Turning Yahoo into an Automatic Web-Page Classifier

Dunja Mladenić[1]

Abstract. The paper describes an approach to automatic Web-page classification based on the Yahoo hierarchy. Machine learning techniques developed for learning on text data are used here on the hierarchical classification structure. The high number of features is reduced by taking into account the hierarchical structure and using feature subset selection based on the method known from information retrieval. Documents are represented as feature-vectors that include n-grams instead of including only single words (unigrams) as commonly used when learning on text data. Based on the hierarchical structure the problem is divided into subproblems, each representing one on the categories included in the Yahoo hierarchy. The result of learning is a set of independent classifiers, each used to predict the probability that a new example is a member of the corresponding category. Experimental evaluation on real-world data shows that the proposed approach gives good results. For more than a half of testing examples a correct category is among the 3 categories with the highest predicted probability.

1 Introduction

Yahoo [2], a well known Web-pages hierarchy is human constructed and designed for human Web browsing. Already classified documents used to build a hierarchy are Web documents, making the hierarchy biased toward human knowledge areas that are represented in Web documents. The Yahoo hierarchy itself (without the top category *'Regional'*) is currently built on about a million Web documents located on the Internet all around the world. We refer to these documents as *actual Web documents*. Hyperlinks to that documents are organized in about 50,000 *Yahoo Web documents* where each document represents one of the included categories with documents representing more general categories closer to the root of the hierarchy. Yahoo documents are connected with hyperlinks forming a hierarchical structure. Each document classified in the Yahoo hierarchy appears only once, but there can be several hyperlinks in the hierarchy leading to it.

The category name is given by the keywords on the path from the root of the hierarchy to the category node (eg., *'Sport'* a subcategory of *'Science'* in Figure 1 is named *'Science: Sport'*). A more specific category is named by adding a keyword to the name of more general category directly connected to it. Some nodes at the bottom of the hierarchy contain mostly hyperlinks to actual Web documents, while the other contain mostly or even only (eg., 'Science: Sport' in Figure 1) hyperlinks to other Yahoo Web documents.

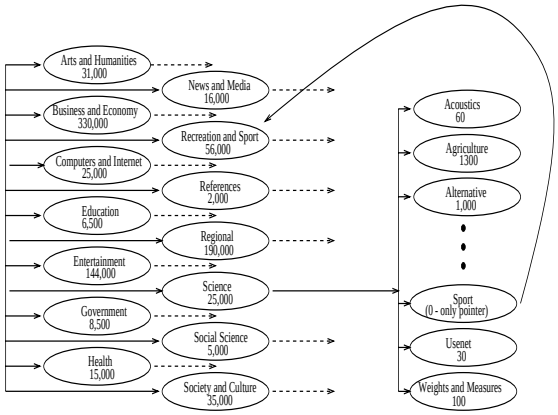

Figure 1. Top level of the Yahoo categorization and the first level of subcategories in 'Science' with the approximate number of documents in each category (site in UK & Ireland, Nov. 1997).

There are currently fourteen top level Yahoo categories whose name includes only one keyword. Figure 1 shows them with the approximate number of actual Web documents located under each category. Each of the top categories is further represented with the hierarchical structure of more specific categories. As example we show in Figure 1 the part of the first-level subcategories in the 'Science' top category ranging from 'Acoustics' to 'Weights and Measures'.

2 Machine learning setting

We use a Naive Bayesian classifier on text documents represented as feature-vectors using the bag-of-words representation as commonly used in learning on text data (eg., [4], [6], [9]). Our document representation additionally includes not only single words (unigrams) but also up to 5 words (1-grams, 2-grams, ... 5-grams) occurring in a document as a sequence (eg., 'machine learning', 'world wide web'). We reduce the high number of features by pruning words contained in the publicly available 'stop-list' of common English words and by pruning low frequency features. We use efficient procedure for feature generation that is performed in n passes over documents, where i-grams are generated in the i-th pass. At the end of each pass over documents all low frequency features

[1] Department of Intelligent Systems, J.Stefan Institute Jamova 39, 1111 Ljubljana, Slovenia

ECAI 98. *13th European Conference on Artificial Intelligence*
Young Researcher Paper
Edited by Henri Prade
Published in 1998 by John Wiley & Sons, Ltd.

are deleted (we check for frequency ≤ 3). Each new pass generates features of length $i+1$ only from the candidate features of length i generated in the previous pass. This process is similar to the large k-itemset generation used in association rules algorithm described in [1].

We divide the whole problem into subproblems each corresponding to the individual category. For each of the subproblems, a classifier is constructed that predicts the probability that a document is a member of the corresponding category. A set of positive and negative examples for each subproblem is constructed from the given hierarchical structure. The final result of learning is a set of specialized classifiers each based only on a small subset of features (similar to the learning a classifier for each split in the Reuters hierarchy [5]).

