A Taxonomy Fuzzy Filtering Approach

S. Vrettos and A. Stafylopatis

***Abstract* - Our work proposes the use of topic taxonomies as part of a filtering language. Given a taxonomy, a classifier is trained for each one of its topics. The user is able to formulate logical rules combining the available topics, e.g. (Topic1 AND Topic2) OR Topic3, in order to filter related documents in a stream. Using the trained classifiers, every document in the stream is assigned a belief value of belonging to the topics of the filter. These belief values are then aggregated using logical operators to yield the belief to the filter. In our study, Support Vector Machines and Naïve Bayes classifiers were used to provide topic probabilities. Aggregation of topic probabilities based on fuzzy logic operators was found to improve filtering performance on the Reuters text corpus, as compared to the use of their Boolean counterparts. Finally, we deployed a filtering system on the web using a sample taxonomy of the Open Directory Project.**

***Index Terms*- Fuzzy Aggregation, Taxonomy Filtering, Web Filtering Systems.**

I. INTRODUCTION

The primary way of interactively finding information on the web is by making a query in a search engine and then browsing a ranked list of possibly related web pages. Alternatively, we can browse a manually organized topic taxonomy to find pages related to the query that we have in mind. Although web taxonomies may be very large, they cover a small portion of the web relative to search engines, primarily because they rely on human effort. Content-based filtering is the task of analysing a stream of information based on a semantic structure, which can be automatically derived or pre-specified [1],[2]. Text/Hypertext categorization promises not only to help maintain updated and large web taxonomies, but to be used in the context of content-based filtering [3]-[8].

The idea is to use topic classifiers that have been trained using the portion corresponding to the well-structured web taxonomy in order to organize the results of a query addressed to the much larger but unclassified web portion indexed by a search engine. Basically, as regards the interface used to include topic information in the query results, it can be topic or list-oriented. In topic-oriented interfaces, results are organized in a flat or hierarchical taxonomy, while, in list-oriented interfaces, the original query list is enriched with topic meta-data. Our work proposes the use of topic taxonomies as part of a filtering language. The user is able to formulate logical rules (filters) combining the available topics, e.g. (Topic1 AND Topic2) OR Topic3, in order to filter related documents or to provide relevance feedback as well [9]-[10].

Typically, classification is a YES/NO assignment, so the Boolean model is a good candidate for the filtering task. Nevertheless, Boolean filtering provides no ordering, which is a drawback to both retrieval effectiveness and man-machine interaction. If perfect classifiers were available, Boolean filtering would be enough. That is because all the true positive documents of the stream and only them would be retrieved. In that case, Boolean filtering would yield recall and precision equal to 1. Unfortunately, no perfect classifiers are available yet and even the best performing classifiers in laboratory text corpora might have poor results in real, noisy environments, like the web. In such cases, ranking according to some suitable measure of classification accuracy is able to improve retrieval performance, either by improving recall through the retrieval of false negative documents that were not included in the answer set, or by improving precision through the ordering of true positive documents higher in the rank, above false positive ones.

In this work, Support Vector Machines (SVM) and Naïve Bayes (NB) classifiers are used to provide topic probabilities. Aggregation of topic probabilities based on fuzzy logic is found to improve filtering performance on the Reuters text corpus with respect to Boolean filtering. Finally, fuzzy aggregation is embedded in a web filtering system based on a sample taxonomy of the Open Directory Project.

