Feed Distillation Using AdaBoost and Topic Maps

Wai-Lung Lee, Andreas Lommatzsch, Christian Scheel
DAI-Labor
Technical University Berlin
Ernst-Reuter-Platz 7
Berlin, Germany
{wai-lung.lee, andreas.lommatzsch, christian.scheel}@dai-labor.de

February 7, 2008

Abstract

This paper retains the experiences by participating in TREC 2007 Blog Track 'Feed Distillation'. To perform the run various classifiers are combined, which analyze title-, content- and splog-specific features to predict the relevance of a feed related to a topic, based on the idea of AdaBoost. The implemented classifiers utilize keywords retrieved from different thesauri such as Wordnet and Wortschatz, as well as from websites providing hierarchical organized 'ontology' such as the 'Open Directory Project' and Yahoo Directory. To structure the keywords, Topic Maps are utilized according to ISO/IEC 13250:2000.

1 Introduction and Motivation

Nowadays blogs are the most used media to express individual's experiences, emotions and opinions related to a topic. Both the rapid growth[1] and the underlying influence of bloggers on different areas of our society and online community are fascinating [5, 14]. Commonly, the blogs are aggregated to a feed, which provides a clean, fine-grained structure and improves the machine readability. However, since bloggers not only compose blogs related to the same topic, it is time-consuming for interested readers to judge the topic-related relevance of a feed.

To face this problem, a novel approach using AdaBoost [16] and Topic Map [2] is utilized to estimate a feed's relevance concerning a certain topic.

According to the ISO/IEC 13250:2000, a Topic Map is used to represent knowledge and concurrent views related to a topic and contains main parts such as association, occurrence and topic. Topics represent subjects in the real world and are identified by names. Occurrences refer to relevant resources of a topic. The relations between the topics are depicted by associations [2].

AdaBoost is a learning algorithm and frequently used in classification issues. AdaBoost has been introduced by Freund and Schapire [16], and belongs to ensemble-based systems in decision making [40]. The idea of AdaBoost is to build a strong learner using a set of weak learners, which individually are slightly better than a random guessing algorithm [16]. As previous investigations have shown that boosting algorithms commonly outperformed other learners such as SVM (Support Vector Machine) [35] or neural networks [55], AdaBoost is combined with Topic Maps implementing a set of 16 weak classifiers analyzing title, content and spam blog (splog)[2] specific features of blogs within a feed. The work in this paper is also inspired by investigation efforts in regard with 'data fusion' and analyses of multiple evidence combination [24]. Some efforts reveal that combining dif-

[1] http://www.sifry.com/alerts/archives/000436.html

[2] http://en.wikipedia.org/wiki/Spam_blog

ferent sources of evidence or strategies can improve retrieval performance. In this context, Lee additionally figures out that rank works better than using similarity in some circumstances [24]. Moreover, the analysis of Scheel et. al. on detecting redundant and avoiding strategies, can also improve retrieval effectiveness. They suggest that strategies have to be diverse as possible while maximizing their individual quality [49]. In other words, strategies complementing one another can leverage the retrieval results.

2 Overview of the Paper

The remaining part of this paper is structured as follows. Related work is referred in Section 3. Section 4 points out the disadvantages and advantages of AdaBoost by comparing neural networks and SVMs relating to performance and evaluation issues, followed by Section 5 depicting the conceptual framework for participating in this annual TREC Blog Track 'Feed Distillation'. In doing so, the basic assumptions using AdaBoost and Topic Maps as well as diversity measures between the implemented classifiers are presented. In Section 6, the implemented architecture is briefly overviewed. In Section 7, some training and experimental settings are highlighted. Section 8 deals with the evaluation results judged by the participating groups and reflects the proposed approach for reperforming an inoffiicial blog track run. This paper concludes with a discussion and an outlook on further research and performance improvement.

3 Related Work

Blogs have different facets related to a topic, analyzed by [33]. In this context, Paradis already reveals that the notion of topic includes three aspects from the view of linguistic and discourse representation theories: theme (statutory analysis), given information (topic-focus articulation), and intentions (discourse representation) [36]. In doing so, Paradis suggests that using linguistic and discourse structures to derive topics can improve the subjective relevance of documents towards user's information need. Indeed, selecting the right amount of topic-related keywords and concepts is a great challenge and many mining techniques exist. While some mining and retrieval techniques refer to proven remedy such as identification of finite mixtures, latent semantic indexing, independent component analysis [3]. Other researchers devote to topic identification from external resources [58, 52]. These approaches base on query-based expansion [56] theory, and consider lexical-semantic meanings of certain topics using different thesauri [29] or hierarchical structured ontology provided by 'Open Project Directory'. Additionally, relating to the blog sphere there is a modern approach to mining theme patterns, which incorporate spatiotemporal properties of blogs to detect topics within the blog sphere [31]. However, the present approach in regard with topical features extraction is aligned to the topic retrieval from external resources.

Determining topical relevance of feeds is something new to research in terms of classification and ranking problems. Previous work refers to topic distillation of authoritative web sites. These techniques base on the hyperlink analysis, and use DOM (Document Object Models) [7] or exploit the correlation between the outgoing and incoming hyperlinks to predict the topical relevance of documents. [39]. A more modern approach, patent-registered by Google, assesses the topical relevance of blog using author's reputation score, which is secured by digital signature system of Google[3]. Apart from these, boosting algorithms on text classification have been proved as excellent ranking and categorization techniques, e.g. RankBoost [21] and AdaBoost.MHKR [50]. Since SVMs result an excellent precision but poor recall, boosting SVMs is recently a very active area of machine learning research [51].

4 Preliminaries

In this section, similarities and distinctions between commonly used learning algorithms are briefly overviewed. In the research literature, many comparisons exist between SVM and AdaBoost or AdaBoost and Feed-forward Neural Network (FNN). However,

[3] http://searchengineland.com/070209-164512.php

the most comparative study on the three learning algorithms is conducted by [46]. Inspired by their empirical work, using AdaBoost as appropriate learning algorithm is justified in the context of classification issue in the following sections.

For those who are familiar SVM, FNN and AdaBoost, it may be advisable to continue reading in Section 5

4.1 Comparison of Feed-forward Neural Network, SVM and AdaBoost

Nowadays, learning algorithms such as AdaBoost, Support Vector Machine (SVM) and Feed-forward Neural Network (FNN) have recently attracted popularity in different domains such as handwritten character recognition, face detection, and especially in text classification [13, 51].

Commonly, all these algorithms are used to learn boundaries between positive and negative examples. The functions of the learners for determining the decision boundaries are different from the view of geometrical and mathematical basics. Principally, they all base on the marginal-theory [15] to gather a suitable function which can maximize the margin for separating classes by avoiding the overfitting problem. In this regard, the measures such as generalization problem, error rate, the size of training data sets are most used concepts to evaluate the different performance output of such learning and classification systems [34]. To obtain a better understanding for operating modi of these learning algorithms, literatures and researches in the domain of machine learning are recommended [55, 32].

4.1.1 AdaBoost

The idea of AdaBoost is to produce a highly accurate classification rule by combining a set of classifiers (or weak hypotheses), each of which may be only moderately inaccurate [15, 48]. In the context of learning process, the weak learners are trained sequentially, one at a time. Principally, at each iteration a weak classifier is inaccurate to classify the examples, which were most difficult to classify by the previous weak hypotheses. After a certain number of iterations, the resulting weak hypotheses are linearly combined into a single prediction rule, so-called combined hypothesis.

The generalized AdaBoost algorithm for binary classification [48] maintains a vector of weights as a distribution D_t over training data set. At round t, the objective of AdaBoost is to estimate a weak hypothesis $h_t : X \rightarrow R$ with moderately low error in regard with to the weights D_t. In this setting, weak hypotheses $h_t(x)$ make real-valued confidence-rated predictions [46]. Initially, the distribution D_t is uniformly initialized, which means that all instances are equally weighted. The boosting algorithm increases (or decreases) the weights $D_t(i)$ when $h_t(x_i)$ making a bad (or good) prediction of instances, with a variation proportional to the confidence $h_t(x_i)$. The final hypothesis, $f_T : X \rightarrow R$, calculates its prediction using a weighted vote of the weak hypothesis $f_T(x) = \sum_{j=1}^{T} \alpha_j h_j(x)$. The updating rule can be expressed as

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i) * e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \\ &= \frac{e^{-\sum_{j=1}^{t} \alpha_j y_i h_j(x_i)}}{M * \prod_{j=1}^{t} Z_j} \\ &= \frac{e^{y_i f_t(x_i)}}{M * \prod_{j=1}^{t} Z_j} \\ &= \frac{e^{-mrg(x_i, y_i, f_t)}}{M * \prod_{j=1}^{t} Z_j} \end{aligned} \quad (1)$$

Schapire and Singer [48] already prove that the training error of the AdaBoost algorithm exponentially decreases with the normalization factor Z_t computed at round t. This property is utilized in designing of the weak learner, which attempts to figure out a weak hypothesis h_t that minimizes $Z_t = \sum_{i=1}^{M} D_t(i) * exp^{(-\alpha_t y_i h_t(x_i))}$.

