UDC:004.6
Survey paper

HOW TO APPROACH DATA ANALYSIS OF TEXTS

Dunja Mladenič
J.Stefan Institute, Jamova 39, 1000 Ljubljana, SLOVENIA
Dunja.Mladenic@ijs.si

Abstract: *Analysis of large text data sets is gaining popularity providing the users some insights into their own (potentially even very unstructured) data sets that where difficult to get using the standard methods. This kind of data analysis differs from the standard analysis in the following three directions: (1) the used methods for data analysis differ from the standard statistical methods, (2) the data we are analyzing have different characteristics than the standard, structured data bases, and (3) the users of the data analysis results have different needs and requirements than the usual users of common analytical services (statistics, data-mining, OLAP). This paper gives a brief idea of the area addressing that kind of data analysis commonly referred to as Text-Mining. It is a growing area placed at the intersection of Information-Retrival (IR), Data-Mining (DM), Machine-Learning (ML), Natural-Language-Processing (NLP). The problems usually addressed in Text-Mining are topic detection and tracking, document categorization, visualization of document collections, user profiling, information extraction, construction and updating of hierarchical indices and document collections, intelligent search.*

Keywords: *text data analysis, data mining, example applications of text mining, personalized information delivery.*

1. INTRODUCTION

Different problems involve analysis of large text data sets, one of the most typical being filtering of text information, as performed for discarding spam e-mails or selecting interesting news messages from a large set of divers messages submitted to a number of news groups. A kind of text information filtering is also performed to help the user browsing the Web based on the generation and analysis of the user profile, as described in Section 2. When given a set of document talking about different topics, TextMining can be used to group (cluster) the document according to the similarity of their content and to assign the content category or keywords to a new document based on the already classified documents. Larger document can be automatically divided in content segments, a long stream of news can be addressed by identification of a new topic being reported on and tracking the raise and disappearance of the topic from the time series of documents (news).

A number of intelligent agents involve some kind of text data analysis, such as intelligent agent for news filtering, agent searching for experts from a particular area,

personal browsing assistant described in Section 2. For more information about analysis of text data sets and related intelligent agents see (Mladenic 1999).

The rest of this paper is organized as follows. Section 2 gives a brief idea of the data characteristics and the methods used in Text Mining. An example of analysis of text data used in user profiling by personal browsing assistant is given in Section 3. Section 4 describes an approach to automatic document categorization in hierarchical document collections. Section 5 concludes by brieflydescribing some recent research results having a great potential for real-world applications.

2. BASIC METHODS

There is a lot of research work in the area of text data analysis, a large part of it being related to the Web. One possibility is to discuss it through the prism of three important methodological questions: (1) what representation is used for text documents, (2) how is the high number of words (usually referred to as features) dealt with and (3) which algorithm is used.

2.1 DOCUMENT REPRESENTATION

The frequently used document representation in Information Retrieval and Machine Learning on text data is so called vector representation. It is a *bag-of-words representation*: all words from the document are taken and no ordering of words or any structure of the text is used. When having a set of documents, each document is represented as a bag-of-words including all the words that occur in the set of documents (see Figure 1). We all agree that there is additional information in text documents that could be used, for example, some Natural Language Processing information about the structure of the sentences, word type and role, position of the words or neighboring words. The question is how much can we gain considering additional information in the data analysis (and what information to consider) and what is the price we have to pay for it? There is currently no established comparison or directions for text document representation that we are aware of. There is some evidence in information retrieval research, that for long documents, considering information additional to the bag-of-words is not worth the efforts.

There is also work on document classification that extends the bag-of-words representation by using word sequences (n-grams) instead of single words (Mladenic and Grobelnik 1998). This work suggests that the usage of single words and word pairs as features in the bag-of-words representation improve performance of classifiers generated from short documents.