II. PRODUCING TOPIC PROBABILITIES

Support Vector Machines

The SVM model was proposed by Vapnik in 1979 and gained much popularity in the recent years due to its strong theoretical and empirical justification [11]. In the simplest linear form, a SVM is a hyperplane that separates a set of positive examples from a set of negative examples with maximum margin (the minimal distance τ from the separating hyperlane to the closest data point).A separating hyperplane is a linear function capable of separating the training data in the classification problem without error [12]. Suppose that the training data consist of m samples (documents) that belong or not to a given category: $(d_1,y_1),\ldots,$ (d_m,y_m), $d_i \in R^d$, $y_i \in \{+1,-1\}$, which can be separated by a hyperplane decision function

$$D(d_i) = (w \cdot d_i) + w_0 \qquad (1)$$

with appropriate coefficients w and w_0. A separating hyperplane satisfies the constraints that define the separation of data samples:

$$y_i[(w \cdot d_i) + w_0] \geq 1,\ i = 1,\ldots,m \qquad (2)$$

It is intuitively clear that a larger margin corresponds to better generalization. maximizing the margin τ is equivalent to minimizing the norm of w. An optimal hyperplane is one that satisfies condition (2) above and additionally minimizes

School of Electrical and Computer Engineering National Technical University of Athens157 80 Zographou, Athens, Greece vrettos@cslab.ntua.gr, andreas@cs.ntua.gr

$$\eta(w) = \|w\|^2 \tag{3}$$

with respect to both w and w_0.

The data points that exist at the margin, or –equivalently- the data points for which (2) is an equality, define the location of the decision surface and are called the *support vectors*. In the case of training data that cannot be separated without error, it would be desirable to separate the data with a minimal number of errors. To do so, we penalize examples that fall on the wrong side of the decision boundary introducing positive slack variables $\xi_{i,}$ $i=1,\ldots,m$, to quantify the nonseperable data in the defining condition of the hyperplane. For a training sample d_i the slack variable ξ_i is the deviation from the margin border corresponding to the given category. Slack variables greater than zero correspond to nonseparable points, while slack variables greater than one correspond to misclassified samples.To relieve the problem of nonlinear separability, a nonlinear mapping of the training data into a high–dimensional feature space is usually performed according to Cover's theorem on the separability of patterns [13]. The constrained optimization problem is solved using the method of Lagrange multipliers. In this work, we used the OSU SVM Classifier Matlab Toolbox to train classifiers [14].

For each test example d_i , an SVM classifier outputs a score that is the distance of d_i from the hyperplane learned for separating positive from negative examples. The sign of the score indicates whether the example is classified as positive or negative. In our approach, we want to have a measure of confidence (belief) in the prediction To provide an accurate measure of confidence, a parametric approach was proposed by Platt for SVM, which consists of finding the parameters of a sigmoid function, mapping the scores into probability estimates [15].

Naïve Bayes classifiers

The decision of whether an unseen document d_i belongs or not to a category c_j is based on the estimated belief of d_i belonging to class c_j.

Bayes theorem can be used to estimate the probability Pr(c_j |d_i), that a document d_i is in class c_j.

$$\Pr(c_j \mid d_i) = \frac{\Pr(d_i \mid c_j)\cdot \Pr(c_j)}{\sum_{l=1}^{c} \Pr(d_i \mid c_l)\Pr(c_l)} \tag{4}$$

where Pr(c_j) is the prior probability that a document is in class c_j and Pr(d_i | c_j) is the likelihood of observing document d_i in class c_j. $\hat{\Pr}(c_j)$, the estimate of Pr(c_j), can be calculated from the fraction of the training documents that is assigned to this class. The probability of observing a document like d_i in class c_j is based on the naive assumption that a word's occurrence in class c_j is independent of the occurrences of the other words. Therefore Pr(d_i|c_j) is:

$$\Pr(d_i \mid c_j) = \prod\nolimits_{k=1}^{n} \Pr(t_k \mid c_j), t_k \in d_i \tag{5}$$

where t_k represents the k^{th} term of the collection document. The estimation of Pr(d_i| c_j) is now reduced to the estimation of Pr(t_k| c_j) (Laplace estimator), which is the likelihood of observing t_k in class c_j:

$$\Pr(t_k \mid c_j) = \frac{1 + tf(t_k, c_j)}{n + \sum_{l=1}^{n} tf(t_l, c_j)} \tag{6}$$

where $tf(t_k, c_j)$ is the number of occurrences of the word t_k in category c_j and n is the number of the terms of the corpus.

