WISE: Hierarchical Soft Clustering of Web Page Search Results based on Web Content Mining Techniques

Ricardo Campos
Centre of Human Language Technology and Bioinformatics
University of Beira Interior
ricardo.campos@ipt.pt

Gaël Dias
Centre of Human Language Technology and Bioinformatics
University of Beira Interior
ddg@di.ubi.pt

Célia Nunes
Centre of Human Language Technology and Bioinformatics
University of Beira Interior
celia@mat.ubi.pt

Abstract

Typically, search engines are low precision in response to a query, retrieving lots of useless web pages, and missing some other important ones. In this paper, we study the problem of the hierarchical clustering of web pages search results. In particular, we propose an architecture called WISE [1], a meta-search engine that automatically builds clusters of related web pages embodying one meaning of the query. These clusters are then hierarchically organized and labeled with a phrase representing the key concept of the cluster and the corresponding web documents. The system which is a web-based interface (soon available at wise.di.ubi.pt), introduces some interesting new ideas, such as the pre-selection of the retrieved web pages, the capacity to statistically detect phrases within documents and the representation of documents based on their most relevant key concepts by using web content mining techniques. The final step of the system is supported by a graph-based overlapping clustering algorithm which groups the selected documents into a hierarchy of clusters.

1. Introduction

Current search engines return lists of ranked urls with their title and a short description, known as snippet. One of their main problems is that their induced relevance may not satisfy the user's needs when browsing for relevant web pages, as search engines fail to present the results in an appropriate manner. In this context, some scientific literature has been published, but all have ignored the potential of using web content mining techniques to semantically analyze a web page. Without this analysis, systems are not ready to completely understand the contents of web pages. As a consequence, the ambiguity and the synonymy problem remain unsolved. To tackle these drawbacks, we developed a meta-search engine called WISE [1]. Through the use of web content mining techniques introduced in the context of the Webspy software [16] and statistical methodologies for phrase detection with the SENTA software [4] to semantically represent the content of web documents, the system generates soft hierarchical clusters on the fly, without pre-defined groups or pre-built knowledge bases, by applying a clustering algorithm called PoBOC [3], which is graph-based and allows a document to be in multiple clusters (overlap). In order to do so, we use the whole text of the web pages, instead of web snippets, which are based on unknown retrieval models that can not guarantee a correct definition of the content of the web pages and are too short to produce semantic knowledge.

In summary, our results are innovative as the system as a whole, and not just part of it, is language and topic independent, unlike most of the methodologies proposed so far, showing interesting capacity to find, analyze and semantically understand the content of the web pages, disambiguate the query and organize all the information retrieved by any search engine in response to a query, by combining hierarchical soft clustering and phrases with the use of web content mining techniques. In this paper we provide a summary of the literature, explain our main contributions defining the overall architecture, draw some conclusions and propose some ideas for future work.

2. Related Work

Lots of different scientific papers have been published related to web snippet clustering. In this section, we present the different approaches proposed so far. Table 1 summarizes all these works.

Table 1. Different approaches of web snippet clustering.

	Flat Clustering	Hierarchical Clustering
Single Words	[8], [9]	[2], [6]
Phrases	[15], [17],[18]	[5], [7], [11], [12], [13], [14], [19]

The first step in this kind of process is to select a set of keywords to describe the documents. For such, the different approaches use vector space model ([5], [6], [15]), shared n-grams ([9], [14], [17], [18], [19]), concept lattice ([2]), topicality ([12]) or lexical affinity ([13]).

They also distinguish themselves by the match of the two following items: (1) simple words or phrases and (2) flat or hierarchical clustering as shown in Table 1. The simplest case is the match between flat clustering (only one-level partitioning of the data) and single words, whilst the general case and best approach [5] is the match between hierarchical clustering and phrases.

All the works presented so far suffer from the same problems: (1) they are not able to capture the real contents of the documents as they do not analyze them and (2) as a consequence, ambiguity and synonymy problems remain unsolved. Besides, all the works, contrary to our work, use stop words and stemming algorithms, turning their solutions faster, but putting in question their applicability to other languages when dealing with the entire web as [7] refers. Many existing clustering algorithms also require the user to specify the number of clusters as an input parameter and/or use thresholds to define if a document belongs to a cluster or not. To deal with these drawbacks, we propose to use soft hierarchical clustering with phrases, with the use of web content mining techniques to semantically extract key concepts from documents.

