Keyword Extraction from a Single Document Using Centrality Measures

Girish Keshav Palshikar

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

Abstract. Keywords characterize the topics discussed in a document. Extracting a small set of keywords from a single document is an important problem in text mining. We propose a hybrid structural and statistical approach to extract keywords. We represent the given document as an undirected graph, whose vertices are words in the document and the edges are labeled with a dissimilarity measure between two words, derived from the frequency of their co-occurrence in the document. We propose that central vertices in this graph are candidates as keywords. We model importance of a word in terms of its centrality in this graph. Using graph-theoretical notions of vertex centrality, we suggest several algorithms to extract keywords from the given document. We demonstrate the effectiveness of the proposed algorithms on real-life documents.

1 Introduction

In information retrieval (IR), given a collection (corpus) of documents, *index terms* are extracted from each document. The set of index terms for each document helps in indexing, searching and retrieving documents relevant to a given query. The automatic index term extraction techniques in IR identify terms that are neither too specific (i.e., occur only within that document) nor too general (i.e., occur in all documents).

There is a related but different problem of automatically extracting keywords from a *single* document, such as an article, a research paper or a news item. A document is 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 describe a major topic discussed in the document. Typically, only a few keywords (10 or 20) are associated with each document, whereas IR associates a large number (hundreds) of index terms with each document in a collection. Moreover, 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.

A. Ghosh, R.K. De, and S.K. Pal (Eds.): PReMI 2007, LNCS 4815, pp. 503–510, 2007.

The *keyword extraction problem* consists of extracting k keywords from the given document, where $k \geq 1$ is a given integer. We assume that some preprocessing steps are performed on the document before giving it as input to the keyword extraction algorithm. Most approaches to keyword extraction are considered *statistical* in nature. For example, a naive approach is to find the k most frequent words in the document. However, this does not usually work (even after removing stopwords) because keywords are important (meaningful), not necessarily frequent.

In this paper, we propose a graph-theoretical (structural) notion to capture the idea of importance of a word in the given document. We represent the given document as an edge-labeled graph and propose that most keywords would correspond to *central* vertices in this graph. Using appropriate graph-theoretical notions of centrality of a vertex, we then suggest various algorithms to extract keywords from the given document. Since the edge labels are based on the frequency of the words in the document, our approach is not purely structural, but rather a hybrid one, where structural and statistical aspects of the document are represented uniformly in a graph. As a side benefit, the schemes proposed in this paper provide a natural way to order the extracted keywords in terms of their importance (i.e., in terms of the centrality of the corresponding vertices).

Section 2 contains the technical approach and the various keyword extraction algorithms. Section 3 describes results of experiments done to demonstrate the utility of the proposed approach. Section 4 discusses some related work and section 5 contains our conclusions and outlines some further work.

2 Approach

We consider a sentence as an unordered set of words (treating multiple occurrences of a word in a sentence as a single occurrence). Then a document is a collection of sets of words. We simplify the document by applying the following preprocessing steps to it: (i) use of abbreviations (e.g., replace `United Nations` with `UN`) (ii) removal of all numbers (a number is rarely a keyword) (iii) removal of stop words (iv) removal of punctuation symbols (except sentence terminators . ? and !) (v) removal of infrequent words (i.e, words that occur less than a specified number of times in the document) (vi) stemming (e.g., using the Porter stemming algorithm).

2.1 Eccentricity-Based Keyword Identification

Definition 1. *A* term graph *is an undirected edge-labeled graph* $G = (V, E, w)$ *where each vertex in* V *corresponds to a term (i.e, a word) in the document,* E *is the set of edges and* $w : E \rightarrow (0, 1]$ *is the edge weight function. There is an undirected edge between terms* u *and* v *(*$u \neq v$*), with weight* $w(u, v)$*, iff* $0 < w(u, v) \leq 1$*, Edge weights indicate dissimilarity (distance) between terms. We assume that* w *is symmetric i.e.,* $w(u, v) = w(v, u), \ \forall u, w$ *and* $w(u, u) = 0 \ \forall u$.