III. FUZZY LOGIC

In this section, we briefly review some basic concepts in fuzzy sets and fuzzy logic, that will be used in the proposed web filtering approach. Let X be a space of objects and x be an element of X. A classical set A is defined as a collection of elements $x \in A$, such that each x can either belong or not belong to the set A. Therefore, we can represent a classical set A by a set of ordered pairs (x,0) or (x,1), which indicates that $x \in A$ or $x \notin A$, respectively. Extending the definition of the classical set, a fuzzy set is defined as a set of elements that may belong to the set by a membership degree value between 0 and 1. More formally, a fuzzy set A in X is defined as a set of ordered pairs $A = \{(x, \mu(x)), x \in A\}$, where $\mu(x)$ is the membership function (MF) for the fuzzy set A.

Usually, X is referred to as the universe of discourse and may consist of discrete (ordered or unordered) or continuous spaces. Similar to classical set operations of union, intersection and complement, can be defined for fuzzy sets accordingly.

The union of two fuzzy sets A and B is a fuzzy set C, denoted $C = A \cup B$ or C *OR* B, whose MF is related to those of A and B by

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \tag{7}$$

The intersection of two fuzzy sets A and B is a fuzzy set C, denoted $C = A \cap B$ or C *AND* B, whose MF is related to those of A and B by

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \tag{8}$$

The complement of fuzzy set A, denoted by $\neg A$, is defined as

$$\mu_{\neg A}(x) = 1 - \mu_A(x) \tag{9}$$

IV. EVALUATION AND RESULTS

To evaluate fuzzy against Boolean filtering, we used the Reuters-21578 corpus [16]. We consider the flat topic taxonomy that consists of the 10 most frequently assigned topic categories. Using the labels of the topics we are able to formulate filters of the form (Earn) AND (Trade), (Acq) OR (Money-Fx), and use them to find relevant documents in a stream of data. To specify the exact filters to use and measure their effectiveness as regards the logical operators, we considered the "ModApte" split, a standard commonly used partitioning of the Reuters corpus into training and test sets. A search in the test set of the "ModApte" split yielded 213 documents that belong to 2 or more of the specified topics. These documents constitute the set F to be filtered and are mapped to 19 multi-category vectors in the table CM, where cm_{ij} = 1 implies that category c_j exists in multi-category vector m_i (Table I).

Table I The category–multicategory matrix *cm*

	m_1	…	m_i	…	m_{19}
c_1	cm_{11}	…	cm_{1i}	…	$cm_{1,19}$
…	…	…	…	…	…
c_j	cm_{j1}	…	cm_{ji}	…	$cm_{j,19}$
…	…	…	…	…	…
c_{10}	$cm_{10,1}$	…	$cm_{10,i}$	…	$cm_{10,19}$

For every multi-category vector $m_{i,}$ i=1…19, of the filtering set, the logic operator AND is used to combine trained classifiers of all categories having cm_{ji} = 1, in order to find documents in F that belong to all of them. In the same way, the logic operator OR is used to combine trained classifiers of all categories having cm_{ji} = 1, in order to find documents in F that belong to at least one of them. Finally, the logic operator NOT is used in conjunction with OR to combine trained classifiers of all categories having cm_{ji} = 1, in order to find documents in F that do not belong to any of them. As a result, we form 19 filters and obtain their relevant documents in F for every logical operator.

To train NB classifiers, a term-document matrix was created after removing the infrequent and the most commonly used English words [17]. We reduced dimensionality, by applying the Document Frequency (DF) method on the training set leaving the 300 most informative features [18]. We trained and validated one two–class NB classifier for every topic using the "ModApte" split, taking as positive examples all the samples that belong to the topic and as negative examples all the samples that do not. In that way, a document is assigned to a class when its probability of belonging to that class is larger than the probability of belonging to all other classes. In this work, however, we considered that d_i is categorized under category c_j when $\Pr(c_j|d_i)>h_j$ where h_j is a threshold selected to optimize classification accuracy on the filtering set F. In Table II, the classification accuracy of NB on the filtering/test set is displayed.

