ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 257-264

Searching the Web: Learning Based Techniques

M. Diligenti, M. Gori, M. Maggini, F. Scarselli
Dipartimento di Ingegneria dell'Informazione, Università di Siena

Via Roma, 56 - 53100 Siena, Italy
E-mail: {diligmic,marco,maggini,franco}@dii.unisi.it

Abstract. Searching and retrieving information from the Web poses new issues that can be effectively tackled by applying machine learning techniques. In particular, the fast dynamics of the information available on the Internet requires new approaches for indexing; the huge amount of available data is hardly manageable by humans and, on the other hand, can provide large sets of examples for learning algorithms; finally, there is the need of new services like search tools optimized for a specific Web community or even for a single user. Thus, the main areas for the application of these methodologies are the classification of Web documents, the modeling of users' behavior, the personalization of search engines, the automatical extraction of information from Web pages, and the auto-organization of documents on the base of their contents. Many services used to access information on the Web, like Web directories, crawlers, search engines and recommender systems, can be improved by learning from data. This survey reports the main contributions in the application of learning-based techniques to Web searching.

1 Introduction

The dynamic content of the World Wide Web and its rapid size increase make it hard to devise and manage generic-purpose searching tools. For example, maintaining up-to-date the indices of search engines by exhaustive crawling is rapidly becoming impossible, even employing massive crawlers and a large bandwidth. In December 1997, it was estimated that the biggest search engine (*HotBot* at that time) covered 34% of the Web. In February 1999, the same experiment was repeated yielding that the search engine with the largest coverage (*Northern Light*) was indexing only 16% of the Web contents [12]. Machine learning techniques can be used to devise search services tailored to the needs of a specific Web community or even of a single user. These search tools focus their exploration only on the small portion of the Web which contains the information relevant for a specific topic. This approach offers a potential solution to face the fast dynamics of the Internet. Moreover, learning-based search tools can feature a very high precision in retrieving information and can reduce the

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 257-264

need for human efforts for many repetitive tasks (like organizing documents in Web directories). Thus, these techniques can be profitably used for designing Web directories, crawlers, search engines and recommender systems.

This survey describes the current applications of machine learning to searching information on the Internet. In particular, in the next section the techniques for automatic document classification are described. Then, in section 3 the selective gathering of documents from the Internet by a thematic traversal of the Web graph is discussed. Section 4 shows some applications to document clustering. Finally, section 5 reviews some solutions for the problem of information extraction from Web documents.

2 Document classification

In the last few years many efforts have been spent to develop personal assistants that search the Web on behalf of an user. These intelligent agents are able to learn the interests of a single user (or a community of individuals) and then employ the user's profile to search the Internet either by focus crawling, by automatically querying search engines or by filtering Usernets. In order to avoid the need for the user to write explicitly a set of rules which recognize "hot" documents, machine learning has been widely employed for automatically constructing Web document classifiers with reduced human interaction.

WebWatcher [1] is a recommender system which assists users in finding the best path while surfing the Web. It gives suggestions on the hyperlinks to follow and receives positive feedback from the user when the goal of the search is reached. Three different learning-based classifiers for link classification have been tested inside the system: *Winnow*, *Wordstat* using statistics of single words, *TF-IDF* representation of words together with a modified version of the Rocchio distance (cosine correlation between keyword vectors). In a newer version of the system, called *Personal WebWatcher* [17], the user provides a set of "hot" documents which are examples of his/her interests. The Naive Bayes and the k-Nearest Neighbor classifiers have been compared in building the user's profile and scoring the links. k-Nearest Neighbor classification provides a slightly better performance both in accuracy and precision.

Even in the *Syskill and Webert* recommender system [3] a classifier learns the user profile starting from a set of "hot" and "cold" documents. An approach based on a Naive Bayes classifier was selected after a comparison with other six different learning algorithms [19]: k-Nearest Neighbor, ID3, Perceptron, Neural Networks trained by Back-Propagation, PEBLS, and Rocchio. Back-Propagation, Naive Bayes, and Rocchio perform slightly better than the other algorithms on the average. Naive Bayes was used in the final version of Syskill and Webert for its high speed both in learning and prediction, and the easiness to add prior knowledge which increases its accuracy. The recommender system is able to crawl the Web from a starting seed searching target documents, to query automatically a search engine using the most discriminative words in the user's profile, to filter the search engine results, and to keep track of Web

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 257-264

changes monitoring a set of interesting pages and classifying the differences between the current and the earlier version of the page. Feature selection is performed using word cross-entropy computed on the list of "hot" and "cold" pages.

