A new graph based text segmentation using Wikipedia for automatic text summarization

Mohsen Pourvali
Department of Electrical & Computer Engineering at
Qazvin Branch Islamic Azad University
Qazvin, Iran

Ph.D. Mohammad Saniee Abadeh
Department of Electrical & Computer Engineering at
Tarbiat Modares University
Tehran, Iran

Abstract—**The technology of automatic document summarization is maturing and may provide a solution to the information overload problem. Nowadays, document summarization plays an important role in information retrieval. With a large volume of documents, presenting the user with a summary of each document greatly facilitates the task of finding the desired documents. Document summarization is a process of automatically creating a compressed version of a given document that provides useful information to users, and multi-document summarization is to produce a summary delivering the majority of information content from a set of documents about an explicit or implicit main topic. According to the input text, in this paper we use the knowledge base of Wikipedia and the words of the main text to create independent graphs. We will then determine the important of graphs. Then we are specified importance of graph and sentences that have topics with high importance. Finally, we extract sentences with high importance. The experimental results on an open benchmark datasets from DUC01 and DUC02 show that our proposed approach can improve the performance compared to state-of-the-art summarization approaches.**

Keywords- text Summarization; Data Mining; Word Sense Disambiguation.

I. INTRODUCTION

The technology of automatic document summarization is maturing and may provide a solution to the information overload problem. Nowadays, document summarization plays an important role in information retrieval (IR). With a large volume of documents, presenting the user with a summary of each document greatly facilitates the task of finding the desired documents. Text summarization is the process of automatically creating a compressed version of a given text that provides useful information to users, and multi-document summarization is to produce a summary delivering the majority of information content from a set of documents about an explicit or implicit main topic [14]. Authors of the paper [10] provide the following definition for a summary: "A summary can be loosely defined as a text that is produced from one or more texts that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that. Text here is used rather loosely and can refer to speech, multimedia documents, hypertext, etc. The main goal of a summary is to present the main ideas in a document in less space. If all sentences in a text document were of equal importance, producing a summary would not be very effective, as any reduction in the size of a document would carry a proportional decrease in its in formativeness. Luckily, information content in a document appears in bursts, and one can therefore distinguish between more and less informative segments. Identifying the informative segments at the expense of the rest is the main challenge in summarization". Assume a tripartite processing model distinguishing three stages: source text interpretation to obtain a source representation, source representation transformation to summary representation, and summary text generation from the summary representation. A variety of document summarization methods have been developed recently.

The paper [4] reviews research on automatic summarizing over the last decade. This paper reviews salient notions and developments, and seeks to assess the state of-the-art for this challenging natural language processing (NLP) task. The review shows that some useful summarizing for various purposes can already be done but also, not surprisingly, that there is a huge amount more to do. Sentence based extractive summarization techniques are commonly used in automatic summarization to produce extractive summaries. Systems for extractive summarization are typically based on technique for sentence extraction, and attempt to identify the set of sentences that are most important for the overall understanding of a given document. In paper [11] proposed paragraph extraction from a document based on intra-document links between paragraphs. It yields a text relationship map (TRM) from intra-links, which indicate that the linked texts are semantically related. It proposes four strategies from the TRM: bushy path, depth-first path, segmented bushy path, augmented segmented bushy path.

In our study we focus on sentence based extractive summarization. We express that the lexical cohesion structure of the text can be exploited to determine the importance of a sentence. Eliminate the ambiguity of the word has a significant impact on the inference sentence. In this article we will show that the separation text into the inside issues by using the correct concept Noticeable effect on the summary text is created. We have used Word Sense Disambiguation [8] for select correct concept. The experimental results on an open benchmark datasets from DUC01 and DUC02 show that our proposed approach can improve the performance compared to state-of-the-art summarization approaches.

