Strategies for Minimising Errors in Hierarchical Web Categorisation

Wahyu Wibowo Hugh E. Williams
School of Computer Science and Information Technology
RMIT University, GPO Box 2476V
Melbourne, Australia, 3001.
{wwibowo,hugh}@cs.rmit.edu.au

ABSTRACT

On the Web, browsing and searching categories is a popular method of finding documents. Two well-known category-based search systems are the Yahoo! and DMOZ hierarchies, which are maintained by experts who assign documents to categories. However, manual categorisation by experts is costly, subjective, and not scalable with the increasing volumes of data that must be processed. Several methods have been investigated for effective automatic text categorisation. These include selection of categorisation methods, selection of pre-categorised training samples, use of hierachies, and selection of document fragments or features. In this paper, we further investigate categorisation into Web hierarchies and the role of hierarchical information in improving categorisation effectiveness. We introduce new strategies to reduce errors in hierarchical categorisation. In particular, we propose novel techniques that shift the assignment into higher level categories when lower level assignment is uncertain. Our results show that absolute error rates can be reduced by over 2%.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Information Search and Retrieval Search Process; H.3.4 [**Information Storage and Retrieval**]: Information Search and Retrieval Information Filtering

General Terms

Algorithms, Design, Experimentation

Keywords

Categorisation, Hierarchical Categorisation, Error Reduction

CIKM'02, November 4–9, 2002, McLean, Virginia, USA.

1. INTRODUCTION

The number of documents available in digital format continues to grow at an astounding rate. Many documents are now produced in digital formats such as computer, press, and medical reports, and sound or digital images, while others are transformed from paper-based documents. New domains continue to emerge that add to this growth in document production. To manage documents, processing is often required such as attaching a title, summary, or category to each document. However, for large numbers of documents, it is impractical for human assessors to manage documents manually.

Categorisation is the process of assigning category labels to a document. As with paper-based documents, categorisation of digital data is useful to organize documents for storage and retrieval. Documents with the same category or categories can be stored in the same logical location. Using categorisation, the search space for document retrieval can be narrowed to reduce the time taken in meeting user information needs. It can also improve the effectiveness of systems in answering information retrieval questions. An automatic categorisation system is one solution for categorising digital data that avoids the constraints of a manual process with human assessors.

As in traditional library systems, online documents are often hierarchically categorised. For example, Yahoo! provide a browsable hierarchy that can be used to meet an information need. At Yahoo! the category "Richmond Tigers" is a leaf node of the hierarchy. It has a parent of "Clubs and Teams", that in turn has a parent "Australian Football League", which is in turn a child of "Football (Australian)", which is in turn a child of "Sports". The parent of "Sports" is the root of the hierarchy, "Recreation". A user would typically browse the hierarchy to search for "Richmond Tigers" by beginning with the root. As shown by this example, the category labels of parent nodes typically subsume the characteristics of their children. Hence, it is common that the higher the position of category labels in a hierarchy, the more general the characteristics of those category labels. At Yahoo!, documents are manually assigned to categories by human assessors.

Different strategies have been proposed for automatic hierarchical categorisation. Some techniques traverse the hierarchy from root to leaf, deciding which path to follow as each document is categorised. Other approaches develop several

categorisers for a hierarchy, that is, they choose the best category for a document in several steps by categorising a document progressively into different levels of the hierarchy. Traversal of a hierarchy is usually faster than a multiple-categoriser approach when the number of children of each node is high—which is usual in web categories—but it may be slower when the number of children is low. Each approach also has accuracy drawbacks: when traversing a hierarchy, errors are not recoverable, that is, an assignment error that occurs high in the hierarchy is carried down towards the leaf nodes; in contrast, when using multiple categorisers, it is difficult to use the hierarchical relationships to aid in assignment decisions.

In this paper, we investigate automatic categorisation into a web hierarchy and introduce new strategies to reduce assignment errors in hierarchical categorisation. Our aim is to minimise errors by allowing conservative assignment, where assignment to categories that are not at the leaf of the hierarchy is permitted when leaf assignment is uncertain. Our most successful approach uses a bottom-up scheme that we show is more accurate than the flat leaf-level categorisation. Overall, we show that the absolute rate of errors in hierarchical categorisation errors can be reduced by up to 2.6% for web categorisation and up to 4.6% for non-web categorisation.