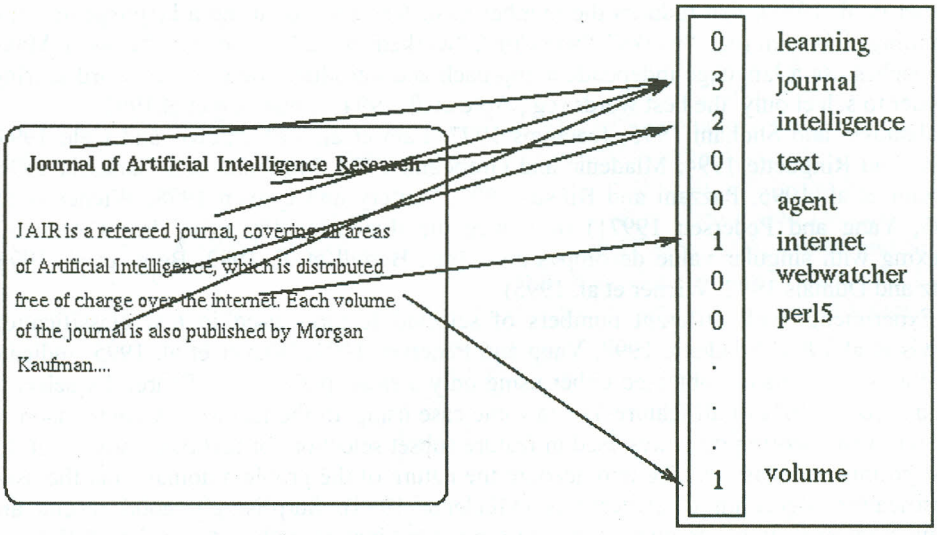

Figure 1: Illustration of the bag-of-words document representation using frequency vector. Each word occurring in a set of documents is mapped into a feature. Document is represented as a vector of features where each feature is assigned a frequency (the number of times it occurs in the document).

Many current systems that learn on text use the bag-of-words representation using either Boolean features indicating if a specific word occurred in a document (eg., Armstrong et al. 1995, Creecy et al. 1992, Cohen 1995, Gelfand et al. 1998, Lewis Gale 1994, Lewis Ringuette 1994, Liere Tadepalli 1998, Maes 1994, Moulinier and Ganascia 1996, Nigam and McCallum 1998, Pazzani et al. 1996, Pazzani Billsus 1997, Shavlik and Eliassi 1998, Slattery and Craven 1998) or the frequency of a word in a given document (eg., Apte et al. 1994, Apte et al. 1998, Balabanovic Shoham 1995, Bartell et al. 1992, Berry et al.1995, Joachims 1997, Joachims 1998, Lam et al. 1997, Lam and Ho 1998, Mladenic 1996, Mladenic and Grobelnik 1998, Yang 1998). There is also some work that uses additional information such as word position (Cohen 1995, Cohen and Singer 1995, Shavlik Eliassi 1998) or word tuples called n-grams (Elligott and Sorensen 1993, Mladenic and Grobelnik 1998, Sorensen and McElligott 1995) (eg., "machine learning" is a 2-gram and "World Wide Web" is a 3-gram).

Some recent work (Slattery and Craven 1998) indicates that the usage of hypertext structure and graph organization of Web pages improves classification results.
There is currently no study that compares different document representations over several domains showing clear advantages of some representations.

2.2 SELECTION OF WORDS

One of the frequently used approaches to reduce the number of different words is to remove words that occur in the *stop-list* containing common English words like "a", "the", "with" (eg., Apte et al. 1994, Balabanovic and Shoham 1995, Lam and Ho 1998, Lewis and Ringuette 1994, Mladenic and Grobelnik 1998, Pazzani et al. 1996, Pazzani and Billsus 1997, Shavlik and Eliassi 1998, Wiener et al. 1995) or pruning the infrequent words (word frequency < min.frequency) (eg., Cohen 1995, Joachims 1997, Joachims 1998, Mladenic Grobelnik 1998). Connected to the particular language is also word stemming, used for example in (Apte et al. 1998, Balabanovic and Shoham 1995, Shavlik and Eliassi 1998,

