Ontological Tree Generation for Enhanced Information Retrieval.

Anwaya Aras
BITS-Pilani University,
Goa Campus,
India
anwayaaras@gmail.com

SakshiPratap
BITS-Pilani University,
Goa Campus,
India
sakshi.2392@gmail.com

Dr. MangeshBedekar
BITS-Pilani University,
Goa Campus,
India
bedekar@goa.bits-pilani.ac.in

Abstract:
Information visualization seeks to leverage human visual processing to make sense of abstract information. One particularly rich class of information structures ripe for visualization are those representable as graphs (i.e. nodes and edges), including organization charts, website linkage, and computer networks. In this paper we propose a methodology to extract information from big data and convert it into a human comprehensible format of graphs to give the reader an objective overall idea of the document content. We put forth the design and implementation details to mapping our data into the Open Directory Project or the DMOZ tree and build a hierarchical ontological tree based on the extracted metadata.

1. Introduction

Large related data, like directories, encyclopedias, books have massive amount of data which is both extensive and important. While it is important to understand and grasp their content, the process of reading such documents might get time consuming and cumbersome. Text in most documents is highly redundant and sometimes irrelevant. Entire documents would need to be read to understand what its content is. Much of such big data, specifically the one which is related can easily visualized using a graphical hierarchical structure and every token representing a node and edge linking related categories .To address the problems effectively, we are trying to use techniques to pull out phrases that seem to characterize a document and then map them to achieve an ontological tree based graph to represent the information.

For the purpose of experimentation, we have used the academic handouts of universities. For initial analysis purpose we are focusing on undergraduate courses as they have a well structured curriculum and range over a wider variety of subjects. Our system effectively generates their respective DMOZ trees [2] give us a fair idea of the depth of the courses taught which are represented by the handout. We then compare handouts of various courses and make notes of the inferences and then correlate them to human drawn conclusions to verify the accurate working of our system.

It is essential to organize the results into ontology, in particular a hierarchical ontology. Ontology is an explicit formal specification of the terms and relations among terms in a domain. It can be achieved by a systematic grouping of domain concepts (e.g., user interests) based on their definitions, in machine-interpretable form [7].

In this paper we are proposing an approach for efficient classification of data into the large topic ontology DMoz. The Open Directory Project, also known as DMOZ, is the largest human edited

directory of Internet sites. The DMOZ directory incorporated 590,000 categories and 6 million quality website content organized into 15 levels. ODP uses a hierarchical ontology scheme for organizing site listings. Listings on a similar topic are grouped into categories which then include smaller categories. One effective technique for the display of such data is a focus+context approach that uses lightweight modeling of user interest to inform the display of information. User interest is modeled using a Degree-of-Interest (DOI) function, which assigns a single number representing the estimated relative interest of the user to each node in the structure. These numbers are used to appropriately layout and render the structure, for example by controlling which nodes are visible and which are elided [9].

2. Brief Methodology

Our proposed plan consists of two parts namely document key word retrival and building of ontological trees.[5] The first part uses techniques of NLP to extract only the important keywords-unigrams,bigrams and trigrams from the large chunk of data. In the second stage, these words are fed into the DMOZ directory and when a particular n gram phrase hits the tree, the entire tree is generated.

2.1. Keyword Extraction

Algorithm:
Efficient extraction of related keywords in the form of n grams models was extremely imporatnt as it directly affects the accuracy of the DMOZ tree. Although,work on summarization of large amounts of data has been effectively done and substantial results have been achieved[5], information extraction from related documents with a characteristic concept flow, requires a different approach as shown by Rahman [11].

Initially, the data is fragmented and divided into multiple tokens. Then a Part Of Speech tagger is then run on each sentence and the words are tagged accordingly. However, this tagging is not accurate enough and hence we have used a novel technique for tagging based on a naturally growing resource which has used concepts demonstrated by Taskaret. al(2012)[3].

Once this tagging is complete, all the important derivatives of speech are put into the output file. We then have hard coded a probability value, to generate unigrams, bigrams and trigrams out of the annotated data.

The following are all the steps that encompass the entire algorithm that runs behind our keyword extraction module. Each of the techniques described below comprise of a separate module running at the back end of our code.

