UnURL: Unsupervised Learning from URLs

Deepak P[1] [1], Deepak Khemani[2]

[1]IBM India Research Lab, Bangalore, India
[2]Dept. of CS&E, Indian Institute of Technology Madras, Chennai, India
deepak.s.p@in.ibm.com, khemani@iitm.ac.in

Abstract

Web pages are identified by their URLs. For authoritative web pages, pages that are focused on a specific topic, webmasters tend to use URLs which summarize the page. URL information is good for clustering because, they are small and ubiquitous, making techniques based on just URL information magnitudes faster than those which make use of the text content as well. We present a system that makes use of only URL information to perform clustering of web search result sets, clustering of general web document corpora and topic identification of topical URL corpora. This research prototype which we call *UnURL* is, to the best of our knowledge, the first attempt on using unsupervised machine learning techniques on URLs.

1. Introduction

With the increasing presence of any topic on the World-Wide-Web, there has been an increased focus on applying machine learning techniques to web documents. Information containers (which depict some useful information about a web page) for web documents include the structure of the document (based on the mark-up language used), the unstructured text content, linkage information in the form of incoming and outgoing links and the URL which uniquely identifies a web document. Of these, unsupervised machine learning techniques such as clustering have never focused on harnessing URL information although their supervised counterparts have been experimented with ([1],[2]). In this demonstration, we present a walk-through of a prototype system, *UnURL*, which uses only URL information to perform clustering of web search result sets and general web document corpora, and topic identification of topical (focused on a topic) web document collections. This demonstration demonstrates the techniques proposed in [3].

Section 2 lays down the motivation behind building a system focusing on only URL information such as *UnURL*. Section 3 provides a concise list of the features demonstrated, and the underlying techniques. Section 4 focuses on the system architecture whereas Section 5 concludes the paper outlining a planned sequence of the demo.

2. Motivation

As already mentioned, we are unaware of any unsupervised learning system which uses URL information, whether or not in combination with other kinds of information. URLs are special due to various reasons. Firstly, URLs present structured information [4], although the structure of URLs is not well understood and there has been no serious study on formalizing such structures. This is partly because the structure has evolved over time and various standards, some of which are particular to specific geographies. Secondly, URL is the easiest information to obtain about a web page. URL information does not incur the cost of loading the web page, since pages that link to a page that hold the URL information for the latter. Even pages that do not exist anymore (broken links) have URLs. Thirdly, URLs tend to be very small entities as opposed to other (useful) knowledge containers for a web page such as the text of the page, the title of the page etc. Techniques which deal with URL information only, tend to be magnitudes faster than those which deal with other information because of the conciseness of URLs and the fact that they are easy to obtain.

International Conference on Management of Data
COMAD 2006, Delhi, India, December 14-16, 2006

[1] This work was done while the author was with Indian Institute of Technology Madras

Lastly, webmasters typically tend to summarize the web page when assigning URLs for authoritative and relatively static web pages (web pages whose content don't change very frequently). Further, information that goes into the URL is usually that part of the information which is relatively permanent. It may be noted that such an assumption is valid only for authoritative web pages, which are focused on a specific topic. To cite an example, the summarization assumption does not hold for pages like blogs[2] which keep on changing in the course of time. Such special properties of URLs coupled with the obvious fact that they contain useful information motivates the need for unsupervised learning from URL information.

3. Techniques Used and Features Demonstrated

We devote this section to enumerate the various features of *UnURL* and explain the techniques used in them with flow charts and explanations. Our system has three features demonstrating the three different techniques, each of which is explained in a section herein.

3.1 Techniques Used

This demonstration uses a host of techniques presented in [3]. This sub-section is a brief walkthrough of the techniques with an overview as to how they are used in this demonstration. URLs are structured entities, although their structure is not well-understood. Firstly, we demonstrate *hierarchical agglomerative clustering*[5] *of URL sets*. URL-Sim is a similarity measure for URL-pairs which takes into account the common nature of the structure of URLs. This demonstration consistently uses the URL-Sim measure for hierarchical agglomerative clustering of URL sets. Secondly, we demonstrate *topic identification from topical URL corpora.* Topical URL corpora are those which contain authoritative web pages focused on a specific topic. A pair wise similarity computation using URL-Sim for all the pairs of URLs in the corpus could be used to score keyword fragments. It has been shown that such keyword fragments, ranked according to their scores, closely approximate the topics for topical URL corpora. We use such fragments as topics to tag topical clusters, wherever we do so in our demonstration. Lastly, our demonstration includes *partitional clustering of URLs*, which is done by representing URLs as vectors of character n-grams [6] where the value of n is fixed. As bigrams have been shown to be most effective for clustering, we use K-Means [7] on bigram vectors for partitional clustering in our demonstration.