II. Related Work

Generally speaking, the methods can be either extractive summarization or abstractive summarization. Extractive summarization involves assigning salience scores to some units (e.g. sentences, paragraphs) of the document and extracting the sentences with highest scores, while abstraction summarization (e.g.http://www1.cs.columbia.edu/nlp/newsblaster/) usually needs information fusion, sentence compression and reformulation [14]. Sentence extraction summarization systems take as input a collection of sentences (one or more documents) and select some subset for output into a summary. This is best treated as a sentence ranking problem, which allows for varying thresholds to meet varying summary length requirements. Most commonly, such ranking approaches use some kind of similarity or centrality metric to rank sentences for inclusion in the summary – see, for example [1].The centroid-based method [3] is one of the most popular extractive summarization methods.

MEAD (http://www.summarization.com/mead/) is an implementation of the centroid-based method for either single- or-multi-document summarizing. It is based on sentence extraction. For each sentence in a cluster of related documents, MEAD computes three features and uses a linear combination of the three to determine what sentences are most salient. The three features used are centroid score, position, and overlap with first sentence (which may happen to be the title of a document). For single-documents or (given) clusters it computes centroid topic characterizations using tf–idf-type data. It ranks candidate summary sentences by combining sentence scores against centroid, text position value, and tf–idf title/lead overlap. Sentence selection is constrained by a summary length threshold, and redundant new sentences avoided by checking cosine similarity against prior ones. In the past, extractive summarizers have been mostly based on scoring sentences in the source document. In paper [12] each document is considered as a sequence of sentences and the objective of extractive summarization is to label the sentences in the sequence with 1 and 0, where a label of 1 indicates that a sentence is a summary sentence while 0 denotes a non-summary sentence. To accomplish this task, is applied conditional random field, which is a state-of-the-art sequence labeling method.

In paper [15] proposed a novel extractive approach based on manifold–ranking of sentences to query-based multi-document summarization. The proposed approach first employs the manifold–ranking process to compute the manifold–ranking score for each sentence that denotes the biased information-richness of the sentence, and then uses greedy algorithm to penalize the sentences with highest overall scores, which are deemed both informative and novel, and highly biased to the given query.

The summarization techniques can be classified into two groups: supervised techniques that rely on pre-existing document-summary pairs, and unsupervised techniques, based on properties and heuristics derived from the text. Supervised extractive summarization techniques treat the summarization task as a two-class classification problem at the sentence level, where the summary sentences are positive samples while the non-summary sentences are negative samples. After representing each sentence by a vector of features, the classification function can be trained in two different manners [7]. One is in a discriminative way with well-known algorithms such as support vector machine (SVM) [16]. Many unsupervised methods have been developed for document summarization by exploiting different features and relationships of the sentences – see, for example [3] and the references therein. On the other hand, summarization task can also be categorized as either generic or query-based. A query-based summary presents the information that is most relevant to the given queries [2] and [14] while a generic summary gives an overall sense of the document’s content [2] , [4] , [12] , [14].

The QCS system (Query, Cluster, and Summarize) [2] perform the following tasks in response to a query: retrieves relevant documents; separates the retrieved documents into clusters by topic, and creates a summary for each cluster. QCS is a tool for document retrieval that presents results in a format so that a user can quickly identify a set of documents of interest. In paper [17] are developed a generic, a query-based, and a hybrid summarizer, each with differing amounts of document context. The generic summarizer used a blend of discourse information and information obtained through traditional surface-level analysis. The query-based summarizer used only query-term information, and the hybrid summarizer used some discourse information along with query-term information. The article [18] presents a multi-document, multi-lingual, theme-based summarization system based on modeling text cohesion.

III. Create Graph and Text Segmentation

The algorithm presented in this paper, at first the input text is pre-processing and the stop words is removed. Then stem of words is found and its (POS) is tagged.

Only verbs and nouns are used in the text, in the way we have presented. The algorithm starts from the beginning of the main text, and take the word, and using Wikipedia knowledge base provides a two-level tree from the links of the word’ abstract. So that root of the word is the same word and tree Children are related words (links) to the target word in the abstract of its web page. Then it searches the children of the target word in the input text and it creates a graph using target word and the words that both are in the children of the previous step tree and input text.