From 1 and the previous expression of Z_t, it can be interpreted that AdaBoost is a stage-wise procedure for minimizing a certain error function, which depends on the functional margin $-mrg(x_i, y_i, f)$. Particularly, AdaBoost attempts to minimize $\sum_i exp^{(-y_i \sum_t \alpha_t h_t(x_i))}$, which illustrates

the negative exponential of the margin of the final classifier. According to Schapire and Singer, the learning bias of Adaboost is proven to be very aggressive at maximizing the margin of the training examples and this makes a clear connection to the SVM learning paradigm [48].

4.1.2 Neural Network

One of well-known Neural Networks is the FNN, so-called Feed-forward Neural Network. The architecture is organized by layers of units, with connections between units from different layers in forward direction [4]. A fully connected FNN with one output unit and one hidden layer of N_h utilizes the computation function:

$$f_{FNN}(x) \quad = \quad \varphi_0 \left(\sum_{i=1}^{N_h} \lambda_i \varphi_i(\omega_i, b_i, x) + b_0 \right), \quad (2)$$

where $\lambda_i, b_i, b_0 \in \Re$ and $x, \omega \in \Re_N$. To simplify, the weights can be separated in $coefficients$ $(\varphi_i)_{i=1}^{N_h}$, $frequencies$ $(\omega_{i=1}^{N_h})$ and $biases$ $(b_i)_{i=0}^{N_h}$. The most used activation functions $\varphi_i(\omega, b, x)$ in the hidden units are sigmoidal, in particular, for Multi-layer Perceptrons (MLP) and radially symmetric for Radial Basis Function Networks (RBFN). Of course, there are many other functions [37, 19]. However, output activation functions $\varphi_0(u)$ are commonly sigmoidal or linear.

The objective of the training process is to determine adequate parameters such as coefficients, frequencies and biases for minimizing a pre-estimated cost function. The most usual sum-of-squares error function is illustrated as follows:

$$E(X) = \sum_{i=1}^{L} \frac{1}{2} (f_{FNN}^{c}(x_i) - y_i)^2. \quad (3)$$

The sum-of-squares error function $E(X)$, illustrated as (3), is an approximation to the squared norm of the error function $f_{FNN}(x) - y(x)$ in the Hilbert space L^2 of squared integrable functions, where the integral refers to probability measure of the problem by X. For C-class problems, architectures with C output units are utilized [46], and the goal is a transform minimizing

$$E(X) = \sum_{i=1}^{L} \sum_{c=1}^{C} \frac{1}{2} (f_{FNN}^{c}(x_i) - y_i^c)^2, \quad (4)$$

where f_{FNN}^{c} is c_{th} component of the output function. The architecture of the network related to the connections, numbers of hidden units and output activation functions is commonly fixed in advance, whereas the weights are trained during the learning procedure.

4.1.3 SVM

As SVM is described by [55, 9, 46], the input vectors of SVM are mapped into a high-dimensional space (inner product) through non-linear mapping ϕ, which is chosen in advance. In this space, the so-called feature space, an optimal hyperplane is designed. The mapping using a kernel function $K(u, v)$ can be implicit, since the inner product expressing the hyperplane can be defined as $\langle \phi(u), \phi(v) \rangle = K(u, v)$ for every two vectors $u, v \in \Re^N$. In the context of the SVM framework, an optimal hyperplane can be described a maximal normalized margin for separating data set. The functional margin of a point (x_i, y_i) with regard to a a data set X is the minimum of the margins of the points in the data set. If f is hyperplane, the normalized (or geometric) margin can be considered as the margin divided by the norm of the orthogonal vector to hyperplane. Thus, the absolute value of the geometric margin is the distance to the hyperplane. Based on Lagrangian and Kuhn-Tucker theory, the maximal margin hyperplane for a binary classification problem given by X can be expressed as:

$$f_{SVM}(x) = \sum_{i=1}^{M} y_i \alpha_i K(x_i, y_i) + b, \quad (5)$$

where the vector $(\alpha)_{i=1}^{M}$ is the solution of the following constrained optimalization problem in the dual space:

$$
\begin{aligned}
Maximize\ W(X) \quad &= \\
-\frac{1}{2}\sum_{i,j=1}^{M} y_i\alpha_i\alpha_j K(x_i,y_j) &+ \sum_{i=1}^{M}\alpha_i \\
subject\ to\ \sum_{i=1}^{M} y_i\alpha_i = 0\ (bias\ constraints), &\qquad (6) \\
0 \le \alpha_i \le C, i = 1,...,M. &
\end{aligned}
$$

To avoid the bias constraints, b is attended apart and fixed a priori in some implementations. A point is well classified if and only if its margin with regard to f_{SVM} is positive signed. The points x_i with $\alpha_i \leq 0$ (active constraints) are support vectors. Relating to their margin value, non-bounded support vectors have margin 1. Conversely, the margin of bounded support vectors are less than 1. The parameter C is utilized to trade off the margin and the number of training errors. One obtains the hard margin hyperplane when setting $C = \infty$. However, the cost function $-W(X)$ including a constant is the squared norm of the error function $f_{SVM} - y(x)$ in the Reproducing Kernel Hilbert Space, which is associated to $K(u, v)$ [9, 46]. The most usual kernel functions $K(u, v)$ are polynomial, Gaussian-like or some particular sigmoids.

4.1.4 AdaBoost vs. SVM

SVM has been emerged as a good technique in classification and categorization issues [13, 51]. SVM provides a good upper bound to generalization of error [55, 13]. While some SVMs achieve an excellent precision, the recall is poor when applying SVMs in text classification. However, SVM algorithms focus on finding the hyperplane as kernel function for maximizing the decision boundary [51]. Since the training kernel matrix grows quadratically, training SVM on a large data set is resource-consuming. In this context, there are improvements relating to this problem [13]. But to prove high effectiveness, SVMs strictly require a large training data set, which does not always exist. In contrast to this, AdaBoost can be used to address this problem using resampling techniques and small sets of training data [40]. Additionally, SVMs are initially designed for binary classification problems, whereas AdaBoost.M1, an extension of AdaBoost, can tackle multiclass problems [16]. In the context of 'Feed Distillation', the proposed approach has to deal with different topic classes and spam-infected blogs within a feed. This problem is addressed in Section 5.1.

From the mathematical perspective, SVM and AdaBoost are different regarding the approach searching the dimensional space and the underlying computation. While SVM refers to quadratic programming, AdaBoost corresponds only to linear programming. Of course, quadratic programming is more computationally consuming than linear one. In doing so, SVM has to deal with the estimation problem of kernel function allowing low dimensional calculations, that are mathematically equivalent to inner products in a high dimensional feature space [16, 55]. More details about the relation between AdaBoost and SVM can be found in [45].

4.1.5 AdaBoost vs. Neural Networks

However, the root of both neural network and AdaBoost can be found in the probably approximately correct (PAC) model, which was introduced by [54]. Neural networks, once introduced by [55], can tackle multiclass and multi-labeled problems as AdaBoost.M1 does. The major disadvantages of neural networks are related to the generalization performance from training to test data, determining the architecture of layers, as well as overfitting problem [32]. In contrast to this, some researchers have empirically observed that AdaBoost does not overfit, even when running for thousands of rounds [16]. However, the fascinating generalization properties outperformed neural network relating to this issue [34]. Additionally, from the computational complexity, it is time-consuming to train neural networks. Conversely, AdaBoost is simple to implement and a stagewise procedure for minimizing a certain error function by focusing on misclassified training items [46].

4.2 AdaBoost.M1

In this subsection, a pseudo-code of AdaBoost.M1 is illustrated in Algorithm 1. AdaBoost.M1 is an extension of AdaBoost and highly robust against multiclass and regression problems [15, 41].

