Predicting content from hyperlinks

Dunja Mladenić
Department of Intelligent Systems
J.Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Dunja.Mladenic@ijs.si

Marko Grobelnik
Department of Intelligent Systems
J.Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Marko.Grobelnik@ijs.si

Abstract

This paper describes an approach to prediction of a document content based on the hyperlink that points to the document. The k-Nearest Neighbor algorithm is used to predict a set of words that appear in the document. Experiments are performed on real-world data obtained from the Web. The proposed approach gives promising results. On the tested data in average about 33% of document words are correctly predicted while among all the predicted words about 15% appeared in the document. The predicted words are chosen from about 4,000 to 8,000 different words and word pairs that appear in the training examples.

1 Introduction

The World Wide Web is becoming not only an important support for exchanging information and communication between people but also a challenge for researchers and developers in different areas of computer science. Documents on the Web are organized in hyper-graphs, where hyperlinks are used to connect documents. Each hyperlink points to one document, while the same document can be pointed to by many different hyperlinks. Each hyperlink consists of underlined text and a pointer (address of a document). Underlined text is supposed to be indicative of the content of document it points to. Reading hypertext also means deciding whether to follow some of the present hyperlinks or not. This decision is usually easier to make if more information about the document the hyperlink points to is available. Since there are many pairs of hyperlinks and the associated documents available on the Web, we can think about using these examples to model a relation that maps a hyperlink to a document. We are currently not aware of any research that would address such problems, where a document content is to be predicted.

This paper describes an approach where machine learning is used for learning the relation between hyperlink and the document it points to. Section 2 describes the addressed problem. Learning algorithm is described in Section 3 and experimental results are given in Section 4.

2 Problem description

One of the motivations and potential application of the proposed approach is in helping user browsing the Web. The possible way to provide help is simply by highlighting interesting hyperlinks on the requested Web documents (more details can be found in (Joachims and Mladenić 1998, Mladenić 1998b). We assume that the interesting hyperlinks are the hyperlinks that are highly probable to be clicked by the user. The problem is defined as predicting clicked hyperlinks from the set of Web documents visited by the user. All hyperlinks that are present on the visited documents are used for constructing machine learning examples. We build descriptions of hyperlinks (HL) from underlined text pointing to a document and some surrounding text, and treat these as shortened documents.

When making a prediction on a new hyperlink, we can either use only the information presented to the user (a part of the text from the document that contains a hyperlink) or try to take advantage of a hyperlink being connected to other document, namely the document it points to. Instead of classifying a given hyperlink, we could first predict the content of the document that is pointed to by the hyperlink and then use such an abstract document to predict if the user is going to click on the hyperlink. We represent the documents the hyperlinks point to as word-vectors using the bag-of-words representation as commonly used in learning on text data (Joachims 1997, Mladenić 1998a, Pazzani and Billsus 1997). This enables using machine learning techniques to predict the frequency of words in the vector. We simplify this problem by predicting word occurrence instead of frequency and reducing the number of words we want to predict. The number of words is reduced by removing words from English "stop-list" and removing infrequent words as commonly performed when learning on text data (Cohen 1995, Joachims 1997). The problem of predicting word occurrence is illustrated in Figure 2 giving a hyperlink and the document is points to, as well as the set of words predicted to occur in the document based on the content of the given hyperlink.

One of the main characteristics of this machine learning problem is having a vector of class attributes instead of only one class attribute, where each of the vector elements corresponds to the occurrence of a particular word in the document. Since a document gives a context for words occurring in it, the occurrence of one word is possibly dependent on other words in the document. Thus, we are dealing with the problem of mutually dependent class attributes. Collecting data for this problem relatively easy, since we are learning from unlabeled data available on the Web.

3 Learning algorithm

We propose a simple modification of the k-Nearest Neighbor algorithm (Aha, Kibler and Albert 1991) to be used for learning on mutually dependent class attributes. We use it in our experiments on predicting word occurrences based on the hyperlink that points to the document.