Let s = {s_1, s_2, ..., s_n} is the set of sentences and $w = \{w_{11}, w_{21}, \dots, w_{ij}, \dots, w_{kn}\}$ is the set of all words are nouns or verbs in the input text. So that w_{ij} shows i-th word in the j-th sentence. Since the goal is to extract sentences with high importance. The sentences are considered as nodes. The relationships between words within a sentence with other sentences words are considered to be edges in the graph. The algorithm is shown in Figure 1.

```
For n=0 to EndOfSentenc
    For i=0 to EndOfSentenc$_n$
        Child[] = CreateTreeInWiki($w_{i_n}$)
            For r=0 to EndOfChild
                For k=0 to EndOfWord
                    If Child[r] == AnyWordOf_W
                        Graph[] = Create_Or_Update_Graph($S_r, S_k$)
                    Endif
                EndFor
            EndFor
    EndFor
EndFor
```

Figure 1. Base algorithm for create the tree and graph

In the above algorithm, ***Child*** is the children of target word tree in the Wikipedia, and ***Graph*** is the constructed graph from the sentences that target word is in them. This algorithm is implemented for all target words in the input text. Finally, we have several independent graphs, that according to the relationship between its nodes, each graph implies a topic in the input text. Figure 2 shows related sentences in the text.

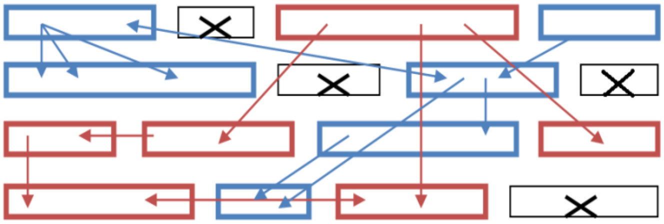

Figure 2. Related sentences and segments, there are two segments with two colors (blow and red)

After extracting the graphs of the input text, the graphs edges were given weight. According to the distance between the words in two sentences, existed in the two sides of the edge, the weighting to the edge is done. To do this we use Average Google normalized distance [19]. NGD takes advantage of the number of hits returned by Google to compute the semantic distance between concepts. The concepts are represented with their labels which are fed to the Google search engine as search terms.

First, using the NGD we define the global and local dissimilarity measure between terms (as shown in [19] the NGD is nonegative and does not satisfy the triangle inequality, i.e. hence isn't distance and consequently in the further it we shall name dissimilarity measure). According to definition NGD the global dissimilarity measure between terms t_k and t_l also is defined by the formula:

$$NGD^{global}(t_k,t_l) = \frac{max\{\log(f_k^{global}),\log(f_l^{global})\}-\log(f_{kl}^{global})}{\log N_{Google}-min\{\log(f_k^{global}),\log(f_l^{global})\}} \quad (1)$$

Where f_k^{global} is the number of web pages containing the search term t_k, and f_{kl}^{global} denotes the number of web pages containing both terms t_k and t_l, N_{Google} is the number of web pages indexed by Google. The main properties of the NGD [19] are listed as follows:

1) *The range of the NGD is in 0 and ∞;*

- If $t_k = t_l$ or if $t_k \neq t_l$ but frequency $f_k^{global} = f_l^{global} = f_{kl}^{global} > 0$, Then $NGD^{global}(t_k,t_l) = 0$. That is, the semantics of t_k and t_l, in the Google sense is the same.
- If frequency $f_k^{global} = 0$, then for every term t_k, we have $f_{kl}^{global} = 0$, and the $NGD^{global}(t_k,t_l) = \frac{\infty}{\infty}$, which we take to be 1 by definition.
- If frequency $f_k^{global} \neq 0$ and $f_{kl}^{global} = 0$, we take $NGD^{global}(t_k,t_l) = 1$.