3.2 Clustered Web Search

In this feature, we provide a web-search interface, which presents search results as clusters of results. Such an interface is becoming popular these days among web search engines[3], although they use all the knowledge containers for a page, which makes our technique different and incomparable to them in that we use only the URL information. The interface takes in two parameters, one being the search query and the other being the type of clustering, whether hierarchical or partitional (by means of a checkbox). If partitional clustering is chosen, the user can enter the number of clusters to be partitioned to (as an additional parameter), which is defaulted to 3. When the query is submitted, the system fetches the results from Google[4] and presents the results as a collapsible tree menu. Each cluster is represented by the set of keyword fragments from the URLs in the cluster. Thus, this feature demonstrates hierarchical clustering (using URLs), partitional clustering (using URLs) and topic identification for topical copora using URLs (which is used for generating the keyword fragment descriptions to describe the sub-clusters).

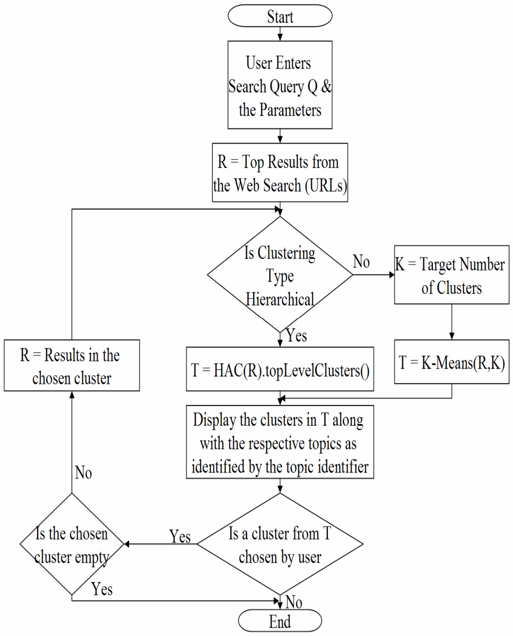
Start
User Enters Search Query Q & the Parameters
R = Top Results from the Web Search (URLs)
Is Clustering Type Hierarchical
No
K = Target Number of Clusters
Yes
R = Results in the chosen cluster
T = HAC(R).topLevelClusters()
T = K-Means(R,K)
Display the clusters in T along with the respective topics as identified by the topic identifier
No
Is the chosen cluster empty
Yes
Is a cluster from T chosen by user
Yes
No
End

Figure 1. Clustered Web Search

The flowchart above depicts the feature discussed. As long as there are results in a chosen cluster, the user can keep choosing the clusters displayed to him unless the chosen cluster is empty. The clustering performed is either partitional (using K-Means on character bigram vectors) or hierarchical agglomerative (HAC) using URL-Sim.

[2] http://en.wikipedia.org/wiki/Blog

[3] http://www.clusty.com

[4] http://www.google.com

3.3 Topic Identification from Web Search Result Sets

This feature demonstrates the keyword identification technique as mentioned in Section 3.1. A topical corpus is a corpus of web pages focused on a particular topic. For our purposes, we focus on topical corpora containing authoritative web pages on the topic involved. Web search engines typically give out authoritative web pages among the top results, and hence, we choose to use such corpora for our keyword identification demonstration. The interface is much like a search engine, with an edit box to enter the query. On submitting the query, the system fetches the top results from Google, uses that as the topical corpus, and uses the keyword (fragment) finder to find fragments approximating the topic of the corpus. As web search engines are known to put the authoritative results for the particular query among the top results, comparing the topics found against the search query entered is a straightforward way of evaluating the goodness of the topic finder.

