# A unified framework for keywords distillation and summarization

Yang Wei

Network Information Center, Shanxi Normal University, Shanxi Linfen, 041004, China

yangw369@126.com

**Keywords:** keywords distillation; document summarization; similarity method; graph based algorithm;

**Abstract.** Keywords distillation and document summarization are important task for many text applications. Existing methods have been conducting these two tasks separately without considering the relationship between the two tasks. In order to capture and make better use of their relationships between these tasks, this paper proposed a unified framework for keyword extraction and summarization. The method is first implemented by constructing a graph which reflect relationship between different size of granularity nodes, and then using graph based algorithm to calculate score of keywords and sentences. Finally, highest score of words and sentences in the document will be chosen as keywords and salient sentence. Experimental results show that our approach outperforms baseline methods.

## Introduction

Key words are defined as the words that express the main idea of a document. Document summarization is a compressed version of a document which covers the main topic of the document. Keywords and summarization are similar because they are both important representations for documents, and they are often sufficiently informative to allow human readers get a feel for the essential topics and main content included in document. Consequently, automatic keywords distillation and summarization have been widely studied for many years. Meanwhile, it is also fundamental to many other natural language processing applications, such as information retrieval, sentiment analysis and so on.

Although we often come across texts form different domains such as scientific papers, news articles and blogs, which are equipped with keywords and summarization by the authors, a large portion of the document are remains untagged. It is beneficial to automatically extract a few keywords and summarization from a given document to deliver the main content of the document [1][2][3]. This paper focus on keyword distillation and summarization for web documents because web document is one of the most popular genres on the web and most web documents have no author-assigned keywords and summarization.

Existing methods mainly use statistical (i.e., frequency of occurrence, inverse document frequency, co-occurrence information) or linguistic information (i.e., term distribution, word position, topics) to extract the most salient words and sentences from document [4]. However, those methods ignore the relationship between different granularity information [5] (i.e., word, sentence, and topic). In other words, the importance of keywords and sentences is determined by the importance of related sentences, topics and words. The idea is borrowed from the phenomena that word is basic unit of document, sentence and topic, and salient words are strongly related to other salient sentences, topic and words.

In this paper, we propose a unified framework to fulfill the above idea. In particularly, the method is first implemented by constructing a graph which reflect relationship between different granularity nodes, and then using graph based algorithm to calculate score of keywords and salient sentences, finally, highest score of words and sentences in the document will be choose as keywords and summarization. Experiments have been performed on dataset consist of 50 news articles and human-annotated keywords and summarization. The results demonstrate the good effectiveness of proposed approach. The use of the relationship between different granularity information can

 

improve both the performance of keywords distillation and document summarization.

The rest of this paper is organized as follows: The related work is presented in section 2. The details of the proposed approach are described in Section 3. Section 4 presents and discusses the evaluation results. Lastly we conclude our paper in Section 5.

## Related Work

The section h Generally speaking, Keywords distillation and summarization methods can be divided into either supervised or unsupervised [6]. In this study, we focus on unsupervised method. Unsupervised methods usually assign each word (or sentence) a saliency score and rank the words (or sentence) in the document. The score is usually computed based on a combination of statistical and linguistic feature, including term frequency, word position, cue words, stigma words topic signature ,lexical chains and so on. Supervised methods are also employed to extract keywords (or sentence). Supervised keywords distillation (or summarization) approaches treat the task as a two class classification problem at the word (sentence) level, where each word is represented by a vector of features. Feature from local content of a document is the key to distillation. The features can be defined form linguistic, such as term significance, term position. Comparably, unsupervised methods rely on a set of heuristic rules to do the extraction.

Summary and keywords distillation can have different forms. Extractive summarization systems collect important sentences form the document in order to generate summary. Abstractive summarization systems try to capture the main concepts in the text, and generate new sentences to represent these main concepts. Document summary can be also either query-relevant or generic. Query-relevant summary should be closed related to the given query, while generic summary should reflect the main topic of the document without any additional clues and prior knowledge. Since creating abstractive summary is a more complex task, most of automatic text summarization systems are extractive systems, in this paper, we focus on abstractive generic document summarization.

An important of unsupervised document summarization and keywords distillation method are that of graph-based methods . Such method assume that the important sentence or paragraphs of a given text document are the most connected entities in more or less semantic structures inside the document. In this direction, the methods construct a graph for each document, with the vertices being the document sentences, and the link being the relationship between document sentences. This method attempt to determine the most connected vertices in the graph. These methods are further classified depending on how the graph are constructed, for example using word co-occurrences, local salience and grammatical relations and so on. These method including TextRank and LexPageRank and so on. The basic idea underlining these methods is that of voting and recommendation. When a sentence links to another one, it is basically casting a vote for that sentence. The higher the number of votes that are cast for a sentence, the higher the importance of sentence is. Moreover, the importance of the sentence casting the vote determines how important the vote itself is. The computation of sentence importance is based on a recursive form, which can be transformed into the problem of solving the principal eigenvector of the affinity matrix.

