Document Summarization using Central Sentences and Keywords

Girish Keshav Palshikar
Tata Research Development and Design Centre (TRDDC),
54B, Hadapsar Industrial Estate, Pune 411013, India.
email: gk.palshikar@tcs.com

Abstract
We propose two distinct approaches to single document summarization. The first approach is based on the notion that important sentences are somehow central in the document. We represent the document as a graph, with sentences as vertices and edges labeled with sentence dissimilarities. We then use graph-theoretical notions of vertex centrality (e.g., eccentricity, closeness etc.) to identify central vertices (sentences). In the second approach, we propose to use any of the well-known keyword identification algorithms to extract characteristic keywords from a document. We then use the number of keywords present in a sentence as a measure of the sentence’s importance. Finally, we propose the use of well-known Condorcet methodology to combine the summaries from several algorithms into a winner summary.

1 Introduction
The advent of fast and cheap computing, storage and communication technologies have given rise to an information explosion. The Web, digital libraries, online newspapers and magazines and enterprise-wide document repositories all make an enormous amount of information easily accessible to people. Document summary is an important method which people use to deal with this information overload. Hence automatic techniques for document summarization are receiving increasing attention.

The summary can be of two types. An extract is a summary which is a subset of sentences from the original documents. An abstract, on the other hand, contains generated sentences, not present in the original documents (e.g., to paraphrase, explain or simplify). Much work in document summarization focuses on the extract identification problem.

A document summarization task is quite involved and can be performed in several ways. In single document summarization, the focus is on creating a summary of a given isolated document. The information in the summary should be (a) complete (should cover all important topics discussed in the document); (b) correctly ordered (c) coherent and consistent (e.g., must not include any ambiguities or redundancies); and (d) understandable for humans. Typically, the length of a summary (i.e., the compression ratio) is between 10% to 20% of the total document length.

Multi-document summarization presents challenges which are quite different from those in single document summarization. First, the documents in a collection usually contain overlapping information; hence redundancy in the generated summary should be avoided. Second, the documents in a collection may contain complementary information (e.g., additional details or different perspectives); hence information cohesion and consistency should be ensured in the generated summary. As a special case, if the documents describe different and partial (but overlapping) event sequences, then the generated summary should combine them into a coherent and consistent event sequence.

The contribution of this paper is as follows. We propose two distinct approaches to single document summarization. The first approach is based on the no-
tion that important sentences are somehow central in the document. We represent the document as a graph, with sentences as vertices and edges labeled with sentence dissimilarities. We then use graph-theoretical notions of vertex centrality (e.g., eccentricity, closeness etc.) to identify central vertices (sentences). In the second approach, we propose to use any of the well-known keyword identification algorithms to extract characteristic keywords from a document. We use the number of keywords present in a sentence as a measure of the sentence’s importance. Finally, we propose the use of well-known Condorcet methodology (originally used to decide winner in an election) to combine the summaries from several algorithms into a winner summary.

This paper is organized as follows. Section 2 outlines related work. Section 3 contains several summarization algorithms based on notions of centrality. Section 4 contains a keyword-based approach to summarization. In section 5, we discuss the use of Condorcet methodology to combine multiple summaries into a winner summary. Section 6 discusses some evaluation of the proposed algorithms. Section 7 contains conclusions and outlines further work.

2 Related Work

Many different approaches for text summarization have been proposed. We focus on single document summarization here; see (Hovy, 2005) for an overview. Several summarization systems consider the position of a sentence to indicate its importance. Presence of cue phrases, such as significantly, also indicates that the sentence may be important (Teufel and Moens, 1997), (Kupiec et al, 2003), (Hovy, 1998). For example, (Kupiec et al, 2003) treats summarization as a statistical classification task. They train a Bayesian classifier from a corpus of documents and associated summaries. They consider sentence position and cue phrases as important features. Presence of words which are somewhat frequent is also a characteristic of important sentences (Edmundson, 1968). One of our algorithms in this paper provides another precise characterization of this approach. We facilitate the use of any of the well-known techniques for keyword identification - e.g., (Matsuo and Ishizuka, 2004) - and use the number of keywords present in a sentence as a measure of the sentence’s importance. (Zha, 2002) is an interesting work that also represents a document as a bipartite graph and combines keyphrase extraction and summarization using the principle of mutual reinforcement and sentence clustering. IR-based summarization methods are closely related to this work (Salton et al, 1994), (Mitra et al, 1997). Like us, they use IR methods, such as TFIDF, to find paragraphs related to a given paragraph. Unlike them, we structure the document as a graph, where the edge weights indicate sentence dissimilarity and then use centrality measures to find important vertices (sentences). Several summarization approaches use some measure of coherence in the text. For example, in (Alam et al, 2003), an approach based on lexical chains is proposed. A lexical chain is a sequence of related words that occurs in a sequence of sentences in the document. A WordNet-based mechanism is used to identify strong chains (based on relationships among the words in a chain such as synonyms, hypernyms etc.). A score is calculated for each sentence based on all chains passing through the sentence and highest ranked sentences are included in a summary. See also (Mani et al, 1998). Summarization has also been treated as an information extraction problem; see e.g., (White et al, 1994).

