Chapter 5

A GRAPH BASED MODEL FOR EXPLORING ROLE AND SENSE OF WORDS FOR AUTOMATIC SUMMARIZATION EVALUATION

5.1 Automatic Summarization Evaluation and its Importance

Human evaluation for text summarization is time consuming, costly, and prone to human variability (Teufel and Van Halteren, 2004); (Nenkova and Passonneau, 2004). Thus, it creates the demand of the automatic evaluation of the machine generated summary.

Evaluation of machine generated summaries has been of importance both in TAC (Text Analysis Conference) and previously DUC (Document Understanding Conference). The main goal is to produce two sets of numeric summary-level scores.

a) All Peers case: a numeric score for each peer summary, including the model summaries. The "All Peers" case is intended to focus on whether an automatic metric can differentiate between human and automatic summarizers.

b) No Models case: a numeric score for each peer summary, excluding the model summaries. The "No Models" case is intended to focus on how well an automatic metric can evaluate automatic summaries.

It is important to note that, the automatic summarization evaluation methods can be easily extended to evaluation of guided answers of variable length (Kumar et al. 2010a).

5.1.1 Problem Definition and Motivation

Exploiting topics covered in the document: It is important to note that summaries may contain more than one topic and each topic may have different levels of importance. The use of the variability in the granularity of the analysis of the summary for summary evaluation by (Nenkova et al. 2007), pyramid method also supports our view. (Kumar et al. 2010a), (Kumar et al. 2010b) also use the topics and their importance in summarization evaluation. But, here our main aim is to limit the use of linguistic resources (i.e., limited to, include only stopwords stemmers and punctuation marks) with effective improvement in quality of result. We use the group average agglomerative clustering (GAAC) to cluster the sentences of document and page rank score of words on reverse directed word graph of sentences, to calculate the
importance of the identified clusters. The page rank score on reverse directed word graph of sentences effectively capture our writing strategies as, we describe the terms after writing it.

**The importance of the role and sense of co-occurring words.** The co-occurring words or sequences may have a different role/sense in a model and machine generated summary. Thus, neglecting the sense of co-occurring words/sequences may misguide the evaluation scheme. It will clear from following reference sentences, taken from (Lin and Hovy, 2003).

- S1. Police killed the gunman.
- S2. Police kill the gunman.
- S3. The gunman kill the police.
- S4. The gunman kill police.
- S5. Gunman the killed police.

In these five sentences more than one word is common, but their roles are not always same. Now, let us analyze the problems:

I. Suppose we take S1 as the reference and S2 and S3 as candidate summary sentences, then ROUGE-2 (Lin and Hovy, 2003), gives same score to S2 and S3. This is just because, both sentences have a common bigram “the gunman”. However, S2 and S3 have very different meaning.

II. To solve this (discussed above) problem ROUGE-S (Skip-Bigram co-occurrence statics) is used by (Lin and Hovy, 2003). But, the potential problem with ROUGE-S is that, it does not give any credit to candidate sentences, if the sentence does not have any word pair co-occurring with its reference, i.e., it cannot properly handle the candidate sentence (S5). To solve this problem an extension of ROUGE-S is proposed (i.e. ROUGE-SU) by the addition of unigram as counting unit. Now, ROUGE-2 and ROUGE-SU4 is used as a benchmark (By TAC “Text Analysis Conference”) in automatic evaluation of the machine generated summary. But, we believe that there should be a single metric to handle all such issues.

**Information loss due to phrase length related restriction:** as, discussed earlier, the predefined phrase patterns creates problem of information loss due to phrase length related restrictions. For example, ROUGE-L (which uses LCS “Longest Common Sub-Sequence”), suffers with this disadvantage. It only counts main in-sequence words, therefore, other LCSes and shorter sequences are not reflected in the final score. In the example, in the sentence “S4”, using “S1” as reference, LCS counts either “the gunman” or “police killed”, but not both; therefore “S4” has same ROUGE-L score as “S3”.

