Term Clustering based on Lengths and Co-occurrences of Terms

Michiko Yasukawa and Hidetoshi Yokoo

Department of Computer Science
Gunma University
1-5-1 Tenjin-cho, Kiryu, Gunma 376-8515, Japan

{*michi, yokoo*}*@cs.gunma-u.ac.jp*

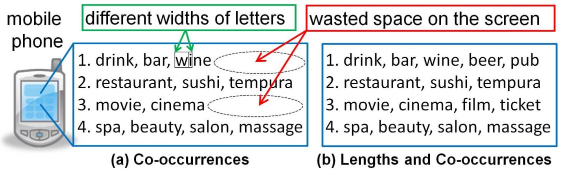
Abstract *Document clustering is useful for addressing vague queries and managing large volumes of documents. However, conventional algorithms for document clustering do not consider the lengths of terms in the cluster labels. Some cluster labels have considerably different lengths. Cluster labels with different lengths result in wasted space on the screen. To counter this problem, we have developed a new method for term clustering. Our method considers both lengths and co-occurrences of terms while clustering them. Therefore, our method can achieve an efficient document search even with limited area on the screen.*

Keywords Information Retrieval, Web Documents

1 Introduction

A single-term query is usually ambiguous, and it results in a large number of documents. Search result clustering is very effective in managing such a large number of searched documents[1][2][3]. We have developed a model for classifying a set of searched documents into clusters of related terms[4]. The developed system was found to be useful for PC users but not for the users of mobile terminals. This is because the number of terms in each cluster label varies. Further, the number of letters in each term varies. For example, the number of letters in *cafe* is less than half the number of letters in *restaurant*. The situation worsens when we use a proportional font to represent the cluster labels. In a proportional font, the space required to represent the letter "w" is larger than that required for "i," thereby resulting in wasted space on the screen (Figure 1 (a)). In order to make optimal use of the limited space on mobile terminals, we propose a new clustering method. Our proposed method generates a set of related-term clusters that fit in a rectangular region (Figure 1 (b)). The related-term clusters are based on the co-occurrence of related terms and are supposed to be intuitively better understood by users than randomized related terms. This is because co-occurrent terms in documents are supposed to be terms associated with each other. According to Meyer and Schvaneveldt[5], pairs of associated terms such as (BREAD-BUTTER) and (NURSE-DOCTOR) are more promptly recognized by users than pairs of unassociated terms such as (BREAD-DOCTOR) and (NURSE-BUTTER). Hence, the proposed clusters are considered to be effective in the selection of preferable terms on the display screen by users.

Proceedings of the 14th Australasian Document Computing Symposium, Sydney, Australia, 4 December 2009. Copyright for this article remains with the authors.



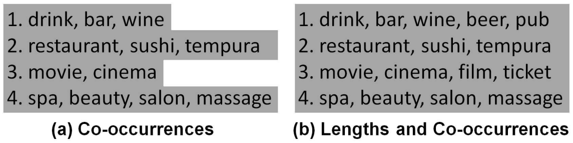
(a) Co-occurrences

1. drink, bar, wine
2. restaurant, sushi, tempura
3. movie, cinema
4. spa, beauty, salon, massage

(b) Lengths and Co-occurrences

1. drink, bar, wine, beer, pub
2. restaurant, sushi, tempura
3. movie, cinema, film, ticket
4. spa, beauty, salon, massage

Figure 1: Comparison between clustering methods



(a) Co-occurrences

1. drink, bar, wine
2. restaurant, sushi, tempura
3. movie, cinema
4. spa, beauty, salon, massage

(b) Lengths and Co-occurrences

1. drink, bar, wine, beer, pub
2. restaurant, sushi, tempura
3. movie, cinema, film, ticket
4. spa, beauty, salon, massage

Figure 2: Comparison between occupied areas

Further, users do not have to input each letter in the terms when using related-terms clusters. Users may simply select preferable terms on the screen. Moreover, users have an option of selecting a number on the screen for accessibility; for example, they can push button "2" to indicate a set of terms "restaurant, sushi, tempura" at once. As shown in Figure 2, clustering (b) can be more informative than clustering (a) because the results of the former occupy a larger area on the screen.

2 Proposed Method

Our proposed method is described as follows. We assume that mobile users will enter a short query (typically just one term such as a location name) and will seek suggested terms in response to the query; The system should present a well-organized menu of various suggestions in response to the query, and the user will then select one of the suggestions in the menu as an expanded requirement. After this, the system will present a number of web pages related to that expanded requirement. The proposed method is explained in the following paragraphs.