Algorithm 1 A Pseudo-code of AdaBoost.M1

Input:
- Sequence of N examples $S = [(X_i, Y_i)]$,
$i = 1, ..., N$ with labels $y_i \in \Omega, \Omega = \omega_1, ..., \omega_c$
- Weak learning algorithm WeakLearn (WL)
- Integer T specifying number of learning iterations

Intialize $D_t(i) = \frac{1}{N}, i = 1, \cdots, N$
for $t = 1, 2, \cdots, T$: **do**
1. Select a training data subset S_t, receive hypothesis h_t drawn from the distribution D_t.
2. Train WL with S_t, receive hypothesis h_t.
3. Calculate the error of h_t :
$\epsilon_t = \sum_{i:h_t(x_i)=y_i} D_t(i) \cdot$ If $\epsilon_t > \frac{1}{2}$ abort
4. Set $\beta_i = \frac{\epsilon_t}{(1-\epsilon_t)}$.
5. Update distribution D_t :

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t \text{ if } h_t(x_i) = y_i, \\ 1, \text{Otherwise} \end{cases}$$

where $Z_t = \sum i D_t(i)$ is a normalization constant chosen so that D_{t+1} becomes a proper distribution function.
end for

Test - Weighted Majority Voting: Given an unlabeled instance x_i
1. Obtain total vote received by each weak learner $V_j = \sum_{t:h_t(x)=\omega_j} \log \frac{1}{\beta_t}, j = 1, \cdots, C$.
2. Choose the class that receives the highest total vote as the final classification.

5 Conceptual Framework

In this section, problems are precisely depicted when performing the Blog track task. In doing so, the assumptions relating to the suitability of AdaBoost and Topic Maps are presented. Particularly, the keyword aspects and the implemented AdaBoost algorithm is described by depicting the applied classifiers and its operation modus. Additionally, the dis-similarities between the keyword scources and the diversity of implmented classifiers are measured. This section concludes with the results related to diversity measures and an initial recommendation for designing the set of classifiers.

5.1 Problem Definition and Basic Assumption

This annual Blog Track task is defined as 'Find me a blog with a principle, recurring interest in X'. As some research efforts show that there is not one universally appropriate definition of relevance, it is important to deal with a definition of topic-related relevance of a feed. Generally, there is a distinction between objective and subjective relevance [18, 1]. Objective relevance can be achieved if a document covers the notion of topicality and aboutness, defined by [30]. But this notion is not the only the crucial factor that contributes to the usefulness of a document [30]. In this context, Cooper and Bookstein reveal that a document is objectively relevant to a query if they both are related to a common topic. Subjective relevance is more aligned with user's information need. A document is subjectively relevant to a query if it covers the information need of the user who issued the query [8, 6]. However, in regard with the TREC Blog Track task, the subjective kind of relevance of a feed is taken into account. Thus, a feed is relevant if this is principally devoted to a topic. In other words, a feed is relevant if this consists of a majority of blog entries which predominately deal with a topic.

The major problem is not only to identify various facets [11] of a topic within a blog and the underlying topical relevance, but also to determine the quantity of topic-related blog entries within a feed. Since bloggers compose blogs related to different topics or seasonal themes, the variety of topical facets is increasingly intensified within a feed. Thus, a feed can be interpreted as a multiclass problem. An additional burden, as the spam detection task of TREC

2006 figures out, is that the TREC blog collection is 'infected' by splogs [27]. In doing so, this problem impedes the topical distillation procedure and has to be considered in the proposed approach. To tackle the problem mentioned above, different hypothesis are combined to predict the topical relevance of feeds by removing splogs simultaneously.

The general idea combining different hypotheses is to predict the relevance of feeds related to a topic. Intuitively, such a hypothesis can be interpreted as an expert related to the certain topic, which can be synonymously called classifier. The goal of using diverse experts is to obtain a high degree of objective relevance by combining the subjective view of each expert. Since the xml-based structure of a feed is fine-grained and provide information about the title, content, etc., these classifiers can make decision based on investigating these features, whether a feed is relevant to the certain topic. However, the approach consists of two parts. First, topic-related keywords are necessary to classify the blogs within a feed. Therefore, the research concerning query expansion [56] and extracting keywords from different keyword sources [29] are followed. In doing so, research of 'data fusion' and evidences combination are referred to cover the concurrent views of relevance and facets of a blog related to a topic. Additionally, to structure these keywords the concept of Topic Map is utilized. Secondly, the importance or weight has to be estimated for each classifier. Since a manual estimation of weights is laborious, AdaBoost is an appropriate learning algorithm to automatically determine such weights.

5.2 Topic Maps and Keyword Aspects

The genesis of Topic Map can be found in the 1990's, developed by Davenport Group, which was discussing ways of interchange of computer documentation [38]. In the past, Davenport Group developed DocBook DTD[4], which is one of most widely used DTDs for authoring SGML[5] and XML[6]. Indeed, using Topic Maps to organize, visualize and navigate knowledge and metadata is not a novel approach, but has recently attracted the interest of some researchers and practitioners [28].

Nowadays, there are different variations, standard formats and query languages alongside the topic map research landscape. In doing so, they are used in different research areas such as E-learning environment, Knowledge Management, Artificial Intelligence, etc [44]. However, the main focus using Topic Map is to incorporate the different facets which are associated to the certain topic. The most important advantage of Topic Map lies in the generalization of this model, that one can describe and structure everything related to a topic and unify all into a single model. In this context, the associations are used to connect to the collections of named entities and concepts retrieved by the different thesauri and websites, as illustrated in Figure 1.

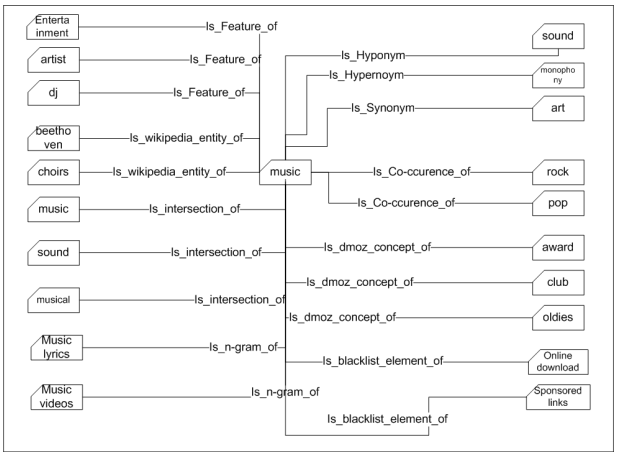

Figure 1: An example of Topic Map for the topic 'music'

To gather topic-related keywords in a Topic Map, the topic word is remitted as a query to existing data sources such as Wikipedia, Dmoz, Yahoo, Google, Wortschatz and Wordnet. In doing so, available APIs (Application Programming Interfaces) of these sources are applied.

[4]http://www.oasis-open.org/docbook/
[5]http://www.w3.org/MarkUp/SGML/
[6]http://www.w3.org/XML/

5.2.1 Wordnet

Wordnet[7] is a well-known resource for lexical and semantic keywords. In this context, terms considering hypernyms, hyponyms and synonyms are extracted.

5.2.2 Wortschatz

Wortschatz[8] is a German online webservice providing access to a large set of corpora in different languages. Additionally, this webservice offers statistical information about co-occurrences related to a topic [43].

5.2.3 Google Suggest

Google suggest[9] provides n-grams, based on most searched query phrases related to a topic. Frequently, this API offers 2-, 3-, 4- or 5-grams with information in terms of query frequencies associated with the keywords combination.

5.2.4 Yahoo

By querying **Yahoo**[10] every topic is combined with the word 'shop' to gather product- or topic-related features from the perspective of shop-providers. In doing so, TF-IDF to select those terms is utilized, which are used by twenty shop sites offering topic-related products. In this regard, the Yahoo API is utilized to extract additional n-grams. Yahoo n-gram API varies from 'Google suggest' in the missing information about the frequencies.

5.2.5 Open Directory Project

Dmoz[11], so-called 'Open Directory Project', is a human-edited and -maintained category-based search engine. Since Dmoz provides a hierarchical organized 'ontology', keywords of hierarchical related categories are used, e.g. relating to 'Solaris' category labels within the categories 'administration' and 'software' are retrieved.

5.2.6 Wikipedia

The terms of the topic are adressed as query to **Wikipedia**[12] for matching site titles via its underlying API. In this context, links related to categories or themes concerning the certain topic are retrieved. The problem is sometimes that there is no title which can be precisely matched by the given query word(s). In this case, the first hit of this query is taken.

5.2.7 Intersection

Apart from the black-listed keywords, a Topic Map also contains an intersection of all keywords, which occur in all the data sources mentioned above. In doing so, these keywords are removed from the remaining classifiers to avoid redundancy.

5.2.8 Blacklist

In addition to the (semi-)automatic retrieved keywords, a hard-coded blacklist of spam-specific and query-independent keywords is deployed, which frequently are used for spam blogs. In doing so, the keywords are weighted with a numerical scoring system, as illustrated in Figure 2

```
buy cheap#10, buy sell#10, compare prices#10, view details#10, sponsored links#10,
sponsored results#10, instant access#10, save money#10, online download#5, adult
video#5, listings#3, free#3, price#3, rating#3, register#3, subscribe#3, account#2,
sign#5, specials#2, sponsored#3, store#3, bid#3, http#3, xxx#3, teen#3, girl#2,
purchase#3, www#5, com#5, contact#3, trial#3, saving#3, attorney#4
```

Figure 2: A query-indepedent blacklist for removing splogs

5.3 AdaBoost Classifiers

In this section, the implemented classifiers and their purpose for completing the 'Feed Distillation' task are described.