**Use of closeness centrality measure:** In this chapter, we introduce the use of closeness centrality measure to:
Identify the sense of co-occurring words and
Remove the chances of information loss due to fixed length sequences.

For this, we use word graph of sentences, which helps in maintaining the inter-word cohesion. Then we use closeness centrality measure, which helps in calculating the information propagation strength of words (as, words are treated as nodes in a word graph of sentences). The information propagation strength of word (i.e., closeness centrality) is a global measure w.r.t., local sequence matching. Thus, it gives a better prediction about the role of co-occurring words in every identified topic. The experimental results on TAC-2011 dataset also support our way to solve the problem. The following contains the conceptual work flow of the system.

![Conceptual workflow of the system](image)

**Figure 5.1** Conceptual workflow of the system

### 5.2 Calculating Importance of Words

The calculation of the importance of words is very important. We use the importance of words in calculation of importance of topics in the next step. But, first of all, we apply document cleaning steps. To clean any document, we remove the noisy entries and stopwords and stem the entire text by using porter
stemmer. Finally, we filter the sentences. The rest of the process to calculate the importance of words is given below:

To calculate the importance/weight of words, we prepare reverse directed word graph of sentences and then calculate the page rank score of every distinct word. The way to prepare reverse directed word graph of sentences and calculation of page rank is given below:

![Figure 5.2: reverse directed word graph of sentences.](image)

**Preparing reverse directed word graph of sentences:** For a given set of sentences, i.e. \( S = \{S_1, S_2, \ldots, S_n\} \), we add a reverse directed link for every adjacent word pair of every given sentence. See Figure-1. We denote \( G = (V, E) \) as a directed graph, Where, \( V = \{V_1, V_2, \ldots, V_n\} \) denotes the vertex set and \( V \in C \) and link set \( \{(V_j, V_i) \in E \text{ if there is a link from } V_j \text{ to } V_i\} \).

**Calculating Page Rank Score:** we use the page rank scheme defined by Page et al. (1999) to calculate the page rank score of every word. For any given vertex \( V_i \), let \( IN(V_i) \) be the set of vertices that point to it (predecessors), and let \( OUT(V_i) \) be the set of vertices that vertex \( V_i \) points to (successors). Then the page rank score of vertex \( V_i \) can be defined as:

\[
S(V_i) = \frac{(1-\lambda)}{N} + \lambda \sum_{j \in IN(V_i)} \frac{S(V_j)}{OUT(V_j)}
\]

(5.1)

Where, \( S(V_i) = \text{Rank/score of the word/vertex } V_i \), \( S(V_j) = \text{rank/score of word/vertex } V_j \), from which incoming link comes to word/ vertex \( V_i \). \( N = \text{Count of number of words/vertex in word graph of sentences.} \)

\( \lambda = \text{Damping factor (we use a fixed score for damping factor, i.e., “0.85” as used in (Mihalcea and Tarau, 2004))}. \)
5.3 Identifying Topics and Calculating their Importance

To identify the topics covered in the document we use Group average agglomerative clustering scheme (GAAC). In our case the topic is considered as a set of sentences related to the same concept. To apply the GACC on sentences we use a sentence vector representation of the entire document. Here, each row represents a sentence and each column represents a term. We use the threshold “0.4” in the entire evaluation (same as used in Section 3.5, Chapter 3).

Calculating Importance of Sentence clusters or topics: To calculate the weighted importance of any Sentence cluster or topic, we calculate the Sum of weighted importance of all words in the given sentence cluster. It can be given as:

\[ w(c) = \sum W_{wd} \]  

(5.3)

Where, \( w(c) \) = weight of given sentence cluster ‘C’, \( \sum W_{wd} \) = weight of all words in the given sentence cluster. (See equation 5.1 to calculate the weight of words).

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

\[ \%w(c) = \left( \frac{w(c)}{\sum w(c)} \times 100 \right) \]  

(5.4)

Where, \( \%w(c) \) = percentage weight of given sentence cluster ‘C’. \( \sum w(c) \) = sum of the weight of all identified sentence cluster. \( w(c) \) = weight of given sentence cluster ‘C’.

