 2005 multi-document summarization tasks. From the results given in table 4, it is clear that our system shows high correlation with ROUGE-2 and ROUGE-SU4.

Table 4: Spearman correlation and Pearson correlation of Our Devised system vs. macro averaged ROUGE for the DUC 2005 multi-document summarization tasks

Metric	Spearman coefficient	Pearson coefficient
ROUGE-2	0.932	0.943
ROUGE-SU4	0.971	0.977

5. Experiments with Variable Length Texts

In this experiment we used Wikipedia based topics discussion as model answer and evaluated the texts with similar topic obtained from different source (considered as test document). Here the length of both documents i.e. model document (i.e. Wikipedia document) and test document (document related to similar topic and from different source) may vary. The main aim of this scheme is to evaluate the information content in variable length text documents i.e. test document w.r.to corresponding Wikipedia document. The details of dataset, evaluation metrics and results are given below.

5.1 Details of dataset

We randomly downloaded total 26 Wikipedia articles related to different topics. We used these Wikipedia article as model answer and then one article related to each topic from different source and considered it as test document. The details are given in Table-5, this table contain name of the topic and link for that article. We considered Wikipedia article of same name as model answer.

5.2 Evaluation Strategies:

In the evaluation process, we have used two human evaluators; every one independently evaluated the articles given in Table-6, w.r.t. corresponding Wikipedia article. The main focus of evaluation was to check the (1) information content and (2) flow of information in every article w.r.t. corresponding Wikipedia article. The evaluation score was an integer between 1 and 5, with 1 being the least in information content and flow of information and 5 being the most informative and best flow of information w.r.t. corresponding Wikipedia article. We calculated the average of scores obtained by both human evaluators and converted it into percentage. In Table-6, we present the score obtained from our devised system and average of scores obtained by human evaluators. All results are in percentage. As there is only one model document (i.e. Wikipedia document), so we did not use the ROUGE evaluation, because for effective evaluation, ROUGE uses (1) more than one baseline or model answer and (2) even it is not tested for documents, where there is a huge variation in length occurs.

5.3 Experimental Result

From the experimental results, given in Table-6 it is clear that our system generated score is near to the human evaluation score. The Pearson correlation coefficient between the human evaluation score and our system generated score is "0.97". This shows the effectiveness of our system in evaluation of variable length documents. The less number of experiments (i.e. only 26 different topics are used) may be a weakness of this experimental setup. Actually generating evaluation score by reading and analysing every document is very tough task and this is the main reason for less number of experimental topics. To overcome this issue, we tried to randomly select the documents from different domains (see Table-5).

Table 5: Name and source of documents that are evaluated against Wikipedia document of same title

(1) Bacteriophage :http://pathmicro.med.sc.edu/mayer /phage.htm (2) Blue Whel : http://animals.nationalgeographic.com /animals/ mammals/blue-whale.html, (3) Chinese Civil War: http://www. globalsecurity. org/ military/ops/chinese-civil-war.htm, (4) Cloud computing : http://www.salesforce.com/ cloudcomputing/, (5) Congo River :http://rainforests.mongabay.com /congo/congo_river.html (6) Cyclone :http://cyclone.thelanguage.org/wiki/ Introduction% 20to% 20Cyclone, (7) Dengu Fever :http://denguefeverinformation.com/ (8) Ganga :http://www.indianetzone.com/2/ganga_river.htm, (9) Great Pyramid of Giza: http://www.gizapyramid.com /overview.htm, (10)

Greenhouse effect: http://www.physicalgeography.net/ fundamentals / 7h.html, (11) MAC_OSX: http://osxbook.com/book/bonus /ancient/ whatismacosx/, (12) Malaria: http://www.medicinenet.com/malaria /discussion-182.htm, (13) Milky Way: http://seds.org/messier/more /mw.html, (14) Nile: http://www.touregypt.net/egypt-info/ magazine-mag05012001-magf4a.htm, (15) Nuclear reactor technology: http:// www. world-nuclear.org / info/inf32.html, (16) Russian Revolution 1917: http://www.st-petersburg-life.com / st-petersburg/1917-russian-revolution, (17) Social network: http://www.whatissocialnetworking .com/, (18) solar eclipse :http://www.colorsofindia.com/ eclipse/ whatsolar.htm, (19) Solar energy :http://edugreen.teri.res.in /explore /renew/solar.htm, (20) Taj Mahal: http://www.angelfire.com/in/ myindia/tajmahal.html, (21) Tiger Shark: http://animals. nationalgeographic.com/animals/fish/tiger-shark.html, (22) Tigerfish : http://members.mweb.co.zw/fish/species/ tiger.htm, (23) Twitter : http: //twitter.com/about, (24) Virus Structure :http://micro.magnet.fsu.edu /cells/ virus.html, (25) Windows 7 :http://www.frankps.net/2009/01/ introduction-to-windows-7/, (26) Yamuna :http://www.indianetzone. com/2/ yamuna_river.htm