Based on the idea of ensemble-based systems, diverse AdaBoost classifiers are sequentially developed,

[7] http://wordnet.princeton.edu/
[8] http://wortschatz.uni-leipzig.de/
[9] http://google.com/complete/search?output=toolbar
[10] http://www.yahoo.com/
[11] http://www.dmoz.org/
[12] http://en.wikipedia.org/w/api.php

at one time. Diversity, more detailed in Section 5.4.2, is an essential requirement by developing ensemble-based systems. In doing so, the order, in which the classifiers are trained, has impacts on the weights to estimate. However, these classifiers can be simple or complex ones and coarsely categorized in title-, content-based and splog-specific classifiers. As optimalization tricks for blogs reveal, that a blog with key phrase including topic word(s) in the title can better placed in the search engine hits, a simple classifier has to be able, e.g. to analyze the existence or absence of the topic word in the title. In contrast to a simple classifier, a complex classifier is based on topic-related keywords and exceeding a threshold to predict the relevance of a blog. If the similarity score between the keywords and the blog entry exceeds a threshold, the analyzing blog document is predicted as relevant by the classifier of these keywords.

Since the topical similarity has to be estimated based on a committee of different classifiers by removing splogs simultaneously, the AdaBoost.M1 scoring is modified. Originally, the relevance estimation of AdaBoost.M1 is based on the linear combination of hypotheses which are positive signed. In the case of feed distillation, the splog-specific and title- and content-based classifiers are different. While the splog-related classifiers are negative signed, content- and title-based ones are positive signed. Thus, estimating topical relevance is based on the sum of negative and positive weighted classifiers. In the following subsections, each implemented classifiers based on title-, content-, as well as splogs-specific retrieval strategies are depicted.

5.3.1 Title-based classifiers

The IntersectionTitleFilter contains the intersection mentioned above and is used to check the existence of those keywords in the blog's title. Moreover, the assumption is that the intersection is a minimal evidence for the relevance of a blog relating to a topic.

The CategoryTitleFilter predicts the blog's relevance analyzing the occurrence of the topic word(s) in the title of a blog.

5.3.2 Content-based classfiers

The IntersectionBodyFilter is applied to the content of a blog and determines the blog's topical relevance depending on the existence of all intersection keywords.

The CategoryBodyFilter works in the same manner as CategoryTitleFilter does and refers to the content of a blog.

The RelevanceWikipediaFilter is threshold-based classifier and contains a set of keywords retrieved from links. This classifier is applied on the content of the blog by analyzing the similarity between the blog's content and topic-related keywords retrieved by Wikipedia. Since both Weblog and Wikipedia are emerging technologies of Web 2.0, one can assume that there are similar relations between the keywords in both media.

RelevanceDmoz1Filter, RelevanceDmoz2Filter, and RelevanceDmoz3Filter are also threshold-based classifiers. Since these classifiers are based on category labels, the intention is to observe the similarity between the blog's content and hierarchical organized 'ontology' related keywords which are biased by human-beings.

The RelevanceWordnetFilter is also a threshold-based classifier and consists of synonyms, hypernyms, hyponyms, etc. retrieved by Wordnet. The idea using these keywords to consider topical facets of blog from the view of lexical aspects, especially, to bridge the gap of concept problems, e.g. if various name entities refer to the same concept.

The RelevanceWortschatzFilter is also a threshold-based classifier and works with co-occurrences retrieved by the webservice of Wortschatz. In doing so, the topical facets can be covered from the statistical point of view.

The RelevanceYahooFilter also uses a threshold to separate classes and contains keywords retrieved by Yahoo. Particularly, features are interesting from the commercial-intended view of properties relating to a topic, e.g. concerning the topic 'iPod' features such as battery, cables and other accessory items are expected.

5.3.3 Splog-specific classifiers

SimilarityPatternFilter. Previous work on detecting splogs has shown [26], that spam blogs frequently have similar structures. In doing so, the link-based and temporal aspects are discarded. Based on the basic idea, this classifier can predict a blog non-relevant if the similarity score between title and content does not exceed the threshold.

The NGramTitleFilter scrutinizes the occurrence of one of Google Suggest's keywords and is used for splog detecting.

NGramBodyFilter. Historically, commercial websites use most frequently searched phrases of search engines to be better ranked by the search results. Based on this fact, the commercial-intended bloggers can also exploit this knowledge to better place their blogs in the search hits. Thus, the NGramBodyFilter classifier judges a blog as non-relevant when exceeding the threshold by the similarity score between the n-grams and the blog's content. On the one hand, this classifier can be used for removing splogs. On the other hand, it can also be applied for the content-based retrieval. For instance, while 'mobile phone ring tones' as 4-gram relating to the topic 'mobile phone' is more commercial-intended, 'machine learning approaches' are highly topic-specific related to the topic 'machine learning'. However, first one is utilized.

The BlacklistTitleFilter is a simple classifier that has a set of hard-coded features to detect splogs, and analyzes the title on splog-related keywords.

The BlackBodyFilter is similar to BlacklistTitleFilter, but is threshold-based and complex one. As mentioned in Section 5.2, the blacklist utilizes the numerical points as a weighting system. If the threshold is exceeded by respective sum of points, the classifier assesses the analyzing blog as non-relevant.

5.4 Diversity of Classifers and Dissimilarity of Keyword Sources

The present section deals with diversity measures for designing a ensemble of classifiers and prompts the question, whether dis-similarity of the used key sources is crucial factor or whether diversity measure of classifiers is final criteria for improving the retrieval performance. Based on the extensive findings of Kunchewa and Whitaker [23], a general recommendation is derived for specifying the present implementation with regard to learning procedure and applying parameters. Since the architecture has been already implemented when running the TREC track task, this section provides information which is attended after submitting the results to TREC. Thus, the architecture and its underlying procedure is reflected in Section 8.2.1.

5.4.1 Dis-similarity of the Keywords References

This subsection is devoted to an overview of the mutual dis-similarities between the lists structured in the Topic Maps. The objective of this overview serves to underline the desirable effects of the keyword references in terms of complementing each another. To do so, the cosinus-similarity is mutually measured between all lists pairwise within the Topic Map for a certain topic. However, the similarity of a list itself is not considered. Also the hand-coded and intersection are not taken into account. To compare the similarities between lists over all topics suggested in TREC, a standardization is utilized as follows:

$$sim_{t_i} = \frac{1}{T}\frac{1}{N-1}\sum_{t\in T}\sum_{j\in L\setminus\{i\}} sim(l_i, l_j), \qquad (7)$$

where N is number of set of lists L within a Topic Map, T is number of topics suggested in TREC, l with indices i and j are elements of set L.

As illustrated in Figure 3, the average mutual similarities are relative low. Also there is not much of difference between the similarities with and without Porter's stemming algorithm (see Section 6.1 for more details). Since the keyword resources are dis-similar to each other, one could conjecture that these keywords from various sources could complement each another. However, that illustration does not finally testify to leveraging the retrieval effectiveness and requires keywords that are adequate topical relevant to match the various facets of a blogs and to predict

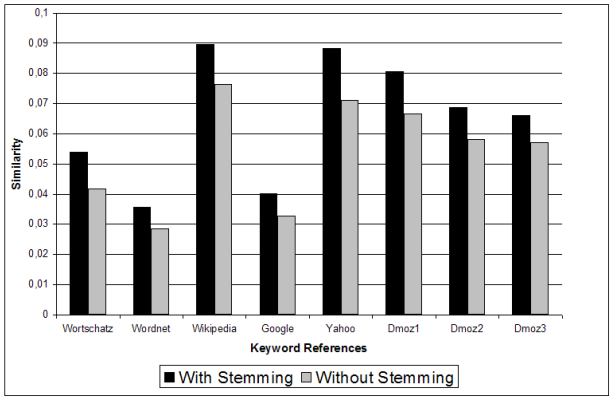

Figure 3: An Overview of Average Mutual Similarities between the Keyword References for All Topics Suggested in TREC

the relevance of a feed. In this context, the diversity required by ensemble systems cannot be justified yet, based on the computed dis-similarities illustrated above. The dis-similarity can be interpreted as dependency between keyword sources. To address the question whether the dis-similarity between the keyword sources is a crucial factor for the retrieval effectiveness and classification performance, measuring the output performances by adaptive addition of classifiers can be helpful. Additionally, it is also interesting to find out, how the diversity differs from a set of dependent and independent classifiers. However, these questions can be answered with regard to the diversity of AdaBoost classifiers in the following subsections.