An agent, which searches the Web and selects the best pages to present to the user is presented in [2]. After receiving feedback from the user about the relevance of the recommended pages, the system refines the classifier and updates the search. A set of weights, associated to vocabulary words, are used to classify the documents. Relevance feedback is used to update the weights according to the user interaction.

Traditional search engines rank the results using predefined criteria that do not take into account users' needs. In [5] an automatic procedure to optimize the ranking criteria of a search engine by analyzing users feedback is proposed. The parameters of the classifier are learned using the *modified downhill simplex* variant of simulated annealing.

WebCobra is a collaborative filtering system [9] for recommending documents among users with similar interests. Machine learning allows the system to dynamically refine the boundaries of the communities, taking into account changes in the users' interests. WebCobra creates an user profile according to the relevant documents. The system generates accurate user profiles by employing a Natural Language Processing technique, based on the *HT-summarization* paradigm. Users' profiles are grouped by a clustering module based on the *AutoClass* classifier.

The *WAWA* system [22] can be either used as a recommender system, crawling the Web from a set of user-provided seeds, or as a focused crawler able to construct vertical portals. The characteristics of the interesting documents are described by the users with a specific language. The system stores the user profile in a Neural Network and then refines the knowledge learning from examples. While crawling the Web, the Neural Network is used to score pages and links.

Amalthea [18] learns the users' interests by analyzing a list of example documents. Interesting Web pages are collected by automatically querying search engines, by filtering information environments and by monitoring a fixed set of URLs supposed to contain "hot" information. The system is composed by two sets of *filtering* and *discovery agents*. The filtering agent models the user's interests, adapting itself if the user's interests evolve over time. The discovery agent optimizes the search of the relevant information by adapting the management of the information sources. Agents are described by an evolvable *genotype* and a *phenotype* that is fixed. The genotype of a generic filtering agent contains the weighted keyword vector representing the model of the user's interest for that agent. The genotype of a discovery agent collects the URL of the search engine to query, the number of keywords to use, etc. Agents evolve in an artificial ecosystem, that implements a reward/penalty strategy based on the user's feedback. Only agents that obtain a high "credit" are allowed to reproduce.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 257-264

The *TAPER* system, proposed by the IBM research labs, automatically classifies Web pages into hierarchical topic taxonomies [21]. Classification is performed using a set of Naive Bayes classifiers. In order to increase the performance, the document representation includes terms which encode the connectivity among documents (terms representing documents linked by or linking the current document).

In [24] an implementation of mobile software agents for document retrieval and classification is proposed. Three different trainable classifiers are compared: a simple classifier based on TF-IDF feature representation and *cosine correlation*, a Naive Bayes Classifier and a *DistAI* Neural Network classifier. The cosine distance and the Neural Network classifier outperform the Naive Bayes classifier on almost all the datasets.

3 Focus Crawling

A generic crawler discovers new documents starting from a set of pages, called *seeds*, and then recursively following the hyperlinks contained in the collected pages. Since the goal of the crawler is to cover the largest possible portion of the Web, no crawling strategy is applied. A unfocused crawler is time and resource demanding: even the crawlers used by popular search engines require many weeks to perform a complete Web spidering. On the other hand, a focused crawler efficiently seeks documents about a specific topic and guides its search using both the content and link structure of the Web [6]. The crawling strategy requires a module which associates a score with each link found in a Web page. Then, the links are inserted into a priority queue. A best–first search is performed by popping the next URL to retrieve from the head of the queue [7]. The rules which define the spidering policy depend on the particular topic. When a new topic is selected, a new set of rules must be designed from scratch. Machine learning has been widely employed in automatically reconfiguring the crawling strategy with a reduced human interaction.

In [16] a population of agents perform crawling through the Web. Genetic recombination is applied between agents when they reproduce. A Neural Network encoding the user's interests is used by each agent to select the most promising links. Users can provide a feedback which is used to adapt the parameters.

The simplest focused crawlers use a partial modeling of the paths leading to interesting documents. The policy is based on the evaluation of the current page for topic relevance and each link contained in the page is assigned the same score [6]. A major problem which affects this strategy is that successful paths may include off-topic documents. Basically a reliable strategy should model paths rather than the single pages.