One simple scheme to define the weights in the term graph is as follows. Let $c(\{u, v\})$ denote the number of sentences in which terms u and v both occur together. Then

$$w(u, v) = \begin{cases} 0 & \text{if } c(\{u, v\}) = 0 \\ \frac{1}{c(\{u,v\})} & \text{otherwise} \end{cases}$$

If terms u and v do not co-occur in at least one sentence, then the weight $w(u, v) = 0$ and the edge uv is absent in the term graph. Otherwise, the weight $w(u, v) = 1/c(\{u, v\})$. For example, for the collection of sets $\{\{a, b\}, \{a, b\}, \{a, b, c\}, \{a, b\}, \{a, c\}, \{a\}\}$, the $w(a, b) = 0.25$, $w(a, c) = 0.5$ and $w(b, c) = 0.5$. Clearly, lower weight indicates higher strength of the (co-occurrence) relationship between the terms.

In general, the term graph may be disconnected. If the term graph is disconnected, we apply the keyword identification procedure to each component and then return the union of the keywords from each component (alternatively, we may select more keywords from a larger component). To simplify matters, we assume in the following that the term graph is connected.

Consider the following news item posted in TIME magazine's issue for Nov. 21, 2006 (`www.timeasia.com`), with the headline
`Nepal, rebels sign peace accord`.

> Nepal's government and Maoist rebels have signed a peace accord, ending 10 years of fighting and beginning what is hoped to be an era of peaceful politics in the Himalayan kingdom. In a ceremony, Nepali Prime Minister Girija Prasad Koirala and Maoist leader Prachanda signed the agreement on Tuesday, which brings the rebels into peaceful multiparty democratic politics.
>
> "The politics of violence has ended and a politics of reconciliation has begun," Koirala said after the signing. Last week, the Maoists agreed to intern their combatants and store their arms in camps monitored by the United Nations. Nepal's Maoist rebels have been fighting an armed rebellion for 10 years to replace the monarchy with a republic. More than 13,000 people have been killed in the fighting. According to the agreement, any use of guns by the rebels will be punished. The democratic government and the Maoists have agreed to hold elections in June 2007 for constituent assembly that will decide the fate of the monarchy.
>
> "This is a historic occasion and victory of all Nepali people," Chairman of the Communist Party of Nepal Prachanda said at the signing ceremony, witnessed by political leaders, diplomats, bureaucrats and the media. "A continuity of violence has ended and another continuity of peace has begun," Koirala said. "As a democrat it was my duty to bring non-democrats into the democratic mainstream. That effort is moving ahead towards success. "The peace agreement is an example for the whole world since it is a Nepali effort without outside help," he added. The challenge Nepal now faces is holding constituent assembly elections in a peaceful manner.
>
> Meanwhile, Maoist combatants continued to arrive in seven camps across the country Tuesday, albeit without United Nations monitoring. A tripartite agreement between the government, Maoists and the U.N. has to be signed before the U.N. can be given a mandate to monitor arms and combatants. "I