5.4.2 Diversity of the AdaBoost Classifiers

Diversity has been recognized as a very important characteristic in the research area of combining classifiers [10, 47, 21]. In this context, the key success of ensemble systems, as AdaBoost belongs to, is to build a set of diverse classifiers. However, as the study of Kuncheva and Whitaker reveals [23], there is no strict separation between diversity, dependence, orthogonality or complementarity of classifiers, the focus of this subsection is to deal with some measures of diversity. Since many metrics exist, which can be categorized in pair-wise and non-pairwise ones, the present goal is not to extensively describe and compare all of these measures. Even if Kuncheva concludes, that there is no diversity measures that consistently correlates with higher accuracy, Entropy of the Votes and Q-statistic can be used for reflecting and arranging the present approach and system.

First of all, as many authors use the concept of diversity in terms of correct/incorrect (oracle) outputs, a denotation for the most simple pair-wise measure such as Q-statistic is utilized and illustrated in Table 1.

	h_i is correct	h_i is incorrect
h_j is correct	a	b
h_j is incorrect	c	d

Table 1: Denotation for Pairwise Measuring Diversity of Classifiers

For T classifiers, $T(T-1)/2$ pairwise diversity measures can be computed and an overall diversity of a set of classifiers can be calculated by averaging these pairwise measures. Based on given hypotheses h_i and h_j, Q-statistic can be expressed as

$$Q_{i,j} = (ad - bc)/(ad + bc) \tag{8}$$

Q results positive (negative) values if the same instances are correctly (incorrectly) classified by both classifiers. Maximum diversity is obtained if Q is equal to 0.

Conversely to pairwise measuring, Entropy of the Votes (Entropy Measure) assumes that the diversity is highest if half of the classifiers are correct, and the remaining ones are incorrect. This measure is defined as

$$E = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T - \lceil T/2 \rceil} min\{(\zeta_i, (T - \zeta_i)\}, \tag{9}$$

where $\lceil \cdots \rceil$ is ceiling operator, N is the dataset cardinality, ζ is number of classifiers, which incorrectly classifies unlabeled instance x_i. The Entropy Measure varies between 0 and 1, where 0 indicates

no difference between the classifiers and highest diversity in the team of classifiers.

5.4.3 Measuring Diversity of Implemented Classifiers

In this section, both the diversity of a set of independent and dependent classifiers are separately measured. To do so, this subsection hat its focus on the comparison and initial recommendation for redesigning of the present setting in terms of applying the learning parameters of the classifiers and its underlying set of keywords. To declare the meaning of dependency and independency of classifiers, a set of dependent classifiers has low similarity between the sources of keywords, as illustrated in Figure 3. Otherwise they are independent and the order, in which they are reduced can be found in subsection 6.6. To do so, classifiers based on the sources such as Intersections, Blacklist and Similarity are not taken into account, since they are query-independent and have similar settings in terms of F_1-measure or precision, etc. for all topics suggested by TREC's participating groups.

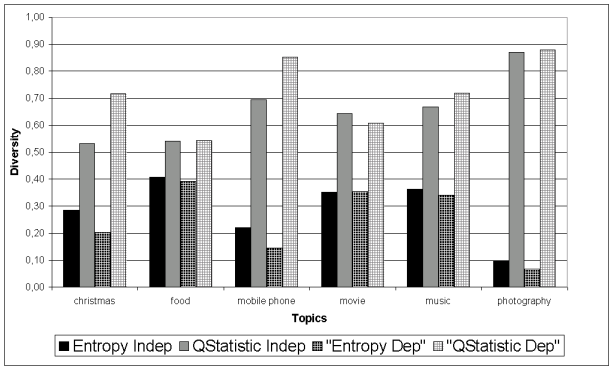

Figure 4: Diversity Measuring set of independent and dependent Classifiers

As illustrated in Figure 4, there are remarkable differences between both the non-pair wise and pair wise measures, which should be more individually detailed as follows.

First, relating to the dependent and independent set of classifiers, both measures are relative inconsistent in terms of diversity. While the Q-statistic refers to that dependent classifiers are more diverse than independent ones on an average, the Entropy disagrees with Q-statistic in this issue. However, the common agreement of both diversity measures is that the differences between independent and dependent set of classifiers are proportional relative similar over the six topics when not considering the algebraic sign.

Second, both the diversity measures reveal very contradictory results. While the Q-statistic arguments that the weak learner are low diverse and they incorrectly and correctly classify the same instances on an average, the Entropy Measure reveals that the classifiers are relative diverse and half of those are incorrect and correct when considering the value 0 as criteria of highest diversity in both cases. In particular, the extreme values can be found in the topics 'mobile phone' and 'photography'. The result returned for 'photography' can be biased by characteristics related to photography. On the one hand, since photography related blogs capture both blogs describing the techniques and methods of photography and photos with low content. On the other hand, the sources using Dmoz's categories are manually chosen and the resulting keywords are not retrieved by query using the terms contained in the topic's name entity, as described in Subsection 5.2.5.

Relating to the topic 'mobile phone', another assumption is taken up. Since the topic 'mobile phone' consists of two self-contained terms, which commonly can be independently used and refer to concepts within a concept group. However, they not only have a great intersection of common keywords, but also use separate terms for querying the feed collections, which results relative highly diverse segments of blogs. For instance, while querying 'phone' results blogs related to address- and phonebook resources, using 'mobile' to query the blog collections returns blogs relating to interest areas such as 'mobile phone', 'wireless computing', 'downloads of tone', 'movie for mobile', etc. Moreover, the sources such as Wordnet or Wortschatz, as described in subsections 5.2.1 and 5.2.2, are vulenerable to queries of combined concepts and terms. However, these are just possible assumption for the conflicting findings resulted by both the diversity measures, in particular, for the extreme val-

ues.

5.4.4 Initial Recommendation for (Re-)Designing the Architecture

The inconsistent findings measuring the diversity of implemented classifiers and the dis-similarities between sources cannot be utilized as the final criteria correlated to the crucial factor of the present approach. However, the Q-statistic measure can be initially used to detect common blogs for resampling training data sets and to imagine, how well these can complement each another and to combine these classifiers. Moreover, the Q-statistic has been proposed as measure of (dis-)similarity in the numerical taxonomy literature [53]. In contrast to that, Entropy of the Votes can be utilized for an overall performance evaluation, based on the overall correctness and incorrectness of the classifiers.

However, an important review of the present setting of measurement is advisable, based on the disagreements of both the diversity measures. The classifiers have not similar setting of accuracy parameter values, as Kuncheva and Whitaker utilize and recommend in their extensive analysis of diversity measures. In this context, they use classifiers with high similarity in terms of accuracy characteristics, e.g. related to precision or recall, for spanning the largest possible interval for the diversity measures. Based on this idea, classifiers should be used considering similar accuracy peculiarities.

6 Architecture

To imagine how the proposed approach is implemented at the time of performing the TREC Blog run, the architecture and workflow is illustrated in Figure 5. The general recommendation derived, as mentioned above, is taken up in Section 8.2.1

First, as Mishne reveals that indexing RSS content is better than the entire HTML content [33], feeds are indexed with Lucene[13] for proposing topics and storing blog contents. Second, to gather and to structure keywords from various sources in a Topic Map, keywords are extracted from different sources via their underlying API. Third, to determine the thresholds of complex classifiers, the threshold estimation procedure is iterated by trading off precision and recall. Thus, the threshold is taken when the F_1-measure cannot be improved.

The weighting and testing of these classifiers is the most labour-intensive part of the implemented architecture. The weighting procedure can be described as the sequential training of all classifiers, whereas the weight of one classifier depends on the weight distribution of previous ones. Therefore, the order, in which the classifiers are trained, influences the combined hypothesis and results. In this regard, some efforts have to be invested in experimenting the order in the weighting and testing process.

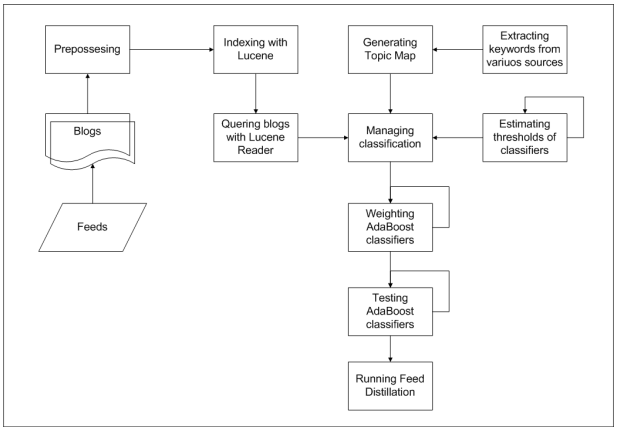

Figure 5: Workflow and architecture of the proposed approach

6.1 Prepossessing

To improve Information Retrieval (IR) performance, stemming and filtering methods are used for the implemented prepossessing steps. The filtering step removes stop words using hand-coded stop-list for English. Among the others, Stop words' removal is used for analyzing the blog entries within the Lucene In-

[13] http://lucene.apache.org

dexing Tool (Luke)[14] and similarity measuring process between the keywords of the Topic Maps and the content of each blog within a feed.