3 Centrality-based Summarization

We consider the following single document summarization problem: given an integer $k \geq 1$ and given a text document (a finite set of sentences), identify a subset of $k$ sentences in the document. The selected $k$ sentences should constitute a reasonable summary of the document.

We propose an approach for text summarization based on the notion that a summary should identify and include central sentences in the document (in some well-defined sense). We represent the given document as a sentence graph and formalize the notion of the centrality of a sentence in a document in terms of the eccentricity of the corresponding vertex in the sentence graph.

3.1 Sentence Graph

We first define a notion of distance between two sentences in a document, based on the well-known
TFIDF approach. Basically, we treat each sentence in the given document as a separate “document” and compute the TFIDF distance between two sentences.

**Definition 1** Let a given document \( D \) consist of \( N \) sentences \( S_1, S_2, \ldots, S_N \). Let \( R = \langle u_1, u_2, \ldots, u_M \rangle \) denote an arbitrarily ordered list of all \( M \) words that occur in \( D \). Then a sentence \( S_j \in D \) can be viewed as an \( M \)-component document vector \( V_j = \langle s_{1j}, s_{2j}, \ldots, s_{Mj} \rangle \) where the quantity \( s_{ij} \) is defined as follows:

\[
s_{ij} = \frac{tf_{ij} \times \log \left( \frac{N}{n_k} \right)}{\sqrt{\sum_{k=1}^{M} (tf_{kj})^2 \times (\log \left( \frac{N}{n_k} \right))^2}}
\]

where \( tf_{ij} \) = the number of times the \( i \)-th term in \( R \) appears in sentence \( S_j \)
\( N \) = total number of sentences in \( D \)
\( M \) = total number of words in \( D \) (\( M = |R| \))
\( n_k \) = number of sentences that contain \( k \)-th term \( u_k \)

**Definition 2** Given two document vectors \( V_i \) and \( V_j \), the cosine distance between them, denoted \( \text{cosine}(V_i, V_j) \), is defined as:

\[
\text{cosine}(V_i, V_j) = \frac{V_i \cdot V_j}{||V_i|| \times ||V_j||}
\]

where \( V_i \cdot V_j \) denotes the dot product of two vectors and \( ||V_i|| \) is the magnitude of the vector \( V_i \).

For a given document \( D \), we define an undirected edge-labeled sentence graph, where the sentences in \( D \) are the vertices and edges are labeled with the cosine distance between the two vertices.

**Definition 3** Let a given document \( D \) consist of \( N \) sentences \( S_1, S_2, \ldots, S_N \). We define an undirected, edge-labeled sentence graph \( G_D = (V, E, \phi) \) for \( D \) where the set \( V \) of vertices contains one vertex for each sentence \( S_i \) (\( |V| = N \)), the set \( E \) of edges contains an edge from every sentence \( S_i \) to every other sentence \( S_j \) labeled with \( \phi(S_i, S_j) = \text{cosine}(V_i, V_j) \) where \( V_i, V_j \) are the document vectors corresponding to sentences \( S_i, S_j \), provided \( \text{cosine}(V_i, V_j) > 0 \).

Since cosine is a distance (or dissimilarity) measure between two sentences, lower score indicates sentences which are more similar.

Consider the following news item published on 4th June, 2007 in The Economic Times (http://economictimes.indiatimes.com) with the headline Oil prices lower after key US gas pipeline reopens. The sentences are numbered for later reference.