5.4 Preparing Evaluation Sets

At this stage, we prepare evaluation sets. Every evaluation set consists of two sets, i.e., (1) set-1: contains set of sentences from the identified sentence cluster (also denoted as topic) and (2) set-2: contains the uniquely matching sentences from the target or candidate document, which matches with set-1. Thus, the number of evaluation set depends on the number of the identified sentence clusters in the reference/model summary. Briefly, each evaluation set contains an identified topic (i.e. Sentence cluster from reference or model summary) and uniquely mapped set of sentences from target summary/ machine generated summary). See figure 5.3 for a sample evaluation set.
5.5 Using Closeness Centrality Measure

We use closeness centrality measure to predict the sense of co-occurring words in both sets (i.e. Set-1 and Set2) of every given evaluation set. Due to (1) topic boundary of sentences in every evaluation set (as achieved in Set-1 and Set-2 of evaluation set), and (2) global nature of closeness centrality score, the proposed system effectively predicts the sense of co-occurring words w.r.t., local measures based on sequences. Now, we remove the words which do not co-occur in both sets (i.e. Set-1 and Set-2) of the given evaluation set (See Figure 5.4, which contains co-occurring words of Set-1 and Set-2 of evaluation set given in Figure-5.3). Next, we calculate the closeness centrality score of (1) co-occurring words of Set-1 and (2) Co-occurring Words of Combined Graph of Set-1 and Set-2. The details are given below:

5.5.1 Calculating Closeness Centrality score of Co-occurring words of Set-1

We prepare a Bi-directional graph of co-occurring words of the Set-1 of given evaluation set and calculate their closeness centrality score. To prepare Bi-directional graph we add a bidirectional link for every adjacent word pair in every sentence of Set-1 (see figure-5.5).

Graph Theoretical Notation: We denote $G = (V, E)$ as a directed graph of Set-1 of the given evaluation set. Where, $V = \{v_1, v_2, ..., v_n\}$ denotes the vertex set and link set $\{v_j, v_i\} \in E$ if there is a link from $v_j$ to $v_i$.

Path Length: In this scheme we use link strength to calculate the path length between any two nodes. As, the link strength of any link between two nodes in the word graph of sentences depends upon the number of times the adjacent words co-occur in the given text. The Link Strength between two adjacent nodes can be calculated as:

$$\text{Link\_Stringht} = 2 \times \min(\# \text{Forward\_Link}, \# \text{Backward\_Link})$$

(5.5)

Where, $\# \text{Forward\_Link} = \text{count of forward links between two adjacent nodes.}$ $\# \text{Backward\_Link} = \text{count of backward links between two adjacent nodes.}$

Note: As, we consider only Bi-directional links, so we use “multiply by 2” in the above equation.

In general case, with the increase of count of co-occurrences, the similarity between words increases. We believe that with the increase of similarity between words the edge weight between words should decrease. We use this fact in the calculation of path length. Now, according to this scheme, path length can be calculated as (e.g., see figure-5.7, for path length of different paths):
Strength Link Length Path

\[
Path \_ Length = \frac{1}{Link \_ Strength}
\]  \hspace{1cm} (5.6)

**Calculating Closeness Centrality Score:** The closeness centrality of any node \( V_i \) is defined as the mean geodesic distance (i.e., the shortest path) between a node \( V_i \) and all of the nodes reachable from \( V_i \) as follows, where \( n \geq 2 \) is the size of the connected component reachable from \( V_i \).

\[
C_C(V_i) = \frac{(n-1)}{\sum_{t \in V/V_i} d_G(V_i,t)}
\]  \hspace{1cm} (5.7)

Where, \( C_C(V_i) \) = closeness centrality of node / vertex \( V_i \), \( d_G(V_i,t) \) = sum of geodesic distance from \( V_i \) to \( 't' \), we use the path length obtained from above step in the calculation of all geodesic distances (see Figure 5.6 for a sample calculation of geodesic distances and Figure 5.9, for closeness centrality scores).