Wiener et al. 1995), that reduces the number of different words using a language-specific stemming algorithm (eg., "works", "working", "worked" are all replaced by "work"). Many approaches use a language independent approach and introduce some sort of word scoring in order to select only the best words (eg., Apte et al. 1994, Armstrong et al.1995, Balabanovic and Shoham 1995, Joachims 1997, Lam et al. 1997, Lewis and Gale 1994, Lewis and Ringuette 1994, Mladenic and Grobelnik 1998, Moulinier and Ganascia 1996, Pazzani et al. 1996, Pazzani and Billsus 1997, Slattery and Craven 1998, Wiener et al. 1995, Yang and Pedersen 1997}) or reduce the dimensionality using latent semantic indexing with singular value decomposition (eg., Bartell et al. 1992, Berry et al. 1995, Foltz and Dumais 1992, Wiener et al. 1995).

Experiments with different numbers of selected features used in text classification (Lewis et al 1996, Mladenic 1998, Yang and Pedersen 1997, Wiener et al. 1995) indicate that the best results are obtained either using only a small percentage of carefully selected features (up to 10% of all features) or in some case using all the features. A comparison of different word scoring measures used in feature subset selection for text data shows that the most promising measures take into account the nature of the problem domain and the used classification algorithm characteristics (Mladenic 1998). Surprisingly good results are obtained using a simple frequency measure in a combination with a "stop-list" (Mladenic 1998, Mladenic and Grobelnik 1998, Yang and Pedersen 1997).

2.3 ALGORITHMS FOR DATA ANALYSIS

In Information Retrieval, one of the well-established techniques for text document classification is to represent each document using bag-of-words as a TFIDF-vector in the space of words that appear in training documents (Salton and Buckley 1987), sum all interesting document vectors and use the resulting vector as a model for classification (based on the relevance feedback method Rocchio 1971). Each component of a document vector $d^{(i)} = TF(W_i, d)IDF(W_i)$ is calculated as the product of Term Frequency (TF) – the number of times word W occurred in a document and Inverse Document Frequency $IDF(W_i) = \log \frac{D}{DF(W_i)}$, where D is the number of documents and document frequency DF(W) is the number of documents word W occurred in at least once. The exact formulas used in different approaches may slightly vary but the idea remains the same. A new document is then represented as a vector in the same vector space as the generated model and the distance between these two vectors is measured (usually using the cosine similarity measure) in order to classify the document. This technique is commonly used as a baseline when testing performance of some machine learning algorithms (Mitchell 1997) on text data. TFIDF classification has already been used in Machine Learning experiments on the World Wide Web data (eg., Armstrong et al. 1995, Balabanovic and Shoham 1995, Berry et al. 1995, Joachims 1997, Pazzani et al. 1996, Pazzani and Billsus 1997) and in most cases shown to be inferior to the tested machine learning methods.

An extension of TFIDF proposed in (Joachims 1997) called Probabilistic-TFIDF takes into account document representation and was shown to achieve results better than TFIDF and comparable to the Naive Bayesian classifier. The Naive Bayesian classifier and the k-Nearest Neighbor are two classifiers commonly used in text-learning and reported to be among the best performing classifiers for text data. For instance, the Naive Bayesian classifier was used in (Joachims 1997, Lewis and Ringuette 1994, Mladenic 1996, Mladenic and Grobelnik 1998, Pazzani et al. 1996, Pazzani and Billsus 1997). The Nearest Neighbor algorithm was used in (Mladenic 1996, Pazzani et al. 1996, Pazzani and Billsus