- **Stemming**

We use stemming to narrow our overall word matrix so as to help with the lack of similar words per text. We have used the standard Porter's stemming algorithm for the purpose. Overall, we see mixed results for stemming.While stemming shows improvement it also negatively affects the overall accuracy when combined with other aspects of descriptive texts.

•**Stop Word Removal**
As in much text, that there were many common, short function words, often prepositional terms and other grammatical syntax fillers that were found specially in descriptive texts. We have not yet attempted to build a domain-specific we used a standard set of Porter stem words.
Ex: the, is, at, who, which, on
Stop words removal turned out to drastically reduce the length of passage and descriptive texts thereby giving huge accuracy boost.

•**Named entity recognition**
Entity identification is a very important part of summarization especially because it tells the system extremely relevant data about the text data. Though NER systems are known to be brittle, our system has been specifically designed for the big data domain and thus it rightly extracts the necessary words depending upon the type and setting of the question.

•**Punctuation Removal**
Removing punctuation was another result of looking at our data and noticing that written texts have a very large variance in phrasing and word choice. This is especially true for words that may have multiple accepted forms, or words with punctuation in them. Also, because we parse the item descriptors on spaces, any punctuations that are in the phrase are left in, including ellipses, periods, exclamation points, and others. In addition, words that are concatenated are often used differently. Punctuation removal was the also an effective feature normalization method used for summarization.

•**Lowercasing**
While writing in English, texts tend to have the first word capitalized. In addition, different writing styles will capitalize different words intentionally or otherwise, depending on their intent interpretation of the word, of choice of capitalizing acronyms. This is generally not a useful normalization for the system to understand as it deteriorates the performance of the POS tagger.
Ex. President, president, CD, cd, Windows, windows

2.2 Keyword synthesis
After application of the above techniques, the system generates a relevant set of words from the document. However, tp precisely understand what the document is trying to convey, the frequency of each word, specific occurrence with other words and the context of its usage is equally important. Hence, once the raw word data set is available, we further analyze the words and the information they convey. Three modules are used for the same and the words are further divided as :

1. **Unigramlist**: These are the words which occur frequently in the document i.e occur more times than some threshold value. This threshold value is decided on the basis of the total number of words, the count of the word which has highest occurrence and the count of the word which has lowest occurrence.

2. Bigram list: These are the words which always occur together in the document. For example "operating systems" or "computer networks". Their pairing and occurring together is much meaningful than the word occurring isolated and hence they carry more weight in word extraction analysis. Two words get included in the bigram list if their count of occurring together or at an offset of at most 3 words is more than a predetermined threshold value.

3. Trigram list: If three words are seen to occur together then they are included in the trigram list. Occurrences of such words are rare, but if they occur they covey very subtle information and help greatly in the DMOZ tree building.

2.3. Ontological tree generation

The problem of retrieving the complete category link for an interest and ranking the categories based on their importance is resolved through the use of DMoz category hierarchy. Every category in the DMoz dump consists of a listing and description of external pages associated with that category. Our DMoz based approach works as follows: each interest is searched in the DMoz RDF dump. The categories under which one or more of the external page descriptions contain the concerned interest are selected.

Thus, when searching for an interest such as "programming" in the dump, we see that "Computers" occurs frequently in the external page description of the category link. All such category links under which the interest is found are extracted. To engineer the ontology, we use only the top level categories. This avoids large scale duplication of the interest instances. Furthermore, with DMoz, it becomes possible to retrieve the complete category link associated with the interest. Thus, for example, the interest "programming" in DMoz belongs not only to the category "Computers" but to its complete category link, which is Computers->Internet->Cloud Computing->Programming We parse the complete category link and every term in the link becomes a node in the ontology. The interest is made a child of the lowest node in the hierarchy. As seen in the figure, all interests are accurately grouped under the respective categories to which they belong.Simultaneously the depth of the node from the start point is attached at every level to gauge the importance especially when the levels are more than 10.

