Predlog teme doktorske disertacije:

**Machine Learning on non-homogeneous, distributed text data**
**(Strojno učenje na nehomogenih, distribuiranih tekstovnih podatkih)**

Dunja Mladenić

## 1 Motivation

With the growing availability of information sources, especially large, non-homogeneous, distributed sources like the World Wide Web, there is also a growing interest in tools that can help in making a good and quick selection of information we are interested in. Recent work that arises at the intersection of Information Retrieval and Machine Learning offers some novel solutions to this problem, as well as the work in Intelligent Agents and Intelligent User Interfaces. For example, Armstrong et al. [3] developed WebWatcher, a system that assists user in locating information on the World Wide Web taking keywords from the user, suggesting hyperlinks and receiving evaluation. Balabanović et al. [4] developed "a system which learns to browse the Internet on behalf of a user". It searches the World Wide Web taking bounded amount of time, selects the best pages and receives an evaluation from the user. The evaluation is used to update the search and selection heuristics. Pazzani et al. [26] collect ratings of the explored Web pages from the user and learn a user profile from them. Pages are separated according to their topic and a separate profile is learned for each topic. Mitchell et al. [25] proposed a system connected to the user's electronic calendar, that generates sets of rules capturing the user's scheduling preferences and some other information about individual attendees of meetings. It uses these rules to provide advice to the user for new, unscheduled meetings. Lang [21] developed a system for electronic news filtering that uses text-learning to generate models of user interests. Krulwich and Burkey [20] proposed "The ContactFinder agent" that reads and responds to bulletin board messages, assists the users by referring them to other people who can help them, and categorizes messages and extracts their topic areas. Maes [23] described "interface agents" that learn from the user as well as from other agents. As examples of such agents they developed agents for: electronic mail handling, meeting scheduling, electronic news filtering and entertainment recommendation. Some of these agents use the context of documents and adopt Information Retrieval approaches (eg. news filtering). Others rely on the correlation between different users — performing "social learning" (eg., entertainment recommendation). Holte and Drummond [16], Drummond et al. [11] designed a system that assists browsing of software libraries, taking keywords from the user and using a rule-based system with forward chaining inference, assuming that the library consists of one type of items and the user goal is a single item. Etizioni and Weld [13] offer an integrated interface to the Internet combining a UNIX shell and the World Wide Web to interact with Internet resources. Their agent accepts high-level user goals and dynamically synthesizes the appropriate sequence of Internet commands to satisfy those goals. Hammond et al. [15] and Burke et al. [9] developed a system that uses a "natural language question-based interface to access distributed text information sources" and helps the user to find answers to her/his question in a databases such as FAQ files.

One of the available information sources is the World Wide Web and it is currently growing quickly, attracting many users with different interests. Since the interaction with the World Wide Web (WWW) is by means of computer, one can use computers to "observe" and record user actions, to communicate with users and use all the collected information to help the users find their way in the WWW. From the machine learning point of view, the problem is in handling large, non-homogeneous, distributed sources of text data.

The proposed work is mainly inspired by WebWatcher: "a Learning Apprentice for the World Wide Web" [3], [17] and some other work related to learning apprentice and learning from text [18], [19], [21], [25]. The idea of a learning apprentice is to automatically customize to individual users, using each user interaction as a training example.

## 2 Thesis area and related approaches

The proposed thesis is in the area of machine learning where learning is over text documents that are mostly written in the HTML-format. Even though the way of applying machine learning techniques to that problem is to be precisely defined as a part of the research work in the thesis, it is already clear that it will be characterized by a high number of features (several tens of thousands). Additionally, documents are connected to each other in directed graphs via HTML-*hyperlinks*. The source of data is non-homogeneous, distributed and dynamically changing, there is potentially an infinite number of training examples and user interests can be changing over time. The problem is not only space intensive but also time demanding, since we have to do the classification while the user is waiting to get the requested document.

Most of the work in learning on Web documents defines a machine learning problem similar to information filtering, asking users to rank given documents and generating model of user interests based on users' ranking (Armstrong et al. [3], Balabanovic et al. [4], Pazzani et al. [26]). The proposed thesis aims at a user customized system that is present whenever the user is browsing the Web but does not demand any additional efforts from the user (no ranking or search keywords required).

There is quite some work in the area of learning over text documents that is not necessarily related to the Web. We will take a look at some of this work through the prism of three questions we also have to deal with in the thesis: (1) what representation is used for documents, (2) how is the high number of features dealt with and (3) which learning algorithm is used.

(1) Many current systems that learn from text use the bag-of-words representation using either Boolean features indicating if a specific word occurred in a document (eg., [3], [10], [22], [26], [32]) or frequency of a word in a given document (eg. [2], [4] [5], [6], [18], [32]). There is also some work that uses additional information such as the word position [10] or word tuples called n-grams [31].