In order to apply k-Nearest Neighbor algorithm on mutually dependent class attributes we modified the algorithm to handle examples that consist of $HyperLink$-Doc pairs, where $HyperLink$ includes the actual hyperlink from HTML-document and some of its context and Doc is a text document pointed to by the hyperlink. When applied on the Yahoo hierarchy (Yahoo 1997), each hyperlink description $HyperLink$ we build to serve as a part of machine learning example includes all the text in the hyperlink item included in the Yahoo document and the words from the document category name (written on the top of the Yahoo document that includes the hyperlink). The idea is that the algorithm finds a group of neighboring documents that are 'close' when the corresponding $HyperLinks$ are similar. $HyperLink$ and Doc are represented by two word-vectors. It is possible that multiple hyperlinks point to the same document. In this case, multiple copies of the document are included and no connection between them is made. The document is considered as neighboring if any of the hyperlinks pointing to it is one of the nearest neighbors of the hyperlink under classification. The similarity of two $HyperLinks$ is measured by cosine similarity between their word-vectors.

$$cos\theta(\vec{X},\vec{Y}) = \frac{\sum_i X_i Y_i}{\sqrt{\sum_j X_j^2 \sum_l Y_l^2}}$$

Notice that our 'neighbor' relation is not transitive. Namely, if two hyperlinks H_1 and H_2 are neighboring, there can be the third hyperlink H_3 that is neighboring to only one of them, eg., H_1. This is due to the fact that the reason that two hyperlinks are neighboring are their common words. The common words of H_1 and H_2 can be completely different from the common words of H_1 and H_3, and at the same time there can be an empty set of the common words of H_2 and H_3.

The detected neighboring documents are then represented by an abstract document whose word occurrence is computed as an average of word occurrences of the neighboring documents. For each class (word occurrence) a threshold is set for mapping to a binary-valued class such that length of the abstract (predicted) document is as close as possible to the average length of neighboring documents from which it was generated. Figure 1 illustrates the construction of the abstract document. Mutual dependence of class attributes influences the value of Threshold set such that $Length(Abs.Doc) \doteq \frac{1}{k}\sum_i^k Length(Ex_i.Doc)$, where $Abs.Doc$ is the vector representing abstract document.

4 Experiments

4.1 Domain description

The procedure for getting domain data from the Yahoo domains given in Table 1 is the following. From all the examples included in the text hierarchy, 500 were randomly selected and used in 5-fold cross validation. Each example consists of two parts: hyperlink description and document the hyperlink points to. The hyperlink part includes an item from the Yahoo document and the corresponding category name this Yahoo document represents. The second part of example is the actual document pointed to by the hy-

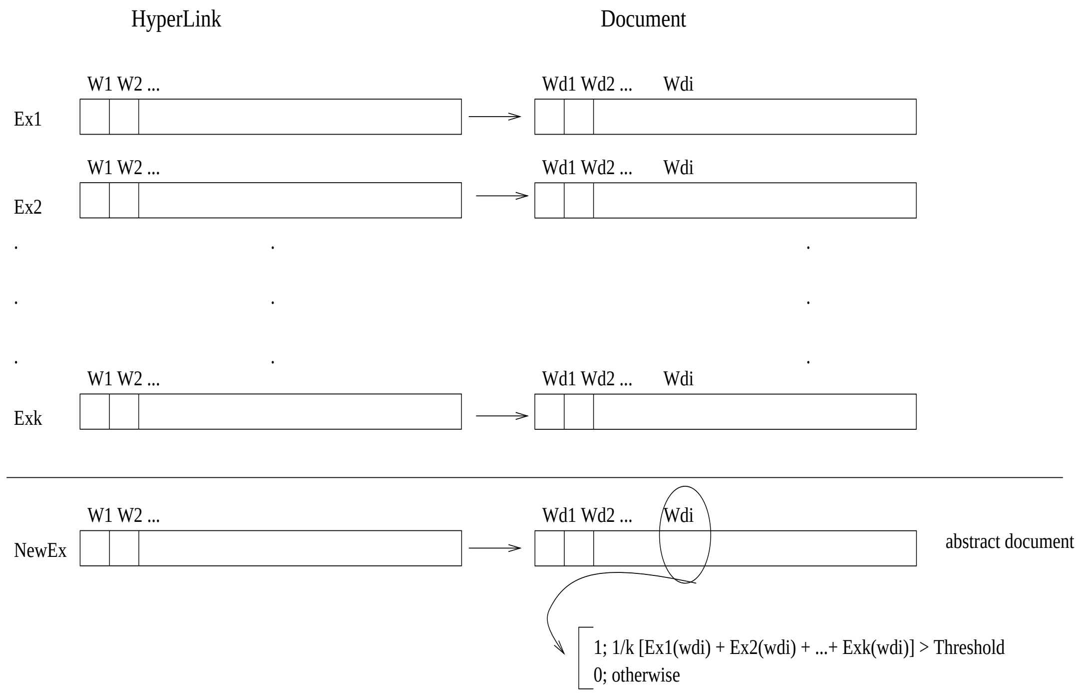

Figure 1: Illustration of document content prediction using the abstract document construction based on the hyperlink that points to the document. The approach is based on the k-Nearest Neighbor algorithm. The k document pointed to by the k most similar hyperlinks are combined to get an abstract document.