Our method different form precious method is that not only relationship between words is considered in the ranking model, but relationship between topic and sentence are also exploited to calculate the affinity score of word. We believe the use of interaction between different sizes of granularity can improve the keywords distillation and summarization performance.

## Our Approach

### Framework.

Given a specified document for keywords distillation and summarization, the proposed approach first split the document into different size of granularity including words, sentence and topic. In this study, clustering algorithm is explored to produce topic clusters. The proposed approach is intuitively based on the follow ideas: (1) If word A is heavily linked with other words, topic or sentence, then the word A should be salient. (2) If sentence B is heavily linked with other sentence, topic or word, then

the sentence B should be salient. (3) If topic C is heavily linked with other word, sentence, then the topic C should be salient. These assumptions are similar to PageRank [4] algorithm since they make use of mutual recommendations between different sizes of granularity to rank objects.

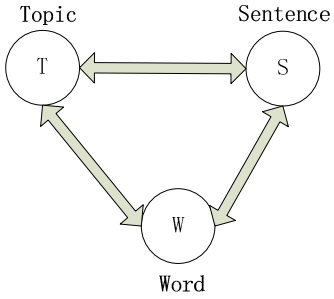


Fig. 1 Heterogeneous Relationship Graph

Figure 1 gives an illustration of above assumptions, where object T, W and S represent topic, word and sentence respectively, and the link between objects represent recommendation relationship. The proposed approach first builds the heterogeneous graph to reflect the above relationships respectively, and then iteratively computes the saliency scores of word, sentence and topic based on the graph. After the algorithm converges and each word and sentence gets its score, the word and sentence with high score are chosen to be keyword and summarization candidate.

**Keyword Distillation and summarization.**

Heterogeneous Graph Building: Given the words collection W of a document, we will create an affinity heterogeneous graph with different sizes of granularities. We use agglomerative clustering algorithm [11] to produce topic clusters. Because it is hard to predict the actual cluster number, we typically set the number k of expected clusters as follows: .Where is the number of all sentences in the document set. Agglomerative is a bottom-up hierarchical clustering algorithm and start with the word as individual clusters, and at each step, merge the most similar or closest pair of clusters, until the number of clusters reduces to desired number k. The similarity between clusters is computed using average link method.

In this study, word (W), topic (T) and sentence (S) are consider as nodes, the document can be modeled as an undirected heterogeneous graph by generating an edge between two nodes if their content similarity exceeds 0. Thus, there are 6 relationships to reflect the influence between different sizes of granularities, Specifically, we define the graph as G=(VW ∪ VT ∪ VS, EWW ∪ EWT ∪ EWS ∪ ESS ∪ ETT ∪ ETS, where VW is a set of word nodes, VT is a set of topic nodes, VS is a set of sentence nodes, and EWW is the link between words, EWT is the link between word and topic, other links are defined similar to EWW and EWT. In the following, we will introduce the methods which calculate the link between words, topic and sentence.

Word to word link: Given the word collection of documents, the semantic similarity between two words can be computed using approaches that are either knowledge-based or corpus-based. Many approaches have been proposed to measure semantic relatedness based on WordNet [12]. In this study, we simply use extend lesk algorithm to model the relatedness between two words.

Word to sentence link: Given the sentence collection and word collection of a document, the link similarity is average similarity between words contained in sentence. The formula is as follows:

$$E_{WS} = \frac{\sum_{Wj \in S} simi(w_i, w_j)}{\sum_{Wk \in D} simi(w_i, w_k)} \qquad (1)$$

Where is the similarity between word wi and wj , wj is the word contained in sentence S, and wk is the word contained in whole document D.

Word to topic link: Given the word collection and topic collection of a document, the link similarity is average similarity between word and topic contained in document. The formula is as follows:

$$E_{WT} = \frac{\sum_{wj \in T} simi(w_i, w_j)}{\sum_{wk \in TC} simi(w_i, w_k)} \qquad (2)$$

Where wj is the word contained in topic T, and wk is the word contained in whole topic collection TC.

Sentence to sentence link: Given the sentence collection of document, the cosine measure is used to compute similarity between two sentences. It is noteworthy that other measures can also be exploited to compute the content similarity between sentence, and we simply choose the cosine measure in this study.