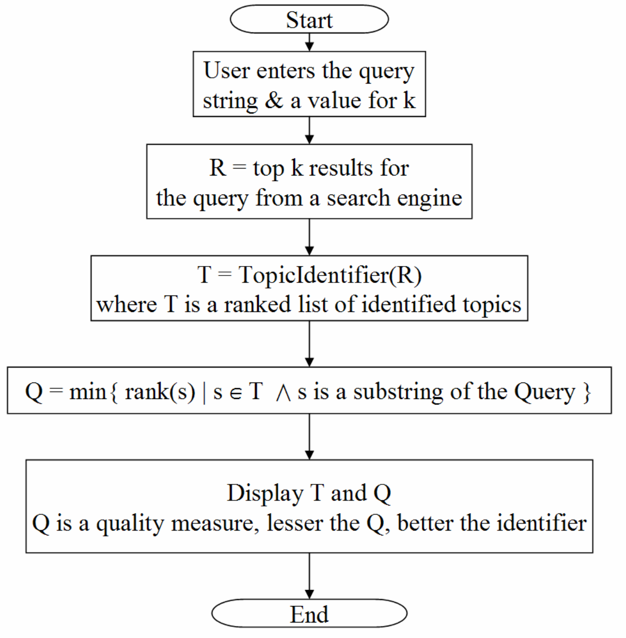
Start
User enters the query string & a value for k
R = top k results for the query from a search engine
T = TopicIdentifier(R) where T is a ranked list of identified topics
Q = min{ rank(s) | s ∊ T ∧ s is a substring of the Query }
Display T and Q
Q is a quality measure, lesser the Q, better the identifier
End

Figure 2. Topic Identification

Figure 2 depicting the topic identification feature is mostly self-explanatory. The quality measure Q depicts how low the rank of the search query is, in the ranked list T. As is obvious, a lower value of Q would validate the goodness of the topic identifier. It may be noted here that a high value of Q does not necessarily mean a bad performance. For instance, a topical corpus for the query “Indian Institute of Technology” may lead to a corpus where the topic identifier scores “iit” (the abbreviation of the query) very high even though it is not a substring of the search query.

3.4 Multiple Query Result Clustering

The goodness of the clustering obtained is demonstrated by this feature. As observed in Section 3.3, web search engines typically give a topical corpus among the top results in response to a search query. Multiple such topical corpora could be used to perform clustering, thereby providing a labelled corpus (each topical corpus corresponding to a search query labelled with the query) which enables us to measure the goodness of the clustering using extrinsic quality measures such as entropy and purity. This feature demonstrates just that. It provides an interface that allows the user to enter multiple search queries, performs the clustering by means of both the partitional and hierarchical techniques as explained in Section 3.1. Each cluster is described using the topic identification method, as in Section 3.3. Further, it displays the extrinsic quality measures, purity and entropy [8], for clusters. Ideally, we would explain a partitional clustering fed with the union of k topical corpora (each attached to a search query) to give k clusters wherein the topic identifier identifying the words close to the corresponding search queries for each of topical corpus involved. The flowchart for this feature as in Figure 3 is largely self-explanatory, and hence, we choose not to explain it.

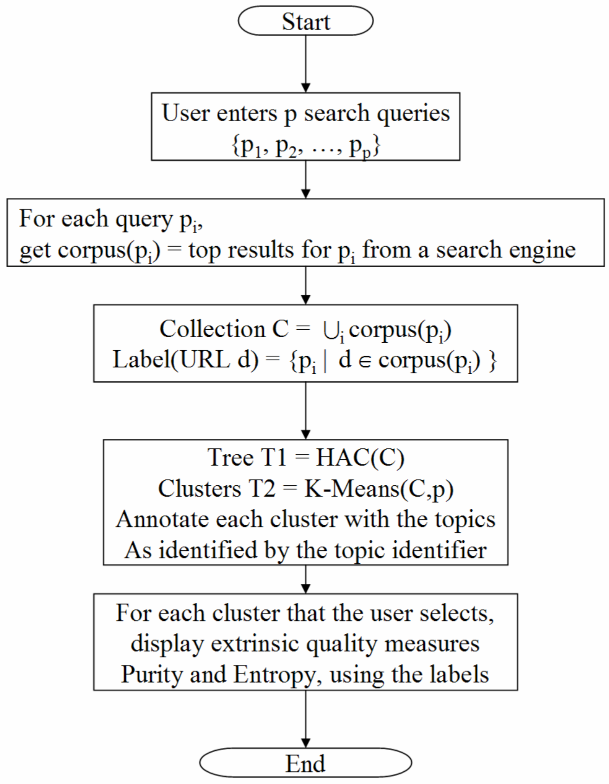
Start
User enters p search queries {p_1, p_2, …, p_p}
For each query p_i, get corpus(p_i) = top results for p_i from a search engine
Collection C = $\cup_i$ corpus(p_i)
Label(URL d) = {p_i | d ∊ corpus(p_i) }
Tree T1 = HAC(C)
Clusters T2 = K-Means(C,p)
Annotate each cluster with the topics As identified by the topic identifier
For each cluster that the user selects, display extrinsic quality measures Purity and Entropy, using the labels
End

