UDC: 007.52
Preliminary communication

ASSIGNING KEYWORDS TO DOCUMENTS USING MACHINE LEARNING

Dunja Mladenić

Department of Intelligent Systems, J.Stefan Institute, Ljubljana, Slovenia
E-mail:Dunja.Mladenic@ijs.si

Marko Grobelnik

Department of Intelligent Systems, J.Stefan Institute, Ljubljana, Slovenia
E-mail:Marko.Grobelnik@ijs.si

This paper describes the usage of machine learning techniques to assign keywords to documents. The large hierarchy of documents available on the Web, the Yahoo hierarchy, is used here as a real-world problem domain. Machine learning techniques developed for learning on text data are used here in the hierarchical classification structure. The high number of features is reduced by taking into account the hierarchical structure and using a feature subset selection based on the method used in information retrieval. Documents are represented as word-vectors that include word sequences (n-grams) instead of just single words. The hierarchical structure of the examples and class values is taken into account when defining the subproblems and forming training examples for them. Additionally, a hierarchical structure of class values is used in classification, where only promising paths in the hierarchy are considered.

Keywords: machine learning, assigning keywords, Yahoo hierarchy, document categorization, F1-measure, F2-measure.

1. INTRODUCTION

Text documents can be characterized by a set of keywords giving an idea about the document content. This can be seen as additional information about the documents or as a kind of document abstraction. For example, many conferences require that each paper submission is accompanied by a title page containing a set of keywords describing the area to be discussed. Usually the authors are asked to select keywords from the predetermined set of keywords given in the conference call for papers. Here we will describe the usage of machine learning techniques for the problem of automatically assigning keywords to documents. Our set of keywords is defined by the domain, that is in this case the Yahoo hierarchy [4].

2. DOMAIN DESCRIPTION

We use the existing Web hierarchy as an example domain for learning document keywords. The keywords used in the Yahoo hierarchy for naming categories are

selected to describe category content. The categories being used have been constructed by humans and were designed for humans to browse the Web. Documents that are already classified and used to build a hierarchy are Web documents, making the hierarchy biased toward human knowledge areas that are represented in these Web documents.

The Yahoo hierarchy itself (without the huge category 'Regional') is currently built on approximately 900,000 Web documents located all around the Internet. Hyperlinks to those documents are organized in about 50,000 Yahoo Web documents. Each Yahoo document represents one of the included categories named by a set of keywords. These documents are connected with hyperlinks, forming a hierarchical structure with the more general categories closer to the root of the hierarchy. A category is denoted by keywords that describe category content and that appear on the path from the root of the hierarchy to the node representing the category. In other words, a more specific category is named by adding a keyword to the name of the more general category directly connected to it (one level higher in the hierarchy).

There are currently fourteen top level Yahoo categories each named with only one keyword. Each of the top categories is further represented by a hierarchical structure of more specific categories. For example, *'Machine Learning'* is under the top category *'Science'*, it is named *'Science: Computer Science: Artificial Intelligence: Machine Learning'* and thus we assign it four keywords *Science, Computer Science, Artificial Intelligence* and *Machine Learning*.

The domains we have generated from the Yahoo hierarchy of Web documents represent five out of the fourteen top level Yahoo categories: *'Entertainment', 'Arts and Humanities', 'Computers and Internet', 'Education'* and *'References'*. The data was obtained from the publicly accessible Yahoo Web site. Table 1 gives the domain characteristics including information about the number of nodes in the domain hierarchy, the number of features showing how many features represent the different length of word sequences, the number of examples (documents), the average distance between nodes measured as the average number of connections between any two nodes in the hierarchy and the average number of features in positive documents. The average number of features in positive documents is calculated as the average over the defined domain subproblems. For instance, the domain *'Entertainment'* has 8,081 nodes with an average distance of 7.56 between them, 30,998 features consisting of 15,144 1-grams, 11,21 2-grams, 2,970 3-grams, 1,059 4-grams, 505 5-grams and 79,011 actual Web documents.

From the data characteristics given in Table 1 it can be seen that the number of unique words (1-grams) varies from 701 for the domain *'References'* to 15,144 for the domain *'Entertainment'*. The set of negative examples is the same for all the subproblems in one domain. The main reason for this decision is efficiency supported by the fact that each subproblem can be seen as a differentiation of its positive examples from the mass of examples. The set of positive examples changes for each of the subproblems. For most of the subproblems the probability of a positive class value is $< 0{:}1$, and this means that we have domains with an unbalanced class distribution.