2. CATEGORISATION

A document processing system often requires categorisation to assign labels to documents. A categoriser to perform this task can be built automatically through a learning process, where a collection of pre-categorised documents are used to *train* the categoriser to recognise the features of specific categories. Once training is complete, a categoriser can assign uncategorised documents to categories automatically, with little or no manual intervention.

In this research work, we use a *similarity-based linear categoriser* that determines whether a document should be assigned to a category based on the computation of a linear function [10]. This approach is effective, and can be used efficiently on large scale datasets on general-purpose hardware [15]. Moreover, these categorisers are *term-based*, that is, tokens such as words are assigned weights that represent their importance in each category. It is less clear how the techniques we propose could be applied to other state-of-the-art hierarchical categorisers such as support vector machines (SVMs) [2, 5]. However, it is also unclear whether SVMs are suited to categorising complete documents in large hierarchies because of the nature of SVM as a binary categoriser and the high computational requirements of training; recent work has considered reducing the features used in categorisation of web documents by using only summaries [5], feature selection [8, 11], and using only frequent words from each category [5].

We investigate the performance of our techniques with a Rocchio [10, 12] categoriser. We have also experimented with Probabilistic and TFIDF categorisers [7, 9], and we report these results in brief.

Rocchio Categoriser

A Rocchio categoriser is a training-based categoriser developed from the Rocchio relevance feedback model [7, 10, 12]. For each training document, the weight of the vector of each category is modified so that when a document is a category member, the weights of category terms are increased. Likewise, when a document is not a category member the weights of the terms in the category vector are decreased. The result is a vector for each category w of length t, where t is the number of distinct terms—usually words—in the collection; in this way, all category vectors are of length t and contain all terms. Terms that are representative of a category have positive weights and terms that are not have negative weights.

Given the Rocchio measure, it is a simple batch process to derive a category feature vector for a given set of categories and training documents. Having derived the vectors for each category, categorisation proceeds by computing the similarity of a vector derived from each document with the category vectors by using the cosine measure [14]. The document is then assigned to the statistically most similar category; an alternative—which is not applicable to our web hierarchy—is multiple category assignment, where all categories that exceed a threshold score are assigned to a document.

TFIDF Categoriser

Joachims [7] has proposed a direct application of the well-known TF.IDF ranking measure to categorisation. If d is a new document to categorise, C is a set of categories, and F is a set of features (such as words), then the TFIDF score of document d on category c can be computed as follows:

$$\text{TFIDF}(d,c) = \frac{\sum_{w \in F}(TF(w,d).IDF(w)).(TF(w,c).IDF(w))}{\sqrt{(\sum_{w \in F, c \in C}(TF(w,c).IDF(w))^2)}}$$

where (w, d) refers to term w in document d, and (w, c) refers to term w in all documents of category c.

The category of a document can be assigned by computing the TFIDF scores for all categories. The category with the highest score can be selected as the document category—we call this single category assignment—or a category can be assigned if the score is above a threshold point for that category resulting in multiple category assignment.

Hierarchical Categorisation

Most categorisation systems assume independence between the categories that may be assigned to a document that is being categorised. We refer to these as *flat* categorisers because a document is assigned by selecting one or more categories from the complete set, and the assignment decision does not consider any relationships between categories.

Hierarchical categorisation is the process of assigning category labels from a set that are organised into a directed, acyclic tree structure. For example, a hierarchy of Usenet news categories is shown in Figure 1. In this example, there are five *leaf nodes* that represent the most specific categories: `rec.sport`, `rec.hobby`, `rec.music`, `talk.politics`, and `talk.religion`. The leaf nodes, in turn, have parent categories that subsume the characteristics of their children. The leaf nodes `talk.politics` and `talk.religion` have the parent category `talk`, while `rec` is the parent of the leaf nodes `rec.sport`, `rec.hobby`, and `rec.music`. The parents of the leaf nodes, in turn, share a single parent that is the *root node* of the hierarchy. In contrast to a hierachical categorisation approach, a flat categoriser would consist of only the five leaf node categories.