2) *NGD (tk,tk) = 0 for every tk. For every pair t_k and t_l, we have $NGD^{global}(t_k,t_l) = NGD^{global}(t_l,t_k)$: It is symmetric.*

Formula 3 is the dissimilarity measure between sentences S_i and S_j.

$$\mathrm{diss}_{\mathrm{NGD}}^{\mathrm{global}}(S_i,S_j) = \frac{\sum_{t_k \in S_i}\sum_{t_l \in S_j} \mathrm{NGD}^{\mathrm{global}}(t_k,t_l)}{m_i m_j} \quad (2)$$

That m_i and m_j are the number of words in i-th and j-th sentences. Then, the weighting of the graph, we are selecting the heavier graph (the graph that has heavy nods and light edges). Using the following formula a weight is assigned to each graph.

$$V_g = \frac{1}{L} \times \frac{\sum_{i=1}^{L} F_i \times d_i}{\sum_{j=1}^{E} e_j} \quad (3)$$

That L is number of nodes and E is number of edges in any graph, d_i is the degree of i-th node.

IV. Sentence Extraction

Finally, the graph which is higher than other graphs contains the main topic of the text. In formula 1, sentences can be extracted according to the percent of summarization. If we want to have the summarization of other topics in addition to main topic in the text we extract important sentences from the important graph according to the summarization percent. Using the following formula, each node is evaluated according to its number of incoming and outgoing edges.

$$F_i = \frac{(I_i+O_i)}{L} \times \sum_{t=1}^{m} W_{t_i} \quad (4)$$

Where O_i is number of outputs from i-th sentence and I_i is number of inputs to i-th sentence. We use following formula to calculate the weight of the word W_{ti} .

$$w_{ti} = TF_{ti} \times ISF_{ti} \quad (5)$$

That TF_{ti} is the number of occurrences phrase t in the sentence S_i, and ISF is:

$$ISF = \log(\frac{N}{N_t}) \quad (6)$$

N_t is the number of sentences the word t_i has occurred in it.

V. Experiments and Results

In this section, we conduct experiments to test our summarization method empirically.

A. Datasets

For evaluation the performance of our methods we used two document datasets DUC01 and DUC02 and corresponding 100-word summaries generated for each of documents. The DUC01 and DUC02 are an open benchmark datasets which contain 147 and 567 documents-summary pairs from Document Understanding Conference (http://duc.nist.gov). We use them because they are for generic single-document extraction that we are interested in and they are well preprocessed. These datasets DUC01 and DUC02 are clustered into 30 and 59 topics, respectively. In those document datasets, stop words were removed using the stop list provided in ftp://ftp.cs.cornell.e-du/pub/smart/english.stop and the terms were stemmed using Porter's scheme [9], which is a commonly used algorithm for word stemming in English.

B. Evaluation metrics

There are many measures that can calculate the topical similarities between two summaries. For evaluation the results we use two methods. The first one is by precision (P), recall (R) and F1-measure which are widely used in Information Retrieval. For each document, the manually extracted sentences are considered as the reference summary (denoted by Summ_{ref}). This approach compares the candidate summary (denoted by $\text{Summ}_{\text{cand}}$) with the reference summary and computes the P, R and F1-measure values as shown in formula (9) [12].

$$\text{P} = \frac{|\text{summ}_{\text{ref}} \cap \text{summ}_{\text{cand}}|}{|\text{summ}_{\text{cand}}|} \tag{7}$$

$$\text{R} = \frac{|\text{summ}_{\text{ref}} \cap \text{summ}_{\text{cand}}|}{|\text{summ}_{\text{ref}}|} \tag{8}$$

$$\text{F}_1 = \frac{2\text{PR}}{\text{P}+\text{R}} \tag{9}$$

The second measure we use the ROUGE toolkit [5], [6] for evaluation, which was adopted by DUC for automatically summarization evaluation. It has been shown that ROUGE is very effective for measuring document summarization. It measures summary quality by counting overlapping units such as the N-gram, word sequences and word pairs between the candidate summary and the reference summary. The ROUGE-N measure compares N-grams of two summaries, and counts the number of matches. The measure is defined by formula (10) [5], [6].