perlink. It is represented as a vector of class attributes and this part is to be predicted. This two parts are connected by relation $HyperLink \rightarrow Doc$. The hyperlink parts in *'Reference*, *'Education*and *'Computers'* have in average 9 ± 0.15, 12.2 ± 0.09 and 12.4 ± 0.11 word sequences respectively. The document parts in *'Reference*, *'Education*and *'Computers'* have in average 51.6 ± 0.96, 53.8 ± 1.16 and 75.1 ± 1.96 sequences respectively. This means that for instance, on domain *'Computers'* we are on average predicting the occurrence of 75.1 word sequences based on the occurrence of 12.5 word sequences.

The hyperlink parts and the document parts are represented by word sequences of length up to two (1-grams and 2-grams) as described in (Mladenić and Grobelnik 1998) and given in Table 1. Table 1 gives for each domain the average number of word sequences that occur in the hyperlink parts and the average number of word sequences that occur in the document parts. Notice that for all three domains that originally include different number of examples ranging from 2,000 to 23,000 (see Table 1), 500 randomly selected examples are used in the described experiments. This is also the reason why the number of class attributes in the used data set (# features in Table 1) is rather similar for all the three domains.

4.2 Experimental results

Our experiments are performed using our recently developed machine learning system **Learning Machine** (Grobelnik and Mladenić 1998) 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 measure the performance of the induced model as the average Precision and Recall per document on the predicted document words. Precision can be seen here as the accuracy calculated on the predicted words, while Recall is the proportion of document words that are among the predicted words (both having values in interval [0..1]). We also report the average value of F_2-measure as commonly used evaluation measures for text data (Lewis 1995). $F - measure$ is a combination of Precision and Recall $F_\beta = \frac{(1+\beta^2) Precision \times Recall}{\beta^2 Precision + Recall}$. In cases when we care more about Recall than about Precision, the value of parameter β is set to 2, as set here. It can be seen from the experiments that even though the achieved Preci-

Yahoo category	# of sub-ctgs. (nodes)	# of all examples (actual docs)	Avg. node distance	# of features (1-grams+2-grams)	Avg. # of features in HL	in DOC
Comp.	2,652	23,105	6.77	8327 (5994+2333)	12.4 ± 0.11	75.1 ± 1.96
Edu.	349	5,406	4.54	4447 (3229+1208)	12.2 ± 0.09	53.8 ± 1.16
Ref.	129	1,995	4.29	5488 (4001+1487)	9 ± 0.15	51.6 ± 0.96

Table 1: Domain characteristics for three domains formed from the three top Yahoo categories. We give for each domain (from left to right) its name, the number of included subcategories, the number of all examples among which 500 were randomly selected (actual Web documents the category is based on), the average distance between two nodes in the hierarchy (length of the shortest path between the two nodes), the number of features generated from training examples, the average number of features that appear in the hyperlink parts (attributes) and the average number of features that appear in the document parts (class-attributes).

sion is lower than the achieved Recall, many of the predicted words that are not present in the document (and thus decrease Precision) are not very different from the correct words. We observe the influence of the number of neighbors varied from 1 to 30 on independent set of randomly selected testing examples.

The reported results are averaged over 5 runs using 5-fold cross validation method. Results of experiments on three domains generated from Yahoo hierarchy are given in Table 2. We report the results obtained by using the k Nearest Neighbor algorithm on three real-world domains. Result of combining k randomly selected training examples is given as a baseline result. It can be seen that in average by combining the k Nearest Neighbors about 33% of document words are correctly recognized (Recall 0.33) while among all the predicted words only about 15% are present in the document that we wanted to predict (Precision 0.15). When combining the k random examples the average recall is about 0.28 and the average precision about 0.11. The best predictions are made when 30 nearest neighbors are considered.

5 Discussion

Figure 2 shows an example pair of hyperlink-document and the words predicted by our system. The used hyperlink is given in the upper left corner in Figure 2. The predicted words (word sequences) are shown as two sets of words. The first set contains the predicted words that also occur in the actual document (true positive). In Figure 2 these words are shown on the actual document in boldface (**Information Systems, Asia, Web, research**). The second set contains other predicted words sorted according to their predicted probability (false positive). In Figure 2 these words are given in the right hand-side. It can be seen that among the other predicted words that don't occur in the document, the highest probability is assigned to the words that occur in the category name and describe the document content (eg., information retrieval, library information). Recall that each hyperlink description we build includes all the text in the hyperlink item and the words from the document category name.