**NOTE:** In some cases the path from node \( V_i \) to \( 't' \) may not exist. In such cases, we consider that word at the node \( V_i \) is not related to word at node \( 't' \). So, in such cases we consider the geodesic distance from \( V_i \) to \( 't' \) as the count of the total number of nodes in the graph.

Thus, in this case, \( d_G(V_i,t) \) = count of total number of nodes in given graph. By using this scheme, we calculate the closeness centrality of every node / word of given graph.

**Figure 5.3:** Evaluation Set Containing Set-1 and Set-2, here uppercase letters represent the words

<table>
<thead>
<tr>
<th>Evaluation Set 1</th>
<th>Evaluation Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C D</td>
<td>M N B D</td>
</tr>
<tr>
<td>B N D</td>
<td>C Q D</td>
</tr>
<tr>
<td>W X B C</td>
<td>A B C</td>
</tr>
</tbody>
</table>

**Figure 5.4:** Only common words from both sets are taken

<table>
<thead>
<tr>
<th>Common words from both sets</th>
<th>Set-1</th>
<th>Set-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Words</td>
<td>A B C D</td>
<td>N B D</td>
</tr>
<tr>
<td>Common Words</td>
<td>B N D</td>
<td>C D</td>
</tr>
<tr>
<td>B C</td>
<td>A B C</td>
<td></td>
</tr>
</tbody>
</table>
5.5.2 Calculating Closeness Centrality Score of Co-occurring Words of Combined Graph of Set-1 and Set-2

For this, we prepare a combined directed graph by using co-occurring words of Set-1 and Set-2 of the given evaluation set. Here the main aim is to exploit the differences in the information flow of adjacent common words of both sets of given evaluation set in predicting the sense of co-occurring words. The details are given below:

**Graph Construction:** We take every adjacent pair of words in every sentence of Set-1 of the given evaluation set and adds a forward directed link. Next, we take every adjacent pair of words of every
sentence of Set-2 of the given evaluation set and add a backward directed link to the existing graph. Thus the constructed graph represents the flow of information of pair of adjacent words from both sets (e.g., see Figure 5.6).

**Path Length:** Similar to scheme discussed in sub-section 5.5.1., we use link strength to calculate the path length between any two nodes. But, due to the use of combined graph, Bi-directional links may not exist for some nodes. For this, we make some change in strategy. We calculate the link strength by using two additional cases i.e.

Now, the link strength between two adjacent nodes when Bi-directional links between nodes do not exist can be calculated as:

- If only Forward link(s) exist.
  \[ \text{Link\_Strength} = \#\text{Forward\_Links} \]  

- If only backward link(s) exist.
  \[ \text{Link\_Strength} = \#\text{Backward\_Links} \]  

The rest of the process for calculation of the path length is same, as discussed in sub-section 5.5.1. (See equation-5.6). We use this path length in the calculation of geodesic distances between every pair of nodes. See Figure 5.8 for a sample calculation of pairwise geodesic distances for the graph shown in Figure 5.6.

**Calculating Closeness Centrality Score:** The process of calculating the closeness centrality is same as discussed in previous sub-section 5.5.1 (See equation 5.7). (e.g., See Figure 5.9)

### 5.5.3 Identifying Co-occurring Words having Similar Sense

At this stage, we identify the co-occurring words in set-2, having same sense w.r.t. Set-1. As, earlier it is discussed that there may be some words co-occur in both sets, i.e. Set-1 and set-2 (of evaluation set), but the role of these words in set-2 may be different w.r.t. Set-1. To check this, we use closeness centrality scores of common words of set-1 and set-2 (see sub-section 5.5.1., and 5.5.2., for calculation of the centrality score of the node / word of graphs). Next, we calculate the “%” difference in centrality scores between every co-occurring word (node) of word graph of set-1 and combined word graph of set-1 and set-2 of the given evaluation set. The scheme is given below:

\[
\%\text{Diff}(V_i) = \left( \frac{C_C(V_i) - C_C(V'_i)}{C_C(V_i)} \right) \times 100
\]  

(5.9)
Where, \( C_c(V_i) \) = Closeness centrality score of the node/word \( V_i \) in “Word graph of Set-1”. \( C_c(V'_i) \) = Closeness centrality score of the node/word \( V'_i \) in “combined word graph of Set-1 and Set-2”. Here \( V_i = V'_i \), i.e. It denotes the same word representing the node on different graphs.