1. SINGAPORE: Oil prices fell in Asian trade as gasoline (petrol) supply concerns eased after a key US pipeline reopened over the weekend, dealers said.
2. At 10:30 am (0230 GMT), New York’s main contract, light sweet crude for July delivery was 40 cents lower at 64.68 US dollars a barrel from 65.08 dollars in late US trades Friday.
3. Brent North Sea for July delivery dropped 34 cents to 68.73 dollars.
4. The re-opening of a pipeline carrying gasoline supplies to the US east coast by the Colonial Pipeline Company Sunday was the main factor behind the price decline, dealers said.
5. “The pipeline restarted on Sunday and the pipeline is a main distribution pipeline to get gasoline to the east coast,” said Victor Shum, an analyst with energy consultancy Purvin and Gertz in Singapore.
6. “That is back on line and is reassuring because gasoline supply worries in the US have been driving up crude oil,” he said.
7. US gasoline demand has entered its peak demand period as the US driving season gets underway with millions of Americans due to take to the nation’s highways en route to summer holiday hot spots.
8. Gasoline supply concerns will remain a key factor influencing the market after the US Department of Energy showed petrol reserves rose by 1.3 million to 198 million barrels in the week ending May 25, lower than market expectations for an increase of 1.5 million barrel.
9. US crude reserves fell 2.0 million barrels to 342.2 million barrels in the week ending May 25.
10. That caught many analysts, who had forecast an increase of 1.0 million barrels, off guard.
11. Below-average stock levels of gasoline have forced retail gasoline prices higher across the United States.

Fig. 1 shows the sentence graph for this news item, which contains 11 vertices (one for each sentence) and 37 edges (pairs of sentences with non-zero distance between them).

**3.2 Sentence Centrality**

The graph-theoretical notion of eccentricity measures how central a vertex in a graph is. We have also used other notions of centrality of a vertex in a graph - e.g., closeness and betweenness - from social network analysis.
Definition 4: Given an edge-labeled graph $G$, the distance $d(u,v)$ between two vertices $u$ and $v$ is the sum of the edge weights on a shortest path from $u$ to $v$ in $G$. Eccentricity $\epsilon(u)$ of a vertex $u$ in $G$ is the maximum distance from $u$ to any other vertex $v$ in $G$, i.e., $\epsilon(u) = \max\{d(u,v) | v \in G\}$. A vertex $u$ is a center of the graph $G$ if it has the smallest eccentricity among all vertices in $G$.

In general, a graph may have more than one center vertex. Computing the eccentricity of a given vertex is easy. Dijkstra’s single source shortest path algorithm efficiently computes the shortest paths from a given vertex $u$ to all other vertices. Then the eccentricity $\epsilon(u)$ of $u$ is the length of the longest path among these paths. Intuitively, one may expect low eccentricity sentences (in a sentence graph) to be more important. One approach to automatic selection of summary sentences is now clear. List the sentences in increasing order of their eccentricities in $G$ and pick the first $k$ sentences (having the least eccentricity). This approach is formally specified in the algorithm EccSen.

**Algorithm EccSen**

input document $D$, integer $k \geq 1$
output Set $A$ containing $k$ sentences of $D$

Form sentence graph $G = (V,E,\phi)$ for $D$
for each vertex $u \in V$ do
Find and store eccentricity $\epsilon(u)$ of $u$
end for
Find and store degree $\delta(u)$ of $u$
end for
Select and return top $k$ vertices in increasing order of the values of $\frac{\epsilon(u)}{\delta(u)}$

Table 1: Centrality measure for the vertices.

<table>
<thead>
<tr>
<th>$u$</th>
<th>$\delta(u)$</th>
<th>$\epsilon(u)$</th>
<th>$f(u) = \frac{\epsilon(u)}{\delta(u)}$</th>
<th>$C(u)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>0.310</td>
<td>25.81</td>
<td>0.451</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>0.282</td>
<td>28.37</td>
<td>0.564</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.347</td>
<td>2.88</td>
<td>3.104</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>0.310</td>
<td>22.58</td>
<td>0.450</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>0.305</td>
<td>26.23</td>
<td>0.429</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>0.309</td>
<td>25.89</td>
<td>0.471</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>0.307</td>
<td>26.06</td>
<td>0.424</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0.318</td>
<td>28.30</td>
<td>0.519</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>0.347</td>
<td>17.29</td>
<td>0.784</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.306</td>
<td>16.34</td>
<td>0.560</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>0.312</td>
<td>19.23</td>
<td>0.462</td>
</tr>
</tbody>
</table>

Table 1 shows the eccentricity $\epsilon(u)$ for the sentence graph in Fig. 1. If $k = 3$, then vertices 2, 5 and 10 are ranked lowest in terms of the values of $\epsilon(u)$. The summary consists of sentences 2, 5, 10.