Given a hierarchical relationship of categories, at least two approaches to developing categorisers are possible. First,

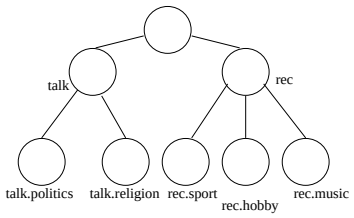

Figure 1: ***A example hierarchy for five Usenet news categories.***

several flat categorisers can be trained, each for a subset of the hierarchy and these used to make decisions as to where in a hierarchy a document is assigned [5, 8, 13]. Second, a single categoriser can be developed but the categorisation decisions can be made by considering the similarity of documents to nodes in different levels of the hierarchy. The second approach is a variation of flat categorisation, but hierarchical information is considered when making assignments, and we consider this approach in detail later.

Different approaches can also be used for hierarchical categorisation. Assignment may only be permitted at the leaf level or, alternatively, could be permitted anywhere in the hierarchy. In addition, multiple category assignment may be permitted or, alternatively, only one category may be allowed. Most previous work in hierarchical categorisation focuses on leaf-level assignment, and most accuracy results report the difference in performance of assigning documents to leaf-level categories using flat and hierarchical categorisers.

Perhaps the most intuitive method of categorisation is a *top-down approach* that begins by determining the similarity of a document to the children of the root node. In our example above, we would consider the similarity of the document to the `talk` and `rec` categories. If, for example, the document is most similar to `rec`, then categorisation proceeds down the hierarchy by considering which of the three children of `rec` is the most similar to the document. The advantage of considering only the children of a parent node is two-fold: first, only a fraction of the nodes at each level are considered, leading to fast categorisation [3]; second, assuming the parent category is correct, the accuracy of child assignment is likely to be higher because there are fewer categories to choose from.

A second approach to categorisation is to calculate the similarity of a document to the categories in each level of the hierarchy and to then consider the results from each level in making an assignment decision. We call these *combination* schemes. Dumais and Chen [5] propose two approaches in this class: first, they categorise documents into a parent level category and the child level category separately, and then multiply the parent and child assignment probabilities, and assign the child category if the result exceeds a threshold; second, they apply a Boolean decision rule where the probability at the parent and child levels must each exceed a threshold, where the threshold for the parent is lower than that of the child.

An advantage of hierarchical categorisation is that it has been shown to improve accuracy over a flat approach. By considering the relationship between categories, better assignment decisions are made. Choosing the correct categories in higher levels of the hierarchy has been shown to be more reliable—for the reason that the categories are broader and more distinct than lower-level categories—and this aids assignment at the leaf nodes [17]. Perhaps surprisingly, these improvements are usually small: in recent work, Dumais and Chen [5] showed that a hierarchical approach is around 4% more accurate than a flat approach, a result that is consistent with those reported elsewhere [3, 4, 16]. However, the results we show in Section 4 contradict these results; as we discuss later, we believe this is because our hierarchy is a more realistic test set than those previously used.

3. MINIMISING ERRORS

In this section, we propose novel strategies to categorise web documents into hierarchies. Hierarchical categorisation is usually considered as an alternative to flat categorisation, where the goal is achieving more accurate leaf-level assignment. However, while hierarchical categorisation offers small improvements in accuracy, it also has the potential to reduce errors in assignment. We explore this novel idea in this section.

One possible application of hierarchical categorisation is to be conservative in assignment and thus avoid completely incorrect categorisation. Specifically, when there is uncertainty in assigning a leaf category, a parent category can be chosen instead. Correct assignment to parent categories is not as specific as assignment to a leaf, but it is still useful: by suggesting a parent category, a human assessor can manually categorise the document into a leaf category or, alternatively, the document can be permanently associated with the parent as occurs in many web hierarchies such as DMOZ and Yahoo!. Clearly, however, categorising all documents at or near the root of the hierarchy would not be considered useful.

A typical top-down categorisation begins by considering which category at the top level of the hierarchy best matches the document being categorised. The best category is selected and the categorisation continues by considering the child categories of the selected category. This traversal continues until a leaf level category is selected, and the leaf category is assigned to the document.

We have investigated the role of a hierarchy in categorisation with the goal of minimising error and maximising leaf assignments. A new approach that permits parent assignment to reduce errors is a modified top-down approach. In our modified approach—which we call *top-down with parent preference*—the relationships are used to allow assignment to occur into nodes that are not only at the leaf. Our aim is to minimise errors when there is uncertainty in the assignment process. As in the unmodified top-down approach, the best top level category is selected. The categorisation then considers all categories in the next level of hierarchy—not only the children of the best parent—and the best category is selected. If the best top-level category is the parent category of the best lower-level category, then categorisation continues to the next level. However, if the best lower-level category is not a child of the best top-level category, then categorisation stops and the parent category is assigned to the document. This is a variation of the Boolean decision approach of Dumais and Chen [5] but without thresholds and with a goal of minimising errors.