1997, Yang 1998). In addition to using the Naive Bayesian classifier and Nearest Neighbor, Pazzani et al. (Pazzani et al. 1996, Pazzani and Billsus 1997) as well as Lewis and Ringuette (Lewis and Ringuette 1994) performed experiments on text data with symbolic learning using Decision Trees. Moulinier and Ganascia (Moulinier and Ganascia 1996) experimented using Decision Rules. Yang (Yang 1998) compared the performance of Linear Least Square Fit (LLSF) and a variant of k-Nearest Neighbor, reporting that similar results are achieved by both classifiers. Creecy et al. (Creecy et al. 1992) and Maes (Maes 1994) used Memory-Based reasoning. Apte et al. (Apte et al. 1994) used Decision Rules and in (Apte et al.1998) boosted Decision Trees. Cohen (Cohen 1995, Cohen and Singer 1995) used Decision Rules, the Sleeping experts algorithm and two Inductive Logic Programming (ILP) algorithms FOIL and FLIPPER. Slattery and Craven (Slattery and Craven 1998) used the Naive Bayesian classifier and two ILP algorithms FOIL and FOIL-PILFS (FOIL with Predicate Invention for Large Feature Spaces). Lewis et al. (Lewis and Gale 1994, Lewis et al 1996) used a combination of the Naive Bayesian classifier and logistic regression, the Widrow-Hoff algorithm and Exponential Gradient (EG).

Wiener et al. (Wiener et al. 1995) performed experiments showing that Neural Networks achieve slightly better results than Logistic Regression. McElligot and Sorensen (Mc Elligott and Sorensen 1993, Sorensen and McElligott 1995) used a connectionist approach combined with Genetic Algorithms. Lam et al. (Lam et al.1997) used Bayesian Network Induction. Gelfand (Gelfand et al. 1998) used Semantic Relationship Graphs (SRG) to represent documents based on the WordNet lexical database. Classification is performed in a similar way TFIDF is used; each class is defined by a group of training documents and represented as a union of their SRG representation. Armstrong et al. (Armstrong et al. 1995) used the Winnow algorithm and a statistical approach they called WordStat that assumes mutual independence of words. Shavlik and Eliassi (Shavlik and Eliassi 1998) used theory-refinement on Neural Networks, where the user provides an initial advice that is compiled into Neural Network and refined during the interaction with the user based on the users page ratings and additionally provided advises. Active learning was used by Liere and Tadepalli (Liere and Tadepalli 1998) where a committee of Winnow learners is used. Active learning was also used in (Nigam and McCallum 1998) in a combination of Query by Committee and the Expectation Maximization (EM) algorithm.

There is currently no strong evidence about the superiority of any of the given algorithms for text-learning over different domains. Most experiments show the superiority of the tested algorithm over the TFIDF classification. A comparison of some learning algorithms given in (Pazzani and Billsus 1997) indicates that a document representation including feature selection is more promising approach to classification accuracy improvement than finding a better learning algorithm.

In their experiments, the Naive Bayesian classifier, Nearest Neighbor and Neural Networks achieve the best results on the tested data. Similar observations about a good performance of k-Nearest Neighbor, Neural Networks and Linear Least Square Fit was reported in (Yang 1998), while in the same experiments the Naive Bayesian classifier didn't perform well. On the other hand, in (Yang and Pedersen 1997, Joachims 1998) is reported that on their domains feature subset selection was not crucial for the classifier performance. Joachimas (Joachims 1998) reported that Support Vector Machines outperform the Naive Bayesian classifier, while in (Apte et al. 1998) even better results are reported using boosted Decision Trees. Lam et al. (Lam et al. 1998) observed that the Generalized Instance Set algorithm achieved better results than either k-Nearest Neighbor or linear classifiers (Rocchio, Widrow-Hoff).