$$\text{ROUGE} - \text{N} = \frac{\sum_{S\in \text{summ}_{\text{ref}}} \sum_{\text{N-gram}\in S} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{S\in \text{summ}_{\text{ref}}} \sum_{\text{N-gram}\in S} \text{Count}(\text{N-gram})} \tag{10}$$

Where N stands for the length of the N-gram, $\text{Count}_{\text{match}}$ (N-gram) is the maximum number of N-grams co-occurring in candidate summary and a set of reference–summaries. Count (N_gram) is the number of N-grams in the reference summaries. We use two of the ROUGE metrics in the experimental results, ROUGE-1 (unigram-based) and ROUGE-2 (bigram-based).

C. Simulation strategy and parameters

The parameters of our method are set as follows: depth of tree that is created for any word, *n=3*; extra value for *Lesk* algorithm, $\lambda = 5$; Finally, we would like to point out that algorithm was developed from scratch in C#.net 2008 platform on a Pentium Dual CPU, 1.6 GHz PC, with 512 KB cache, and 1 GB of main memory in Windows XP environment.

D. Performance evaluation and discussion

We compared our method with four methods CRF [12], NetSum [13], Manifold–Ranking [15] and SVM [16]. Tables 1 and 2 show the results of all the methods in terms ROUGE-1, ROUGE-2, and F1-measure metrics on DUC01 and DUC02 datasets, respectively. As shown in Tables 1 and 2, on DUC01 dataset, the average values of ROUGE-1, ROUGE-2 and F1 metrics of all the methods are better than on DUC02 dataset. As seen from Tables 1 and 2 Manifold–Ranking is the worst method, In the Tables 1 and 2 highlighted (bold italic) entries represent the best performing methods in terms of average evaluation metrics. Among the methods NetSum, CRF, SVM and Manifold–Ranking the best result shows NetSum. We use relative improvement $\frac{(our\ method - other\ methods)}{other\ methods} \times 100$ for comparison. Compared with the best method NetSum, on DUC01 (DUC02) dataset our method improves the performance by 3.43% (4.82%), 7.15% (16.30%) and 3.12% (4.28%) in terms ROUGE-1, ROUGE-2 and F1, respectively.

TABLE I. AVERAGE VALUES OF EVALUATION METRICS FOR SUMMARIZATION METHODS (DUC01 DATASET).

Methods	**Av.ROUGE-1**	**Av.ROUGE-2**	**Av.F1-measure**
Our method	0.48021	0.18962	0.48743
NetSum	0.46427	0.17697	0.47267
CRF	0.45512	0.17327	0.46435
SVM	0.44628	0.17018	0.45357
Manifold–Ranking	0.43359	0.16635	0.44368

TABLE II. AVERAGE VALUES OF EVALUATION METRICS FOR SUMMARIZATION METHODS (DUC02 DATASET).

Methods	**Av.ROUGE-1**	**Av.ROUGE-2**	**Av.F1-measure**
Our method	0.47129	0.12986	0.48259
NetSum	0.44963	0.11167	0.46278
CRF	0.44006	0.10924	0.46046
SVM	0.43235	0.10867	0.43095
Manifold–Ranking	0.42325	0.10677	0.41657

VI. Conclusion

We have presented the approach to automatic document summarization based on creating graph and text segmentation and extraction of sentences using Wikipedia. Our approach consists of two steps. First creates a two-level tree from the links of the word's abstract, and then creates graph using of previous phase, and finally selects important segments which were created using of previous graph. When comparing our methods with several existing summarization methods on an open DUC01 and DUC01 datasets, we found that our methods can improve the summarization results significantly. The methods were evaluated using ROUGE-1, ROUGE-2 and F1 metrics. In this paper we also demonstrated that the summarization result depends on the similarity measure. Results of experiment have showed that proposed by us NGD-based dissimilarity measure outperforms the Euclidean distance.