For example, see Figure 5.8 (“%Diff”). Now, we calculate the median of “%Difference” of scores of all nodes / (distinct words) and put a minimum threshold as:

\[
\text{“% Diff” should be} = \begin{cases} 
<50\% & \text{if (median}<50\%) \\
<\text{median} & \text{if (median}\geq 50\%)
\end{cases}
\] (5.10)

We use this threshold value in identification of words in Set-2 (which are common to both sets of given evaluation set) and (2) whose role in Set-2 is similar to the corresponding word of Set-1. We use this information in scoring in the next section. The threshold used in equation 5.10, is fixed after a lot of observation on the TAC-2009 and TAC-2010 dataset.

**Additional Note:** If in any case all the nodes of graph shows \( %Diff(V_i) \geq 50\% \), then it means, common words of both sets show very high diversities in roles. So, we will not consider such nodes (words) of that graph as valid nodes.

**Example:** For the evaluation set given in Figure 5.3, our system predicts that only two words, i.e. ‘A’ and ‘C’ have similar roles in both sets, i.e. Set-1 and Set-2 (See Figure 5.4).

Similarly for sample sentences given in sub-section-5.1.2., our system predicts “3” valid matches when comparing S2 w.r.t. S1 (i.e. 75% score on [0-100] scale), “1” valid match when comparing S3 w.r.t. S1 (i.e. 25% score), “2” valid matches when comparing “S4”, w.r.t. “S1” (i.e. 50% score) and “No” valid matching when comparing “S5” w.r.t. “S1” (i.e. 0% score). The comparison of these scores with ROUGE scores also supports our view.

**Note:** to evaluate these sentences we applied above discussed rules and do not use stemming and stopwords removal, as used in (Lin and Hovy, 2003).
5.6. Final Scoring

At this step, we take every evaluation set one by one and check, if set-2 is not null, then we calculate the score for every such evaluation set. Now we apply following formula to calculate the weighted score in any given evaluation set $S_i$.

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

(5.11)

Where, $Score(S_i)$ = Evaluation score for the given evaluation set $S_i$. $\sum Count_{match}(word)$ = count of co-occurrences of all such words in Set-1, (1) which co-occur in both sets, i.e. Set-1 and Set-2 and (2) pass the minimum frequency threshold related criteria. As described earlier, (see sub-section 5.5.3., equation 5.9 and 5.10). $\sum Count(word)$ = Count of all words in Set-1 of the given evaluation set.

Note: In any given evaluation set, if there does not exist any mapped sentences in Set-2, then we set the evaluation score of that evaluation set to zero.

$$Score(S_i) = 0;$$

(5.12)

Calculating Final Score: For this we just add the evaluation scores of all evaluation set.

5.7 Algorithmic Steps

Input: (1) reference / model summary, (2) candidate / machine generated summary, both in ASCII format.

Output: %score, which can be further normalized to “0-1” scale.

Algorithm:

Step1. Apply pre-processing and input cleaning for reference / model summary and candidate / machine generated summary and calculate the weight of every word of reference / model summary. (See Section-5.2.).

Step2. Identify the sentence clusters in reference / model document and calculate the weighted importance of every identified sentence cluster (See Section-5.3.).

Step3. Prepare separate evaluation sets by using every identified sentence cluster of reference / model summary by uniquely mapping the sentences from candidate / machine generated summary (See Section-5.4.).
Step 4. Use closeness centrality measure to identify the co-occurring words having the same sense in both sets of given evaluation set (See Section 5.5).

Step 5. By using (1) co-occurring words having the same role in both sets of every evaluation set and (2) importance of sentence cluster (i.e., Set-1 of the given evaluation set), calculate the final evaluation score of the given machine generated summary (See Section 5.6).