Figure 3. Multiple Query Result Clustering

4. System Architecture

Although we have explained the working of the different features by means of the flowcharts presented in the preceding section, we present a system architecture diagram, which shows a brief overview of the major modules in the system in Figure 4. We have included each feature described above as a module in the system to enable easy mapping to the preceding section. The edges indicate a "uses" relation for the feature involved.

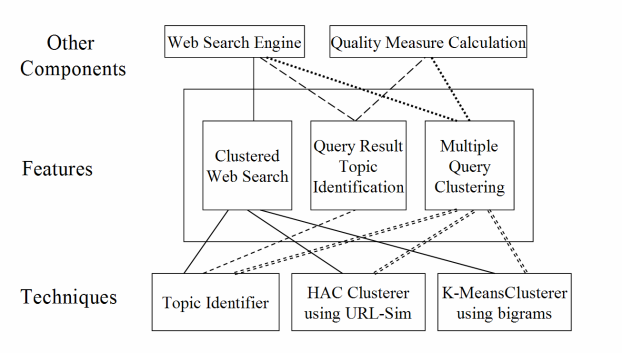
Other Components
Web Search Engine
Quality Measure Calculation
Features
Clustered Web Search
Query Result Topic Identification
Multiple Query Clustering
Techniques
Topic Identifier
HAC Clusterer using URL-Sim
K-MeansClusterer using bigrams

Figure 4. System Architecture

5. Demonstration

To the best of our knowledge, *UnURL* is the first attempt on using URL Information for unsupervised learning tasks. We primarily use the Google as the search engine for this demonstration, but would like to emphasize that, as is obvious, the techniques are largely independent of the search engine used. We present a walkthrough of the features of *UnURL*, illustrating how URL information can prove to be very useful and efficient for the learning tasks such as the ones in *UnURL.*

References

1. Min-Yen Kan and Hoang Oanh Nguyen Thi (2005) Fast webpage classification using URL features. To appear in Proc. of Conf. on Info and Knowledge Management (CIKM 2005). Bremen, Germany, November 2005. Poster Paper.
2. Min-Yen Kan (2004) Web Page Classification without the Web Page. Poster, 13th World Wide Wed Conference, 2004 (WWW 2004), NY
3. Deepak P, Deepak Khemani, "Unsupervised Learning from URL Corpora", In the Proceedings of the 13th Intl. Conference on Management of Data (COMAD 2006), 2006, Delhi, India
4. T. Bernes Lee, Masinter, McCahill, "Uniform Resource Locators", RFC-1738, Network Working Group, 1994 http://www.ietf.org/rfc/rfc1738.txt
5. Willet, "Recent Trends in Hierarchical Document Clustering: A Critical Review", Information Processing and Management, 1988
6. Canvar, Tenkle, "N-Gram based text categorization", Proc of the 3rd Annual Symposium on Document Analysis and Information Retrieval, SDAIR, 1994
7. MacQueen, JB, "Some methods for classification and analysis of multivariate observations", Proc. Of the 5th Symposium on Math, Statistics and Propability, Berkeley, CA, 1967
8. Zhao, Karypis, "Criterion Function for Document Clustering: Experiments and Analysis", Dept. of CS, University of Minnesota, TR#01-40