(2) One of the frequently used approaches to reduce the number of different words is to use a "stop-list" containing common English words (eg. a, the, with) or pruning the infrequent and/or very frequent words (eg., [10], [18]). Connected to the particular language is also word stemming that reduces the number of different words using a language-specific stemming algorithm. Many approaches use a language independent approach and introduce some sort of word weighting in order to select only the best words (eg. [2], [3], [4], [18], [22], [26]) or reduce the dimensionality using latent semantic indexing with singular value decomposition (eg. [5], [6], [14]).

(3) One of the well-established techniques for learning over text in Information Retrieval is to represent each document as a $TFIDF$-vector in the space of words that appeared in the training documents [30], sum all the interesting document vectors and use the resulting vector as a model for classification (based on [29] relevance feedback method). This technique has already been used in Machine Learning experiments on the World Wide Web data (eg. [3], [4], [6], [18], [26]).

Armstrong et al. [3] used a statistical approach they called WordStat that assumes the mutual independence of words. Pazzani et al. [26] used in their experiments a Naive Bayesian classifier on Boolean vectors, Nearest Neighbor and symbolic learning using Decision Trees. A variant of k-Nearest Neighbor was also used by Yang [32]. Joachims [18] introduced Probabilistic $TFIDF$ that takes into the account document representation. Apté et al. [2] used Decision Rules. Cohen [10] used Decision Rules and the Inductive Logic Programming systems FOIL and FLIPPER. Lewis and Gale [22] used a combination of a Naive Bayesian classifier and logistic regression. Maes [23] used Memory-Based reasoning, McElligot and Sorensen [12], [31] used a connectionist approach combined with Genetic Algorithms.

## 3 Proposed methodology

The proposed thesis focuses on the modification of machine learning techniques to handle large, non-homogeneous, distributed sources of text data with sparse coverage of example space. As example study we use documents that are present on the Web and define the target problem as predicting the hyperlinks that the user is going to click on (the clicked hyperlinks).

### 3.1 On-line learning for highlighting hyperlinks

Imagine a user sitting behind a Web browser and requesting documents, most of the time by clicking on the hyperlinks on already requested documents that are presented to the user. Predicting and highlighting the clicked hyperlinks is one of the ways to help users finding their way on the Web. In the machine learning setting proposed in this thesis, all the hyperlinks that were shown to the user are considered as training examples with a Boolean class (clicked or un-clicked). Training examples are collected on-line from each interaction with the user. Data collected for a single user represents a domain to be handled by machine learning techniques. The thesis will use the recently developed machine learning approach to learning on large datasets and on-line learning [8] where subsets of incoming examples are taken and used to generate a new model. All generated models are stored and used for classification (eg., combined by voting).

It is not clear what information to include in each example. The proposed thesis will compare the performance of the two solutions: (1) include information presented to the user, that is a part of the text from the document that contains a hyperlink and (2) include information that was not presented to the user, that is the content of documents pointed to by hyperlinks. (1) Generating examples from the information presented to the user on the requested document means using machine learning to model the function $User_{HL} : HyperLink \rightarrow \{pos, neg\}$, where $HyperLink$ ($HL$) is defined to include the acctual hyperlink from HTML-document and some of its context. (2) Retrieving documents pointed to by hyperlinks on a requested document means that the model will be induced from documents $User_{DOC} : Document \rightarrow \{pos, neg\}$. The reason we can afford thinking about the second solution is that the classification model will be induced "off-line", meaning that the induction is not so time critical as classification. The time critical part is classification, since usually the user is sitting behind the Web browser waiting for the requested document that we are using in the classification process. In the both cases, classification will be performed on $HyperLinks$ that in the second solution can be viewed as a kind of small HTML-documents. In (1) we will be using the model induced from $HyperLinks$ and in (2) the model induced from the documents pointed to by hyperlinks.

In the previously mentioned arcing approach to on-line learning that will be used in the thesis, subsets of incoming examples are taken and used to generate a new model. In this way we can handle large data sets with the existing machine learning algorithms. For the problem of learning over text documents the appropriate size of the subsets of training examples is unknown, since it is not clear how many examples are needed for a reasonable behaviour of induced model. The thesis will experimentally estimate the sufficient number of examples and show the influence of the changing number of examples to the performance of the induced model.

A single model will be induced using one of the known machine learning algorithms such as k-Nearest Neighbours ([1], [28], [24]), Naive Bayesian Classifier [24], Decision Tree Induction [27].