5.8 Experiments

We use TAC-2011, AESOP dataset to evaluate our system. The details of the dataset, evaluation strategies, metrics, baselines and results are given below.

Evaluation Strategies, Metrics and Baselines: For automatic evaluation of summary quality, we consider two cases, i.e. (1) All peer and (2) No-Model cases with both parts, i.e., an initial summary and update summary. Thus we have total four evaluation results, i.e. (1) All-Peer evaluation with initial summary, (2) All-Peer evaluation with update summary, (3) No-Model evaluation with initial summary and (4) No-Model evaluation with update summary.

Metrics: To judge the summary quality, we calculate (a) Pearson's, (b) Spearman's, and (c) Kendall's correlations with (1) Pyramid, (2) Overall Responsiveness and (3) Readability.

Pyramid score: The Pyramid method is a novel semi-automatic evaluation method (Harnly et al. 2005). Its basic idea is to identify summarization content units (SCUs) that are used for comparison of information in summaries. The annotation starts with identifying similar sentences and then proceeds with finer grained inspection that can lead to identifying related subparts more tightly. SCUs that appear in more manual summaries will get greater weights, so a pyramid will be formed after SCU annotation of manual summaries. At the top of the pyramid there are SCUs that appear in most of the summaries and thus they have the greatest weight. The lower in the pyramid the SCU appears, the lower its weight is because it is contained in fewer summaries. The SCUs in peer summary are then compared against an existing pyramid to evaluate how much information agrees between the peer summary and manual summary.

Overall Responsiveness: In this case, the assessor gives an overall responsiveness score to each summary. The overall responsiveness score is based on both content (coverage of all required aspects) and readability/fluency\(^5\).

**Readability:** In this case the assessor gives a readability/fluency score to each summary. The score reflects the fluency and readability of the summary (independently of whether it contains any relevant information) and is based on factors such as the summary's grammaticality, non-redundancy, referential clarity, focus, and structure and coherence.

**NOTE:** Readability and Overall Responsiveness will each be judged on the following 5-point scale:
1. Very Poor
2. Poor
3. Barely Acceptable
4. Good
5. Very Good

**Baselines:** We consider TAC-2011 AESOP baseline and TAC-Best Score, for comparison purpose. The details are given below:

i. **Baseline-1:** ROUGE-2: Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows.

\[
\text{Rouge-N} = \frac{\sum_{S \in \text{ReferenceSummaries}} \sum_{\text{gram}_n \in S} \text{count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{ReferenceSummaries}} \sum_{\text{gram}_n \in S} \text{count}(\text{gram}_n)}
\]  

Where \( n \) stands for the length of the n-gram, \( \text{gram}_n \), and \( \text{count}_{\text{match}}(\text{gram}_n) \) is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. To get the ROUGE-2 evaluation score, we put ‘N’ = 2 and apply the setting which includes stemming and keeping stopwords.

ii. **Baseline-2:** ROUGE-SU4: It represents the skip-bigram with a maximum gap length of 4. Skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps. Skip-bigram co-occurrence statistics measure the overlap of skip-bigrams between a candidate translation and a set of reference translations. To get the ROUGE-SU4 evaluation score, we apply the stemming process and keeping stopwords.

iii. **Baseline-3:** Basic Elements (BE). Summaries were parsed with Minipar, and BE were extracted and matched using the Head-Modifier criterion. It applies the modules twice, in two phases: preparation and scoring. In the preparation phase, the first module breaks up the reference summaries into a list of reference BEs; the second module considers all reference BEs and merges semantically identical ones; and the third module assigns a score to each of the reference BEs. In the second (scoring) phase,
the first module breaks up the summary to be scored into a separate list of BEs; the second compares each BE to the list of reference BEs; the third assigns a score to each BE to be rated and computes the final overall score of all the BEs contained in the summary to be rated.

iv. TAC-Best Score: It contains best TAC-2011 scores on every evaluation metric by TAC-2011 participants/benchmarks.