Topic to topic link: Given the topic collection of document, the cosine measure is used to compute similarity between two topics. The formula is as follows:

$$E_{TT} = \frac{\vec{t_i} * \vec{t_j}}{\left\|\vec{t_i}\right\| * \left\|\vec{t_j}\right\|} \qquad (3)$$

Where and are the corresponding term vector of topic and

Topic to sentence link: Given the sentence collection and topic collection of a document, cosine similarity is used to compute relatedness between sentence and topic contained in document. The formula is as follows:

$$E_{WT} = \frac{\vec{t_i} * \vec{t_j}}{\left\|\vec{t_i}\right\| * \left\|\vec{t_j}\right\|} \qquad (4)$$

Where is the term vector of topic T, and is the term vector of sentence S.

Thus, we construct a weighted graph G to reflect the relationships between word, topic and sentence in the document, and use an adjacency matrix with each entry corresponding to the weight of a link defined above to describe the heterogeneous graph.

Affinity score computing: The computation of important score is based on the above following three intuitions. Based on the above intuitions, the importance for word can be deduced from those of all other granularities linked with it and further formulated in a recursive form as follows:

$$sc(w_i) = \varepsilon \sum_j \mathbf{WS}_{ji} sc(s_j) + \phi \sum_j \mathbf{TW}_{ji} sc(t_j) + \gamma \sum_j \mathbf{WW}_{ji} sc(w_j) \qquad (5)$$

Where , and are adjacency relationship matrix for word to sentence, word to topic and word to word respectively. , and are affinity score of word , topic and sentence respectively. , and are weight corresponding to different relationships and must subject to . Similarly, affinity score of sentence and topic can be calculated as follows:

$$sc(s_i) = \alpha \sum_j \mathbf{WS}_{ji} sc(w_j) + \beta \sum_j \mathbf{TS}_{ji} sc(t_j) + \delta \sum_j \mathbf{SS}_{ji} sc(s_j) \qquad (6)$$

$$sc(t_i) = \mu \sum_j \mathbf{WT}_{ji} sc(w_j) + \nu \sum_j \mathbf{TS}_{ji} sc(s_j) + \pi \sum_j \mathbf{TT}_{ji} sc(t_j) \qquad (7)$$

And the matrix from is:

$$\mathbf{w} = \varepsilon \times \mathbf{WS} \times \mathbf{s} + \phi \times \mathbf{TW} \times \mathbf{t} + \gamma \times \mathbf{WW} \times \mathbf{w} \qquad (8)$$

$$\mathbf{s} = \alpha \times \mathbf{WS} \times \mathbf{w} + \beta \times \mathbf{TS} \times \mathbf{t} + \delta \times \mathbf{SS} \times \mathbf{s} \qquad (9)$$

$$\mathbf{t} = \mu \times \mathbf{WT} \times \mathbf{w} + \nu \times \mathbf{TS} \times \mathbf{s} + \pi \times \mathbf{TT} \times \mathbf{t} \qquad (10)$$

We can use the Markov chain model and random reader to formulate the iterative process. In the Markov model, G is treated as the Markov chain, each word as a state and each edge as a transition from one state to another. Usually the convergence of iteration algorithm is achieved when the difference between the scores computed at two successive iteration for any words, sentences and topics falls below a given threshold. A threshold can be set to control the time of iteration. Finally, the highest score of sentence and words are chosen as keywords and summarization.Do *not* print page

numbers: Please number each sheet toward the middle near the bottom (outside the typing area) with a soft pencil.

## Experiments

The proposed approach is compared with the baseline methods relying only on the relationships between words. The baseline uses the graph-based ranking algorithm to compute the words and sentences scores for each document. Table1 give the comparison results of the baseline methods and the proposed methods. Seen form table 1, our approach outperform the baseline method over all there metrics. The results demonstrate the good effectiveness of our approach.

Single-document summarization has been one of the fundamental tasks in DUC2001 and DUC2002, and we used the task for evaluation. DUC2001 provided 30 document sets and DUC2002 provided 59 document sets and generic abstracts of each document set with lengths of approximately 100 words or less were required to be created . The documents were news articles collected from TREC-9. The sentences in each article have been separated and the sentence information has been stored into files. The summary of the two datasets are shown 1 table 1.

TABLE I. summary of data set

| | DUC 2001 | DUC 2002 |
|---|---|---|
| Number of documents | 409 | 667 |
| Number of clusters | 30 | 59 |
| Data source | TREC-9 | TREC-9 |
| Summary length | 200 words | 200 words |

We use the ROUGE toolkit for evaluation, which has been widely adopted by DUC for automatic summarization evaluation. It measure summary quality by counting overlapping units such as n-gram, word sequences and word pairs between the candidate summary and the reference summary.