For the training of SVM, a term-document matrix was created after removing the infrequent and the most commonly used English words [17]. To reduce dimensionality, we applied Principal Component Analysis (PCA) on the training set leaving the 300 most informative features. PCA lead to better classification results than DF on the test set. We trained and validated one two–class SVM classifier for every topic using the "ModApte" split, taking as positive examples all the samples that belong to the topic and as negative examples all the samples that do not. To obtain the best possible classification accuracy we optimized the hyperparameters of non-linear SVM on the filtering set F. Table II shows the classification accuracy of SVM on the filtering/test set. The output of each SVM was transformed to probability using maximum likelihood, as described in Section II.

Operators are used to create filters that decide of the membership of a document to a conjunction, a disjunction or a negation of topics.

Table II Classification accuracy on the test/filtering set

Topic	SVM	NB	Samples in F
Earn	0.9812	0	4
Acq	0.9812	0.7	10
Money-fx	0.9531	0.9184	49
Crude	0.9624	0.9348	47
Grain	0.9484	0.9825	115
Trade	0.9635	1	13
Interest	0.9108	0.8626	44
Ship	0.9531	0.8409	46
Wheat	0.9014	0.9849	71
Corn	0.9624	0.9643	56
Average	0.9518	0.82	

For every logic operator, we summarize and compare the performance of Boolean and fuzzy filtering over all filters. In the case of Boolean filtering, every document d in F was assigned a crisp value {0,1} depending on whether it belongs to class c_j or not. For every filter, the related decisions were aggregated using Boolean logic. The classic notions of *precision* (Pr) and *recall* (Re) are used to measure classification effectiveness. For categorization into category c_j, let tp_j, fp_j, tn_j, fn_j be the true positive, false positive, true negative and false negative documents respectively. Precision is defined as the estimated probability that, if a random document d is categorized under category c_j, this decision is correct, that is:

$$\Pr = \frac{tp_j}{tp_j + fp_j} \tag{10}$$

Analogously, recall is defined as the estimated probability that, if a random document d should be categorized under category c_j, this decision is actually taken, that is:

$$\text{Re} = \frac{tp_j}{tp_j + fn_j} \tag{11}$$

Precision and recall are related in an inverse manner. High levels of precision can be achieved by keeping recall low and vice versa.

Because no ordering is available, Boolean filtering was evaluated by averaging recall and precision of a logical operator over all filters (Table III). On the contrary, fuzzy filtering provides ordering, so a standard recall-precision diagram of a logical operator can be constructed [19].

In all cases, fuzzy aggregation succeeded in improving retrieval performance (Fig. 1 and 2, Table III). In the case of OR and NOT operators the improvement was due to higher precision. This means that true positive documents are placed high in the ranked answer set. In the case of AND operators fuzzy aggregation managed to improve both recall and precision.

Table III Average recall and precision for all filters and operators

	SVM		*NB*	
Operator	Precision	Recall	Precision	Recall
AND	0.36	0.39	0.21	0.15
OR	0.93	0.86	0.96	0.54
NOT	0.75	0.80	0.75	0.99

