Managing Diversity and Redundancy in Summaries with the Aid of Wikipedia Categories

Abdullah Bawakid
Faculty of Computing and Information Technology, University of Jeddah, Jeddah, Saudi Arabia

ABSTRACT
This paper presents a novel multi-document summarization system that relies on specific features extracted from Wikipedia. The system utilizes mainly two aspects from Wikipedia, its articles titles and categories network. It does not employ the inner content of the articles in Wikipedia, nor inter or inner links. The implemented summarizer includes a module that focuses on managing diversity and redundancy when generating summaries. We describe in this paper how this module is applied on the subtopic clusters which are also generated during the summarization process. The evaluation we performed on the system illustrate its competitiveness when compared against others in the literature.

Keywords: Summarization, Diversity, Redundancy, Wikipedia, Wikipedia Categories

1. INTRODUCTION
The continuously increasing amount of available information online over the past few decades has emphasized the importance of intensive research in the area of information retrieval (IR). Among the main challenges in that area is the task of text documents summarization. The main purpose of this task is to provide users with condensed, relevant and short version of the original text documents in an automatic way. The process can be applied to single or multi source documents and may be driven by a user query shifting the focus of the summary to the interests of the user.

We describe in this paper the design and implementation for an automatic documents summarizer which we built. The summarizer relies on specific features from Wikipedia to help discover the directly mentioned topics within the source documents in addition to the inferred ones. In addition, the main algorithm of the system helps with establishing links between the discovered topics and determines the links strength among the different topics based on how related they are. We also explain how subtopic clusters are built. In addition, we describe how and why we built a module in the system that handles information diversity and redundancy. The evaluation we performed illustrate the effectiveness of the summarization system and puts focus on the effect of the diversity/redundancy handling module as well by showing the effect of its introduction to the system.

In the next section we talk about background and related work. In section 3 we describe the methodology of the system and how it was constructed. In section 4 we explain how we evaluated the system. Finally in section 5 we give the conclusion.

2. BACKGROUND AND RELATED WORK
In most cases, text documents contain information of varying degrees of importance to the readers. The task of automatic summarization involves the distinction of the most important and relevant information parts in the documents from the rest and providing them to the user in a condensed and suitable form. In general, work for this task can be divided into two categories based on the generated summaries type.

One is extractive summaries where important parts from the original source documents are selected to form the summary without altering the inner content of the chosen sentences. Another type is abstractive summaries when a restricting of the selected sentences takes place to either shorten their length or include more information in the summary. There have been a number of attempts in the literature at the creation of abstracts as summaries as in [1], [2] and [3]. However, work in this direction is very limited when compared against other work that focuses on extractive summaries. In our work in this paper, we also focus on extractive summaries.

It is also possible to look at the summarization problem from another perspective based on the methodology used for generating summaries. For instance, some algorithms employ surface-level features extracted mainly from the source documents to be summarized. An example for this is the usage of words frequencies, commonly referred to as bag-of-words methods, as in the work of [4]. Others utilized the position of sentences by giving leading sentences more weight in the summaries than the rest of the sentences as in [5] and [6]. The performance of these surface-based methods was found to be lacking when compared against others in the literature that rely on external ontologies as reported in [7]. However, their advantage is in their simplicity as they do not often require preprocessing and or large processing resources when implemented.

Other work in the literature utilized statistical and machine learning approaches for summarization. For instance, the work in [8] and [9] used different variations of the Hidden Markov Models (HMM) for summarization. Neural Networks were employed In [10] and [11], too. Also in the literature, others looked at the summarization problem from another perspective by attempting to capture the implicit semantic relationship between different text fragments in the source document based on their contextual occurrences. Among the first attempts in this direction is the Latent Semantic Analysis (LSA) technique. It was employed in the task of summarization in [12] and [13].
The usage of external ontologies and knowledge bases was also investigated in the literature. In [14], WordNet was used a semantic dictionary for a simplified Lesk algorithm in a learning-based summarization system. The internal links of Wikipedia was employed in [15] to device a semantic relatedness measure. This measure was later used for automatic documents summarization in [16] and [17]. Another measure utilizing a personalized PageRank on a Wikipedia-based graph was proposed in [18]. It was later employed for summarization in [19].

In our work here, we also rely on an external ontology, namely Wikipedia. We use an algorithm previously covered in [20] for detecting direct and inferred relationship between text fragments with the aid of term-categories vector. We also construct subtopic clusters within the source documents during summarization. Additionally, we implement a diversity and redundancy management module. This module checks candidate sentences before being added to the summary by evaluating the relevancy, diversity and redundancy of the information they contain against what is already in the summary.