3. The Architecture of Wise

The architecture of WISE is simple, and each part of it can be easily used in separate by other researchers. WISE is a web search interface system, that in response to a query returns a page with a set of clusters and their associated key concepts which are keywords representing the web documents. Below each keyword there exists a list of urls, in one or more clusters, so that the user can easily choose the web page he wants to see. Our algorithm can be divided into five steps: (1) Search results gathering, (2) Selection of relevant web pages, (3) Document processing for phrase extraction, (4) Document processing for key concept extraction and (5) Hierarchical clustering and labeling. The first step is formalized in 1:

$$R = r_s(d_i \mid q),\ \forall_i, i = 1,...,n \qquad (1)$$

where q is the query, d_i a document, r_s a function of a search engine s, that calculates the relevance between q and d_i. In the second step, we apply a selection function over R, defined in Equation 2:

$$R' = select(R) \qquad (2)$$

where *select* is the algorithm that selects the most relevant documents over the set of web pages search results. In the third step, each document of R' is processed to extract phrases. This step is defined in Equation 3:

$$S = senta(d_j), d_j \in R', j = 1,...,\#R' \qquad (3)$$

where S is the set of documents in which all phrases replace the relevant sequences of single words. In the fourth step, formalized in Equation 4, we represent each document d_j of S, as a vector of key concepts $kc_{j.} = (kc_{j1}, kc_{j2,...,} kc_{jm})$, $j=1,...,\#R'$

$$kc_{j.} = webspy(d_j \mid q), d_j \in S \qquad (4)$$

where m is the number of key concepts for d_j retrieved by Webspy [16] taking into account the query q. At this step, we achieve flat clustering, although we go further achieving hierarchical clustering. Flat clustering can be seen as defining each key concept kc_h, $h = 1,...,u$, as a cluster C_h, so that it contains all the documents d_j, where d_j is relevant through Webspy to q and kc_h, as given by 5:

$$C_h = \{d_j \mid d_j \in S \wedge dt(d_j, kc_h)\} \qquad (5)$$

where *dt* is the decision tree model that is implemented in Webspy and defines if a key concept is relevant to any document. Following, we define the list of flat clusters such as $FC = \{C_h \mid h = 1,...,u\}$. In the fifth step, formalized in Equation 6 and 7, we first need to represent each document d_v, $v=1,...,\#C_h$, of each cluster C_h, as a vector of key concepts $kc_{v.} = (kc_{v1}, kc_{v2,...,} kc_{vt})$

$$kc_{v.} = webspy(d_v \mid kc_h), d_v \in C_h \qquad (6)$$

where t is the number of key concepts for d_v retrieved by Webspy taking into consideration the query kc_h. This number can be different for different documents as mentioned earlier. Then, we apply the PoBOC algorithm [3] to obtain the set of hierarchical clusters HC:

$$HC = poboc(\{sim(kc_{v.}, kc_{v'.}), v \neq v' \wedge v, v' = 1,...,\#C_h\}) \qquad (7)$$

where *sim* is the well known cosine similarity measure. Finally, the label of the hierarchical clusters is obtained by *HC'*, applying the label function

$$HC' = label(HC) \qquad (8)$$

3.1. Selection of relevant web pages

We developed a new extraction method of relevant web pages that ignores some of the retrieved documents, the fewer relevant and adds some others. The algorithm selects as a relevant document (1) any absolute url retrieved as such (i.e. the web domain) and (2) any url which number of occurrences exceeds the *average_relevance* value defined in Equation 9:

$$average\_relevance = \frac{\#R}{\#\text{different absolute urls}} \qquad (9)$$

where *#different absolute urls* is the number of distinct absolute retrieved urls (i.e. web domains), and, number of occurrences is the number of times that for each retrieved url, its web domain appears in the list of the results.

Additionally, the algorithm extends the list of pages previously selected, by adding a set of pages not caught by the system, so far, but related to the query, re-running the search engine with the same query, over the absolute urls. As a consequence, our system turns out to be more robust as it considers more relevant documents than the original list.

3.2. Phrase Extraction

As [17] refers, disregarding the use of phrases is a lost of valuable information, that would benefit the task of search engines in classifying search results [5], in order to return more intuitive key concepts and cluster titles because of its phrasal natural order [7]. Nevertheless, most traditional search engines keep treating each document as a bag of single words. The detection of

phrases could be done by using syntactic, statistical/machine learning or hybrid methodologies. In order to keep the system language-independent, we use SENTA [4], a statistical methodology that uses three basic concepts: positional n-grams, the association measure Mutual Expectation and the GenLocalMaxs algorithm.

3.3. Web Mining Key Concept Extraction

Web content mining is a process that extracts web knowledge, not necessarily the most frequent terms, by analyzing the content of the documents [10]. So in our work, each phrase of the document is represented by a set of 8 properties calculated during the step 4 of the architecture i.e. by using WebSpy. The features from 1 to 6 characterize the importance of the phrase in the document and the features 7 and 8 evaluates the relationship between the phrase and the query in the document collection. In the following, we represent the current phrase as *x,* the documents d_i, *i=1,...,n,* as being all the documents where *x* occurs, $\#d_i$ as being the total number of phrases in the document d_i, and the query as *q*.

