Hierarchical Text Classification

Ashwin K Pulijala **Susan Gauch**
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS, 66045
{ashwin, sgauch}@ittc.ku.edu

ABSTRACT

The vast quantity of information on the Web has resulted in the proliferation of topic hierarchies that allow users to browse, rather than search, for Web pages. In our research on conceptual retrieval, we classify documents during indexing so that they can later be retrieved by a combination of keyword and conceptual match. These and other applications have created a need for tools that automatically classify new documents with respect to such hierarchies. Most approaches use flat classifiers that ignore the hierarchical structure, treating each topic as a separate class. Although these flat classifiers are computationally simple, they fail to exploit the information inherent in the structural relationship between topics.

This paper explores the use of hierarchical structure for classifying a large, heterogeneous collection of Web content. Use of the hierarchical structure during classification has resulted in a significant improvement of 45.4% in exact match precision when compared with a flat classifier.

Keywords: Text Classification, Hierarchical Models, Conceptual Search.

1. INTRODUCTION

With the exponential growth of information on the Internet, it is becoming increasingly difficult to find relevant material. One of the underlying reasons for this is that most search engines find matches based on keywords, regardless of their meanings.

To overcome this problem and provide the user with more useful information, a conceptual search engine called KeyConcept was developed [4]. In this system, documents are automatically classified during indexing so that they can later be retrieved by a combination of keyword and conceptual match.

However, the classification approach currently in use ignores the hierarchical structure and treats each concept separately, thus "flattening" the class structure.

2. OBJCTIVE AND MOTIVATION

The goal of this research is to explore the use of hierarchical structure for classifying a large, heterogeneous collection of Web content. Rather than building a single massive classifier, our aim is to construct a hierarchy of classifiers that increase accuracy by focusing only on a small set of classes at each level, those relevant to task at hand.

By utilizing the known hierarchical structure, the classification problem can be decomposed into a smaller set of problems corresponding to hierarchical splits in the tree. One first distinguishes among classes at the top level, and then the lower level distinctions are determined only within the subclasses of the appropriate top level class. Each of these sub problems can be solved much more accurately and efficiently as well [2]. Moreover, greater accuracy is possible because the classifiers can identify, and ignore, commonalities between subtopics of a specific class, and concentrate on those features that distinguish between them [1].

3. RELATED WORK

The document classification problem is one of assigning newly arriving documents to one or more of the pre-existing classes. In the case of *flat classification,* predefined classes are treated in isolation and there is no structure defining the relationships among them. *Hierarchical classification* refers to assigning a document to a suitable concept from a hierarchical concept space.

The focus of the research is the case where there is a hierarchy of preexisting classes. While, in general, a lower level class may belong to more than one higher-level class and the classes form a directed acyclic graph, in the problem we consider here the hierarchy is a *classification tree*. In a classification tree, each class can belong to at most one parent class and documents can be assigned to both internal and leaf classes.

There are two basic hierarchical classification methods, namely, the big-bang approach and the top-down level-based approach [7]. In the big-bang approach, the classifier assigns a document to a class in one single step whereas in the top-down level-based approach, the classification is accomplished with the cooperation of classifiers built at each level of the tree. The test document starts at the root of the tree and is compared to classes at the first level. The document is assigned to the best matching level-1 class and is then compared to all sub-classes of that class. This process continues until the document reaches a leaf or an internal class below which the document cannot be further classified. One of the obvious problems with top-down approach is that a misclassification at a parent class may force a document to be mis-routed before it can be classified into child classes.

In our approach, we adopt a top-down, level based approach that utilizes the hierarchical topic structure to decompose the classification task into a set of simpler problems. The classifiers we use are based on the vector space model.

4. SYSTEM ARCHITECTURE

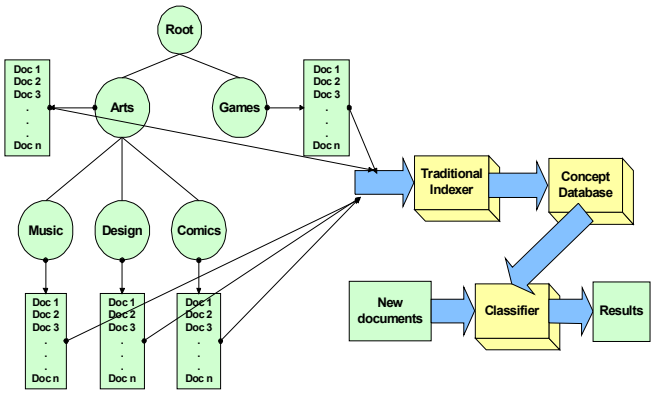

Figure 1. Architecture of the classifier.