Figure 2 shows three examples of top-down with parent preference. At the top of the figure, a sample hierarchy is shown with the best matching categories to a document

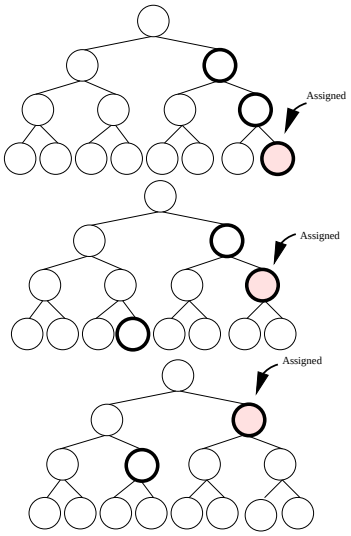

Figure 2: ***Sample assignments using the top-down approach with parent preference***

at each level shown in bold. Because the leaf-node category is the child of the best middle-level category, and the best middle-level category is the child of the best top-level category, then the leaf-level category is assigned; the assignment is indicated by shading. In the middle of the figure, the middle-level category is assigned because it is a child of the best top-level category, but the best leaf-level category is not assigned because it is not the child of the best parent. At the bottom of the figure, the best top-level category is assigned because the middle-level category is not a child; categorisation does not continue to the leaf level in this example.

Our next class of error-minimisation strategies we call the *bottom-up with threshold* categorisation schemes. In bottom-up with threshold categorisation, leaf-level categorisation is considered first, and the best-matching n leaf-level categories are determined. If the best-ranked category has the same parent as any other child in the top n, then the best-ranked leaf category is assigned. Otherwise, its parent is selected and, if the hierarchy has more than two levels, this process is repeated for the next level. If a category has less than n siblings, then n is set to the number of siblings. This approach assumes that categories under the same parent are more closely related than categories that have a different parent.

Our final class of error-minimisation strategies we call the *bottom-up with parent support* categorisation scheme. In bottom-up with parent support categorisation, leaf-level categorisation is considered first. If the best-ranked leaf category has the parent which is the best ranked parent category, then the best-ranked leaf category is assigned. Otherwise, the process repeats with the best parent category. This approach assumes that the best category has a high probability of being the correct category if it is supported by the result from its parent. Figure 3 shows three examples of bottom-up with parent support.

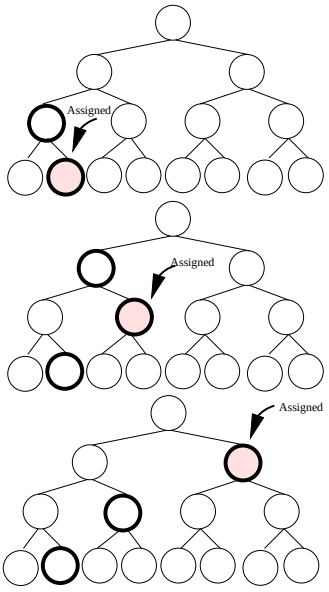

Figure 3: ***Sample assignments using the bottom-up with parent support approach***

4. RESULTS

In this section, we report the results of categorisation into a four level web hierarchy with our error-reduction techniques. For comparison, we report the baseline performance of flat categorisation schemes that only consider leaf-level categories, an unmodified top-down approach, and the combined scores approaches of Dumais and Chen [5].

4.1 Collections

Several test collections have been used as training and test sets for linear text categorisers. Unfortunately, most of the collections do not possess explicit hierarchical identification or the implicit hierarchy is only of depth two. We evaluate the performance of our algorithms on a new web collection that we developed from URLs listed in the *Open Directory Project*.[1]

The Open Directory Project—also known as DMOZ or Directory Mozilla—is hosted and administered by the Netscape Communication Corporation. A large team of volunteers maintain the hierarchical collection of web categories by organising hypertext links into the categories. From the available web directories, we selected 53 categories that are structured into a four level hierarchy. For each category, we downloaded up to 150 websites that were members of that category; each web site was crawled up to a two level depth and a maximum of 10 HTML pages per website were collected. We preprocessed all documents by removing HTML tags and other markup by following the approach of Williams and Zobel [18]; we also removed documents that were unsuccessfully retrieved, for example those that resulted from an HTTP 404 response or "Not Found". The final collection comprised of 5,296 documents. We divided the documents into 3,550 training documents (67%) and 1,747 test documents (33%).