In our proposed clustering method, we first generate a set $L(Q)$ of terms related to the short primary query

Q and determine the relationship between the elements in $L(Q)$. Specifically, we denote the i-th term selected from $L(Q)$ as $t_i(Q)$. For example, for $Q =$ "Shinjuku," $t_i(Q) =$ "restaurant" may be a related term. Next, we define a query consisting of Q and $t_i(Q)$ as $q_i = \langle Q, t_i(Q)\rangle$. From the web pages that are searched by q_i, we extract the adjacent terms of $t_i(Q)$. We call these terms *association terms* of $t_i(Q)$. Let $A_i(Q)$ be the list of association terms of $t_i(Q)$. Note that $A_i(Q)$ may include another related term $t_j(Q)$. This is because the term $t_j(Q) =$ "sushi" may be adjacent to $t_i(Q) =$ "restaurant" in the web pages of $Q =$ "Shinjuku." In order to determine the relationship between the terms $t_i(Q)$ and $t_j(Q)$ with respect to the primary query Q, we define their co-occurrence score, $score_{ij}$, by

$$score_{ij} = (and_{ij}/or_{ij}) * (1 + \log(and_{ij})), \quad (1)$$

where and_{ij} denotes the number of lists of association terms that include both $t_i(Q)$ and $t_j(Q)$ and or_{ij} denotes the number of lists of association terms that include either $t_i(Q)$ or $t_j(Q)$. The equation is defined empirically on the basis of our exploratory experiments. We have observed that in order to consider the co-occurrences of terms, the equation should amplify and_{ij}; however, the amplification must not be excessive.

In the algorithm, we set the minimum and maximum acceptable lengths per line of the display screen to $\ell_{\min}$ and $\ell_{\max}$, respectively.

***Algorithm*—**Rectangular Clustering

(Step 1) Read a list $L(Q)$ of terms related to every query Q. Determine the length of each term in the list $L(Q)$. Here, the length is the actual length of the term on the screen.

(Step 2) For every pair $t_i(Q)$ and $t_j(Q)$ of terms in $L(Q)$, calculate $score_{ij}$ using equation (1).

(Step 3) For every term $t_i(Q)$ in $L(Q)$, select the two highest co-occurrence terms $t_{k_1}(Q)$ and $t_{k_2}(Q)$. Then, merge the selected terms to generate a primitive cluster $c_i = \langle t_i(Q), t_{k_1}(Q), t_{k_2}(Q)\rangle$. Note that terms may overlap in the primitive clusters. Before proceeding to Step 4, calculate the score of c_i as the sum of $score_{ik_1}$ and $score_{ik_2}$.

(Step 4) Remove overlapping terms from clusters. If there are overlapping terms among multiple clusters, retain only those terms that are in the cluster with the highest co-occurrence score. Eliminate all terms that are repeated in other clusters.

(Step 5) Determine the total length of each cluster to alter the cluster. If the total length of a cluster is less than ℓ_{min}, merge the cluster with another cluster. If two clusters c_i and c_j had common terms when they were primitive clusters, they can be merged.

(Step 6) Determine the total length of each cluster to decide whether to select or reject the cluster. If the total length is adequate, select the cluster for a cluster label. If the total length is less than ℓ_{min}, reject the cluster. If the total length is greater than ℓ_{max}, select terms from the cluster as many as possible until the total length is in the range between ℓ_{min} and ℓ_{max}.

(Step 7) Remove the terms used for the cluster labels from the list $L(Q)$. If $L(Q)$ is empty or if no more cluster labels are generated, write out the cluster labels, and end the algorithm. Otherwise, return to Step 3 and continue.

3 Implementation

In order to measure the actual length of a term on the screen, we use Graphviz[1] and IPA font[2]. With this software and font, we can generate the text image of the term. Then, we measure the lengths of terms by using the generated images. In order to calculate $score_{ij}$, we used a tool called GETA[3] for large-scale text retrieval. We used Search API of Yahoo!JAPAN[4] to collect search results of (1) related terms; (2) URLs, titles, and summaries; and (3) web pages. An actual application of the proposed method in a mobile web search system has been demonstrated in [6].

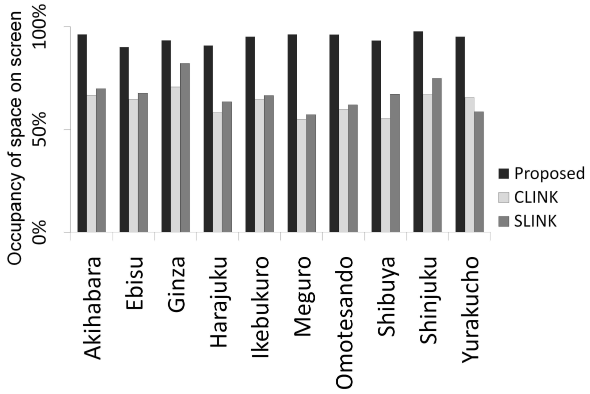


Figure 3: Length of terms on the screen

4 Experiment

We compare the proposed algorithm with two other clustering algorithms— complete-link clustering (CLINK) and single-link clustering (SLINK). These algorithms are widely used conventional algorithms and have been described in detail in [7]. While our algorithm considers both lengths and co-occurrences of terms, these conventional algorithms consider only co-occurrences of terms.