(1) IDF
The Inverse Document Frequency, given by 10, represents the dispersion of a given *x* in d_i, *i=1,...,n*:

$$IDF(x) = \log_2 \frac{\#R'}{n} \quad (10)$$

(2) Normalized Text Frequency
This feature, given by Equation 11, is the average frequency of each word in the set of documents.

$$TF(x) = \frac{\sum_{i=1}^{n} \frac{numOcur(x \mid d_i)}{\#d_i}}{n} \quad (11)$$

where $numOcur(x|d_i)$ is the counts of *x* in d_i.

(3) Normalized Density
This feature is the average of a measure named AD (Average Distance), calculated over all the documents where *x* appears. We define AD, in Equation (12), as the normalized distance between all the occurrences of *x* in the document d_i , where *dist* is a function that calculates the distance between the occurrences of *x*. If *x* appears just one time in the document d_i, $AD(x,d_i) = 0$.

$$AD(x,d_i) = \frac{\sum_{t=1}^{numOcur(x|d_i)} \frac{1}{dist(x_{t+1}, x_t)}}{\#d_i}, i = 1,...,n \quad (12)$$

The normalized density is then defined in Equation (13).

$$ND(x) = \frac{\sum_{i=1}^{n} AD(x,d_i)}{n} \quad (13)$$

(4) Normalized First Position
This feature, given by Equation 14, is the average of the number of phrases that appear before the first appearance of *x* over all the documents.

$$FP = \frac{\sum_{i=1}^{n} \frac{fp(x \mid d_i)}{\#d_i}}{n} \quad (14)$$

where $fp(x|d_i)$ is a function that calculates the first position of *x* in the document d_i.

(5) Size in Characters
The feature size is simply the count of characters of *x*.

(6) Capital Letters
The feature represents the number of capital letters of *x*.

(7) Equivalence Index
Is a co-occurrence measure which evaluates the strength existing between *x* and *q*, and is given by 15:

$$EI(x,q) = \frac{f^2(x,q)}{f(x).f(q)} \quad (15)$$

where $f(x,q)$ is the absolute frequency of occurring *x* and *q* in a window of *n* characters and $f(x)$ and $f(q)$ are respectively the absolute frequencies of *x* and *q*.

(8) Normalized Distance to the Query
Given by Equation 16 and 17, it is the average of a feature named *SD* (Shorter Distance), calculated over all the documents where *x* appears. We define *SD* as the shortest normalized distance between *x* and *q* in the document d_i:

$$SD(x,q,d_i) = \frac{fdist(x,q,d_i)}{\#d_i}, i = 1,...,n \quad (16)$$

where $fdist(x,q,d_i)$ calculates the shortest distance between the occurrences of *x* and *q* in the document d_i. The normalized distance to the query is then defined in 17:

$$DQ(x,q) = \frac{\sum_{i=1}^{n} SD(x,q,d_i)}{n} \quad (17)$$

To combine the given properties into a single score, we apply, for each phrase of a document, a decision tree model (C5.0 class) built over 5 pruned decision trees previously trained using a 5-fold cross validation and applying over-sampling to avoid data sparseness. The representation of the documents is given by a vector of key concepts containing its most relevant scored phrases. Each element of the key concept vector, also known as key concept (*kc*), is a flat cluster with a list of related urls.

It is clear, as many authors have shown that decision trees apply particularly well to text data. The experiments made, showed that the best results over the supervised learning process were obtained using 5-fold cross-validation, with a training set of newspaper articles about sports/politics domain and the testing phase over articles about society domain. The model showed 82.49% average precision over the five test data to classify positive and negative examples, and 60.72% to classify only positive.

Although being an important step in the organization of the results, flat clustering still does not solve the ambiguity problem. To simplify the user's search process, we propose a soft hierarchical structure.

3.4. Hierarchical Clustering

In order to organize the results into clusters, we use the PoBOC soft clustering algorithm [3] which is based on graph theory and has shown encouraging results compared to other clustering algorithm when applied to text data. Moreover, PoBOC can be used on the fly, as it does not depend on any input parameter, like the k-means, the EM, etc. By applying the PoBOC algorithm to the flat clusters, we propose a disambiguation methodology, as key concepts with different meanings should be gathered in different clusters. By doing so, the possible different meanings of the query would be evidenced. Finally, each cluster is labeled using a simple heuristics that chooses the key concept that occurs more often in the vectors of each cluster key concept, taking into account the sum of scores in case of ties.