In general, we found that the algorithm EccSen works reasonably well in identifying central sentences in a document. However, in this example, while sentences 2 and 5 seem acceptable for the summary, sentence 10 is somewhat odd. We propose an improvement to better capture the notion of central sentences.

There is another factor, viz., the degree of a vertex, that can be used to refine the eccentricity-based notion of the centrality of a sentence. A sentence having a larger degree is clearly more important (as a summary sentence) than a sentence with a smaller degree. A sentence with a large degree indicates that it is “related to” more number of sentences and thus is potentially more important. Hence we use the parameter $f(u) = \frac{\delta(u)}{\epsilon(u)}$ to measure the centrality of a vertex $u$, where $\delta(u)$ is the degree of $u$. This parameter favours vertices with lower eccentricity and higher degree. Algorithm EccSen2 is the same as algorithm EccSen except that it sorts the vertices in decreasing values of $f(u)$.

![Figure 1: Sentence graph for the news item.](image-url)
Table 1 shows the degree $\delta(u)$, eccentricity $\epsilon(u)$ and the ratio $f(u) = \frac{\delta(u)}{\epsilon(u)}$ for the sentence graph in Fig. 1. If $k = 3$, then vertices 2, 8 and 5 are ranked highest in terms of the values of the $f(u)$ parameter. The corresponding summary contains sentences 2, 8, 5. Clearly, the new sentence 8 is better than sentence 10 from the earlier summary.

3.3 Using Other Centrality Measures

We used eccentricity as a measure of the centrality of a vertex in a graph and identified those sentences corresponding to the vertices of largest centrality (i.e., least eccentricity) in the sentence graph. Betweenness and closeness are two other notions of vertex centrality, among others, developed in the social network analysis community. Either of these could be used as a measure of vertex centrality to identify central sentences from the sentence graph representation of a given document.

Definition 5 (Freeman, 1977) Given an edge-labeled graph $G = (V, E, \lambda)$, the betweenness $B(v)$ of a given vertex $v$ is defined as:

$$B(v) = \sum_{v \in V, v \neq s \neq t} \frac{n_{st}(v)}{n_{st}}$$

where $n_{st}(v)$ is the number of shortest paths from $s$ to $t$ that pass through $v$ and $n_{st}$ is the total number of shortest paths from $s$ to $t$.

Definition 6 (Wasserman et al, 1995) Given an edge-labeled graph $G = (V, E, \lambda)$, the closeness $C(u)$ of a given vertex $u$ is defined as:

$$C(u) = \sum_{v \in V} d(u, v)$$

where $d(u, v)$ is the length of the shortest path from $u$ to $v$.

Closeness of a vertex $u$ is related to eccentricity. In computing the eccentricity of a vertex $u$, we use the $\max$ operator over the lengths of the shortest paths from $u$; in closeness, we use the summation operator. Thus, computing closeness for a vertex is similar to computing the eccentricity of that vertex. Obviously, vertices with lower value of closeness are more central. So the approach to extract the central sentences is the same as for eccentricity. Algorithm CloSen computes the closeness value for each vertex using Floyd-Warshall all-pairs shortest path algorithm. Then it sorts the vertices in the ascending order of their closeness values and picks the lowest $k$ vertices as central sentences. It uses the vertex degree to break ties among vertices that have the same closeness value (vertices with larger degree are more preferable).

Algorithm CloSen

input document $D$, integer $k \geq 1$
output Set $A$ containing $k$ sentences of $D$
Form sentence graph $G = (V, E, \phi)$ for $D$
Use Floyd-Warshall all pairs shortest-path algorithm to compute matrix $dx$ where $dx[u][v]$ denotes the the length of the shortest path between $u$ and $v$ for each vertex $u \in V$
do Compute and store $C(u)$ for $u \in V$
denotes the the length of the shortest path between $u$ and $v$
end for
Select and return top $k$ vertices in increasing order of the values of $C(u)$ (using vertex degree to break ties)

The summary returned by CloSen contains sentences 7, 5, 4, as these have the least closeness values in the example (Table 1).

Just as for eccentricity, we refine algorithm CloSen into algorithm CloSen2 (not shown here) that takes into account the vertex degree along with the vertex closeness. It then sorts the vertices in the decreasing value of the ratio $\frac{\delta(u)}{C(u)}$ and selects the top $k$ vertices. From Table 1 it is clear that algorithm CloSen2 identifies the sentences 7, 5 and 1 for the example.