2. USER PROFILING FOR BETTER WEB BROWSING

The problem of helping the user browsing the Web can be defined as predicting clicked hyperlinks from the set of Web documents visited by the user. This is performed on-line while the user is sitting behind some Web browser and waiting for the requested document. Our prototype system named Personal WebWatcher (Mladenic 1996) uses some of the methods described in Section 2 on this problem, learning a separate model for each user and using it for highlighting the promising hyperlinks on the requested Web documents. The structure of the system is shown in Figure 2. There is the user on one end and the Web on the other end. Between them is our Personal WebWatcher acting as a so called proxy server. It consists of:

- **proxy** that gets http requests from the browser and fetches the requested Web page,
- **adviser** that gets the original Web page and extracts the hyperlinks from it and composes the modified Web page by highlighting the promising hyperlinks based on the scores assigned to all the extracted hyperlinks,
- **classifier** threats each extracted hyperlink as an example and uses the induced model of the user interests to assign a score to each of them,
- **LEARNER** gets a collection of visited documents and induces a model of the user's interests based on the Web documents.

Browsing the Web is supported here by highlighting promising (ie., interesting) hyperlinks on the requested Web documents. We assume that the interesting hyperlinks are the hyperlinks that are highly probable to be clicked by the user. Our problem is defined as predicting clicked hyperlinks from the set of Web documents visited by the user. All hyperlinks on the visited documents are used for constructing machine learning examples. Each is assigned one of the two class values: positive (user clicked on the hyperlink) or negative. We represented each hyperlink as a kind of small document containing underlined words, words in a window around them and words in all the headings above the hyperlink (the latest hading for each html heading size H1 through H6).

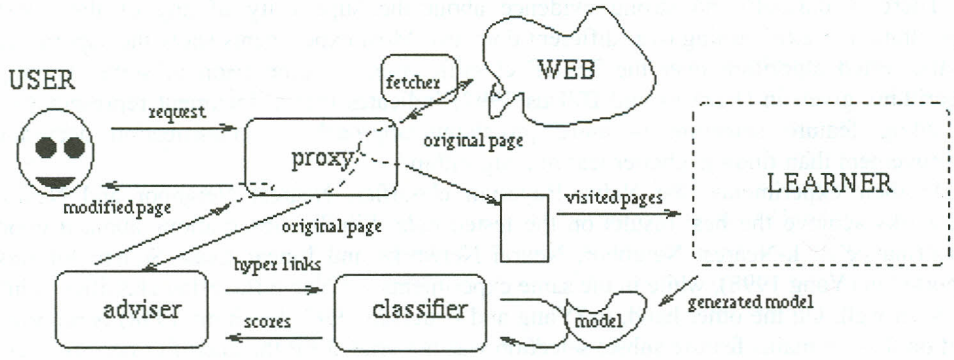

Figure 2: Structure of browsing assistant Personal WebWatcher.

In order to help the users browsing the Web, a profile is induced for each user independently of other users. This profile can be further used to compare different users and to share knowledge between them. For example, instead of having a friend with similar interest sending me the address of some "cool" Web document, the system can automatically suggest documents that were found interesting by the other users that have

interests similar to mine. In order to secure the privacy, only knowledge and not the user identity can be exchanged or even this cooperation could apply only for users that explicitly agreed to take part in knowledge sharing. On the other hand, some users might be interested in "making friends" with similar users and join the list of users whose identity (eg. e-mail) is reviled to similar users from the list. This sharing of knowledge is related to collaborative approach to intelligent agents design (Mladenic 1999) and methods used in multi-agent systems.

A way of cooperation between different users using the same system for user customized Web browsing is on the model induction level. Namely, even though each user has a separate user profile, they have a similar form. If we could infer from the user profiles some higher-level knowledge that is independent of a specific set of documents, that knowledge could be shared between the users. For instance, if we are given some background knowledge, find which part of the given background knowledge is frequently used in different models (what higher-level attributes are useful). That would be especially valuable for new users, where only a small set of documents is available for the model induction.