## 3.2 Predicting a document pointed to by a hyperlink

When making a prediction on a new hyperlink, we can use only the information presented to the user (that is as proposed earlier a part of the text from the document that contains a hyperlink) or decide to take the advantage of a hyperlink being connected to other document. 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. Representing documents as word-vectors (bag-of-words) enables using machine learning techniques to predict the frequency of each word in the vector. We will simplify this problem by predicting word occurrence instead of frequency and reducing the number of words we want to predict.

One of the main characteristics of that machine learning problem is its class being a vector of classes, 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 classes. We will modify the known k-Nearest Neighbours algorithm to handle examples that consist of *HyperLink-Document* pairs, where *HyperLink* includes the acctual hyperlink from HTML-document and some of its context and *Document* is a text document pointed to by the hyperlink. The system will find a group of neighbouring documents that are found as 'close' if the corresponding *HyperLinks* are similar. *HyperLink* and *Document* will be represented each with a word-vector and the similarity of two *HyperLinks* will be defined as cosine similarity measure between their word-vectors.

The detected neighbouring documents will be represented by an abstract document whose word occurrence will be computed as an average of word occurrences of the documents from which the abstract document was generated. For each class (word occurrence) a threshold will be set for mapping to a Boolean class value such that length(doc) = average length(similar documents). The possibility is also to use (instead of modified k-Nearest Neighbours algorithm) a modified regression tree algorithm [7], where instead of minimizing mean-square-error in each node (that is used in a single-class regression trees), the tree would maximize a cosine similarity to an average document vector. We will measure the performance of the induced model as the average cosine similarity between the actual document and the predicted abstract document instead of the classification accuracy used in single-class problems. The induced relation between a hyperlink and the document pointed to by the hyperlink will serve as a background knowledge in learning to predict the clicked hyperlink.

### 3.3 Representation of HTML-documents/hyperlinks

In order to apply machine learning methods on the problem of learning to predict the clicked hyperlink, we need to consider the representation of HTML-hyperlinks and HTML-documents. It is hard to capture the information offered to the user by showing her/him some Web-page. One of the very simple, frequently used but unfortunately rather limited approaches is the bag-of-words text document representation. In this representation, each text document is represented by a vector of numbers, each number corresponds to the occurrence of a particular word in the document. We will enrich such text document representation with new features generated from domain dependent background knowledge; the existing background knowledge such as HTML-tagging information, the relative position of words in the document, English thesaurus and dynamically constructed background knowledge based on the existing Web hierarchies (eg. Yahoo).

## 4 Expected contributions

- Machine learning method for learning over short HTML-documents based on:
  - design of efficient machine learning techniques for large, non-homogeneous, distributed sources of unstructured data (text) with sparse coverage of example space,
  - dynamic construction of background knowledge from the existing Web hierarchies
  - the usage of domain dependent background knowledge to enrich HTML-document representation,
  - feature selection applied on text documents with large number of features (several tens of thousands),
  - study of the influence of the number of training examples used to induce a model to the model performance.
- Machine learning method for learning mutually dependent classes represented with a vector of numbers, that will be used on short HTML-documents.
- Evaluation of the proposed methods on the real data collected from Web users.

## References


[1] Aha, D.W., Kibler, D., Albert, M.K., Instance-Based Learning algorithms, Machine Learning, 6, pp.37-66, Kluwer Academic Publishers, 1991.

[2] Apté, C., Damerau, F., Weiss, S.M., Toward Language Independent Automated Learning of Text Categorization Models, *Proc. of the 7th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dubline, 1994.

[3] Armstrong, R., Freitag, D., Joachims, T., Mitchell, T., WebWatcher: A Learning Apprentice for the World Wide Web, *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, March 1995. URL: http://www.cs.cmu.edu/afs/cs/project/theo-6/web-agent/www/webagents-plus.ps.Z

[4] Balabanović, M., Shoham, Y., Learning Information Retrieval Agents: Experiments with Automated Web Browsing, *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, March 1995. URL: http://robotics.stanford.edu/people/marko/papers/lira.ps

[5] Bartell, B.T., Cottrell, G.W., Belew, R.K., Latent Semantic Indexing is an Optimal Special Case of Multidimensional Scaling, *Proceedings of the ACM SIG Information Retrieval*, Copenhagen, 1992.

[6] Berry, M.W., Dumais, S.T., OBrein, G.W., Using linear algebra for intelligent information retrieval. *SIAM Review*, Vol. 37, No. 4., pp. 573–595, December 1995.

[7] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, P.J., Classification and regression trees, Belmont, CA: Wadsworth International Group, 1984.

[8] Breiman, L., Pasting bites together for prediction in large data sets and on-line, Berkeley University, November 1996.

[9] Burke, R., Hammond, K., Kozlovsky, J., Knowledge-based Information Retrieval for Semi-Structured Text, *Working Notes from AAAI Fall*

*Symposium on AI Applications in Knowledge Navigation and Retrieval*, pp. 19–24, American Association for Artificial Intelligence, 1995.