3. LEARNING DOCUMENT KEYWORDS

The problem of automatic document categorization is well known in information retrieval and is usually tested on publicly available data bases (eg., Reuters, MEDLINE). Machine learning methods have been successfully applied to text data to achieve a better performance than the performance of those information retrieval methods we already know about [6, 15]. Document categorization is usually performed on a set of categories and not on the hierarchical structure of the categories.

Table 1. The domain characteristics for the five domains formed from the five top Yahoo categories. (The problem is keyword assignment that is based on a document hierarchy. We give for each domain (from left to right) its name, the number of included subcategories, the number of features, the number of examples (the actual Web documents the category is based on), the average distance between two nodes in the hierarchy (the length of the shortest path between the two nodes), and the average number of features in positive examples.)

Yahoo category	# of sub-ctgs. (nodes)	# of features (1-grams+. . . +5-grams)	#of examples (actual docs)	Avg. node distance	Avg. pos. features
Entertain-ment	8,081	30,998 (15,144+11,211+ 2,970+1,059+505)	79,011	7.56	60
Arts and Humanities	3,085	11,473 (7,380+3,538+ 463+75+17)	27,765	6.59	65
Computers and Internet	2,652	7,631 (5,049+2,276+ 261+38+7)	23,105	6.77	55
Education	349	3,198 (1,919+1,061+ 184+28+6)	5,406	4.54	100
Reference	129	928 (701+196+ 28+3+0)	1,995	4.29	45

There is some work on hierarchically classifying English documents by Koller and Sahami [7] that includes an evaluation of small, artificially generated hierarchies. Work done by McCallun et al. [9] applies machine learning techniques to some parts of real-world text hierarchies. In their hierarchy all documents are placed at the bottom of the hierarchy, in tree leaves. Many real-world hierarchies include documents in all levels of the hierarchy, meaning that each hierarchy node (and not only the leaf nodes) can include some documents. The Yahoo hierarchy of Web documents contains documents in all its hierarchy nodes. Mladenić [12] proposes the usage of machine learning for automatic document categorization on real-world data from the English Yahoo hierarchy. Work done on feature subset selection for classification, based on

large text hierarchy [14], shows that the large number of features can be drastically reduced by improving classification results.

3.1. Machine learning setting

In order to apply machine learning to text data we represent documents as word-vectors using the bag-of-words representation as commonly used in learning from text data, (eg., [6, 11, 15]). In learning from text data, one of the commonly used approaches to reduce the number of different words used as features that we also apply here is to use a "stop-list" containing common English words, (eg. [3, 6]). Our document representation also 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'). Since we have already removed the words contained in the "stop-list", some features that represent a word sequence can actually capture longer sequences, like for instance, 'Word for Windows' that is represented as a 2-gram 'Word Windows', or 'winners will be posted at the end of each two-week period' that is represented as a 5-gram 'winners posted end two-week period' In this way we can not only capture some characteristic word combinations but we can also increase the number of features (eg. in the whole Yahoo hierarchy from about 70,000 features for 1-grams to about 250,000 features for 5-grams). We can also reduce the high number of features by pruning the low frequency features as suggested in [13] or [16]. The process of feature generation is performed in n passes over documents, where n-grams are generated in the last pass. At the end of each pass over the documents all infrequent features are deleted (here we check for a frequency < 4). 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 the association rules algorithm proposed in [1]. To illustrate this point, in Figure 1 we have shown the accumulated number of features during the process of feature generation on Yahoo documents that compose the Yahoo hierarchy.

The learning algorithm used in our experiments is based on the use of a naive Bayesian classifier on text data as described in [6] and [10]. In their approach, documents are represented as word-vectors where a feature is defined for each word position in the document having a word in that position as a feature value. In a machine learning setting this document representation can be seen as a fixed size vector having a feature for each word from the domain containing a word frequency in a document. The assumption about feature independence used in naïve Bayesian classifier is clearly incorrect. According to [2] this does not necessary mean that this classifier will give a poor performance because of this.

We use the Yahoo hierarchy to learn keyword assignment. We divide the whole problem into subproblems with each corresponding to an individual category. For each of the subproblems, a classifier is constructed that predicts the probability that a document can be characterized by a corresponding set of keywords. A set of positive and negative examples for each subproblem are constructed from the given

hierarchical structure. The final result of learning is a set of specialized classifiers, each based on a small subset of features only.