For comparison, we tested our algorithms using the often-studied Reuters-21578 collection, where the documents have

[1]See `http://www.dmoz.org`

Method	Level 1	Level 2	Level 3	Level 4
Per-level	92.04/100.00	83.51/100.00	78.71/100.00	72.81/100.00
Topdown Leaf-only	—	—	—	68.92/100.00
Combination – Sum	—	—	—	71.78/100.00
Combination – Multiplication	—	—	—	71.61/100.00
Topdown Parent-preference	1.95/ 4.18	2.40/ 6.81	2.80/ 7.67	65.37/ 81.34
Bottom-up Threshold=2	8.30/ 13.17	17.52/ 23.18	16.43/ 21.12	33.09/ 42.53
Bottom-up Threshold=3	2.69/ 3.89	10.82/ 18.37	16.26/ 21.35	43.62/ 56.38
Bottom-up Parent Support	1.26/ 2.18	1.60/ 2.52	2.80/ 7.90	68.12/ 87.41

Table 1: ***Accuracy results for categorisation into the Partial DMOZ four level web hierarchy. Figures for each level are presented as a pair: the first figure indicates the percentage of documents that are correctly categorised into that level, while the second figure indicates the total percentage of all documents that are categorised at that level.***

been arranged into an implicit hierarchical structure. The Reuters-21578 collection is a modified version of one formerly provided by the Carnegie Group, Inc. that was used to evaluate the CONSTRUE system in 1990.[2] The collection has been divided into different training and test sets by many authors, including the *mod-Apte split* [1] that removes all unlabelled documents. Hayes recategorised the 374 categories of the Mod-Apte split into more general divisions [6], and identified 8 broad parent categories for the `Topics` category and 135 child categories under the `Topics` category. We use this hierarchical arrangement of the Reuters-21578 collection in our experiments. The collection contains 7,775 training documents and 3,019 test documents.

Around 505 documents in the Reuters training set and 188 in the test set belong to more than one category. We use the multiply-assigned training set documents to train all categories of which they are members. We also assume in our evaluation—which we descibe next—that documents are correctly assigned if they are assigned to any category of which they are a member.

4.2 Evaluation

Several different approaches have been proposed for quantifying the effectiveness of linear categorisation schemes [19]. *Recall* and *precision* are two measurements widely used to evaluate retrieval systems. Recall measures the ability of a technique to identify the categories of a document, while precision measures the correctness of the assignments that are made. Other measures include *accuracy*, *fallout*, *error*, and the *F measure*.

It is difficult to apply existing measurement techniques to our any level assignment techniques and, to our knowledge, there has been no previous evaluation of similar hierarchical categorisation experiments. The particular difficulty is that—in the case where a parent is assigned instead of a child leaf category—it is difficult to assess the degree of correctness of a non-leaf level assignment. Therefore, we show for each level of a hierarchy a pair of n/m figures, where n shows the percentage of correctly assigned documents and m shows the percentage of the total number of assignments that are at that level. The percentage is measured from the total number of test documents. In addition to showing the degree or accuracy of assignment—correct and incorrect—for each level on the hierarchy, this method of evaluation also shows the distribution of hierarchical assignment amongst the hierarchy levels.

[2]The Reuters-21578 collection is available from `http://www.research.att.com/~lewis/`

4.3 Experiments

Table 1 shows the results of our experiments using the Partial DMOZ collection. The first row of the table shows the baseline performance of the Rocchio categoriser. For example, the level one Rocchio result of 92.04/100.00 indicates that 92.04% of documents are correctly categorised when 100% of documents are categorised into only the first level of the hierarchy. The level four result is a baseline where categorisation is only attempted into the leaf categories, that is, the hierarchy is not considered. At the leaf level, the Rocchio categoriser correctly assigns 72.81% of documents; our TFIDF categoriser achieved 71.84% and the probabilistic categoriser 65.83%.