Relating to the stemming step, a Java[15] library 'Snowball English Stemmer' is utilized, which is based on Porter's stemming algorithm [42]. The Porter's stemming method belongs to conflation algorithms, in particular, to the category of suffix removal. It is intuitive and the most used stemming algorithm, since it is simple with regard to implementation and compactness [57]. Additionally, Hull points out in a detailed evaluation that the Porter's stemmer is the one of best stemming algorithms with regard to average precision and recall metrics [20]. Lennon et al. also reveal, that there are relatively small differences among the conflation methods in terms of compression percentage and retrieval effectiveness [25].

6.2 Indexing and Querying Blogs with Lucene

The TREC collection of documents (feeds) amounts to over 25 GB, as described in [27]. To effectively search and query feeds, in particular, blog entities within feeds, the Lucene text search engine is used. Lucene is implemented in Java and consists of a set API providing high-performance, full-featured ranked searching functionality. Because it is high efficiency and usable on cross-platforms, it has been widely used in many applications to provide full text search functionality [17].

One of the most important advantages refers to the various ways to index a feed. For instance, the content fields such as 'title', 'description', etc. associating to a document can be specified for indexing and storing, while other adjunctive fields such as 'feedno' and 'feedurl', which are irrelevant to the content, can be just stored without being indexed. The indexing process is accompanied by a set of filters such as stemming, removal of stop words and tokenization when it is required.

Another most important advantage is the field search. For example, using 'title:music' as query, one can simply search documents with the title containing the named-entity 'music'. Lucene also provides usual Boolean operands such as 'AND', 'OR' and 'NOT' to make complex queries for matching documents. The querying process of fields also goes through the same set of filters as they have been indexed before. Additionally, the searching speed only amounts to relative few milliseconds by matching 1000 of blog entries. Moreover, there are additional ranking features using default similarity measure and TF-IDF model [22]. Also there are boosting factors for subsequently manipulating the relevance of documents.

However, apart from the feed number and the feed URL, indexing feeds is limited to the level of blogs, additional properties of a feed such as permalinks, etc. are not considered.

6.3 Generating Topic Maps Using TopicMapBuilder

The TopicMapBuilder is responsible for querying and extracting keywords from various sources using their APIs. To do so, the keywords related to a topic are structured in the lists of a Topic Map. Via the Java Architecture for XML Binding (JAXB), it also manages the Topic Maps to serialize Java Objects to XML data (storing) and to deserialize XML data to Java Objects (loading). An additional feature of JAXB is that it can enable to talk to XSLT, DOM, dom4j, XML-aware database, and many existing libraries[16]. Thus, the stored Topic Maps can be re-used for other applications and system in terms of IR.

6.4 Managing Classification

The implemented Class ManageClassification has interfaces to Prepossessing, TopicMapBuilder and Lucene querying API, and unifies their functionalities. ManageClassification can retrieval the indexed blog documents from the TREC collection and the keywords from certain Topic Maps. To do so, the similarity score between terms of a document and

[14] http://www.getopt.org/luke
[15] http://snowball.tartarus.org
[16] http://www.xml.com/pub/a/2003/01/08/jaxb-api.html

the keywords can be measured by using prepossessing steps such as filtering and stemming. Moreover, via XStream API[17], a simple JAVA library to serialize and back again, ManageClassification can load and store the information of the AdaBoost Classifiers. These information refer to the weights and thresholds of each classifier as well as the evaluation result of feeds ranked by the implemented AdaBoost classification algorithm.

6.5 Estimating Appropriate Thresholds of Classifiers

To fix the denotation, the threshold t_r of a classifier, chosen via iterative trading off recall and precision, can be interpreted as a fixed parameter determining the relevance $f_r \in [0,1]$ of a blog document b, based on its similarity to the keywords structured in a certain list l of a Topic Map $sim(b,l)$. Thus, it can expresses as the following formula:

$$f_r = \begin{cases} 1, \text{ if } sim(b,l) > t_r \\ 0, \text{ others} \end{cases} \tag{10}$$

Since thresholds are not only utilized for the estimation of relevant blog documents, but also for spam detection of blogs, it is important to deal with another denotation for this issue. The threshold t_s of a spam-specific classifier is defined as a fixed parameter (chosen as mentioned above) determining the spam-relatedness $f_s \in [0,1]$ of a blog document b, based on its similarity to the spam-specific keywords structured in Blacklist's or Google Suggest's keywords sources l of a Topic Map $sim(b,l)$. Thus, it can denoted as follows:

$$f_s = \begin{cases} 1, \text{ if } sim(b,l) > t_s \\ 0, \text{ others} \end{cases} \tag{11}$$

However, this subsection has its focus on the determining procedure of appropriate thresholds. In fact, the process exploits the optimum performance between the precision and recall using F_1-measure. An algorithm optimizing parameters of each threshold-based classifiers is illustrated in Algorithm 2.

[17] http://xstream.codehaus.org

Algorithm 2 An Algorithm for Determining the Threshold of Classifier

Input : Set of classifiers C and blogs D
Init : $curF1_c = 0.0,\ prevF1_c = 0.0,$
$curThreshold_c = 0.6,$
$prevThreshold_c = 0.0, round = 1$

while $round > 0\ OR\ \|curF1_c - prevF1_c\| > 0.0002$ **do**
  $prevThreshold_c = curThreshold_c$
  $prevF1_c = curF1_c$
  $curF1_c = CalculateF1(c, D)$
  if $prevF1_c > curF1_c$ **then**
    $curThreshold_c = curThreshold_c * 1.15$
  else
    $curThreshold_c = curThreshold_c * 0.97$
  end if
  $round - 1$
end while
$curThreshold_c = prevThreshold_c$

6.6 Weighting AdaBoost Classfiers

This module is responsible for learning weights of the implemented classifiers and is closely connected with iterative changing and arranging parameters in terms of the thresholds of classifiers, training iterations, the size and resampling of training data set. In particular, the focus lies on balancing the appropriate settings for the learning process of the implemented AdaBoost classifiers. Also the variation of the order of weak learner is focal point of this part of the architecture. For more details to the weighting algorithm, see Section 4.2

The order, as mentioned above, in which a committee of classifiers are trained, has impacts on the output of the final classifier. Analyzing weighting output on the set of training data in terms of six topics ('music', 'Christmas', 'movie', 'food', 'mobile phone' and 'photography'), it is observed that using

spam-detection, followed by title-based and content-based, performs a appropriate weighting result. The ascending order of classifiers is illustrated in iii.

i.) Splog-specific Classifiers

1.) BlacklistTitleFilter
2.) BlacklistBodyFilter
3.) SimilarityPatternFilter
4.) NGramTitleFilter
5.) NGramBodyFilter

ii.) Title-based Classifiers

6.) CategoryTitleFilter
7.) IntersectionTitleFilter

iii.) Content-based Classifiers

8.) CategoryBodyFilter
9.) IntersectionBodyFilter
10.) RelevanceWordnetFilter
11.) RelevanceWortschatzFilter
12.) RelevanceWikipediaFilter
13.) RelevanceDmoz1Filter
14.) RelevanceDmoz2Filter
15.) RelevanceDmoz3Filter
16.) RelevanceYahooFilter

6.7 Testing AdaBoost Classfiers

The testing unit utilizes the scores of blogs assigned by the learned AdaBoost classifiers, as illustrated in Section 4.2, to rank the feeds. To do so, the scores of all blogs within a feed is summarized and averaged. Thus, the average score can be used as the criteria for ranking feeds in the test data set and is illustrated as follows:

$$AverageScore_j = \frac{1}{N} \sum_{i}^{N} V_j, \tag{12}$$

where N is number of blogs within a feed, j is an unlabeled instance (feed) of test data and V is weighted majority voting score $V = \sum_{l}^{m} \log \frac{1}{\beta_l}$. As derived from AdaBoost.M1 in Algorithm 1, m is number of classifiers deployed for the current testing and β is the learned weight of a classifier.

Moreover, the testing unit serves to validate the prediction performance of the trained classifiers. As in all classification issues, the most important rule for creating a great predictor is that examples of test data may not exist in training data set. Thus, only applying of trained classifiers to unseen data can assure the generalization of hypotheses and prediction.

7 Training and Experimental Settings

This section deals with the issue in terms of training and test data set. The size of the training and test data set varies from topic to topic, since some topics are preferred by the bloggers and depend on seasonal issues, e.g. 'Christmas'. Additionally, the size of the data set is different between the TREC-related and improved training and testing settings, which are more detailed in the following subsections. However, the iteration for weight learning process for all topics and classifiers is limited to 20 rounds, since AdaBoost's error rate rapidly decreases.