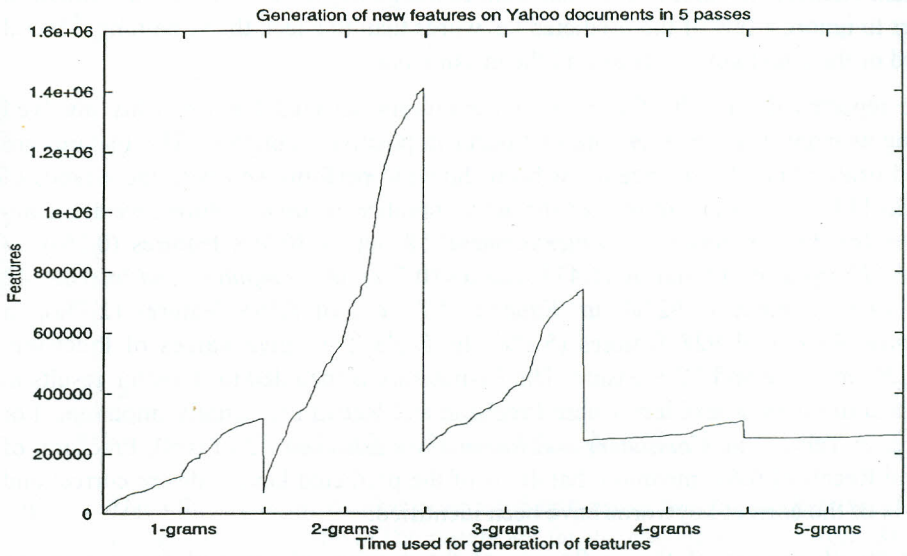

Figure 1. The process of generating new features for Yahoo documents. (At the end of each pass over the documents all the features that occur less than 4 times are deleted.)

4. EXPERIMENTAL RESULTS

Our experiments were performed using our recently developed machine learning system Learning Machine [5] that supports the usage of different machine learning techniques on large data sets with especially designed modules for learning from text and collecting data from the Web. In our experiments we observed the influence of the number of selected features on the system performance. The number of selected features was measured here relative to the category size expressed by the number of features in the positive examples. Recall that we got our categorization results from a set of independent classifiers, with each potentially having a different number of features. A classifier for a larger category uses more features than a classifier for a smaller category, while both classify the same testing example.

Classification results have been reported for the independent set of (500 for the bigger and 300 for the two smaller domains) testing examples selected randomly from the actual Web documents accessible from the corresponding hierarchy domain. The reported results are averaged over 5 repetitions using the hold-out method. To evaluate the results we used Precision, Recall and *F2*-measure as commonly used evaluation measures for text data. We report average Precision and Recall per document calculated for the fixed probability threshold (that is set experimentally to 0.95 [5]). Precision can be seen as the classification's accuracy calculated only for positive

examples, while Recall is the proportion of positive examples the system recognized as positive (the value is in [0..1]). In addition to Precision and Recall, we also report on the *F2*-measure [8] that is commonly used when we care more about Recall than about Precision. Namely, for those users that want to assign keywords to a new document, it is easier to ignore a few of the predicted keywords than to check the remaining several hundred or thousand keywords to find the missing ones.

We reported the results for the best performing selected features (this involved selecting as many features there are that occur in positive examples). The features are selected using Odds Ratio since it has been shown to perform well with these types of problems [13, 14]. To get an idea of the actual number of used features, we are using averages for the categories: in *'Entertainment'* 58 out of 30,998 features (0.2%), in *'Arts and Humanities'* 65 out of 11,473 features (0.7%), in *'Computers and Internet'* 42 out of 7,631 features (0.62%), in *'Education'* 85 out of 3,198 features (2.7%), in *'Reference'* 49 out of 928 features (5.3%). In Table 2 we give values of Precision, Recall, *F1*-measure and *F2*-measure. The *F1*-measure is included to show the results in case we could have a problem where Precision and Recall are equally important. For instance, in Table 2 in *'Computers and Internet'* we achieved *F2* of 0.60, Precision of 0.40 and Recall of 0.84, meaning that 40 % of the predicted keywords are correct and that 84% of the correct keywords have been identified.