In [13] reinforcement learning was used to train a crawler on example Web sites containing target documents. Reinforcement learning allows to model how a target page can be reached and not just how it looks like. Thus the crawling strategy is optimized regarding future reward which can be found many steps

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 257-264

after the current page. Paths of many steps (many clicks on the hyperlinks) leading to interesting pages can be modeled in order to discover interesting information. This crawling strategy has been used to create a collection of more then 50,000 Computer Science papers.

In [10] the focus crawler employs a compact context representation called a *Context Graph* to model the paths leading to target pages. The context is recursively constructed by back crawling the Web from the set of relevant documents. The context graphs are used to construct a pool of Naive Bayes classifiers trained to assign documents to the different categories based on the expected link distance from a target document. The classifiers are used to predict the number of steps to reach a target page from the current document. The crawling policy assigns higher priority to links which are likely to lead to a relevant page in the least number of "clicks".

4 Document clustering

Yahoo or Open Directory are huge Web directories (more then 700,000 nodes) storing Web pages into a topic hierarchy. Leaves of the tree are web documents, whereas intermediate nodes represent a specific topic. Generic topics are associated to nodes at the upper levels of the tree, whereas more specific topics are stored into lower levels. When a user searches in a Web directory, he/she descends a path from the root of the tree, till he/she finds the target documents. Web directories are usually constructed by large teams of human experts. Since human categorization is quite reliable, Web directories respond to users' queries with very good precision. Unfortunately the construction and update process is time consuming and such hierarchies can not follow the fast dynamics of the Web. Thus, Web hierarchies cannot include recent pages. Moreover human construction of huge hierarchies can lead to incongruence in the topic tree reducing the information recall.

Machine learning can be either used to automatically fill an already build Web directory with data, or even to construct the hierarchy structure. In the first case the hierarchy is constructed by human experts, who also provide few examples for each class. Then learning-based document classification techniques are used to associate more documents to each node [14]. In the other case, the hierarchy is both constructed and filled using unsupervised learning techniques. Documents can be incrementally added by automatic classification. Unsupervised learning techniques can recursively create document clusters, starting from a collection of unlabeled data. Hence, automatically generated Web directories can manage a much larger amount of data, resulting in document repositories which are more complete and extensive. Moreover, they can rapidly include new pages following the fast dynamics of the Web.

WEBSOMs [11] extend SOM networks, commonly used to cluster data in pattern recognition, to deal with large collections of text data. Some simplifications and optimizations are proposed in order to make the problem feasible, even if both the document collection and the feature space are large.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 257-264

In [4] the ARHP (Association Rule Hypergraph Partitioning) clustering technique, used in data mining, is applied to clustering Web documents. The technique is compared to other algorithms: *PDDP* (Principal Direction Divisive Partitioning), *HAC* (Hierarchical Agglomeration Clustering) and *AutoClass*.

5 Information Extraction

Machine learning has been proposed to automatically extract information from HTML documents. Automatic extraction of information can be either used to construct domain-specific databases, accessible by search engines, or to create intelligent agents that seek specific information on behalf of users.

WebKB [8] is used to automatically create a knowledge base from Web documents. The system learns to extract information from a training dataset of labeled regions of hypertext. Web pages are classified using a modified Naive Bayes classifier. Ontologies are constructed using a modification of FOIL algorithm.

Hidden Markov Models (HMM) are employed in the *Cora* system to extract information about publications from references in scientific papers [15]. A training set of references has been manually tagged into 13 semantic classes (title, author, journal, pages, publisher, etc.). Emission parameters for the 13 classes are estimated from a large corpus of BibTeX data, downloaded from the Web. The best model correctly labels 92.9% words on 200 test references.

WebFoot [23] parses Web pages into coherent segments to feed *CRISTAL*, a *Natural Language Processing* system that is able to extract knowledge from free text. A set of manually annotated text examples is used to train the classifier.

In [20] a system that learns to extract information from e-commerce Web pages is described. After the learning phase, the system can visit on-line vendors stores, extract information about available goods, and report results to the user. The system is composed by two agents, *SoftBot* and *ILA*. *SoftBot* learns to discover information by iteratively filling HTML forms until the needed information is found. *ILA* learns to parse the output of an information source. For example, querying the white pages Web site, the system learns that the response pages contain information on individuals and in particular on people names, e-mail addresses and telephone numbers. ILA queries the information source with known objects and searches a correspondence between its internal models and the returned text.