Since the idea is that each classifier can distinguish between the documents that should be assigned the category it represents and the other documents, we define a set of negative examples as examples from the whole hierarchy. The set of positive examples is constructed separately for each category from items on the Yahoo document representing the category. These are items containing hyperlinks to other Yahoo documents representing subcategories and items containing hyperlinks to actual Web documents. Our assumption here is that an item on Yahoo document contains words representative for the document it points to. In this way we reduce the time and space needed to collect and store training data. The actual Web documents are used as testing examples.

3 Experimental results

Our experiments are performed using our recently developed machine learning system **Learning Machine** [3] that supports usage of different machine learning techniques on large data sets with especially designed modules for learning on text and collecting data from the Web. In our experiments we observe the influence of different numbers of features (vector size in feature-vector document representation) on an independent set of (300 for *'Computers and Internet'* and 200 for the two smaller domains) testing examples selected randomly from the actual Web documents accessible from the hierarchy domain. We get our categorization results from the set of independent classifiers, each potentially having different number of features. Thus we express the vector size as a factor of the number of features in the positive examples, since the set of negative examples is the same for all classifiers. In this way, a classifier for a larger category is using more features than a classifier for some smaller category, while both classifying the same testing example. For each testing example we observe a list of categories each assigned a probability as a result of consulting a corresponding subproblem classifier. Sorting categories according to that probability gives ranking that we use to get the rank of the correct category. For each testing example we report rank and probability assigned to the correct category. To get summary results over testing examples, we give median with the lower and upper quartile, since some of the testing examples are rather non-typical of their category, containing eg., a welcome page or only one sentence asking for language preference or an error message or a page giving redirection.

We report the results of a Naive Bayesian classifier on 3 domains representing 3 of the top 14 Yahoo categories: *'References'* having 129 categories and 923 (1-grams:697+2-grams:196+3-grams:27+4-grams:3+5-grams:0) features, *'Education'* having 349 categories and 3,215 (1,928+1,067+186+28+6) features and *'Computers and Internet'* having 2,652 categories and 7,631 (5049+2276+261+38+7) features. In all three domains the best performance is achieved when only a small number of features is used and features are selected using the Odds ratio as scoring measure as suggested in [7] and [8]. More specifically, on *'References'* the median of the correct category rank is 2 and the median of the correct category probability over 0.99, ie., for 50% of the testing examples rank 1 or 2 is assigned to the correct category and the probability returned by the correct category classifier, showing its confidence that the example is a member of the (correct) category is > 0.99. On *'Education'* and on *'Computers and Internet'* the median of the correct category rank is 3 and the median of the correct category probability over 0.99, ie., for 50% of the testing examples rank 1, 2 or 3 is assigned to the correct category and probability > 0.99. On *'References'* for the most vector sizes the lower quartile for rank and probability is 1, meaning that for 25% of the testing examples the highest rank is assigned to the correct category and the probability 1. On *'Education'* and *'Computers and Internet'* for the most vector sizes the lower quartile for rank is 1 or 2 and for probability 1.

ACKNOWLEDGEMENTS

This work was financially supported by the Slovenian Ministry for Science and Technology. Part of this work was performed during the authors stay at Carnegie Mellon University. Author is grateful to Tom Mitchell and his machine learning group at Carnegie Mellon University. Author is also grateful to anonymous reviewers. Many thanks to Marko Grobelnik for intensive cooperation on this work.

REFERENCES

[1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I., 1996. Fast Discovery of Association Rules, In Fayyad et al. (eds.), *Advances in Knowledge Discovery and Data Mining* AAAI Press/The MIT Press, pp. 307-328.

[2] Filo, D., Yang, J., 1997. Yahoo! Inc. www.yahoo.com/docs/pr/

[3] Grobelnik, M., Mladenić, D., 1998. Learning Machine: design and implementation, *Technical Report IJS-DP-7824*, SI.

[4] Joachims, T., 1997. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, *Proc. of the 14th International Conference on Machine Learning ICML97*, pp. 143-151.

[5] Koller, D., Sahami, M., 1997. Hierarchically classifying documents using very few words, *Proc. of the 14th International Conference on Machine Learning ICML97*, pp. 170-178.

[6] Mladenić, D., 1996. Personal WebWatcher: Implementation and Design, *Technical Report IJS-DP-7472*, SI.

[7] Mladenić, D., 1998. Feature subset selection in text-learning, *Proc. of the 10th European Conference on Machine Learning.*

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

[9] Pazzani, M., Billsus, D., 1997. Learning and Revising User Profiles: The Identification of Interesting Web Sites, *Machine Learning 27*, Kluwer Academic Publishers.