For an illustration of the influence of the number of selected features to the performance, we have show in Figure 2 the value of the *F2*-measure for the domain *'Reference'*. It can be seen that the highest values are achieved when one selects as many features that are available and that occur in the positive examples (Vector size is 1). We have also shown that the results obtained when a random score is assigned to the features to illustrate how important it is to use the appropriate feature scoring measure (see [13, 14] for more details on feature scoring measure comparison).

Table 2. The results of a keyword prediction for five domains defined in the Yahoo hierarchy. (The average values are given and standard error of 5 hold-out repetitions.)

Domain name	Average on keyword assignment			
	F1-measure	F2-measure	Precision	Recall
Entertainment	0.48 ± 0.006	0.59 ± 0.007	0.44 ± 0.006	0.80 ± 0.006
Arts and Humanities	0.46 ± 0.003	0.59 ± 0.005	0.40 ± 0.002	0.83 ± 0.006
Computers and Internet	0.46 ± 0.006	0.60 ± 0.006	0.40 ± 0.007	0.84 ± 0.005
Education	0.33 ± 0.008	0.48 ± 0.008	0.36 ± 0.010	0.81 ± 0.005
Reference	0.53 ± 0.006	0.64 ± 0.006	0.51 ± 0.007	0.81 ± 0.008

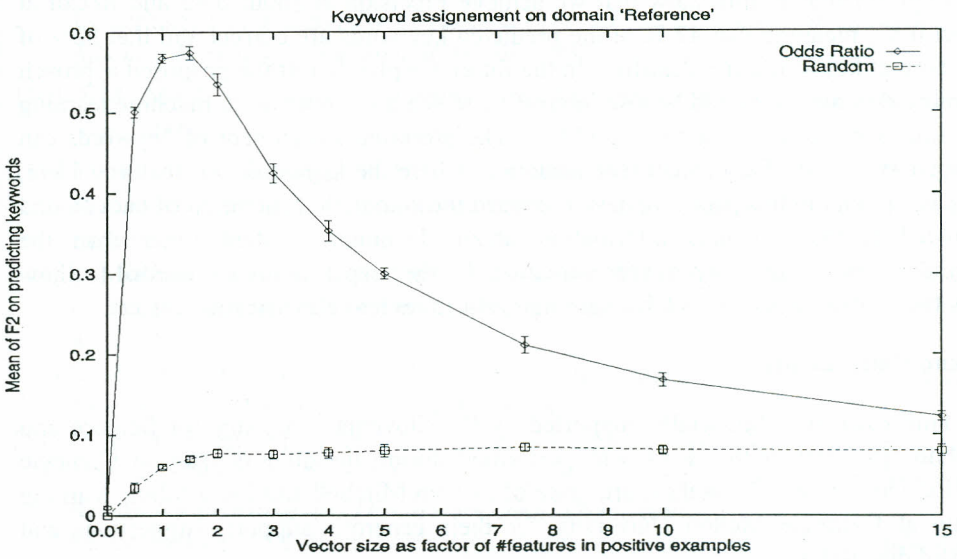

Figure 2. The influence of the number of selected features on the domain *'Reference'* defined in the Yahoo hierarchy. (The results are for the pruning setting = (0.7, 3) and the probability threshold 0.95. We give the mean and standard error of F2-measure.)

5. CONCLUSIONS

Machine learning techniques have been used to automatically assign keywords to a document based on its content. The well known bag-of-words document representation is extended here by using word sequences of up to 5 words. Different domain specifics are considered. One is an unbalanced class distribution, where less than 1%-10% of the examples belong to the target concept. The other has asymmetric misclassification costs, given only implicitly in the problem. These problems are considered during the feature subset selection and by model quality estimates used for the results evaluation. We use a F2-measure that is based on Recall and Precision, to estimate the model quality. This measure makes it possible to take into account the asymmetric misclassification costs by changing the value of a parameter (here set to 2) that regulates the trade-off between Recall and Precision. This measure focuses on the target class value and does not suffer because of unbalanced class distribution. We used Odds ratio as a feature scoring measure appropriate for those domains where the goal is to maximize the performance on one (the target) class value. Additionally, our domains are characterized by a large number of examples and several tens or hundreds of thousands of features requiring efficient methods and careful implementation. The best results are achieved when only a small number of features are used. By this we mean using 50-100 best features or in other words using only 5%-15% of all the features.