[10] Cohen, W.W., Learning to Classify English Text with ILP Methods, *Workshop on Inductive Logic Programming*, Leuven, September 1995.

[11] Drummond,C., Ionescu, D., Holte, R., A Learning Agent that Assists the Browsing of Software Libraries, *Technical Report TR-95-12*, Computer Science Dept., University of Ottawa, 1995.

[12] Mc Elligott, M., Sorensen, H., An emergent approach to information filtering, *Abakus. U.C.C. Computer Science Journal*, Vol 1, No. 4, December 1993.

[13] Etizioni, O., Weld, D., A Softbot-Based Interface to the Internet, *Communications of the ACM* Vol. 37, No. 7, pp.72–79, July 1994.

[14] Foltz, P. W. and Dumais, S. T., Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM*, 35(12), pp.51—60, 1992.

[15] Hammond, K., Burke, R., Schmitt, K., A Case-Based Approach to Knowledge Navigation, *AAAI Workshop on Indexing and Reuse in Multimedia Systems*, pp. 46–57, American Association for Artificial Intelligence, 1994.

[16] Holte, R.C., Drummond, C., A Learning Apprentice For Browsing, *AAAI Spring Symposium on Software Agents*, 1994.

[17] Joachims, T., Mitchell, T., Freitag, D., Armstrong, R., WebWatcher: Machine Learning and Hypertext, *Fachgruppentreffen Maschinelles Lernen*, Dortmund, August 1995.

[18] Joachims, T., A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, *Technical report, CMU-CS-96-118*, School of Computer Science, Carnegie Mellon University, March 1996.

[19] de Kroon, H.C.M, Mitchell, T., Kerckhoffs, E.J.H., Improving Learning Accuracy in Information Filtering, *ICML-96 Workshop Machine learning meets human computer interaction*, 1996. URL: http://www.ics.forth.gr/ moustaki/ICML96_HCI_ML/kroon.ps

[20] Krulwich, B., Burkey, C., The ContactFinder agent: Answering buletin board questions with referrals, *Proceedings of the Thirteenth National Conference on Artificial Intelligence AAAI 96*, pp.10—15, 1996.

[21] Lang, K., News Weeder: Learning to Filter Netnews, *Proc. of the 12th International Conference on Machine Learning ICML95*, 1995.

[22] Lewis, D.,D., Gale, W., A., A Sequential Algorithm for Training Text Classifiers, *Proc. of the 7th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dubline, 1994.

[23] Maes, P., Agents that Reduce Work and Information Overload, *Communications of the ACM* Vol. 37, No. 7, pp.30–40, July 1994.

[24] Mitchell, T.M., Machine Learning, The McGraw-Hill Companies, Inc., 1997.

[25] Mitchell, T., Caruana, R., Freitag, D., McDermott, J., Zabowski, D., Experience with a Learning Personal Assistant, *Communications of the ACM* Vol. 37, No. 7, pp.81–91, July 1994. URL: http://www.cs.cmu.edu/afs/cs/user/mitchell/ftp/cacm.ps.Z

[26] Pazzani, M., Muramatsu, J., Billsus, D., Syskill & Webert: Identifying interesting web sites, *AAAI Spring Symposium on Machine Learning in Information Access*, Stanford, March 1996 and *Proceedings of the Thirteenth National Conference on Artificial Intelligence AAAI 96*, pp.54—61, 1996.

[27] Quinlan, J.R., Constructing Decision Tree in *C4.5: Programs for Machine Learning*, pp.17–26, Morgan Kaufman Publishers, 1993.

[28] Rachlin, J., Kasif, S., Salzberg, S., Aha, D.W., Torwards a Better Understanding of Memory-Based Reasoning Systems, *Proc. of the 11th International Conference on Machine Learning ICML94*, pp. 242—250, 1994.

[29] Rocchio, J., Relevance Feedback in Information Retrieval, in *The SMART Retrieval System: Experiments in Automatic Document Processing*, Chapter 14, pp.313—323, Prentice-Hall Inc., 1971.

[30] Salton, G., Buckley, C., Term Weighting Approaches in Automatic Text Retrieval, *Technical report, COR-87-881*, Department of Computer Science, Cornell University, November 1987.

[31] Sorensen, H., McElligott, M., PSUN: A Profiling System for Usenet News, *CIKM'95 Intelligent Information Agents Workshop*, Baltimore, December 1995.

[32] Yang, Y., 1 Expert Network: Effective and Efficient Learning form Human Decisions in Text Categorization and Retrieval, *Proc. of the 7th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieva*l, Dubline, 1994.