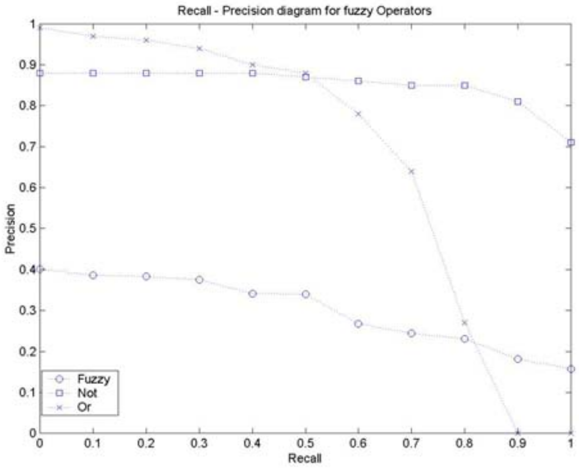

Fig. 1. Results for NB classifiers

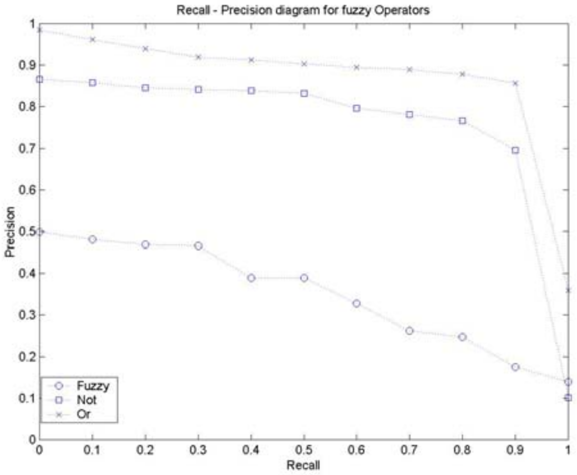

Fig. 2. Results for SVM classifier

V. AN EXAMPLE FILTERING SYSTEM

Generally, an Information Filtering system can be on the server or on the client side (Fig. 3). The proposed fuzzy filtering approach can be used both ways. In client sided filtering, the taxonomy may be in the form of user's bookmarks, for example. The system creates the topic classifiers that the user uses to filter the results of a search engine according to the described method. In server sided filtering, the taxonomy may be in the form of a web directory.

In order to provide an example application, we have developed a server sided filtering system on the web using the Open Directory Project (ODP) [20]. The system basically consists of an Html Wrapper (http://www.do.org/products/parser/) and a Taxonomy Builder located on a server (Fig. 4). In that way we can take advantage of powerful servers for the training of classifiers and the parsing of large quantities of html pages. Clients are in the form of html pages, applets and java applications that communicate with the server using servlets or RMI (Remote Method Invocation). We created NB classifiers for 10 topics related to the Computers/Artificial Intelligence directory of the ODP, using about 40 web pages as training examples for each topic. Through the interface (html client), the user is able to create queries and filters in order to retrieve and filter web pages. Queries are forwarded to ODP and the results are parsed and given topic probabilities according to the taxonomy. Finally, these probabilities are aggregated according to the filters and the results are fed back to the client.

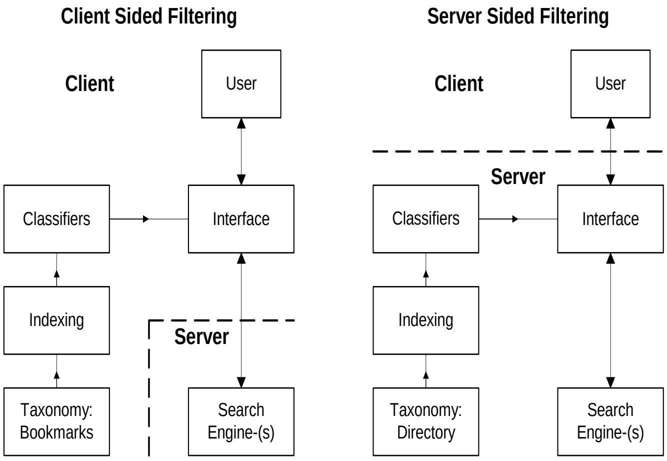

Fig. 3. Client vs server sided filtering systems

<table>
<tr><td colspan="3">CLIENTS (Html – Java Applications – Applets)</td></tr>
<tr><td colspan="3">SERVLETS – RMI</td></tr>
<tr><td>HTML WRAPPER</td><td>TAXONOMY BUILDER</td><td>QUERY-FILTER</td></tr>
<tr><td colspan="3">TCP/IP</td></tr>
<tr><td colspan="3">SEARCH ENGINES</td></tr>
</table>

Fig. 4. The architecture of the filtering system

VI. CONCLUSIONS

We have described and evaluated a framework that can take advantage of a topic taxonomy as part of a filtering language. We used SVM and NB classifiers on the Reuters corpus to create filters that decide of the membership of a document to a conjunction, disjunction or negation of topics. Fuzzy aggregation of the estimated topic probabilities proved to exhibit superior performance than Boolean aggregation for all kinds of filters. Finally, we deployed a filtering system based on this framework using a sample taxonomy of the Open Directory Project.