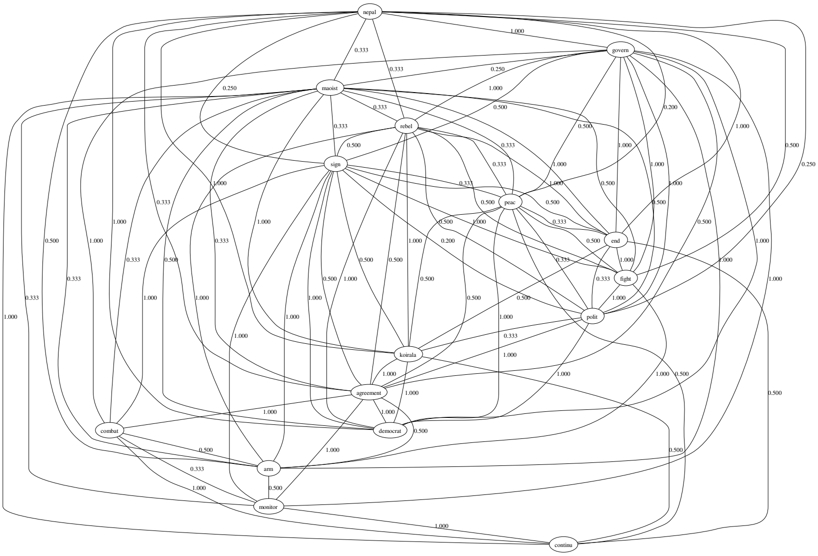

Fig. 1. Term graph for the news item

> hope that we will quickly be able to reach tripatriate agreement on the full modalities for the management of arms and armies clarifying essential detail," said Ian Martin, Special Representative of the United Nations Secretary General in Nepal. The Maoists will now join an interim parliament and an interim government, as early as next week, following the agreement.

The pre-processing steps described earlier were applied to the document which reduces the number of words to 97 (from the original 372). The resulting term graph has 97 vertices (one for each word) and 797 edges. The task now is to choose, say 10, keywords from this set of 97 keywords. Our hypothesis is that the keywords are *central* in some sense in the given document. How does one compute the centrality of a word in a document? Since we have represented the document as a graph, we can now appeal to graph-theoretic notions of centrality of vertices. To simplify the graph drawing, Fig. 1 shows another term graph for the same news item, where we have now retained only those words that occur at least 3 times. The resulting term graph has 16 vertices and 84 edges.

Definition 2. *Given a term graph G, the* distance $d(u, v)$ *between two terms 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 the maximum distance from u to any other vertex v in G i.e., $\epsilon(u) = max\{d(u, v)|v \in G\}$.*

Computing the eccentricity of a given vertex is easy. Dijkstra's single source shortest path algorithm [1] 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

words to be more important. One approach to automatic selection of keywords is now clear. List the terms in increasing order of their eccentricities in G and pick the first k words (having the least eccentricity). Using the proposed algorithm, the eccentricities (and degrees) of the first 16 words (when ordered in terms of increasing eccentricity) in the above news item are as follows:
$\{(nepal, 2.0, 67), (agreement, 2.0, 48), (peac, 2.0, 40), (sign, 2.25, 46),$
$(polit, 2.25, 41), (maoist, 2.333, 54), (rebel, 2.333, 34), (leader, 2.333, 29),$
$(prachanda, 2.333, 29), (ceremoni, 2.333, 29), (end, 2.333, 19), (arm, 2.5, 34),$
$(govern, 2.5, 34), (hope, 2.5, 30), (koirala, 2.5, 23), (fight, 2.5, 21)\}$

While there is no problem in choosing the first 5 keywords, there is some ambiguity in the choice of the last 5 keywords; viz., the 6 words {`maoist, rebel, leader, prachanda, ceremoni, end`} all have the same eccentricity 2.333. How do we choose 5 words from these 6 words? We prefer keywords which have a high degree in the term graph. Then the next 5 keywords are {`maoist, rebel, leader, prachanda, ceremoni`} with degrees 54, 34, 29, 29, 29, which are all higher than the degree 19 of the word `end`. The final set of 10 keywords for the above news item, chosen using this heuristic is: {`nepal, agreement, peac, sign, polit, maoist, rebel, leader, prachanda, ceremoni`}.

2.2 Using Other Centrality Measures

Apart from eccentricity, betweenness [2] and *c*loseness are two notions of vertex centrality, among others, from social network analysis [7]. Either of these could be used as a measure of centrality to identify keywords from the term graph representation of a given document.

Definition 3. *[7] 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.