3. AUTOMATIC DOCUMENT CATEGORIZATION

The problem of automatic text categorization is well known in Information Retrieval community and recently also in Machine Learning community. The problem described here involves using Machine Learning techniques to automatically construct a classifier from a large hierarchy of text documents. Handling a hierarchy an extension of the usually addressed problem of handling a flat structure of categories, such as the well known collection of Reuters news. The existing Web hierarchy we use for this example is *Yahoo* (www.yahoo.com) hierarchy. It is human constructed, regularly updated and captures most of the topics available on the Web. The Yahoo hierarchy itself is currently build on approximately two millions of Web documents located all around the Internet. Hyperlinks to that documents (excluding the top category named *"Regional"*) are organized in about 100,000 *Yahoo Web documents*. Each *Yahoo document* represents one of the included categories denoted by a set of keywords. This documents are connected with hyperlinks forming a hierarchical structure with more general categories closer to the root of the hierarchy. The category is denoted by keywords that appear on the path from the tree root to the node representing category (eg. *"Sport"* a subcategory of *"Scienc"'* is named *"Science: Sport"* and in our approach assigned two keywords: *Science, Sport*). 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 tree). Some nodes at the bottom of the tree contain mostly hyperlinks to actual Web documents, while the other nodes contain mostly or even only hyperlinks to other Yahoo Web documents (nodes in the hierarchy). There are currently fourteen top level Yahoo categories whose name includes only one keyword.

Our goal is to classify an arbitrary text document as accurate and as fast as possible to the right category within Yahoo hierarchy. We observe the list of categories that are assigned the highest probability and the set of keywords that are assigned to the document. The system architecture is shown in Figure 3.

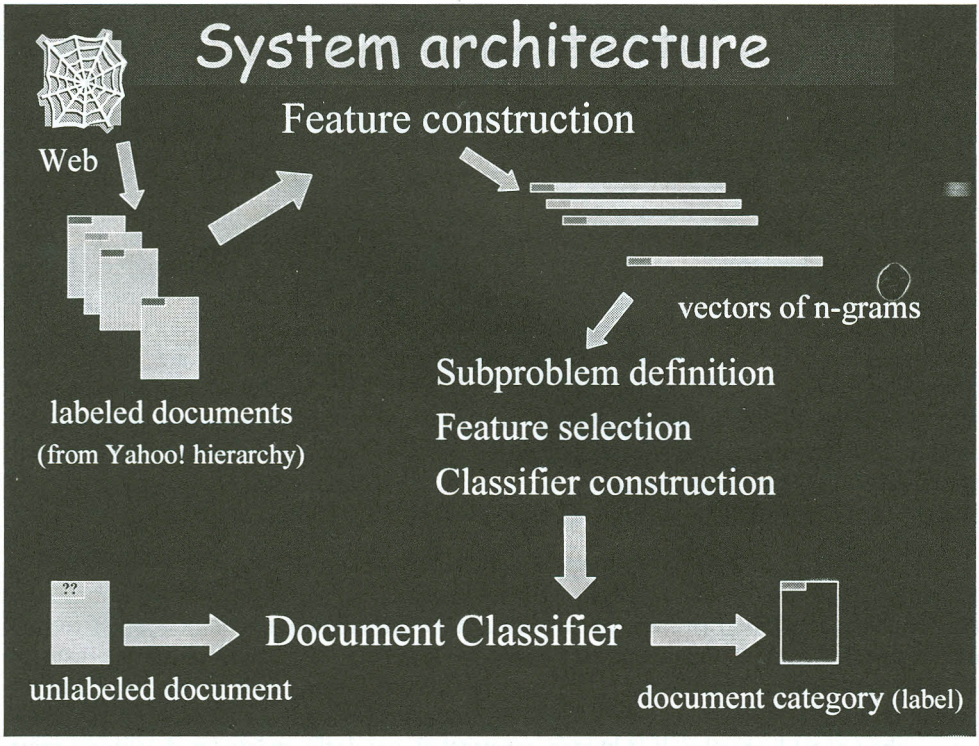

Figure 3: Architecture of the system for automatic document categorization. First a set of labeled Web documents is processed to get the set of potential feature (words and phrases) to be used in document representation. This phase is named *feature construction.* Then, all the documents are represented using the constructed features and a *feature selection* is applied on each of the *defined sub-problems*. For each of the sub-problems, a classifier is constructed and used later for classifying new documents (document categorization).