Table 6: Our Devised system generated score and Human assigned score (all score in percentage)

Document Topic	Our System generated score	Human assigned score average
1. Bacteriophage	40.75	40
2. Blue Whel	15.84	20
3. Chinese Civil War	30.09	30
4. Cloud computing	22.39	20
5. Congo River	28.74	30
6. Cyclone	17.24	20
7. Dengu Fever	29.69	30
8. Ganga	32.81	30
9. Great Pyramid of Giza	40.02	40
10. Greenhouse effect	49.51	50
11. MAC OSX	07.63	10
12. Malaria	15.69	20
13. Milky Way	21.44	20
14. Nile	24.66	20
15. Nuclear reactor technology	12.40	10
16. Russian Revolution 1917	28.69	30
17. Social network	12.11	10
18. solar eclipse	10.73	10
19. Solar energy	08.22	10
20. Taj Mahal	12.20	10
21. Tiger Shark	9.89	10
22. Tigerfish	48.12	50
23. Twitter	21.23	20
24. Virus Structure	30.21	30
25. Windows 7	22.01	20
26. Yamuna	22.44	20
Pearson Correlation: 0.9826		

6. Conclusion and Future Works

In this paper we presented an automatic evaluation system for both (1) text documents summaries and (2) variable length text documents. From the experimental results, it is clear that system is effective in both cases i.e. (1) in the case of text document summaries; it is comparable with ROUGE system and its score comparatively nearer to human evaluation score. (2) In the variable length documents its score is more near to human evaluation score and shows a high correlation with human evaluation score.

Information gap is major problem while evaluating text documents from different sources or machine generated summaries. It will be interesting to extend and test this system by using N-grams and Wikipedia based Information gap reduction method as used in [14].

7. Reference

[1] Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. ACM Trans. Speech Lang. Process., 4(2):4.

[2] Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurance statistics. In Proceedings of HLT-NAACL 2003.

[3] Teufel, S. and H. van Halteren. 2004. Evaluating Information Content by Factoid Analysis: Human Annotation and Stability. Proceedings of the NLP 2004 conference. Barcelona, Spain.

[4] Nenkova, A. and R. Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. Proceedings of the HLT-NAACL 2004 conference.

[5] Hovy, E.H., C.Y. Lin, and L. Zhou. 2005. Evaluating DUC 2005 using Basic Elements. Proceedings of DUC-2005 workshop.

[6] Hovy, E.H., C.Y. Lin, L. Zhou, and J. Fukumoto.2006. Automated Summarization Evaluation with Basic Elements. Full paper. Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 06). Genoa, Italy.

[7] Conroy, J.M. and H.Trang Dang. 2008. Mind the Gap: Dangers of Divorcing Evaluations of Summary Co ntent from Linguistic Quality. Proceedings of the COLING conference. Manchester, UK.

[8] Li, S., Wang, H., Yu, S. (2004) Research on Maximum Entropy Model for Keyword Indexing. Chinese Journal of Computers 27(9), 1192–1197.

[9] Clauset, A., Newman, M. E. J., Moore, C. (2004). Finding community structure in verylarge networks. Physical Review E, 70:066111.

[10] Newman, M. E. J., Girvan, M. (2004). Finding and evaluating community structure in networks. Physical review E, 69:026113, 2004.

[11] Kumar, N., Srinathan, K. (2008). Automatic Keyphrase Extraction from Scientific Documents Using N-gram Filtration Technique. In the Proceedings of ACM DocEng 2008, ACM 978-1-60558-081-4/08/09.6.

[12] Donaway R, Drummey K, Mather L.A Comparison of Rankings Produced by Summarization Evaluation Measures. In Proceeding of ANLP/NAACL Workshop on Automatic Summarization, pages 69-78, 2000.

[13] Porter Stemming Algorithm for suffix stripping, web –link "http://telemat.det.unifi.it/book/2001 /wchange/ download/ stem_porter.html".

[14] Niraj Kumar;Venkata Vinay Babu Vemula;Kannan Srinathan;Vasudeva Varma;EXPLOITING N-GRAM IMPORTANCE AND ADDITIONAL KNOWEDGE BASED ON WIKIPEDIA FOR IMPROVEMENTS IN GAAC BASED DOCUMENT CLUSTERING;KDIR 2010.