3. METHODOLOGY

The algorithm we utilize for summarizing text documents relies on the categories network and the titles of articles in Wikipedia. This bears similarity to our previous work in [20] and [21]. However, we adapt and expand the core algorithm for the task of automatic documents summarization. In particular, we put emphasis on managing the diversity and redundency within the final formed summary will be illustrated in the description below. In the next subsections, we describe the main stages of the implemented summarizer and the step that the system go through to generate a summary.


We used the same Wikipedia preprocessing module which was employed in our work described in [20]. This stage involves the removal of data which is not needed by our algorithm. This includes the text content of each Wikipedia article, the inner links that exist within each article, the tables and figures, and any other meta tags which are attached to each article. We only preserve the titles of all articles in addition to the categories network. Categories which are too broad by having more than 2000 articles are discarded. Too narrow categories having downwards of 10 articles are also discarded. Maintenance and administrative categories were also removed.

b. Creating the Term-Categories Vector

The module implemented for completing this stage is the same as the one used in our previous work in [20]. We construct a vector that defines the relationship strength between all words and the categories that exist in Wikipedia. This relationship strength is reflected in the weight assigned to each term-category pair in the constructed vector. To build the vector, we apply the following formula:

\[ w_t = \frac{1}{|a|} \times \log \frac{|C|}{|C_t|} \]  

(1)

In the above equation, we have the weight of each term represented with \( w_t \), \( |a| \) the number of titles in Wikipedia which contain the term \( t \), \( |C| \) the total number of categories in Wikipedia, and \( |C_t| \) the total number of categories within Wikipedia which contain the term \( t \).

c. Diversity and Redundancy-Influenced Subtopic Clusters

The basic idea here is to create clusters of the subtopics that exist in the source documents. Each cluster would have sentences which are found to be highly related and span related subtopics. Along with the related sentences, each cluster should also contain a number of topics which were found to be highly related. In addition to their need in the implemented algorithm when forming the final query-driven summaries, presenting the subtopic clusters can be useful to the users when typing a query for driving the summary.

This stage starts by forming a representative weighted vector for each sentence \( S \) in the source documents. We begin by computing the frequency of each term \( t \) in \( S \) and represent it with \( Swf \). Next, we select the best representative Wikipedia categories for the sentence \( S \) by attempting to capture the inferred relationship between the titles of Wikipedia articles and their containing categories. This is implemented by applying the following formula:

\[ Sw_a = \sum_{t \in S} (Swf_t \times w_t) \times \frac{1}{|a|} \times \frac{|a|}{|a|} \]  

(2)

Where \( Sw_a \) refers to how strongly the article title is related to the currently scanned sentence, \( Swf \) refers to the frequency of the word in the sentence \( S \), \( w_t \) refers to the word weight, \( |a| \) refers to how many other articles with the same title as that of article \( a \), \( |a| \) refers to the total number of terms that are shared between the current sentence \( S \) and the article \( a \), and \( |a| \) refers to the number of terms in the article \( a \).

We should now have an associated vector of \( Sw_a \) for every sentence in \( S \) that reflects how strongly the sentence \( S \) is related to the different article titles in Wikipedia. Afterwards, we move to the next step which is forming the clusters. Each cluster would focus on a number of related terms and also contain the sentences spanning these topics. We achieve this step by computing the Cartesian product between the associated vectors of \( Sw_a \) for every sentence against the rest of the sentences in the source documents. For any two pair of sentences which are found to have a result exceeding the previously set threshold, they are added together in a cluster. This process is repeated for all sentences resulting in different clusters where each cluster contains semantically related sentences. The topics covered in each cluster are deduced from their contained sentences with the aid of the term-categories vector method described above.

d. Forming the Summary

At this stage, we should have the subtopic clusters already formed. Each cluster would have a list of highly related sentences which were found to cover similar or related topics. The redundency of information in each cluster would be expected. In addition, the diversity of information among the
different clusters is also expected as their topics are not directly related. When forming an extractive summary, it is desired to increase the coverage of information by balancing between the diversity and redundancy of information present in the source text sentences. In our work, we attempt to achieve this by utilizing the previously-formed topic clusters when selecting candidate sentences for the summary.

We create a summary by first assigning a score to all sentences which are present in the different clusters. The score would signify the significance of the sentence among the rest of the sentences in the cluster given a query, if provided by the user. We compute the score for each sentence by aggregating the previously computed weight for the content of this sentence with the aid of the term-categories vector as described above. After scoring all sentences, we rank them in a descending order and select the top ranked sentence for inclusion in the summary.