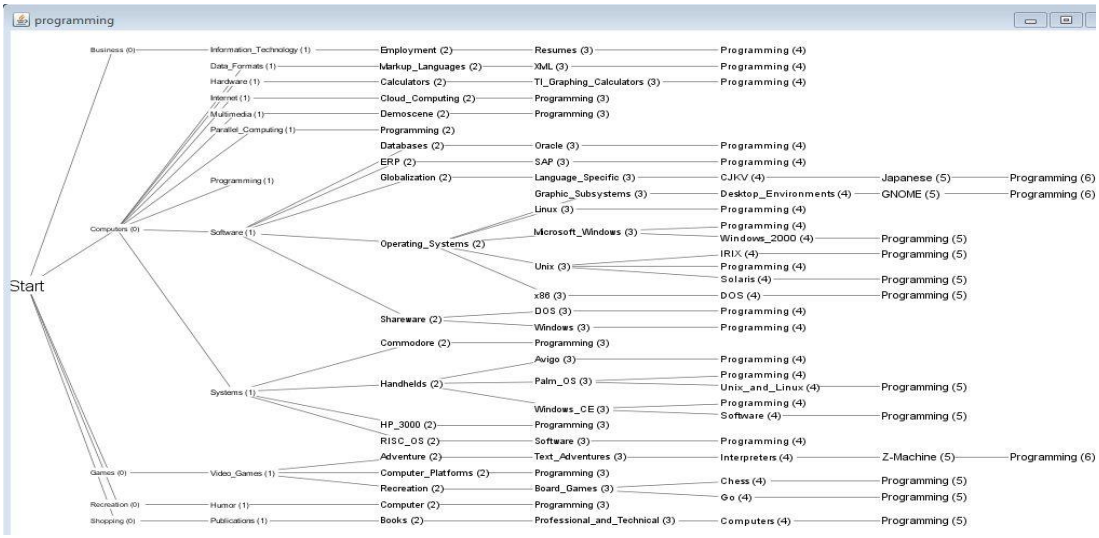


Fig 2.2.1-A view of DMOZ tree output on the word programming without using any filter

The nodes can be then arranged in the required way to allow every user to personalize their space. This can help users to relate certain data, ignore unnecessary information and get more information in a shorter period.[10]

The various features like zooming, panning on mouse and keyboard actions enable only certain data to be visible at a given time and increase readability. Simultaneously the application supports 4 different orientations of the data, top to bottom, bottom to up, right to left, left to right, making it easier for the user to decide depending on the requirement.

The double-click function redirects to open another pop-up window (See figure)which has the options either to delve deeper into the tree of the clicked topic or choose among the links corresponding to its topic. Expanding the tree both hides the other irrelevant nodes not in the path of the clicked data as well as displays its branches .(subtopics arising from the topic) Clicking on any other button opens the corresponding link on the default web-browser. On the left hand side are the topics and on the right hand side is a brief description about each topic.

Figure: Popup window on clicking Oracle.

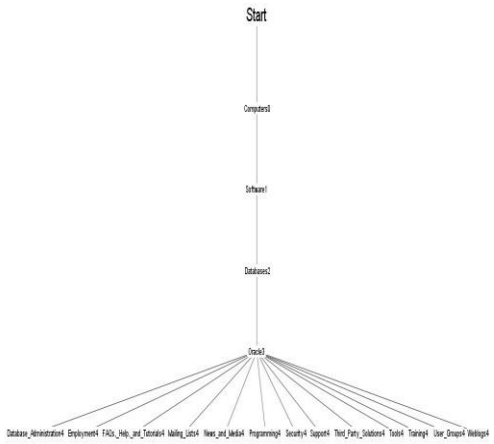


Fig: Result on expanding the tree

Fig: Result on clicking on the website link “Oracle FAQ”

3.Implementation

We have primarily coded in Python and Java to avail the benefits of both the languages. While Python offers an extensive and a fully fledged library for language processing purposes, JavaJava is fast, secure, reliable and with its underlying object-oriented principles provides an excellent platform to work on.

The summarization and key word extraction part has been implemented in Python programming language using NLTK library[1]. Since python contain an extensive library for language processing, it stood out as the best language for development of robust modules for this project. We have developed our own POS tagger,tokeniser and hand codded features for the named entity recognition algorithm based on the dominant aspects of big data analysis.These modules when run together successfully extract the relevant words of our dataand further go on to generating the n gram words for our project.