Rows two to four of Table 1 show the performance of standard hierarchical categorisation techniques. To our knowledge, results of categorisation with a four level hierarchy have not previously been reported. Perhaps surprisingly, the top-down and combination techniques do not improve the accuracy of leaf level categorisation; this result is confirmed by the alternative F_1 accuracy measure. The lack of improvement from top-down categorisation is because assignment errors in higher-level categorisation are not recoverable, that is, once an assignment error is made the traversal continues to a leaf node but the assignment will be incorrect. From another perspective, for an assignment to be correct in top-down categorisation into a four level hierarchy, four correct assignment decisions must be made. Combined score schemes also have a drawback: the final assignment decision may be incorrect when a correct upper-level assignment and an incorrect lower-level assignment (or vice-versa) are combined.

Row five shows the results of our first novel hierarchical scheme—top-down with parent preference—and the accuracy result is around 7% less than the baseline. However, only 81.34% of the documents are categorised into the leaf level. Of the remaining documents, 4.18% are categorised into the first level and almost half (1.95%) are correctly categorised at that level. At the second level, 2.40% of 6.81%, and 2.80% of 7.67% are correct at the third level. In total, 72.52% are correctly categorised into either the correct leaf category, or a parent, grandparent, or great-grandparent of a correct leaf category. Table 2 shows this overall result.

Method	Total	
	Correct	Incorrect
Per-level	**72.81**	27.19
Topdown Leaf-only	68.92	31.08
Combination Rocchio – Sum	71.78	28.22
Combination Rocchio – Multiplication	71.61	28.39
Topdown Parent-preference	72.52	27.48
Bottom-up Threshold=2	**75.33**	24.67
Bottom-up Threshold=3	**73.38**	26.62
Bottom-up Parent Support	**73.78**	26.22

Table 2: ***Overall accuracy results for categorisation into the Partial DMOZ four level web hierarchy. The overall "correct" result is the percentage of documents categorised into the correct leaf node, its parent, its grandparent, or its great-grandparent.***

The results with our bottom-up categorisers are shown in Tables 1 and 2. The bottom-up scheme with a threshold of two is particularly successful in reducing errors: the overall accuracy is around 2.5% better than the baseline but with a significant loss of leaf level accuracy. As expected, higher values of n increase the fraction of successful child assignment but reduce overall accuracy.

Our bottom-up schemes are preferable for reducing errors. Overall, the error-free performance of allowing non-leaf level assignment is good: compared to the baseline, the Rocchio result improves from 72.81% to 75.33%, an absolute increase of 2.52%. A particular advantage of bottom-up schemes is that they are insensitive to hierarchy depth, since categorisation begins with the leaf nodes and proceeds towards the root as inconsistent categorisation occurs. We tested this by reducing the height of our Partial DMOZ collection through removal of level one, and then levels one and two. As expected, we found that top-down results improve as the hierarchy depth decreases. However, bottom-up results are robust: both the leaf categorisation accuracy and the overall result for bottom-up schemes remains roughly constant. We found similar results with probabilistic and TFIDF categorisers.

To give a broader picture of our proposed categorisation approaches, we experimented with a non-web collection, the Reuters Topics Supercategory. We found—in agreement with other experiments with hierarchical categorisation using two level categories—that existing top-down categorisation schemes improve accuracy by around 3.2% and combination approaches improve accuracy by around 2.5%. Our novel top-down and bottom-up categorisation experiments showed similar improvements to those in the Partial DMOZ collection: the leaf level accuracy of top-down with parent preference is around 3% less than the baseline, but the overall accuracy result is 4.5% better. The bottom-up schemes perform similarly: the bottom up with threshold schemes have low leaf level accuracy, but the overall result is 1%–2% better than the baseline; the parent support scheme has leaf level accuracy 3% lower than the baseline, but the overall result is almost 5% better. We confirmed these results with our probabilistic and TFIDF categorisers, and found similar trends and improvements.

5. CONCLUSION

Automatic categorisation is a scalable and practical solution to the problem of organising large numbers of documents. Hierarchical categorisation techniques—which organise documents into tree structures—have been shown to be more accurate than flat categorisation schemes for hierarchies of two levels. Hierarchical schemes are more effective because the relationships between categories can be used to make better category assignment decisions.