4 Keyword-based Text Summarization

In this section, we propose an alternative method for a single document summarization based on keywords. Given the set of characteristics keywords for a document, the idea is to select those sentences which contain maximum number of keywords.

A document (e.g., an article, a research paper or a news item) is typically characterized by a set of keywords (more generally, key phrases). Each keyword indicates an important aspect of the subject matter described in the document. Each keyword may also be thought of as a major topic discussed in the document. Typically, only a few keywords (less than 10 or 20) are associated with each document (unlike the large number of index terms used in information retrieval to index a document in a collection). More-
Table 2: Keywords in sentences.

<table>
<thead>
<tr>
<th>No.</th>
<th>#Key words</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>gasoline, supply, pipeline, price, trade</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>main, barrel, lower, trade</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>gasoline, main, supply, pipeline, price, factor</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>gasoline, main, pipeline, energy</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>gasoline, supply</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>gasoline</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>gasoline, barrel, supply, lower, energy, factor</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>barrel</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>barrel</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>gasoline, price</td>
</tr>
</tbody>
</table>

over, the keywords are usually **ordered** in decreasing order of their importance (keywords that are most characteristic of the document occur first). Alternatively, the keywords may also be ordered in increasing order of their generality (most specific terms occur first). Keywords facilitate classification or categorization of a standalone document whereas the index terms facilitate searching the documents within a collection.

Several algorithms can be used to extract a set of m keywords from a given document D, where m is a user-specified positive integer. Then the following algorithm simply ranks (in descending order) the sentences in D in terms of the number of keywords in W that they contain and returns the first k sentences.

**Algorithm KWSummary**

```plaintext
input document D, k ≥ 1
input set W containing m ≥ 1 keywords
output Set A containing k sentences of D
for each sentence Sᵢ ∈ D do
    Compute number g(Sᵢ) of keywords in W that occur in Sᵢ
end for
Order the sentences in D in decreasing order of their g score
Return set A containing first k sentences
```

Assume that the set of m = 10 keywords for the example is W = {gasoline, main, barrel, supply, pipeline, lower, price, energy, trade, factor}. We treat any variation of these keywords (e.g., plural, past tense etc.) as the occurrence of the same keyword. Then the keywords occurring in each sentence are shown in Table 2.

Algorithm KWSummary identifies sentences 8, 4, 1 since they contain the highest number of keywords. In case of ties, KWSummary can choose the sentence having most different keywords than those in the sentences chosen so far.

One expects that the algorithm KWSummary is sensitive to (a) the keyword extraction algorithm used i.e., to the set of keywords W; and (b) the number k of the keywords used. While this is broadly true, experimentally, we found that the algorithm is quite robust to reasonable changes in both the set W and the number k. For example, for the same keyword extraction algorithm (and the same set of text documents), we found that the summary sentences chosen by KWSummary do not vary much over values of k from 10 to 20.

### 5 Combining Sets of Sentences

We have presented 5 algorithms EccSen, EccSen2, CloSen, CloSen2, KWSummary to summarize a given document. The sentences identified by these algorithms for the example are shown in Table 3.

Note that several sentences are identified by more than one method. For example, sentence 5 is identified by 4 methods and sentences 1, 2 and 8 are each identified by 2 methods. Thus they have a stronger claim to being the “true” summary sentences of the given news item, over other sentences like 7 which were reported by only 1 of the methods. To formalize this notion, we use a well-known method by which we can systematically combine the results of these various summarization algorithms. This method, called the Condorcet algorithm, is often used to decide winner in a poll or an election.

**Definition 7** Let a document D contain N sentences, numbered from 1 to N. Let \( \sigma_i = < S_{i1}, S_{i2}, \ldots, S_{ik} > \) denote the ranked output of \( i - th \) summarization method, where each \( s_{ik} \) denotes a sentence number. The rank of sentence number \( j \) in \( \sigma_i \), denoted \( r(\sigma_i, j) \), is the position at which
that sentence occurs in $\sigma_i$. If sentence number $j$ does not occur in $\sigma_i$ then its rank is set to some large number (say, 100). A method $i$ prefers sentence $x$ over sentence $y$ if $r(\sigma_i, x) < r(\sigma_i, y)$ i.e., $x$ appears earlier than $y$ in the ranked output of method $i$. Let $S = \{\sigma_1, \sigma_2, \ldots, \sigma_M\}$ denote the collection of the outputs of $M$ summarization algorithms.