In order to handle the hierarchical structure, we divided the whole problem into sub-problems, each corresponding to one of the original categories. For each of the sub-problems, a classifier is constructed that predicts the probability that a document is a member of the corresponding category and can thus be assigned the corresponding set of keywords. On each of the sub-problems the Naive Bayesian classifier is used on feature-vector document representation, where each feature represents a sequence of words instead of representing a single word as commonly used when learning on text data. This approach is not limited to Web hierarchy and can be applied on other hierarchies like for instance, thesaurus.

4 FUTURE DIRECTIONS

There is a number of researchers intensively working in the area of Text Mining, mainly guided by the need of developing new methods capable of handling interesting real-world problems. One of such problems recognized in the past few years is on reducing the amount of manual work needed for hand labeling the data. Namely, most of the approaches for automatic document filtering, categorization, user profiling, information extraction, text tagging require a set of labeled data describing the addressed concepts (eg., for automatic document categorization, we start with a set of documents where each document is assigned

to some category based on its content such as in *Yahoo* collection described in Section 4). Using unlabeled data and bootstrapping learning are two directions giving research results that enable important reduction in the needed amount of hand labeling.

In document categorization using unlabeled data, we need a small number of labeled documents and a large pool of unlabeled documents, eg., classify an article in one of the 20 News groups, classify Web page as student, faculty, course, project,... The approach proposed by (Nigam et al. 2001) can be described as combining Expectation Maximization and the Naive Bayesian classifier as follows. First train a classifier with only labeled documents and use the trained classifier to assign probabilistically-weighted class labels to all unlabeled documents. Then train a new classifier using all the documents and iterate until the classifier remains unchanged. It can be seen that the final result heavily depends on the quality of the labels assigned to the small set of hand labeled data, but it is much easier to hand label a small set of examples with a good quality than a large set of examples with medium quality.

Bootstrap learning to classify Web pages is based on the fact that most of the Web pages have some hyperlinks pointing to them. Using that we can describe each Web page either by its content or by the content of the hyperlinks that point to it. First, a small number (eg., 12 documents) of documents is labeled and each is described using the two description. One classifier is constructed from each description independently and used to label a large set of unlabeled documents. A few of that documents for which the prediction was the most confident are added to the set of the labeled documents and the whole loop is repeated. In this way we start with a small set of labeled documents enlarging it through the iterations and hoping that the initial labels were a good coverage of the problem space. This approach was proposed in (Blum and Mitchell 1998) and supported by the computational learning theory in the proposed Co-Training theorem.

Recent work includes also mining the extracted data (Ghani et al. 2000), where Information Extraction is used to automatically collect information about different companies from the Web. Data Mining methods are then used on the extracted data. As Web documents are naturally through the hyperlinks organized in a graph structure, there are also research efforts on using that graph structure to improve document categorization (Slattery and Craven 2000), to improve Web search (www.google.com) and visualization of the Web.

REFERENCES

[1] C. Apte, F. Damerau, S. M. 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 Retrieva}*, Dubline, 1994.

[2] C. Apte, F. Damerau, S. M. Weiss, S.M. Text Mining with Decision Rules and Decision Trees, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[3] R. Armstrong, D. Freitag, T. Joachims, T. Mitchell. WebWatcher: A Learning Apprentice for the World Wide Web, *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, March 1995.

[4] M. Balabanovic, Y. Shoham. Learning Information Retrieval Agents: Experiments with Automated Web Browsing, *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, 1995.

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

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

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

[8] W.W. Cohen, Y. Singer. Context-sensitive learning metods for text categorization, *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, pp.307-315, 1996.