The classification process is comprised of two phases: classifier training and classifying new documents. During classifier training, a fixed number of sample documents for each concept are collected and merged, and the resulting super-documents are preprocessed and indexed using the *tf* * *idf* method. This essentially represents each concept by the centroid of the training set for that concept. New documents are then indexed using a vector-space method and the documents are classified by comparing the document vector to the centroid for each concept.

5. CLASSIFIER TRAINING

For training the classifier, content documents for each concept are concatenated to create a collection of super-documents, D. The super-documents are then pre-processed to remove high-frequency function words (stop words) and HTML tags. Finally, the Porter stemmer [3] is used to reduce each word to its root to decrease the effect of word variations on the classification.

In a vector-space classifier, each concept j is represented by a vector, c_j, containing one entry per term in the vocabulary. The weight for a

given term is a factor of the frequency of the term in the super-document for the concept, tf_{ij}, and the rarity of that term in other concepts, idf_i.

In more detail, the tc_{ij}, weight of term *i* concept *j* is given by:

$$tc_{ij} = tf_{ij} * idf_i \quad (1)$$

where

tf_{ij} = *number of occurrences of* t_i *in* d_j

idf_i = *Log (Number of documents in D / Number of documents in D that contain* t_i*)*

D = the collection of super-documents.

$t_i = i^{th}$ *term in vocabulary.*

d_j = *the* j^{th} *super-document.*

The dimensionality of the concept vectors is very large, one dimension for every word used in any document in the collection. This dimensionality is somewhat reduced by removing stop words and further reduced by stemming. However, since most of the super-documents contain only a small fraction of the possible words and absent terms receive a weight of 0, these concept vectors are very sparse. Because not all documents are the same length, the concepts vary somewhat in the amount of training data. To compensate for this, the term weights in each concept vector are normalized by the vector magnitude, creating unit length vectors.

Thus, ntc_{ij}, the normalized weight of term *i* concept *j* is given by:

$$ntc_{ij} = (tc_{ij} / vector\text{-}length_j) \quad (2)$$

where

$$vector\text{-}length_j = \sum_i tc_{ij} \quad (3)$$

6. OBSERVATIONS USING A FLAT CLASSIFIER

The results from previous experiments conducted using the flat classifier support the following [4]:

i) Using 30 training documents per concept, allows reaching a good compromise between the amount of training data and the classifier precision.

ii) Classifier precision is independent of the number of concepts and does not deteriorate as the number of concepts is increased.

iii) Classifier precision does not depend on the particular concepts chosen for large enough concept sets.

iv) The range for optimum number of top words required to classify a document varies from 20 – 50.

v) The flat classifier has an exact match precision of around 51%.

7. CLASSIFICATION TREE

Because the Open Directory Project hierarchy [6] is readily available for download from their web site in a compact format, it was chosen as the source for classification tree. It is becoming a widely used, informal standard. As of April 2002, the Open Directory had more than 378,000 concepts. With such a fine granularity, subtle differences between certain classes may be apparent to a human but indistinguishable to a classification algorithm. In our research, we are using the classification tree to create user profiles and identify the general topics present in a document for conceptual retrieval [5]. For these purposes, the top few levels of the tree are sufficient. Thus we developed a classifier for the top three levels only, although training data was used from the top four levels. In addition, in order to have sufficient training data, we further restricted the classes to those that had at least 20 documents. There were 15 classes at level-1, 356 classes at level-2, 2,812 classes at level-3, and 3,895 classes at level-4 with at least 20 documents in them.

8. TRAINING DATA

The classifiers at each level were trained using associated documents for that class and all subclasses. Thus, the level-1 classifiers were trained using documents from that class and from subclasses at levels 2 through 4. In

contrast, each level-2 classifier was trained with documents from the appropriate level-2 class and sub-classes for that class in levels 3 through 4.