Results on AESOP Test Dataset: Results are given in Table 5.1, 5.2, 5.3 and 5.4. (1) Table 5.1, shows All-Peer evaluation with initial summary, (2) Table 5.2 shows All-Peer evaluation with update summary, (3) Table 5.3 shows No-Model evaluation with initial summary and (4) Table 5.4 shows No-Model evaluation with update summary.

In all four tables, the first three rows headed as “Baseline-1”, “Baseline-2”, and “Baseline-3” represents the corresponding baseline scores as obtained from TAC-2011 results. “TAC-Best score” shows the best TAC-2011 scores. The last row of all four tables contains the score of our devised system.

The correlation score with (1) Pyramid, (2) Overall Responsiveness and (3) Readability given in Table 5.1, 5.2, 5.3 and 5.4, show that our devised system (1) performs better than all three baseline systems and (2) comparable with TAC-Best Scores. In all four tables highest scores are represented by bold font.

Table 5.1: AESOP-ALL Peers (Initial Summary), correlation with Pyramid, Responsiveness, and Readability

<table>
<thead>
<tr>
<th>System</th>
<th>Correlation with Pyramid</th>
<th>Correlation with Responsiveness</th>
<th>Correlation with Readability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pearson</td>
<td>Spearman</td>
<td>Kendall</td>
</tr>
<tr>
<td>Baseline-1</td>
<td>0.572</td>
<td>0.864</td>
<td>0.703</td>
</tr>
<tr>
<td></td>
<td>0.725</td>
<td>0.779</td>
<td>0.609</td>
</tr>
<tr>
<td></td>
<td>0.663</td>
<td>0.498</td>
<td>0.374</td>
</tr>
<tr>
<td>Baseline-2</td>
<td>0.763</td>
<td>0.886</td>
<td>0.723</td>
</tr>
<tr>
<td></td>
<td>0.733</td>
<td>0.810</td>
<td>0.629</td>
</tr>
<tr>
<td></td>
<td>0.682</td>
<td>0.533</td>
<td>0.400</td>
</tr>
<tr>
<td>Baseline-3</td>
<td>0.781</td>
<td>0.878</td>
<td>0.720</td>
</tr>
<tr>
<td></td>
<td>0.752</td>
<td>0.784</td>
<td>0.590</td>
</tr>
<tr>
<td></td>
<td>0.683</td>
<td>0.531</td>
<td>0.387</td>
</tr>
<tr>
<td>TAC-Best Score</td>
<td><strong>0.975</strong></td>
<td><strong>0.933</strong></td>
<td><strong>0.799</strong></td>
</tr>
<tr>
<td></td>
<td><strong>0.972</strong></td>
<td><strong>0.894</strong></td>
<td><strong>0.740</strong></td>
</tr>
<tr>
<td></td>
<td><strong>0.926</strong></td>
<td><strong>0.674</strong></td>
<td><strong>0.519</strong></td>
</tr>
<tr>
<td>OUR SYSTEM</td>
<td><strong>0.976</strong></td>
<td><strong>0.935</strong></td>
<td><strong>0.799</strong></td>
</tr>
<tr>
<td></td>
<td><strong>0.968</strong></td>
<td><strong>0.895</strong></td>
<td><strong>0.743</strong></td>
</tr>
<tr>
<td></td>
<td><strong>0.926</strong></td>
<td><strong>0.673</strong></td>
<td><strong>0.519</strong></td>
</tr>
</tbody>
</table>

Table 5.2: AESOP-ALL Peers (Update Summary), correlation with Pyramid and Responsiveness