[1] http://www.graphviz.org/

[2] http://ossipedia.ipa.go.jp/ipafont/

[3] http://geta.ex.nii.ac.jp/e/

[4] http://developer.yahoo.co.jp/

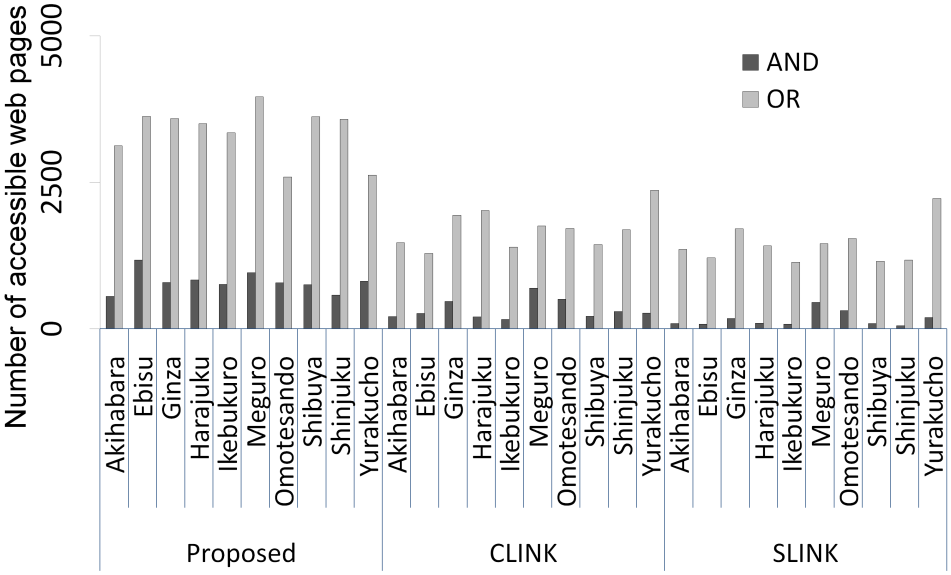


Figure 4: Accessible web pages for different terms on screen

The names of major places in Tokyo were used as queries in the experiment. For each query, 100 related terms and 10,000 web pages were obtained. Term clusters that fit in a rectangular region of 160×160 pixels were generated using the 16-pixel proportional font. In the experiment, the parameters of CLINK and SLINK were adjusted to generate as many clusters as possible with each cluster having two or more terms.

4.1 Area Occupied on Screen

One of the key features of the proposed method is that it takes into consideration the term lengths, thereby optimizing the use of screen space. We investigated the total length ℓ_s of the clusters for each query and then calculated the ratio of the total length ℓ_s of the clusters to the total length ℓ_r of the lines in the rectangular region. In Figure 3, we can observe that the term clusters generated by using the proposed algorithm occupy a larger area on the screen as compared to SLINK and CLINK. Hence, the proposed algorithm is considered to provide more information than others.

4.2 Efficiency of Web Search

Another key feature of the proposed method is its high search efficiency. In Figure 4, “AND” indicates the condition that the web pages include two or more terms in the clusters, e.g., ((restaurant AND sushi) or (sushi AND tempura) or (tempura AND restaurant)). Further, “OR” indicates the condition that the web pages include one or more terms in the clusters, e.g., (restaurant OR sushi OR tempura). The proposed algorithm enables users to obtain desired pages more efficiently than conventional algorithms.

5 Conclusion

We have proposed a new clustering method that enables efficient term clustering in a mobile web search. In the proposed method, a set of primitive clusters are generated on the basis of the co-occurrences of terms. Then, the clusters are altered on the basis of the co-occurrences and lengths of terms. Finally, the clusters are evaluated and adjusted on the basis of the lengths of terms. Term clusters obtained by the proposed method effectively use a small rectangular region on the screen. Hence, the clusters are informative and can aid mobile users to search documents efficiently. In the future, we intend to apply the proposed method to various information retrieval systems.

References

[1] O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *SIGIR*, pages 46–54, 1998.

[2] D. Beeferman and A. L. Berger. Agglomerative clustering of a search engine query log. In *KDD*, pages 407–416, 2000.

[3] S. Osinski. Improving quality of search results clustering with approximate matrix factorisations. In *ECIR*, pages 167–178, 2006.

[4] M. Yasukawa and H. Yokoo. Related terms clustering for enhancing the comprehensibility of web search results. In *DEXA*, pages 359–368, 2007.

[5] D. E. Meyer and R. W. Schvaneveldt. Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, 90:227–234, 1971.

[6] M. Y. Yasukawa and H. Yokoo. Clustering search results for mobile terminals. In *SIGIR*, pages 880–880, 2008.

[7] C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. The MIT PRESS, 1999.