We ran several experiments on the effects of training data on the classifier, varying the number of training documents used per class and including or excluding training documents from level-4. We report here on the best results, those obtained when a single document is used from each class (and subclass) and with level 4 training data included. On average, level-1 classes were trained with approximately 7078 training documents, level-2 classes had 7063 training documents and level-3 classifiers were trained with 6707 documents.

9. TESTING DATA

We tested the accuracy of the classifier by classifying 750 randomly selected level-3 documents. These documents were excluded from the training process and were selected from 750 different level-3 classes. Since we know the class from which the document was selected, we compare the accuracy of our classifier against "truth" by evaluating how often the classifier assigns the test documents to the classes from which they originally came.

10. EXPERIMENT

We created a baseline system by training a flat classifier for all classes in the top 3 levels of the classification tree, ignoring structure. In other words, each of the 3,183 classes was trained using 20 documents from that class alone. After the flat classifier was trained, each test document was classified and the top matching class was recorded.

For the hierarchical classifier, we constructed a set of classifiers, one at each level of the classification tree, using the training method described above. Thus, there was one classifier for level-1 (trained on the 15 level-1 classes), 15 classifiers for level-2 (one for each level-1 class), and 356 classifiers for level-3 (one for each level-2 class). We classified the test document using the level-1 classifier and then, based on the top result, reclassified the document using the appropriate level-2 classifier. After a final classification with the best matching level-3 classifier, the final level-3 class assigned to the test document was recorded.

11. RESULTS

The test documents were initially classified at level-1 using a varying number of words per document where the words were selected based on their *tf* * *idf* weights. The first run used only the highest weighted word for classifying the documents and number of words was increased in each subsequent run until a maximum of 35. The level-1 classifier had a peak accuracy of 79.5% when the top 3 words were used.

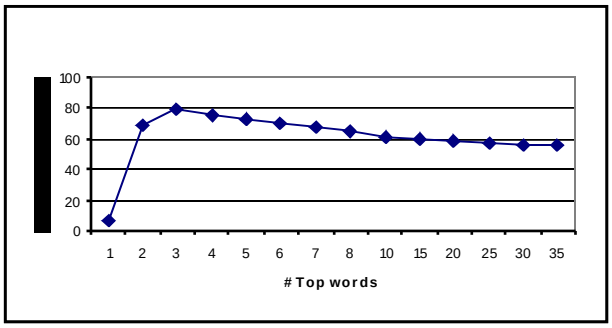

Figure 2. The effect of number of top words selected from the test document on level-1 classification accuracy.

The test documents were then classified at level-2 while again varying the number of top words from 1 to 35. At level-2, the classification process is same as above, but it is constrained to consider only the child concepts of the best matching concept at level-1. The level-2 classifier had a peak accuracy of 71.3% when the top 5 words were selected.

Finally, the test documents were classified at level-3 with the classification process now constrained to consider only the child concepts of the best matching concept at level-2. Since all the test documents came from level-3 classes originally, the accuracy of the

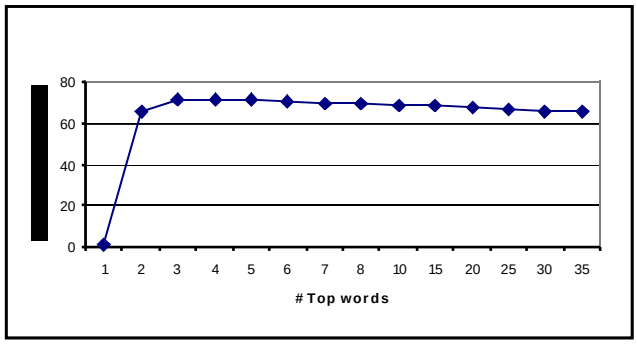

Figure 3. The effect of the number of top words selected from the test document on level-2 classification accuracy.

classifier overall is best judged by the accuracy at level-3.The level-3 classifier had an exact match precision of 70.1% when the top 19 words were used. This means that, from a set of 2,812 classes, the hierarchical classifier assigned 70.1% of documents to their original class.