7.1 TREC-related Training and Testing Settings

Since the TREC document collection already provides categorized feeds, the blogs within the feeds are indexed with Lucene. Thus, a new collection of data set consists of blogs with categories provided by the feed and is created for six topics such as 'music', 'Christmas', 'food', 'movie', 'mobile phone' and 'photography'. However, as [27] reveal that the blogs are infected by splogs, manual analyzing as well as separating splogs and non-relevant blogs are mandatory. Splogs are also important for learning splog-related classifiers, therefore the collection is additionally labeled with a new category 'blacklist' and can be iden-

tified by 'true' and 'false'. Depending on desired setting of the training and test data, the splog-specific and relevance determining classifiers such as title- and content-based ones can be separately trained and tested.

By participating in TREC Blog Track, the competition with other participating groups not only plays a important role, but also the underestimated time. In addition, performing the run requires more than one person. Therefore only one result and first test is submitted when the Blog Track run is due. In this regard, weights and thresholds determined by six topics such as 'Christmas', 'music', 'photography', 'food' and 'mobile phone', are utilized for the remaining 40 topics. Based on the idea of generalizing the proposed approach, the classifiers are initialized with these weights and thresholds mentioned above for the topical classification of the remaining topics.

7.2 Improved Training and Testing Settings

Exploiting the advantage of Lucene's TF-IDF scoring, the training data set varies from 40% to 60%. The assumption is based on the fact, that the classifiers should learn from relative good example set. Moreover, since there is a test data set judged by the participating groups of this annual Blog Track, the test data set should have more evaluative accuracy than the manual labeled one. Also orientation of performance output of the implemented approach towards the average precision of testing output is a great chance for detecting bugs and errors in learning and improving process of both the classifiers and keyword sources.

The thresholds and weights of the classifiers estimated for one of six topics mentioned above are used for all remaining 44 topics. In this context, the best one result of one topic is chosen from all six test series.

Figure 6 points out, how many percent of feeds has been be considered at the time of performing the TREC Blog Track run and submitting the results. Additionally, re-extracting docs feeds does not result in existing complete number of feeds in the present testing environment. In particular, there are many

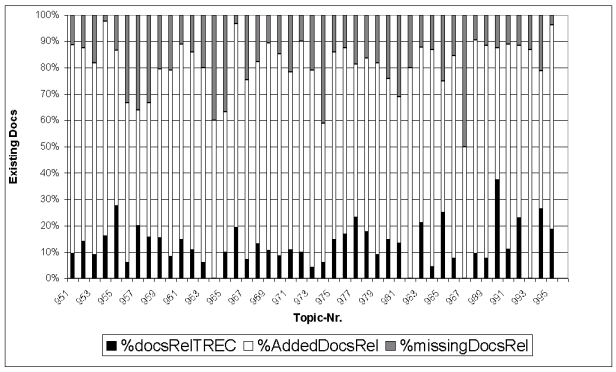

Figure 6: Missing Feeds in Percent

relevant documents, which are still missing. Hence, evaluation of performance of the present prediction system has be taken this into account and has to be modified with regard to evaluation metric such as the number of relative documents for calculating the average precision. This modification is explained more detailed in Section 8.2

8 Evaluation Results

In this section, both the TREC-related and inofficial evaluation results are depicted. Since implementing the proposed approach is accompanied by bugs in terms of quality of the keyword references and implementing AdaBoost classifiers, the differential dealing of both results is advisable. Particularly, qualitative arguments with regard to the evaluation results are illustrated. Also the essential improvements to increase the retrieval and classification results are presented.

8.1 TREC-related Evaluation

Since the implemented system was in beta-version when the TREC was due, the evaluation results were not convincing, as shown in Figure 7. In this context, mistakes by performing the run have been identified and removed in regard with the keyword aspects and weighting AdaBoost classifiers.

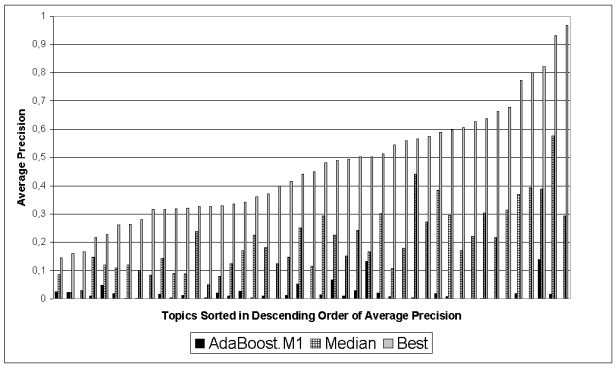

Figure 7: Topics Submitted by TREC and Sorted in Descending Order of Average Precision (AP)

8.2 Inofficial Evaluation and Reconsidering of the Proposed Approach

This subsection points out, which essential arrangements of the implementation are necessary to improve the prediction performance of the AdaBoost classifiers and judge the proposed approach. In the second subsection, performance results are separately illustrated for dependent and independent set of classifiers considering results submitted by other participating groups of the TREC Blog run.

8.2.1 Reflecting Keyword Resources and Implemented Classifiers

This subsection reflects the configuration of classifiers in terms of weighting and testing environments, based on the findings of diversity measures, as depicted in subsections 5.4.

One of the most important finding refers to the original belief that dis-similarity of keyword sources could be equal diversity of classifiers. In this regard, the classifiers utilize the relative dis-similar lists of keywords to compute the similarity and to learn the weights. To do so, the possibility, that the thresholds of certain classifiers are exceeded by the same intersection of similar keywords, is just discarded. In particular, this intersection is greater when taken the stemming process into account. However, even if the similarity of a keyword list to the remaining lists of a Topic Map is relative low, as exhibited in Figure 3, but is greater than the thresholds used by the underlying classifiers. Hence, the question arises, whether the dis-similarity can completely replace the diversity of implemented classifiers or in worst case defeat the boosting effects of AdaBoost. To face this problems, a new set of independent classifiers is created. In this context, all lists of each Topic Map have to be revised in terms of redundancy of retrieved keywords. Therefore, the revised condition is: 'Each keyword of a list must not exist in another remaining list of a Topic Map'. Additionally, since some topics are more vulnerable to spam, e.g. 'mobile phone' or 'music', etc., the hand-coded blacklist's keywords can be used for removing splog-specific keywords from the remaining lists of a Topic Map, which are utilized for the relevance determination. The order, in which a list's keywords are removed, has influences on the selecting and estimating the thresholds and at least on the weights, but it does not matter, if this order is kept in the threshold determining and weight learning process.

However, in Section 8, the results of prediction performance are separately analyzed and illustrated for both dependent and independent set of classifiers.

Relating to the findings of diversity measures the conditions in terms of the size of classifiers' team is arranged and depends on the setting recommended by [23]. In this context, weak learners with similar accuracy characteristics produce the final set of classifiers. Based on experiments and observations related to the parameters of six topics, the conditions of precision and F_1-measure differ from topic to topic, as illustrated in Table 2, where $c_i \in C$ and C is the created set of weak learners. To do so, the final condition is the unification of both ones.

8.2.2 Inofficial Performance Results

Since the comparison of the present prediction performance with the 'Best'-result is not appropriate, the following evaluations refer to the Median of performance results judged by the participating groups of TREC. Additionally, the results are presented in un-revised and revised version of performance out-

Topics	$c_i > F_1$-measure	$c_i >$Precision
Christmas	0.20	0.20
Food	0.45	0.50
Movie	0.20	0.45
Mobile phone	0.20	0.50
Music	0.20	0.50
Photography	0.20	0.45

Table 2: Conditions for Set Formation of Weak Learners

puts. They differ from each other in terms of reduced number of relevant feeds, based on the subtraction of missing number of feeds. Moreover, the evaluation deals with results performed by both the set of indepedent and dependent classifiers.

As illustrated in Figure 8 and 9, both the set of dependent and independent classifiers outperforms the 'Median' in over 14 topics. Also the hard-to-classify topics are better predicted by the implemented classification system. Interestingly, relating to the topics, which have high average precision resulted by Median, the proposed approach only results poor performance.