Computing closeness for a vertex is similar to computing the eccentricity of that vertex. Vertices with lower value of closeness are more central. So the approach to extract the keywords is same as for eccentricity. We compute the closness value for each vertex, sort the vertices in the ascending order of their closeness values and pick the lowest k vertices as keywords. As earlier, we use the vertex degree to break ties among vertices that have the same closeness value (vertices with larger degree are more preferable). The closeness method yoields the following 10 keywords for the news item: {`nepal, peac, maoist, agreement, sign, polit, rebel, arm, govern, leader`}. These keywords are similar to the ones produced by the eccentricity method.

2.3 Proximity-Based Keywords Identification

Yet another approach to keyword extraction is possible by applying the link analysis technqiues, such as the cGraph algorithm [3]. Consider a database of research papers, which can be represented as a collection of sets of authors (each paper is represented as a set of authors). Each paper is considered as a *link* among the co-authors of that paper. The database of papers is then represented as a directed, edge-labeled *collaboration graph*, where each vertex corresponds to an author and there is a directed edge from u to v (and from v to u) if they have co-authored at least one paper together. For any vertex u, the sum of the weights on outgoing edges from u must be 1. The weight of every edge is a real number in the interval $[0, 1]$. There are several ways to compute the weight of the edge from u to v; the simplest way is as follows [3]:

$$w(u, v) = \widehat{P}(v|u) = \frac{\sum_{L:(u,v)\subseteq L} \left(\frac{1}{|L|-1}\right)}{\sum_{L:u\in L} 1}$$

Here, the weight $w(u, v)$ is an estimate $\widehat{P}(v|u)$ of the probability that a link that contains u also contains v. The denominator is the count of links that contain u. For example, for the collection of sets
$\{\{a, b\}, \{a, b\}, \{a, b, c\}, \{a, b\}, \{a, c\}, \{a\}\}$,
the weight $w(a, b) = \frac{1+1+0.5+1}{6} = 0.583$. Here, higher weight indicates higher strength of the co-occurrence (co-authorship) relationship between the authors. To make it a suitable dissimilarity measure, we actually use the inverse of $w(u, v)$ as the edge label. Thus, in the example, $w(a, b) = 1/0.583 = 1.715$.

By considering each sentence as a link, we can analogously construct a collaboration graph for a document, with the words as vertices. We can now apply any of the above algorithms to the collaboration graph to obtain a set of keywords. However, there is an additional possibility: that of using the proximity measure [3] between two vertices u and v, defined as:

$$prox(u, v) = \sum_{p\in V(u,v,m)} \prod_{e_i\in p} P(e_i|a_j \forall j < i)$$

where $V(u, v, m)$ is the set of all non-self-intersecting walks p of length $\leq m$ from u to v and e_i is a directed edge in such a walk p. $P(e_i|e_j)$ is the probability that the edge e_i will be added to a path p from u given that the earlier vertices already in p are $a_1, a_2, \ldots$. The function *prox* considers only paths of length at most m steps. All vertices which are not reachable from u in at most m steps are given a very low proximity. Value of m is usually low, say 3 or 4. In the computation of eccentricity, we could use this proximity measure, instead of the distance of the shortest paths. For a vertex u, we consider all vertices $\{v_1, v_2, \ldots\}$ reachable from u in at most m steps and consider the eccentricity of u to be the largest among these values: $\epsilon(u) = max\{prox(u, v_1), prox(u, v_2), \ldots\}$.