<table>
<thead>
<tr>
<th>System</th>
<th>Correlation with Pyramid</th>
<th>Correlation with Responsiveness</th>
<th>Correlation with Readability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pearson</td>
<td>Spearman</td>
<td>Kendall</td>
</tr>
<tr>
<td>Baseline-1</td>
<td>0.775</td>
<td>0.851</td>
<td>0.684</td>
</tr>
<tr>
<td></td>
<td>0.717</td>
<td>0.869</td>
<td>0.710</td>
</tr>
<tr>
<td></td>
<td>0.712</td>
<td>0.550</td>
<td>0.399</td>
</tr>
<tr>
<td>Baseline-2</td>
<td>0.730</td>
<td>0.883</td>
<td>0.720</td>
</tr>
<tr>
<td></td>
<td>0.675</td>
<td>0.903</td>
<td>0.743</td>
</tr>
<tr>
<td></td>
<td>0.686</td>
<td>0.558</td>
<td>0.405</td>
</tr>
<tr>
<td>Baseline-3</td>
<td>0.740</td>
<td>0.848</td>
<td>0.686</td>
</tr>
<tr>
<td></td>
<td>0.649</td>
<td>0.808</td>
<td>0.637</td>
</tr>
<tr>
<td></td>
<td>0.611</td>
<td>0.415</td>
<td>0.287</td>
</tr>
<tr>
<td>TAC-Best Score</td>
<td><strong>0.953</strong></td>
<td><strong>0.891</strong></td>
<td><strong>0.731</strong></td>
</tr>
<tr>
<td></td>
<td><strong>0.975</strong></td>
<td><strong>0.911</strong></td>
<td><strong>0.762</strong></td>
</tr>
<tr>
<td></td>
<td><strong>0.934</strong></td>
<td><strong>0.663</strong></td>
<td><strong>0.507</strong></td>
</tr>
<tr>
<td>OUR SYSTEM</td>
<td><strong>0.956</strong></td>
<td><strong>0.887</strong></td>
<td><strong>0.733</strong></td>
</tr>
<tr>
<td></td>
<td><strong>0.954</strong></td>
<td><strong>0.921</strong></td>
<td><strong>0.762</strong></td>
</tr>
<tr>
<td></td>
<td><strong>0.934</strong></td>
<td><strong>0.645</strong></td>
<td><strong>0.510</strong></td>
</tr>
</tbody>
</table>
Time complexity and related issue

The running time of the devised system depends upon the running time of the following key techniques:

1. Weighted betweenness centrality (Sub-section-6.3.2): The running time of this algorithm is: \( O(n^2 \log n + nm) \); where, ‘\( n \)’ = Total number of nodes (here distinct words) in the word graph of text and ‘\( m \)’ = Total number of edges in the word graph of text.

2. Calculating normalized pointwise mutual information score (used in sub-section-6.3.4): As, most of the LUCENE based indexing is taken as offline process, so, we just concentrate on calculation of normalized pointwise mutual information score. In a traditional system having 2GB RAM and 2.2 GHz dual core processor, the devised system calculates the normalized pointwise mutual information score of 7000 distinct bigrams.

3. Using Ortiz-Arroyo’s entropy based algorithm for finding sets of key nodes in a graph (Section 5): The running time of this algorithm is: \( O(n^3) \); Where, ‘\( n \)’ = Total number of nodes (here distinct candidate keyphrases/N-grams) in the N-gram graph. However, the count of distinct candidate N-grams lie in the range of [30-40%] of total number of words of the given document.
5.9. Chapter Conclusion and Future Work

In this chapter, we presented a graph based flexible mapping approach to identify the matching word patterns of any type/length and also devised a closeness centrality based scheme, which identify the role of matching words in the entire context of information (globally). In other words, it calculates the role of matching sequences / matching words in both i.e. set of reference sentences and set of candidate sentences. We use both schemes in automatic evaluation of the machine generated summary.

The experimental results on TAC-2011, AESOP dataset shows that, our devised system performs better than TAC benchmarks and is better/comparable with TAC-Best Scores. It is remarkable to note that our devised system does not require heavy linguistic resources and truly unsupervised in nature. Thus, it can be easily extended to multi-lingual platform.

Due to flexible structure, our devised system can be extended to evaluate the short answers, text passages etc.

Techniques Applied: We exploited two basic techniques to develop this application:

- Differentiating role and sense of the words (sub section 1.1.2): we use this technique to properly evaluate the identified evaluation set.
- Integrating importance of words with text mining techniques (sub-section 1.1.4): we use this technique in calculating the importance of every identified topic in the given model summary.