The requirements of our design of DMOZ trees are best supported by Java using Prefuse[8]. Prefuse is a set of software tools for creating rich interactive data visualizations. The original prefuse toolkit provides a visualization framework for the Java programming language. The prefuse flare toolkit also provides visualization and animation tools for ActionScript and the Adobe Flash Player. The label renderer and JPanel components introduce interactivity in the visualization.

4.Integrated System and Results

4.1.Handout Data extraction

The course handouts of Massachusetts institute of technology (MIT) and Birla Institute of

Technology and Science (BITS) have been used for analysis. Using this method all the information of the course can be easily represented in graphical hierarchical form, thus making it possible to get the sense of the topics, sub-topics and the complete course structure. By The handouts used in consideration here are of the course Computer Networks , offered by the Computer Science department of the colleges. Techniques have been used to perform interest-word sense disambiguation . For instance here, "Security" should refer to something in the world of Computers only. Thus, this approach constructs a simple yet effective grouping of user interests.

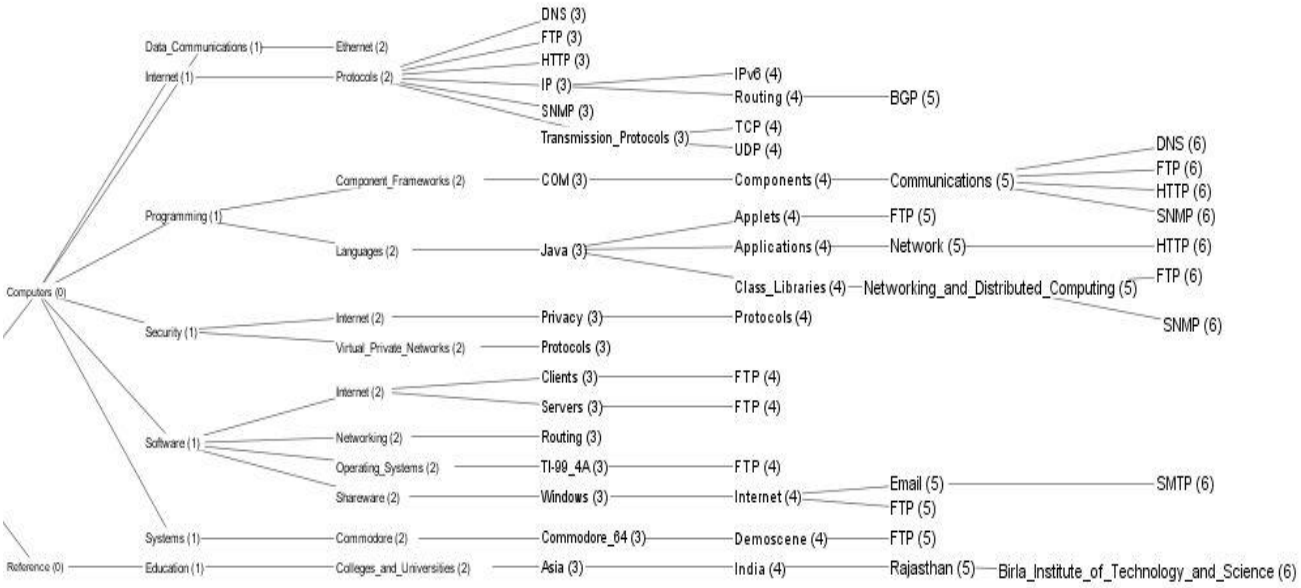


Fig -A view of DMOZ tree output on the handout of BITS.