In our example, referring to Table 3, we have $M = 5$ methods and $\sigma_1 = \langle 2, 5, 10 \rangle$, $\sigma_2 = \langle 2, 8, 5 \rangle$ and so on. The rank of sentence 5 in $\sigma_1$ is $r(\sigma_1, 5) = 2$, since 5 occurs at position 2 in $\sigma_1$. Method 1 prefers sentence 2 to sentence 5.

We form an $N \times M$ matrix $R$, where $N$ is the number of sentences in the given document $D$ ($N = 11$ in the example). The $i$-th row of $R$ corresponds to the $i$-th sentence in $D$. We consider each of the $M = 5$ summarization methods as a voter. Entry $R_{i,j}$ in $R$ indicates the total number of voters (out of $M$) who prefer sentence $i$ over sentence $j$. The matrix $R$ for the example is shown in Table 4, where for brevity we include only those sentences which are identified by at least 1 method. For example, $R_{7,5} = 2$ since 2 methods (3 and 4) prefer sentence 7 over sentence 5.

Condorcet method analyzes the votes of $M$ summarization methods, as represented in the matrix $R$, to decide the winner among the $N$ candidate sentences. For this purpose, it considers $N \times N$ imaginary contests, pitting every sentence $i$ against every other sentence $j$. Sentence $i$ is a winner of this contest if the number of voters that prefer $i$ is greater than the number of voters that prefer $j$. In the example, sentence 8 wins against sentence 5 because $R(8, 5) > R(5, 8)$. When all possible pairings of the sentences are considered, we pick up the $k$ sentences that have won the maximum number of contests. Ties can be broken by means of well-known variations of the Condorcet method; e.g., ranked pairs or Schulze method. In the example, sentences 2, 5 and 7 constitute a winning summary according to this method.

We found that the summarization methods are sensitive to synonyms. In the example, the news item contains synonyms oil, crude and gasoline. If these words are replaced by a single word (say oil) then the summary results improve a lot.

6 Evaluation

We have experimented with the summarization techniques discussed here on several small text document collections. One such collection consists of 64 news items from well-known Indian news magazines, from 5 different categories: environment, economy, defence, health and cinema. In one experiment, we generated the summaries using different algorithms proposed here and measured the overlap among the sentences identified. The algorithms show considerable overlap i.e., many sentences are common in the outputs of the algorithms on the same document. For example, we found that about 55 algorithms EccSen2 and KYSummary, when 5 summary lines were extracted for each document (number of keywords used = 10). That is, on the average about 2 to 3 sentences occur in both summaries for any given document. This considerable overlap allows us to place more confidence in the extracted summary.

Manual evaluation of the summaries generated has also given good feedback - outputs of EccSen2 and KYSummary were generally ranked highest in quality by human users. We are in the process of comparing the results of our algorithms with those of other public-domain and commercial summarizers (e.g., Microsoft Word) on publicly-available text collections. Our initial experiments seem to show encouraging results.

7 Conclusions and Further Work

In this paper, we proposed two distinct single document summarization algorithms. The first approach is based on the notion that important sentences are somehow central in the document. We represent the document as a graph, with sentences as vertices and edges labeled with sentence dissimilarities. We then use graph-theoretical notions of vertex centrality (e.g., eccentricity, closeness etc.) to identify cen-
tral vertices (sentences). In the second approach, we propose to use any of the well-known keyword identification algorithms and then use the number of keywords present in a sentence as a measure of the sentence’s importance. Keywords characterise a document; each keyword indicates an important concept or topic present in the document. We also discussed the use of well-known Condorcet methodology to combine the extraction (summaries) from several algorithms into a winner summary.

Experimentally, we found that these methods produce good summaries. For a given document, their extractions often overlap considerably, thereby allowing us more confidence in the extracted summary. The techniques seem to work reasonably well for documents in a variety of domains.

For further work, we plan to experiment with several other, more complex, measures of centrality of a vertex in a graph, such as the random walk centrality (Noh and Rieger, 2004) and information centrality (Stephenson and Zelen, 1989). We also plan to experiment with other methods to combine the results of different summarization algorithms. One important choice is the well-known Dempster-Schafer method of evidence combination (Schafer, 2001). We are also working on extending these approaches for multi-document summarization. Since these techniques do not use any language specific knowledge, potentially they can be used as language independent summarization techniques. Finally, we plan to modify our methods to use WordNet to take synonyms into account when generating the summaries.

References