Experimental results show that we achieve Precision at about 0.42 and Recall at about 0.82, meaning that 42 % of the predicted keywords are correct and that 82% of the correct keywords are identified. In the future we plan to test the proposed approach on more domains. It would be also interesting to see how some other machine learning algorithms perform on the same problem. The proposed assignment of keywords can be seen as a kind of document representation, where the keywords are features. Here, we can say that in this paper we have proposed the automatic generation of background knowledge, that contains information about document content other than the commonly used bag-of-words representation. Further experiments are needed to show how the incorporation of such background influences text classification results.

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. The authors are grateful to Tom Mitchell and his machine learning group at Carnegie Mellon University for their generous support, suggestions and fruitful discussions.

REFERENCES

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A.I. Verkamo. Fast discovery of association rules. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, eds. *Advances in Knowledge Discovery and Data Mining.* AAAI Press/The MIT Press, 1996, pp. 307-328.

[2] P. Domingos and M Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning,* 29, 1997, pp. 103-130.

[3] W.W. Cohen. Learning to classify English text with ILP methods. *Workshop on Inductive Logic Programming*, 1995.

[4] D. Filo and J. Yang. Yahoo! Inc. http://www.yahoo.com/docs/pr/, 1997.

[5] M. Grobelnik and D. Mladenić. Learning Machine: design and implementation. *Technical Report IJS-DP-7824.* Department for Intelligent Systems, J.Stefan Institute, Ljubljana, 1998.

[6] T. Joachims. A Probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *Proc. of the 14th International Conference on Machine Learning ICML97*, 1997, pp. 143-151.

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

[8] D.D. Lewis. Evaluating and optimizating autonomous text classification systems. *Proceedings of the 18th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval,* 1995, pp.246-254.

[9] A. McCallum, R. Rosenfeld T. Mitchell and A. Ng. Improving text classification by shrinkage in a hierarchy of classes. *Proc. of the Eleventh 14th International Machine Learning Conference ICML-98,* 1998.

[10] T.M. Mitchell. *Machine Learning*. The McGraw-Hill Companies, Inc., New York, 1997.

[11] D. Mladenić. Personal Web watcher: implementation and design. *Technical Report IJS-DP-7472.* Department for Intelligent Systems, J.Stefan Institute, Ljubljana, 1996.

[12] D. Mladenić. Turning Yahoo into an automatic Web-page classifier. *Proceedings of the 13th European Conference on Artificial Intelligence ECAI'98,* 1998, pp. 473-474.

[13] D. Mladenić. Feature subset selection in text-learning. *Proc. of the 10th European Conference on Machine Learning ECML98,* 1998.

[14] D. Mladenić and M. Grobelnik. 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*, 1998.

[15] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting Web sites. *Machine Learning,* 27, 1997, pp. 313-331.

[16] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. *Proc. of the 14th International Conference on Machine Learning ICML97*, 1997, pp. 412-420.

Received: 19 October 1999
Accepted: 17 December 1999

Dunja Mladenić
Marko Grobelnik

DODJELJIVANJE KLJUČNIH RIJEČI DOKUMENTIMA TEHNIKOM STROJNOG UČENJA

Sažetak

U radu se opisuje primjena tehnike strojnog učenja u dodjeljivanju ključnih riječi dokumentima. Kao problemska domena korištenja je Yahoo hijerarhija, najveća hijerarhija dokumenata raspoloživa na Web-u. Upotrebljena tehnika strojnog učenja razvijena za učenje na tekstu temeljena je na hijerarhijskoj klasifikacijskoj strukturi. Velik broj obilježja reduciran je korištenjem svojstava hijerarhijske strukture, identificiranjem podskupova za ispitivanje pomoću metoda poznatih iz dohvaćanja informacija. Dokumenti su reprezentirani vektorima riječi, koji sadrže nizove riječi umjesto jednostavnih riječi. Prilikom formiranja primjera za učenje, uzeta je u obzir hijerarhijska struktura primjera i vrijednosti klasa, a kod razmatranja samo pojedinih obećavajućih puteva u hijerarhiji u klasifikaciji je korištena dodatna hijerarhijska struktura vrijednosti klasa.

Ključne riječi: strojno učenje, pridruživanje kjučnih riječi, Yahoo hijerarhija, kategorizacija dokumenata, mjera F1, mjera F2.