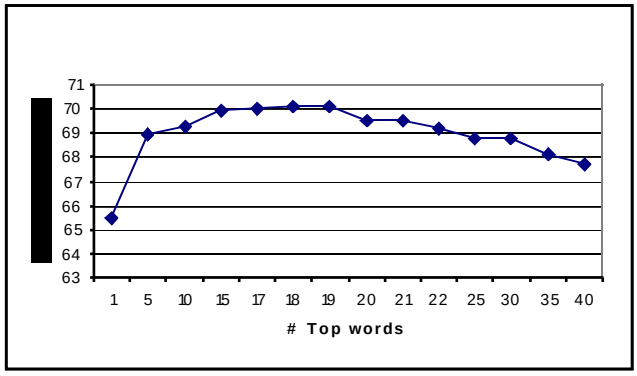

Figure 4. The effect of the number of top words selected from the test document on level-3 classification accuracy.

It is interesting to note that, as we move down the hierarchy, the classifiers perform better with more words extracted from the test documents. This is because they need more information in order to make finer-grained distinctions between the classes.

12. COMPARISON BETWEEN FLAT CLASSIFIER AND HIERARCHICAL CLASSIFIERS.

To compare the relative performance of the flat classifier and the hierarchical classifiers, the same set of test documents was used and the results are graphically represented below. As shown in the Fig 5, the flat classifier produced an exact match precision of only 48.2% whereas, with the hierarchical classifiers, 70.1% of documents classified had an exact match. As the graph indicates, the use of hierarchy for text classification results in a significant *(p<2.56E-06)* improvement of 45.4% in exact match precision.

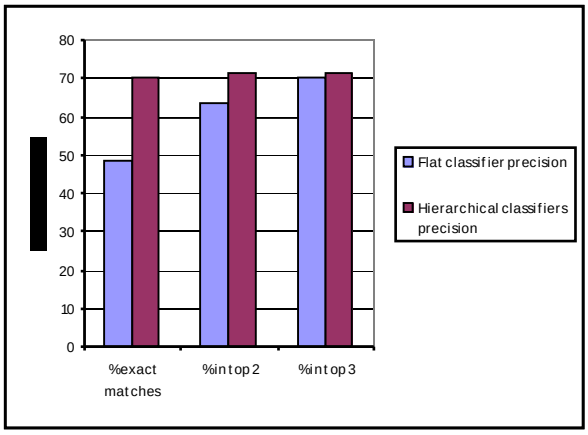

Figure 5. Comparison between the flat classifier and the hierarchical classifier.

13. CONCLUSIONS AND FUTURE WORK

The research described in this paper explores the use of hierarchical structure for classifying a large, heterogeneous collection of web content. An approach is presented which utilizes the existing rich hierarchical structure in order to facilitate the process. Rather than building a single massive classifier, a hierarchy of classifiers is generated which increase the

classifier precision by focusing only on a small set of classes, those relevant to task at hand. As it is shown, exploiting the relationships among classes and utilizing the hierarchical topic structure results in a considerable increase in the classifier precision.

14 .REFERENCES

[1] S.D'Alessio, K.Murray, R.Schiaffino, and A.Kershenbaum, The Effect of Using Hierarchical Classifiers in Text Categorization, In Proceeding of RIAO-00, 6th International Conference "Recherche d'Information Assistee par Ordinateur", pages 302--313, Paris, 2000.

[2] S.Dumais and H.Chen, Hierarchical Classification of Web Content, In SIGIR 2000.

[3] W.B Frakes, R.Baeza-Yates, Information Retrieval: Data Structures and Algorithms.

[4] S.Gauch, D.Ravindran, S.Induri, J.Madrid and S.Chadalavada, Internal Technical Report on KeyConcept - Information and Telecommunication Technology Center, University of Kansas.

[5]S.Gauch, J.Chaffee and A.Pretschner, Ontology-Based User Profiles for Search and Browsing, Web Intelligence and Agent Systems

[6] Open Directory Project. http://dmoz.org.

[7] A.Sun, E.Lim, and W.Ng, Performance Measurement Framework for Hierarchical Text Classification, Journal of the American Society for Information Science and Technology,54(11):pp.1014-1028, 2003.