6 Conclusions

From the example in Figure 2 and the results of experiments in Table 2 it can be seen that there are quite some predicted words that does not occur in the actual document. The number of predicted words is influenced by the threshold used for mapping the predicted probabilities into classifications. Currently this threshold is set so to get the number of predicted words approximately equal to the average number of words in the neighboring documents (document the nearest neighbor hyperlinks point to). Further experiments are needed to check if changing the threshold to predict smaller number of words will improve the system results.

References

Aha, D.W., Kibler, D. & Albert, M.K. (1991). Instance-Based Learning algorithms. *Machine Learning, 6*, 37-66. Kluwer Academic Publishers.

Domain name	# of nbs.	Random neighbors precision	recall	F_2	Nearest neighbors precision	recall	F_2
Comp.	1	0.098 ± 0.003	0.087 ± 0.005	0.072 ± 0.003	0.202 ± 0.003	0.209 ± 0.003	0.188 ± 0.004
	5	0.089 ± 0.003	0.174 ± 0.005	0.11 ± 0.001	0.125 ± 0.004	0.314 ± 0.005	0.178 ± 0.003
	10	0.096 ± 0.003	0.223 ± 0.007	0.126 ± 0.002	0.124 ± 0.004	0.327 ± 0.006	0.182 ± 0.002
	15	0.096 ± 0.004	0.247 ± 0.005	0.136 ± 0.002	0.121 ± 0.005	0.328 ± 0.005	0.182 ± 0.002
	20	0.096 ± 0.004	0.275 ± 0.007	0.147 ± 0.002	0.112 ± 0.004	0.336 ± 0.004	0.184 ± 0.002
	25	0.110 ± 0.003	0.275 ± 0.003	0.152 ± 0.002	0.120 ± 0.004	0.349 ± 0.004	0.189 ± 0.002
	30	0.101 ± 0.003	0.296 ± 0.004	0.157 ± 0.001	**0.122** ± 0.004	**0.357** ± 0.004	**0.193** ± 0.002
Edu.	1	0.099 ± 0.004	0.091 ± 0.009	0.092 ± 0.004	0.161 ± 0.004	0.150 ± 0.006	0.138 ± 0.005
	5	0.089 ± 0.003	0.216 ± 0.005	0.128 ± 0.003	0.139 ± 0.004	0.246 ± 0.003	0.171 ± 0.003
	10	0.106 ± 0.002	0.220 ± 0.004	0.146 ± 0.002	0.146 ± 0.004	0.280 ± 0.0018	0.189 ± 0.002
	15	0.106 ± 0.002	0.239 ± 0.004	0.152 ± 0.002	0.146 ± 0.004	0.287 ± 0.002	0.192 ± 0.003
	20	0.113 ± 0.003	0.259 ± 0.002	0.164 ± 0.002	0.147 ± 0.003	0.230 ± 0.002	0.202 ± 0.003
	25	0.118 ± 0.002	0.254 ± 0.003	0.171 ± 0.002	0.152 ± 0.003	0.309 ± 0.003	0.206 ± 0.003
	30	0.118 ± 0.002	0.282 ± 0.004	0.177 ± 0.002	**0.152** ± 0.003	**0.315** ± 0.003	**0.211** ± 0.003
Ref.	1	0.115 ± 0.007	0.132 ± 007	0.115 ± 0.003	0.268 ± 0.006	0.202 ± 0.004	0.196 ± 0.005
	5	0.112 ± 0.003	0.194 ± 0.004	0.137 ± 0.002	0.171 ± 0.005	0.243 ± 0.003	0.185 ± 0.002
	10	0.123 ± 0.004	0.232 ± 0.002	0.161 ± 0.001	0.198 ± 0.005	0.265 ± 0.002	0.206 ± 0.002
	15	0.128 ± 0.003	0.246 ± 0.003	0.173 ± 0.003	0.210 ± 0.004	0.271 ± 0.001	0.214 ± 0.002
	20	0.132 ± 0.002	0.268 ± 0.003	0.184 ± 0.003	0.210 ± 0.004	0.279 ± 0.002	0.221 ± 0.003
	25	0.131 ± 0.002	0.281 ± 0.003	0.189 ± 0.003	0.212 ± 0.004	0.287 ± 0.002	0.227 ± 0.003
	30	0.138 ± 0.003	0.288 ± 0.003	0.194 ± 0.002	**0.217** ± 0.003	**0.295** ± 0.002	**0.235** ± 0.003

Table 2: Domain name, the number of neighbors used in predictions and the results of two experiments: one by using k random examples and the other by using k nearest neighbors. For each prediction we give the average Precision, Recall and F_2-measure and the standard error. The results are obtained using 5-fold cross validation on 500 examples for each of the three domains. The addressed problem is predicting occurrence of words in the document based on the hyperlink that points to the document.