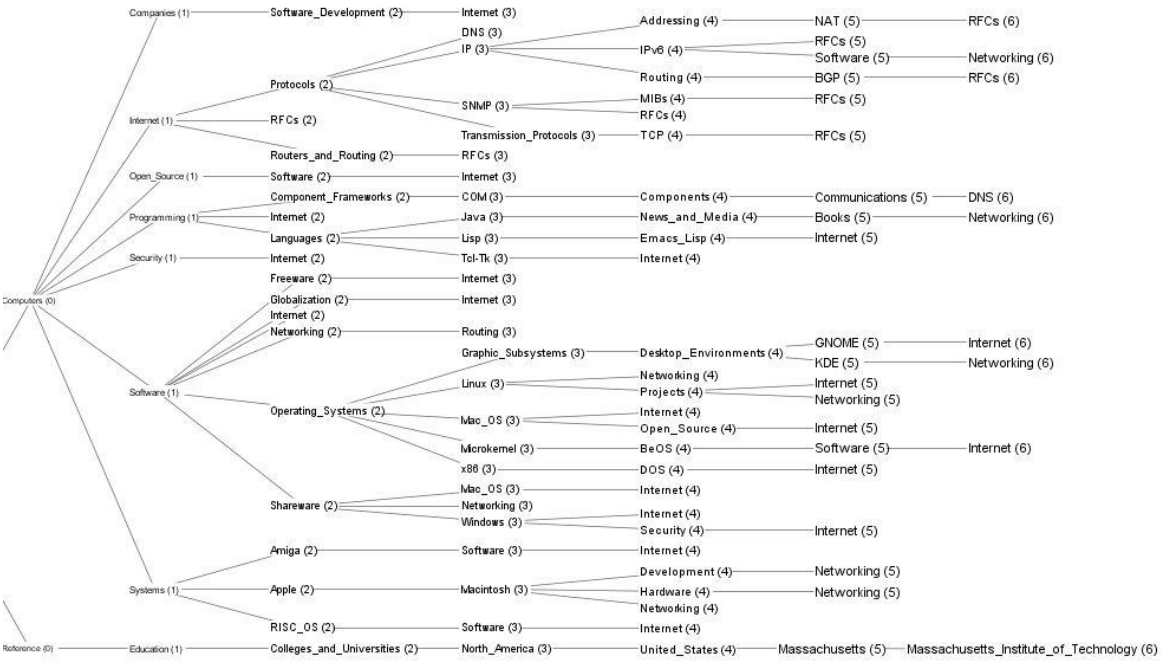


Fig -A view of DMOZ tree output on the handout of MIT.

4.2 Results:

The tree gives complete information of the course structure, the linkage between categories, classification, the volume and depth of each topic that will be covered.

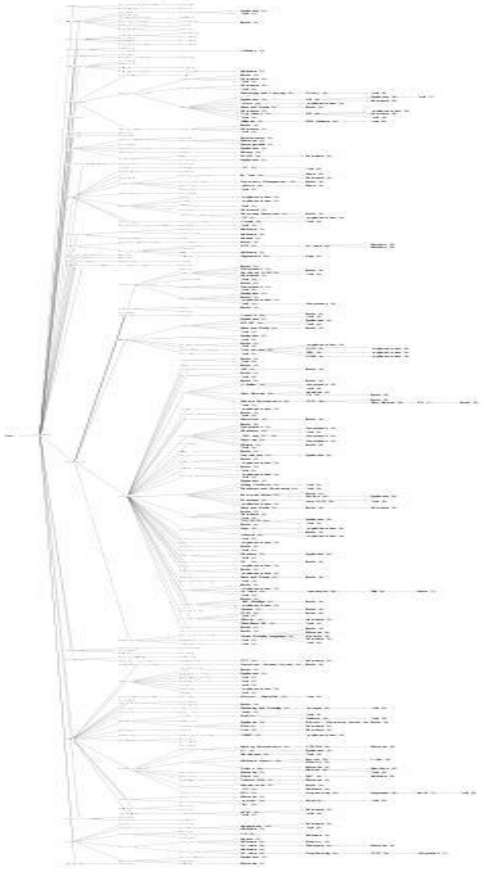
It can be gauged if only the overview or details of particular topics would be covered in the duration of the course. For instance in BITS, the node ethernet has no children, thus it is less likely that the topic will be taught in much detail. Similarly we can even compare two topics and safely say that protocols will be focused more than ethernet in BITS.