After adding the first candidate sentence to the summary, we recomputed the score for the remaining ranked sentences to consider the inclusion of the top ranked sentence. This step is applied by utilizing the following equation:

$$w_{Sc} = \omega \text{Sim}(S_c, S_j) - (1 - \omega) \text{MaxSim}(S_c, S_j) \quad (3)$$

Where $w_{Sc}$ refers to the new weight of the current sentence $S_c$, $S_j$ is a candidate sentence that has not been added to the summary yet, $\text{MaxSim}$ refers to the maximum semantic relatedness score between the current sentence $S_c$ and the rest of the sentences $S_j$ that have already been added to the summary, and $\omega$ is a number between one and zero that defines the balance of the weight between $\text{Sim}$ or $\text{MaxSim}$ in the above formula.

It can be noted from equation 3 that is generally divided into two parts. The first part computes the similarity between the current sentence and the rest of the candidate sentences. The second part considers the similarity between the current candidate sentence against the rest of the previously chosen sentences to be added to the summary. In effect, this function puts focus on both diversity and redundancy by penalizing sentences which are similar to those already added in the summary and promoting those have relevant and divergent information.

4. EXPERIMENTS AND EVALUATION

In order to evaluate our system, we used the documents dataset provided by NIST for the Text Analysis Conference (TAC) multi-document summarization task for 2010. The task involved asking participants to generate summaries not exceeding in length a previously determined number. Each summary would be derived from ten newswire documents and should consider a specific given query.

Automatic evaluation for the generated summaries takes place through the usage of the Basic Elements (BE) and the ROUGE metrics. When evaluating our system, we obtained the scores illustrated in Table 1. For the sake of comparison, the results of two additional runs are also illustrated in the table, namely baseline-1 and baseline-2. In baseline-1, a simple summarization system was implemented that works by selecting leading sentences in the source documents to form the summary. The other baseline is the MEAD [22] summarizer which was also used as a baseline by the TAC task organizers.

Table 1: Evaluation Scores Obtained for Different Runs in the TAC10 Summarization Task

<table>
<thead>
<tr>
<th></th>
<th>ROUGE-2</th>
<th>ROUGE-SU4</th>
<th>BE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline-1</td>
<td>0.053515</td>
<td>0.086865</td>
<td>0.029415</td>
</tr>
<tr>
<td>Baseline-2</td>
<td>0.060965</td>
<td>0.093925</td>
<td>0.035485</td>
</tr>
<tr>
<td>Run1</td>
<td>0.064851</td>
<td>0.097324</td>
<td>0.037712</td>
</tr>
<tr>
<td>Run2</td>
<td>0.068924</td>
<td>0.099051</td>
<td>0.040181</td>
</tr>
</tbody>
</table>

To test the effectiveness for the inclusion of the diversity and redundancy handling part in our system, we created two runs. In run1, the system selects the top ranking sentences to be included in the summary without the inclusion of the redundancy/diversity handling module described above, while in run2 the system does. The results we obtained with run1 shows that the basic algorithm we utilized which simply relies on the categories network in Wikipedia in addition to its title can provide competitive results especially when compared against other Bag-of-Words methods as in baseline-2. With the results we obtained for run2 we can see an improvement obtained with all metrics by the introduction of the diversity and redundancy management module.

5. CONCLUSION

In this paper, we described the implementation of a multi-document automatic summarization system. The system utilizes only certain aspects from Wikipedia, namely its articles titles and the categories network. It relies on the usage of a term-categories vector which guides the core system algorithm through defining how strongly a term is related to the different categories in Wikipedia. In addition, this vector was used previously in certain other related applications including documents clustering [20] and linking named entities [21].

The described summarization system in this paper is extractive and produces summaries that contain unaltered sentences taken from the original source documents after a selection process. The selection process involves creating subtopic clusters within the source documents. Afterwards, a module that manages redundancy and diversity during the generation of the summary takes charge. This module examines the different subtopic clusters and adds candidate sentences to the summary only after verifying the redundancy and diversity of the information contained in the new candidate sentence. We also explained how we evaluated the implemented system with the aid of TAC10 summarization task and provided the obtained results. The evaluation results show the competitiveness of the system especially when compared against others in the literature as was shown in the evaluation section above. In the future, we intend to test the system by introducing it to real-life users and examining their behavior especially when viewed the generated summaries along with the subtopic clusters. We also intend to explore the possibility of expansion on the core algorithm by introducing it to the information retrieval domain.
REFERENCES