[9] R. M. Creecy, B. M. Masand, S. J. Smith, D. L. Waltz. Trading MIPS and Memory for Knowledge Ebgineering, *Communications of the ACM*, Vol. 35, No.8, pp.48--64, August 1992.

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

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

[12] R. Ghani, R. Jones, D. Mladenic, K. Nigam, S. Slattery. Data Mining on Symbolic Knowledge Extracted from the Web, *In KDD-2000 Workshop on Text Mining, 2000.*

[13] T. Joachims. 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, 1997.

[14] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features, *Proceedings of the 10th European Conference on Machine Learning ECML98*, pp. 137-142, 1998.

[15] W. Lam, K. F. Low, C. Y. Ho, C.Y. Using Bayesian Network Induction Approach for Text Categorization, *15th International Joint Conference on Artificial Intelligence IJCAI97*, pp. 745-750, 1997.

[16] D. D. Lewis, W. A. Gale. 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.

[17] D. D. Lewis, M. Ringuette. Comparison of two learning algorithms for text categorization, *Proc. of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, 1994.

[18] D. D. Lewis, R. E. Schapire, J. P. Callan, R. Ron Papka. Training Algorithms for Linear Text Classifiers, *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*}, pp.298-306, 1996.

[19] R. Liere, P. Tadepalli. Active Learning with Committees: Preliminary Results in Comparing Winnow and Perceptron in Text Categorization, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

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

[21] T. M. Mitchell. *Machine Learning*. The McGraw-Hill Companies, Inc.,1977.

[22] D. Mladenic. Personal WebWatcher: Implementation and Design, *Technical Report IJS DP-7472*, http://www-ai.ijs.si/DunjaMladenic/papers/PWW/

[23] D. Mladenic. Machine Learning on non-homogeneous, distributed text data, *PhD thesis*, University of Ljubljana, Slovenia, October, 1998, http://www.cs.cmu.edu/~TextLearning/pww/

[24] D. Mladenic. Text-learning and related intelligent agents. *IEEE EXPERT, Special Issue on Applications of Intelligent Information Retrieval, 1999*.

[25] D. Mladenic, M. Grobelnik. Word sequences as features in text-learning. *Proceedings of the Seventh Electrotechnical and Computer Sc. Conference ERK'98*, Ljubljana, Slovenia: IEEE section, 1998, pp. 145-148.

[26] D. Mladenic, M. Grobelnik. Feature selection for unbalanced class distribution and Naive Bayes, *Proceedings of the 16th International Conference on Machine Learning ICML-99*, Morgan Kaufmann Publishers, San Francisco, CA, 1999. pp. 258 - 267.

[27] I. Moulinier, J. G. Ganascia. Applying an Existing Machine LEarning Algorithm to Text Categorization. In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, (S.Wermter, E.Riloff, G.Scheler Eds.), Springer-Verlag, 1996.

[28] K. Nigam, A. McCallum. Pool-Based Active Learning for Text Classification, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[29] K. Nigam, A. McCallum, S. Thrun, T. Mitchell. Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning Journal,* 2001.

[30] M. Pazzani, J. Muramatsu, D. Billsus. 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.

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

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

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

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

[35] S. Slattery, M. Craven. Learning to Exploit Document Relationships and Structure: The Case for Relational Learning on the Web, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

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

[37] E. Wiener, J. O. Pedersen, A. S. Weigend. A Neural Network Approach to Topic Spotting, *Proc. of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.

[38] Y. Yang, J. O. Pedersen, J.O. A Comparative Study on Feature Selection in Text Categorization, *Proc. of the 14th International Conference on Machine Learning ICML97*, pp. 412-420 1997.

[39] Y. Yang. An evaluation of statistical approaches to text categorization, *Journal of Information Retrieval*, No.2, 1998.

Received: 14 June 2004
Accepted: 08 December 2004