In this paper, we have investigated the performance of hierarchical schemes on a deeper web hierarchy, and whether hierarchies can also be used reduce errors in categorisation. Assignment into broader categories that are not at the leaves of the hierarchy is less error-prone, but has the disadvantage that it is less specific. We argue that correct assignment into less-specific categories is better than incorrect assignment into specific categories: after assigning documents into higher levels of a hierarchy, human assessors can select the correct child from a short list or the higher-level categorisation can be tolerated as is the case in many web hierarchies.

We have used a four level web category in this work. With this hierarchy, we have shown that existing top-down and combination schemes do not improve the accuracy of categorisation. This result is not surprising: for a hierarchy of four levels, four correct assignment decisions must be made for the overall result to be correct. We have proposed several new schemes that work well for deeper hierarchies. These schemes have lower leaf level accuracy, but permit conservative assignment into the correct branch of the hierarchy. We plan to extend this work to include deeper hierarchies of web documents. We also believe our techniques would be as effective in improving the performance of other categorisers and other collections.

6. REFERENCES

[1] C. Apte, F. Damerau, and S. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.

[2] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, July 1992. ACM Press.

[3] 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, FR, 2000.

[4] S. D'Alessio, K. Murray, R. Schiaffino, and A. Kershenbaum. Category levels in hierarchical text categorization. In *Proc. of EMNLP-98, 3rd Conference on Empirical Methods in Natural Language Processing*, Granada, Spain, 1998. Association for Computational Linguistics, Morristown.

[5] S. T. Dumais and H. Chen. Hierarchical classification of Web content. In N.J. Belkin, P. Ingwersen, and M.-K. Leong, editors, *Proc. ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 256–263, Athens, Greece, 2000. ACM Press, New York.

[6] P.J. Hayes and S.P. Weinstein. CONSTRUE/TIS: a system for content-based indexing of a database of news stories. In A. Rappaport and R. Smith, editors, *Proceedings of IAAI-90, 2nd Conference on Innovative Applications of Artificial Intelligence*, pages 49–66. AAAI Press, Menlo Park, 1990.

[7] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In D.H. Fisher, editor, *Proc. of the 14th International Conference on Machine Learning*, pages 143–151, Nashville, 1997. Morgan Kaufmann, San Francisco.

[8] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In D.H. Fisher, editor, *Proc. of the 14th International Conference on Machine Learning (ICML97)*, pages 170–178, Nashville, 1997. Morgan Kaufmann, San Francisco.

[9] L.S. Larkey and W.B. Croft. Combining classifiers in text categorization. In H.-P. Frei, D. Harman, P. Schauble, and R. Wilkinson, editors, *Proc. ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 289–297, Zurich, Switzerland, 1996.

[10] D.D. Lewis, R.E. Schapire, J.P. Callan, and R. Papka. Training algorithms for linear text classifiers. In H.-P. Frei, D. Harman, P. Schäuble, and R. Wilkinson, editors, *Proc. ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 298–306, Zurich, Switzerland, 1996.

[11] D. Mladenic and M. Grobelnik. Feature selection for classification based on text hierarchy. In *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, Pittsburg, USA, 1998.

[12] J.J. Rocchio. Relevance feedback in information retrieval. In *The Smart Retrieval System — Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Englewood, Cliffs, New Jersey, 1971.

[13] M. E. Ruiz and P. Srinivasan. Hierarchical neural networks for text categorization. In M.A. Hearst, F. Gey, and R. Tong, editors, *Proc. ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 281–282, Berkeley, CA, 1999.

[14] G. Salton. *Automatic Text Processing*. Addison Wesley, Massachusetts, 1989.

[15] V. Shanks and H.E. Williams. Fast categorisation of large document collections. In *8th International Symposium on String Processing and Information Retrieval (SPIRE2001)*, pages 194–204, San Rafael, Chile, 2001.

[16] A.S. Weigend, E.D. Wiener, and J.O. Pedersen. Exploiting hierarchy in text categorization. *Information Retrieval*, 1(3):193–216, 1999.

[17] W. Wibowo and H.E. Williams. On using hierarchies for document classification. In *Proc. Australian Document Computing Conference*, pages 31–37, Coffs Harbour, Australia, 1999.

[18] H.E. Williams and J. Zobel. Searchable words on the web. *International Journal of Digital Libraries*. To appear.

[19] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90, 1999.