REFERENCES

[1] 1. B.U. Oztekin, G. Karypis and V. Kumar, "Expert Agreement and Content Based Reranking in Meta Search Environment using Mearf," *Proc. WWW2002*, Honolulu, May 2002.

[2] S. Vrettos and A. Stafylopatis, "A Fuzzy Rule-Based Agent for Web Retrieval – Filtering," *Proc. of First Asia-Pasific Conference on Web Intelligence* (*WI 2001)*, Maebashi City, Japan, October 2001, pp. 448-453.

[3] Y. Yang, "An evaluation of statistical approaches to text categorization, *Journal of Information retrieval*," vol. 1,1999, pp. 69-90.

[4] Y. Yang and X. Liu, "A re-examination of text categorization methods," *Proc. of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, 1999.

[5] H. Chen and S.T. Dumais, "Bringing order to the Web: Automatically categorizing search results," *Proc. of ACM SIGCHI Conference on Human Factors In Computing Systems(CHI)*, 2000, pp. 145-152.

[6] S. Dumais, E. Cutrell and H. Chen, “Optimizing Search by Showing Results In Context,”, *Proc. of SIGCHI*,Seattle, WA, USA,March 31,2001, pp. 69-90.

[7] C. Chekuri, M. Goldwasser, P. Raghavan and E. Upfal, “Web search using automated classification,” Proc. *of Sixth International World Wide Web Conference , Poster POS725*,Santa Clara, California,April 1997.

[8] S. Chakrabarti, B. Dom, R. Agrawal and P. Raghavan, “Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies,” *The VLDB Journal*, vol. 7, 1998, pp. 163-178.

[9] Vrettos, S. and Stafylopatis A, “A Fuzzy logic Framework for Web Page Filtering,” *Proc. 6th Seminar on Neural Networks Applications in Electrical Engineering NEUREL 2002*, September 2002, pp. 47-51.

[10] M.A. Hearst, “Improving Full-Text Precision on Short Queries using Simple Constraints,” *Proc. of SDAIR*,Las Vegas, NV, April,1996.

[11] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami, “Inductive learning algorithms and representations for text categorization,” *Proc. of ACM Conference on Information and Knowledge Management*, 1998, pp. 148-155.

[12] Vladimir Cherkassky and Filip Mulier, Learning from Data, (Jown Wiley & Sons, 1998).

[13] Simon Haykin, Neural Networks a Comprehensive Foundation (Prentice – Hall, 1999).

[14] OSU SVM Classifier Matlab Toolbox (ver 3.00) http://eewww.eng.ohio-state.edu/~maj/osu_svm/.

[15] John C. Platt, “Probabilities for SV Machines,” *Advances in Large Margin Classifiers*, (The MIT Press, Cambridge, 2000).

[16] The Reuters – 21578 collection: www.reasearch.att.com/~lewis/reuters21578.html

[17] 17. T. Joachims, “A probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization,” *Proc. of the 14th International Conference on Machine Learning ICML97*, 1997, pp. 143-151

[18] Yiming Yang and Jan O. Pedersen, “A Comparative Study on Feature Selection in Text Categorization”, *Proc. of ICML-97, 14th International Conference on Machine Learning*, 1997.

[19] Ricardo Baeza-Yates and Berthier Ribeiro – Neto, Modern Information Retrieval, (ACM Press Books, 1999).

[20] Open Directory Project. [http://dmoz.org/]