3 Experimental Results

We have received very positive feedback on the keywords generated by the algorithms on specific documents. Most users agreed on most of the documents that the keywords generated by the algorithms were good. However, we carried out the following experiment to get a more objective feedback about the quality of the generated keywords. We collected 64 news stories from well-known English news magazines in India, under 5 categories: environment, economy, defence, health and cinema. Each news also had a headline. The everage size of the stories was 1352 words and 8208 characters (without the headline). We generated 10 keywords for each news story (without its headline) and computed how many of these 10 keywords also occurred in the headline. The idea is that the headline would generally contain the main keywords in the news story. For example, the headline `Nepal rebels sign peace accord` of the earlier news story contains 4 keywords `Nepal rebels sign peace`. Note that the a synonym `agreement` of the fifth word `accord` in the headline is a keyword. Nevertheless, we add that the headline is often written to be catchy and exciting, so that it does not always contain the main keywords. Hence we expect only a partial overlap between the generated keywords and the words in the headline. We extracted 10 keywords for each of the 64 news items and compared them with the corresponding headlines. On the average, about 3 to 4 keywords (out of 10) appeared in the headline. We consider this result as rather satisfactory and supporting our keyword extraction algorithms. We also carried out a similar experiment on a set of 300 smaller (one paragraph) financial news items, with similar results.

4 Related Work

[6] presents a closely related approach called KeyGraph for extracting keywords from a document. In Keygraph, a document is represented as a graph, whose vertices are then clustered so that each cluster represents a concept or topic in the document. Then highest ranking words from each cluster are extracted and returned as keywords. In [5], a cognitive process based on priming and activation is used for extracting keywords. The readers's mind is a network of concepts and reading a document activates some concepts, which in turn activates the related concepts and so on. Vertices in a graph represent words and edges denote the associations between words. A mathematical model for activation spreading is specified, based on word frequency and recency of activation. At the end, most highly activated terms are returned as keywords. In [4], a probability distribution of co-occurrences between frequent words and all other words is analysed for bias using χ^2-measure and terms with unusual bias are selected as keywords.

5 Conclusions and Further Work

Keywords are used to characterize or summarize the main topics in a document. Extracting a small set of keywords from a single document is an important problem in text mining. Keyword extraction techniques are beginning to be used in other applications such as web-page clustering and discovering emerging topics by analyzing co-citation graphs. In this paper, we proposed a hybrid structural and statistical approach to extract keywords. We represent the given document as an undirected graph, whose vertices are words in the document and the edges are labeled with a dissimilarity measure between two words, derived from the freuqnecy of their co-occurrence in the document. We then propose that central vertices in this graph are candidates as keywords, where we model the importance of a word in terms of its centrality in this graph. Using appropriate graph-theoretical notions of centrality of a vertex - such as eccentricity, closeness, betweenness and proximity - we suggested several algorithms to extract keywords from the given document. The proposed keyword extraction algorithms appear to be effective when tested on a set of real-life news stories. We found that reducing the number of words (vertices) by retaining only those words that appear at least a user-specified minimum number of times does not decrease the effectiveness of the extracted keywords.

For further work, we are working on refining the proximity-based approach. We also wish to modify our algorithms to take into account other factors such as the position of words, rather than rely purely on their frequencies. We are trying to define an objective measure for the goodness of the extracted keywords, so as to facilitate the comparison of various keywords extraction algorithms. Currently, the number of keywords to be extracted is specified by the user; we are trying to automatically arrive at this number.

Acknowledgements. I thank colleagues in TRDDC for useful discussions and help. Thanks to Dr. Manasee Palshikar for her support and encouragement.

References

1. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms 2/e. MIT Press, Cambridge (2001)
2. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry 40, 35–41 (1977)
3. Kubica, J., Moore, A., Cohn, D., Schneider, J.: Finding underlying structure: A fast graph-based method for link analysis and collaboration queries. In: Proc. 20th Int. Conf. on Machine Learning (ICML 2003) (2003)
4. Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. Int. Journal on AI Tools 13(1), 157–169 (2004)
5. Matsumura, N., Ohsawa, Y., Ishizuka, M.: Pai: Automatic indexing for extracting assorted keywords from a document. In: Proc. AAAI 2002 (2002)
6. Ohsawa, Y., Benson, N.E., Yachida, M.: Keygraph: automatic indexing by co-occurrence graph based on building construction metaphor. In: Proc. Advanced Digital Library Conference (ADL 1998), pp. 12–18 (1998)
7. Wasserman, S., Faust, K., Iacobucci, D.: Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge (1995)