Cohen, W.W. (1995). Learning to Classify English Text with ILP Methods. *Proceedings of the Workshop on Inductive Logic Programming.* Leuven.

Filo, D., & Yang, J. (1997). Yahoo! Inc. http://www.yahoo.com/docs/pr/

Grobelnik, M., & Mladenić, D. (1998). Learning Machine: design and implementation. *Technical Report IJS-DP-7824.*

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

Joachims, T., Mladenic, D., (1998). Browsing-Assistenten, Tour Guides und Adaptive WWW-Server *KI, Kunstliche Intelligenz, Special Issue Intelligent Information Agents*, vol. 3, pp. 23-29, September 1998.

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

Mladenić, D., & Grobelnik, M. (1998). Word sequences as features in text-learning. *Proceedings of the Seventh Electrotechnical and Computer Sc. Conference ERK'98.* Ljubljana, Slovenia: IEEE section.

Mladenić, D. (1998a). Turning Yahoo into an Automatic Web-Page Classifier. *Proceedings of the 13th European Conference on Aritficial Intelligence ECAI'98* (pp. 473-474).

Mladenić, D. (1998b). Feature subset selection in text-learning. *Proceedings of the 10th European Conference on Machine Learning ECML98* (pp.95-100).

Pazzani, M., & Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning, 27*, 313-331. Kluwer Academic Publishers.

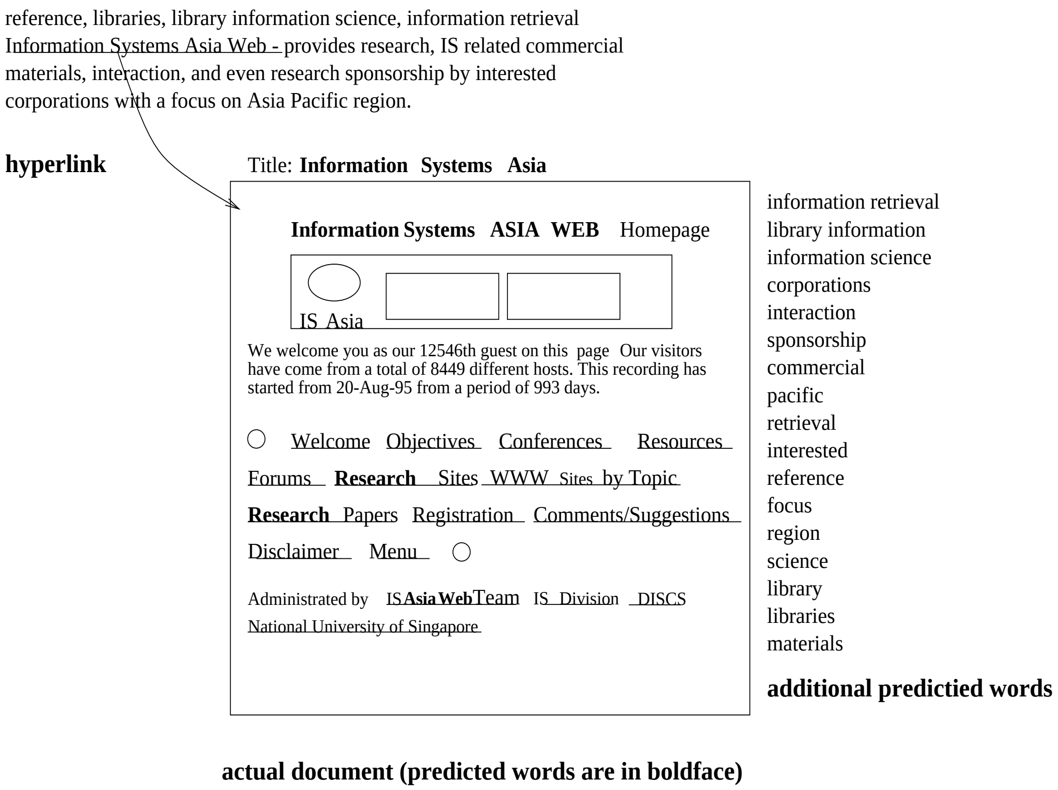

Figure 2: Example hyperlink-document pair and the set of word sequences predicted to occur in the document. The prediction is made from the hyperlink that points to the document. The predicted words are shown in two parts: correctly predicted words shown as boldface words on the document and the other predicted words shown in the list of words on the left side of the document.