References

[1] Robert Armstrong, Dayne Freitag, Thorsten Joachims, and Tom Mitchell. WebWatcher: A learning apprentice for the World Wide Web. In *AAAI Spring Symposium on Information Gathering*, pages 6–12, 1995.

[2] Marko Balabanovic and Yoav Shoham. Learning information retrieval agents: Experiments with automated Web browsing. In *Proceedings of the*

AAAI Spring Symposium on Information Gathering from Heterogenous, Distributed Resources, pages 13–18, 1995.

[3] D. Billsus and M. Pazzani. Learning probabilistic user models. In *Proceedings of Sixth International Conference of User Modeling UM97*. Springer Wien New York, Vienna, New York, 1997.

[4] D. Boley, M. Gini, R. Gross, E. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Partitioning based clustering for web document categorization. *Decision Support Systems*, 27(3):329–341, 1999.

[5] J. Boyan, D. Freitag, and T. Joachims. A machine learning architecture for optimizing Web search engines. In *Proceedings of the AAAI Workshop on Internet-based Information Systems*, 1996.

[6] S. Chakrabarti, M. van der Berg, and B Dom. Focused crawling: a new approach to topic-specific Web resource discovery. In *Proceedings of the 8th International World-Wide Web Conference (WWW8)*, 1999.

[7] Junghoo Cho, Hector Garcia-Molina, and Lawrence Page. Efficient crawling through URL ordering. In *Proceedings of the 7th World-Wide Web Conference (WWW7)*, 1998.

[8] Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew K. McCallum, Tom M. Mitchell, Kamal Nigam, and Seán Slattery. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1-2):69–113, 2000.

[9] O. de Vel and S. Nesbitt. A collaborative filtering agent system for dynamic virtual communities on the Web. In *Working Notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[10] M. Diligenti, F. Coetzee, S. Lawrence, L. Giles, and M. Gori. Focus crawling by context graphs. In *Proceedings of the International Conference on Very Large DataBases, 11-15 September 2000, Il Cairo, Egypt*, pages 527–534, 2000.

[11] Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Salojrvi, Vesa Paatero, and Antti Saarela. Organization of a massive document collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3):574–585, May 2000.

[12] Steve Lawrence and C. Lee Giles. Accessibility of information on the Web. *Nature*, 400:107–109, 8 July 1999.

[13] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Building domain-specic search engines with machine learning techniques. In *Proceedings of the AAAI Spring Symposium on Intelligent Agents in Cyberspace*, 1999.

[14] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Nigam. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the 15th International Conference on Machine Learning, (ICML-98)*, pages 359–367, 1998.

[15] A. Kachites McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[16] Filippo Menczer and Richard K. Belew. Adaptive retrieval agents: Internalizing local context and scaling up to the Web. *Machine Learning*, 39(2/3):203–242, 2000.

[17] D. Mladenic. Personal WebWatcher design and implementation. Technical Report IJS-DP-7472, School of Computer Science, Carnegie Mellon University, 1996.

[18] Alexandros Moukas. Amalthaea: Information discovery and filtering using a multiagent evolving ecosystem. In *Proceedings of the Conference on Practical Applications of Agents and Multiagent Technology*, April 1996.

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

[20] M. Perkowitz, R. Doorenbos, O. Etzioni, and D. Weld. Learning to understand information on the internet: An example-based approach. *Journal of Intelligent Information Systems*, 8(2):133–153, 1997.

[21] S. Chakrabarti R. Agraval, P. Raghavan and B. Dom. Scalable feature selection, classification and signature generation for organizing large text databases into topic taxonomies. *VLDB journal*, 7(3):163–178, 1998.

[22] J. Shavlik and T. Eliassi-Rad. Building intelligent agents for Web-based tasks: a theory-refinement approach. In *Working Notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[23] S. Soderland. Learning to extract text-based information from the World Wide Web. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, 1997.

[24] J. Yang, P. Pai, V. Honavar, and L. Miller. Mobile intelligent agents for document classification and retrieval: A machine learning approach. In *14th European Meeting on Cybernetics and Systems Research. Symposium on Agent Theory to Agent Implementation, Vienna, Austria.*, 1998.