REFERENCES

[1] Alguliev, R. M., & Alyguliev, R. M. (2007). Summarization of text-based documents with a determination of latent topical sections and information-rich sentences. *Automatic Control and Computer Sciences , 41*, 132–140.

[2] Dunlavy, D. M., O'Leary, D. P., Conroy, J. M., & Schlesinger, J. D. (2007). QCS: A system for querying, clustering and summarizing documents. *Information Processing and Management , 43*, 1588–1605.

[3] Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research , 22*, 457–479.

[4] Jones, K. S. (2007). Automatic summarizing: The state of the art. *Information Processing and Management , 43*, 1449–1481.

[5] Lin, C. -Y. (2004). ROUGE: A package for automatic evaluation summaries. *In Proceedings of the workshop on text summarization branches out*, (pp. 74–81). Barcelona, Spain.

[6] Lin, C. -Y., & Hovy, E. H. (2003). Automatic evaluation of summaries using N-gram co-occurrence statistics. *In Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology (HLT-NAACL 2003)*, (pp. 71–78). Edmonton, Canada.

[7] Mihalcea, R., & Ceylan, H. (2007). Explorations in automatic book summarization. *In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL 2007)*, (pp. 380–389). Prague, Czech Republic.

[8] Navigli, R., & Lapata, M. (2010). An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. *IEEE Computer Society , 32*.

[9] Porter, M. (1980). An algorithm for suffix stripping. *Program , 14*, 130–137.

[10] Radev, D., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summarization. *omputational Linguistics , 22*, 399–408.

[11] Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). Automatic text structuring and summarization. *Information Processing and Management , 33*, 193–207.

[12] Shen, D., Sun, J. -T., Li, H., Yang, Q., & Chen, Z. (2007). Document summarization using onditional random fields. *In Proceedings of the 20th international joint conference on artificial intelligence (JCAI 2007)*, (pp. 2862–2867). Hyderabad, India.

[13] Svore, K. M., Vanderwende, L., & Burges, C. J. C. Enhancing single-document summarization by combining RankNet and third-party sources. *In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL 2007)*, (pp. 448–457). Prague, Czech Republic.

[14] Wan, X. (2008). Using only cross-document relationships for both generic and topic-focused multi-document summarizations. *Information Retrieval , 11*, 25–49.

[15] Wan, X., Yang, J., & Xiao, J. (2007). Manifold-ranking based topic-focused multidocument summarization. *In Proceedings of the 20th international joint conference on artificial intelligence (IJCAI 2007)*, (pp. 2903–2908). Hyderabad, India.

[16] Yeh, J-Y., Ke, H-R., Yang, W-P., & Meng, I-H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing and Management , 41*, 75–95.

[17] McDonald, D. M., & Chen, H. (2006). Summary in context: Searching versus browsing. ACM Transactions on Information Systems, 24, 111–141.

[18] Fung, P., & Ngai, G. (2006). One story, one flow: Hidden Markov story models for multilingual multi document summarization. ACM Transaction on Speech and Language Processing, 3, 1–16.

[19] Cilibrasi, R. L., & Vitanyi, P. M. B. (2007). The Google similarity measure. IEEE Transaction on Knowledge and Data Engineering, 19, 370–383.

[20] Alguliev, Rasim; Aliguliyev, Ramiz. "Evolutionary algorithm for extractive text summarization". http://www.highbeam.com/doc/1G1-214205320.html.

[21] Stergos Afantenos, Vangelis Karkaletsis, Panagiotis Stamatopoulos. "Summarization from medical documents: a survey".

[22] Xiaojun Wan. "Towards a Unified Approach to Simultaneous Single-Document and Multi-document Summarizations".