TABLE II. experimental result of Duc2001 dataset

| System | ROUGE-1 | ROUGE-2 | ROUGE-W |
|---|---|---|---|
| Baseline | 0.38741 | 0.06742 | 0.21752 |
| Our Approach | 0.39855 | 0.06931 | 0.31203 |

TABLE III. experimental result of Duc2002 dataset

| System | ROUGE-1 | ROUGE-2 | ROUGE-W |
|---|---|---|---|
| Baseline | 0.36741 | 0.07542 | 0.11752 |
| Our Approach | 0.38855 | 0.08831 | 0.13203 |

The proposed method is compared with the baseline model which only uses sentence relationship to deduce score of each sentence. Table 2, 3 shows the comparison results on DUC2001 and DUC2002. Seen form the table, our approach can outperform the baseline over almost all three metrics on DUC2001 datasets. The results demonstrate the good result of the proposed model. Moreover, the use of different size of granularity is validated to be effective.

In order to better understand the relative contributions from the sentence nodes, topic nodes and word nodes. We optimize the parameter $\alpha$ , $\beta$ and $\delta$ so as to maximize the guideline on dataset. Without loss of generality, we maximize the GOUGE-1, GOUGE-2 and GOUGE-w on DUC2001 and DUC2002. This approach has been used in many applications and proved to be very effective. Let us denote the 3 parameter by a vector $\theta = [\alpha, \beta, \delta]$, and each dimension of the $\theta$ is denoted as 1,2,3. The optimization problem can be cast as the multi-dimensional function optimization algorithm. The procedure is work as follows: 1,2,3 are taken as a set of directions. Line search moves along the first direction while keeping the other unchanged, so as to maximize the GOUGE-1 score; then it moves form there along the second direction to maximize the score, and so on.

Cycling through the whole set of direction as many time as necessary, until the stops to increase, we obtain the values of the parameters. This method is intuitive and efficient, but it may converge to different local maxima with different start points. Therefore, we perform the procedure multiple times with random start point, and select the parameter that produces the best guideline. The experimental

result is as follows:

TABLE IV. experimental result of parameter on Duc2001 dataset

| guideline | $\alpha$ | $\beta$ | $\delta$ |
|---|---|---|---|
| ROUGE-1 | 0.34 | 0.32 | 0.34 |
| ROUGE-2 | 0.38 | 0.32 | 0.30 |
| ROUGE-W | 0.35 | 0.33 | 0.32 |

TABLE V. experimental result of parameter on Duc2002 dataset

| guideline | $\alpha$ | $\beta$ | $\delta$ |
|---|---|---|---|
| ROUGE-1 | 0.31 | 0.34 | 0.35 |
| ROUGE-2 | 0.31 | 0.32 | 0.37 |
| ROUGE-W | 0.33 | 0.34 | 0.33 |

Seen form the table 4 and 5, the best performance was achieved when all the parameter value almost the same, and performance decrease when one of the parameter is too large or too small. The result demonstrates that both the contribution from sentence, topic and word are important for ranking sentence; moreover the contributions are almost equally important. Loss of either contribution will much deteriorate the final performance.

## Conclusion

In this paper, we proposed a novel approach for keyword distillation and summarization by leveraging an graph based algorithm. The approach first employs a heterogeneous graph to reflect the relationships between word, sentence and topic, and then using an iterative algorithm to calculate score of each candidate keywords. Experimental result on real words dataset demonstrates the good effectiveness of our method.

In this study, we use PageRank as ranking algorithm. In the future work, we will investigate other ranking algorithm such as Hits and Degree, moreover we will do more experiments in larger dataset to test our method.

## References

[1] Turney, P. D. 2000. Learning algorithms for keyphrase extraction.Information Retrieval, 2:303-336.

[2] Barker, K., and Cornacchia, N. 2000. Using nounphrase heads to extract document keyphrases. In Canadian Conference on AI.

[3] Frank, E.; Paynter, G. W.; Witten, I. H.; Gutwin, C.; and Nevill-Manning, C. G. 1999. Domain-specific keyphrase extraction. Proceedings of IJCAI-99, pp. 668-673.

[4] Mihalcea, R., and Tarau, P. 2004. TextRank: Bringing order into texts. In Proceedings of EMNLP2004.

[5] Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1998. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Libraries

[6] Muñoz, A. 1996. Compound key word generation from document databases using a hierarchical clustering ART model. Intelligent Data Analysis, 1(1).