On formation of the tree we can easily extract useful comparison from the structure of courses:

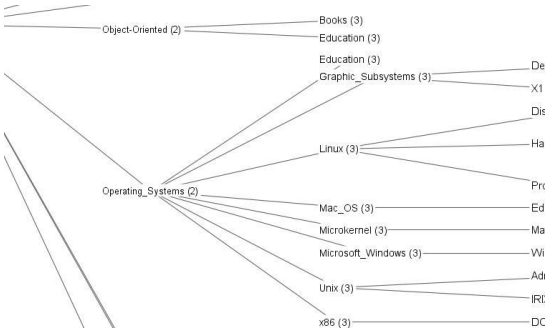
- Here both MIT and BITS have topics till the level 6, thus providing equal information on considering depth and detail level of the course taught.
- We can easily find out the broad topics being studied, which topics overlap and differences. Here the topic Data communications is covered in BITS, including the sub-topic ethernet while MIT does not have it.
- If we go into the depth of the tree, we see that protocols is taught in much more depth in MIT focusing on RFCs, NAT while in BITS the focus is on studying various types of protocols without going into intricate details.

We similarly created a tree of all the subjects of interest to get a complete picture of the subjects

ordered by only important key topics, thus providing details on the whole course in one graph.Zooming and panning provides easy navigation to any topic and its linked branches. For details regarding a particular topic, we can double click the interest node and go into further depth or web links relevant to the category.We aim to use highlighting and navigation techniques to check overlapping information between subjects, check particular courses itself and thus provide a clear picture.



Complete tree



Zooming in a particular area

5. Future Work and Conclusion

The results provided by our application were satisfactory and gave precise information as well as provided clarity in regard to the actual topics being covered and depth and those as resulting from our application. The technique and methodology used provided upto 90% accuracy in most cases. One of the future scope that stands for this project is increasing the number of relevant key words extracted by the system and thus increasing the recall value for key word extraction module. Specifically, the system tends to reject acronyms as it identifies them as stray words and overlooks their instances. Robust feature extractors for training the system using NER are needed to be built to overcome this problem and get an even more efficient system.
The interface is being made more appealing by adding additional graphics and features toincrease interactivity. The project is being deployed as a web app to allow instant result display

on any file that is provided.

In future, similar procedure can be employed to other databases and directories like Wikipedia to improve visualization and to provide exhaustive results as well as compare how the results are mapped in each of the given dumps. More number of colleges as well as schools' curriculum is being compared .Analysis on how the information is related at every stage and usefulness and relevance of subjects when compared to specialization in a particular field in that branch is being done. The same Information Visualization Techniques and concepts can be used to create precise memory maps and business analytics techniques can be formulated.

6.References

[1] Natural language toolkit. URL http://www.nltk.org.

[2] Dmoz dump: http://dmozimporter.codeplex.com/

[3] S. Li, V. Graca and B.Taskar. 2012 .Wiki-ly Supervised Part-of- Speech Tagging. In Proc.EMNLP 2012 .

[4] C. cheng Lin and H. chun Yen, "On balloon drawings of rooted trees," pp. 12–14, 2005.

[5] Laender, A. H. F., Ribeiro-Neto, B. A., da Silva,A. S., and Teixeira, J. S.2002. A briefsurvey of web data extraction tools. SIGMOD Rec.31,2, 84

[6] Kushmerick, N.2002. Finite-state approaches to web information extraction. Proc. of 3rd Summer Convention on Information Extraction.

[7] Thorsten JoachimsMandarHaridas and DoinaCaragea: Exploring Wikipedia and DMoz ,2009. Knowledge Bases for Engineering a User Interests Hierarchy for Social Network Applications, USA.

[8] Jeffrey Heer,Stuart K. Card and James A. Landay: Prefuse:2004 A toolkit for interactive information visualization

[9] Jeffrey Heer and Stuart K. Card: Efficient User Interest Estimation in Fisheye Views, ACM Human Factors in Computing Systems (2003)

[10] Graphviz. http://www.research.att.com/sw/tools/graphviz/

[11] A. F. R. Rahman, H. Alam and R. Hartono. "Understanding the Flow of Content in Summarizing HTML Documents". In Int. Workshop on Document Layout Interpretation and its Applications, DLIA01, Sep., 2001.

[12] Marko Grobelnik, Dunja Mladenić,2005."Simple classification into large topic ontology of Web documents" In Journal of Computing and Information Technology.