4. Conclusion and Future Work

This paper explores web page hierarchical soft clustering as an alternative method of organizing search results. With WISE, we provide a topic and language-independent real-world web architecture, returning quality results through an organized and disambiguated structure of concepts. WISE has a strong theoretical basis, but its architecture is simple. It is based on (1) an algorithm ignoring less relevant documents and adding relevant ones; (2) statistical phrase extraction to define concepts; (3) web content mining techniques to semantically represent the web pages; (4) an overlapping clustering algorithm to organize results into a hierarchy of concepts and a classical labeling process.

Experimental results, demonstrate the correctness of the clusters, the quality of the labels, concept disambiguation and language-independence, but formal evaluation is still needed. As future work, we propose a evaluation scheme within HARD TREC. We are currently improving the performance of the system, to provide an on-line free version (*wise.di.ubi.pt*) and developing a web warehouse to save the key concepts of the documents, in order to avoid running the system whenever a query is performed and thus providing a new indexing structure following the new Web Farming paradigm.

5. Acknowledgments

Work financed by FCT (project SITE-O-MATIC POSC/EIA//58367/2004).

6. References

[1] Campos, R., Dias, G.: *Automatic Hierarchical Clustering of Web Pages*. In ELECTRA Workshop associated to the 28th Annual International ACM SIGIR Conference, Salvador, Brazil, August 19, 83-85, (2005).

[2] Carpineto, C., Romano, G.: *Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO*. In Journal of the Universal Computer Science, (2004).

[3] Cleuziou, G., Martin, L., Vrain, C.: *PoBOC: an Overlapping Clustering Algorithm. Application to Rule-Based Classification and Textual Data*. In 16th ECAI, Spain, 440-444, (2003).

[4] Dias, G.: *Extraction Automatique d' Associations Lexicales à partir de Corpora*. PhD Thesis. DI/FCT New University of Lisbon (Portugal) and LIFO University of Orléans (2002).

[5] Ferragina, P., Gulli, A.: *A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering*. In 14th International Conference on Data Mining, USA, May, (2003).

[6] Fung, B., Wang, K., Ester, M.: *Large Hierarchical Document Clustering using Frequent Itemsets*. In SIAM International Conference on Data Mining, USA, May, (2003).

[7] Hannappel, P., Klapsing, R., Neumann. G.: *MSEEC - a multi search engine with multiple clustering*. In Information Resources Management Association International Conference, Hershey, Pennsylvania, May, (1999).

[8] Hearts, M., Pedersen, J.: *Re-examining the Cluster Hypothesis: Scatter/Gather on Retrieval Reuslts*. In 19th Annual International SIGIR Conference, Switzerland, 76-84, (1996).

[9] Jiang, Z., Joshi, A., Krishnapuram, R., Yi, L.: *Retriever Improving Web Search Engine Results using Clustering*. In Managing Business with Electronic Commerce, (2002).

[10] Kosala, R., Blockeel, H.: *Web Mining Research: a Survey*. In ACM SIGKDD Exploration 2(1), 1-15, (2000).

[11] Kummamuru, R., Lotlikar, R., Roy, S., Singal, K., Krishnapuram, R.: *A hierarchical monothetic document clustering algorithm for summarization and browsing search results*. In WWW13, (2004).

[12] Lawrie, D., Croft, B.: *Generating hierarchical summaries for web searches*. In 26th Annual International SIGIR Conference, Toronto, Canada, (2003).

[13] Maarek, Y., Fagin, R., Ben-Shaul, I., Pelleg, D.: *Ephemeral document clustering for web applications*. Technical Report RJ 10186, IBM Resarch, (2000).

[14] Martins, B., Silva, M.: *Web Information Retrieval with Result Set Clustering*. In NLTR 2003 - Natural Language and Text Retrieval Workshop, (2003).

[15] Osinski, S., Stefanowski, J., Weiss. D.: *Lingo: Search results clustering algorithm based on Singular Value Decomposition*. In Intelligent Information Systems Conference 2004, Zakopane, Poland, (2004).

[16] Veiga, H., Madeira, S. and Dias, G.: *Webspy*. Technical Report nº 1/2004. http://webspy.di.ubi.pt, (2004).

[17] Zamir, O., Etzioni, O.: *Web Document Clustering: A Feasibility Demonstration*. In 19th Annual International SIGIR Conference, 46-54, (1998).

[18] Zeng, H., He, Q., Chen, Z., Ma, W.: *Learning to Cluster Web Search Results*. In 27th Annual International Conference on Research and Development in Information Retrieval, Sheffield, UK, 210-217, (2004).

[19] Zhang, D., Dong, Y.: *Semantic, Hierarchical, Online Clustering of Web Search Results*. In 6th Asia Pacific Web Conference (APWEB), China, April (2001).