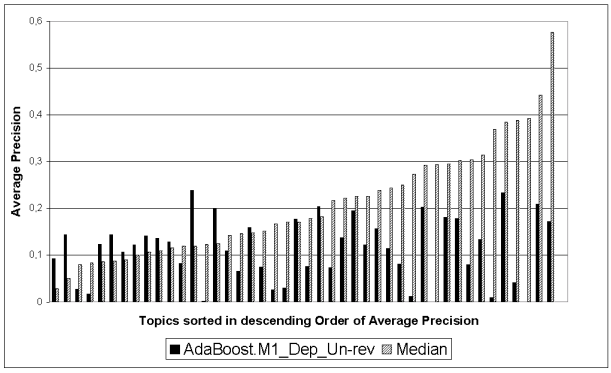

Figure 8: Topics judged by the Committee of Dependent Classiferis and Sorted in Descending Order of Average Precision (AP)

However, from the view of the average precision, the 'Median' has 11 more topics than the prediction system, based on the set of dependent classifiers, when considering topics with average precision over

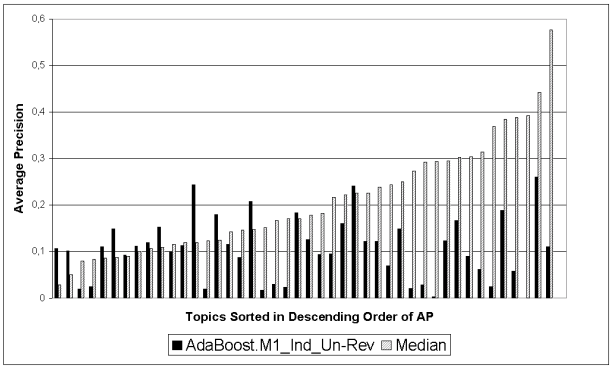

Figure 9: Topics Judged by the Committee of Independent Classifiers and Sorted in Descending Order of Average Precision (AP)

0.24625. Additionally, apart from the range between 0.14775 and 0.24625 in Figure 10, there are not much of difference between the revised and un-revised version of average precision. The system of independent classifiers achieves more poor performance of average precision than the system of dependent ones, as exhibited in Figure 11. However, the differences between dependent and independent versions are critical in terms of performance output in the range between 0.04925 and 0.24625 when comparing both the figures 10 and 11.

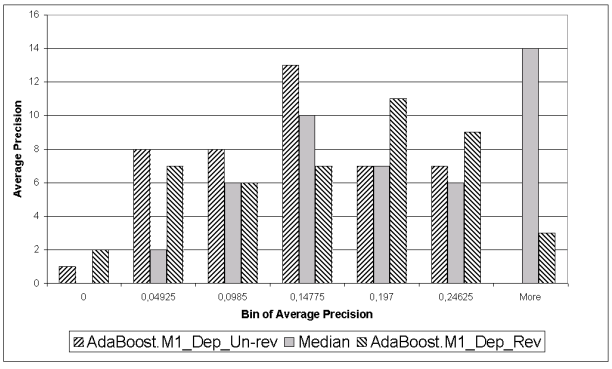

Figure 10: Prediction Performance Results Divided in AP Bins for Comparing the Dependent and Independent Classifiers Considering the Median

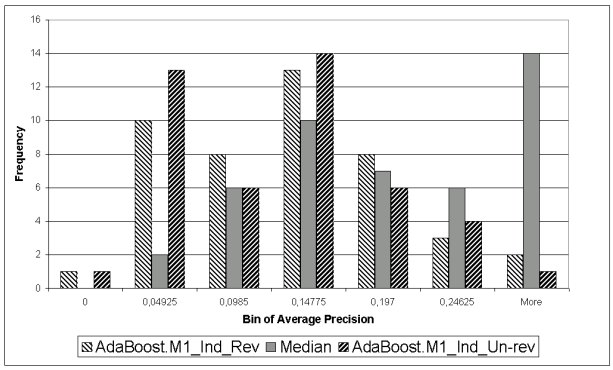

Figure 11: Difference between Un-revised and Revised Performance Results Related to the Median Considering the Set of Independent Classifiers

Apart from 8 topics, the remaining topics are better predicted by the committee of dependent classifiers, as shown in Figure 12. However, high differences between both the sets can be found in 9 topics.

Based on the view of separated bins of average precision, as illustrated in Figure 12 and Figure 11, the dependent prediction system achieves a better performance output when considering the last three bins. However, the disadvantage is that this illustration discards the topic-specific characteristics and splog-specific vulnerability.

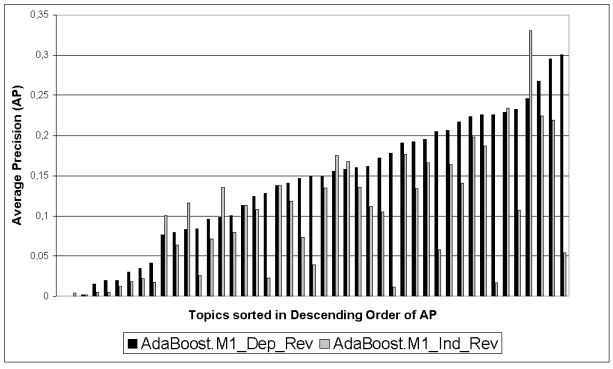

Figure 12: Difference of Revised Prediction Performance Results between the Set of Dependent and Independent Classifiers, Based on the Descending Order of AP

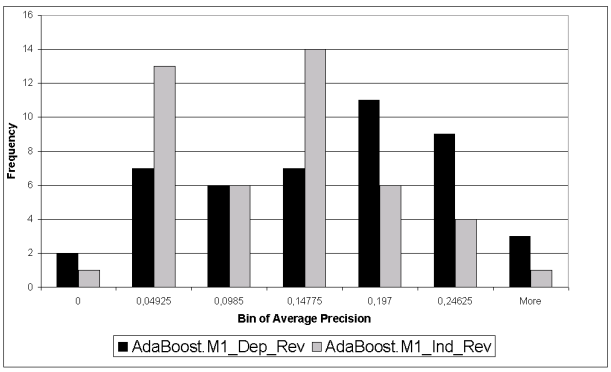

Figure 13: Difference of Revised Prediction Performance Result between the Set of Dependent and Independent Classifiers, Based on the View of AP Bins

Definitely, Figure 14 and Figure 15 reveal that there is no correlation between Median and the implemented system for both the independent and dependent sets. To address the question how the independent and dependent set of classifiers are related to each another, the interpretation of Figure 16 can be helpful. Given the average precision of a dependent set of classifiers one can conjecture that the average precision of independent classifiers has a similar characteristic. In other words, both the systems archieve more or less similar performance results when discarding the question to which parts of the topics they generally match and better perform.

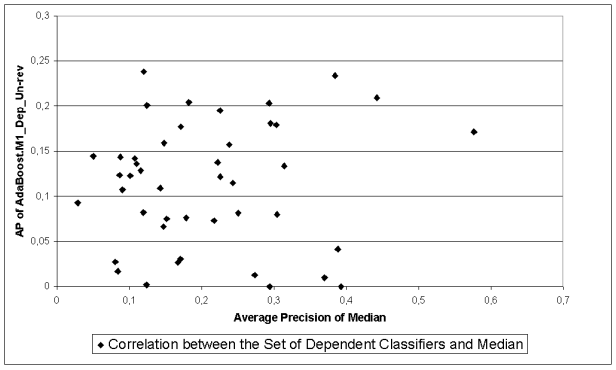

Figure 14: Correlation of Prediction Performance Results between the Set of Dependent Classifiers and Median

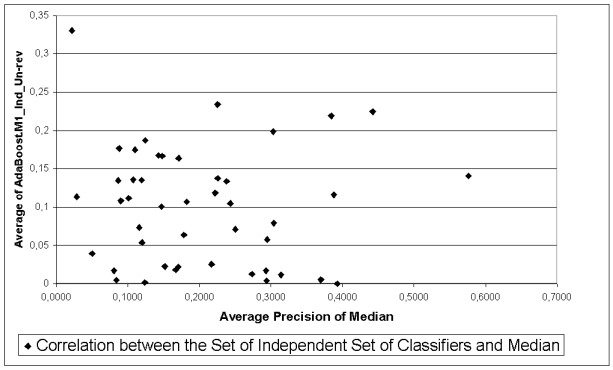

Figure 15: Correlation of Prediction Performance Results between the Set of Independent Classifiers and Median

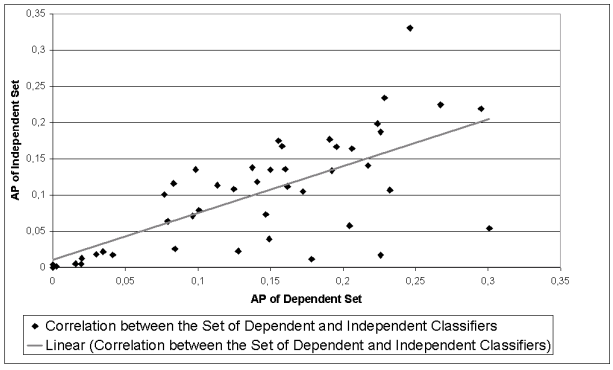

Figure 16: Correlation of Prediction Performance between the Set of Dependent and Independent Classifiers

9 Conclusion and Future Work

Although the presented evaluation results are not convincing, one can assume that the present approach using APIs of emerging technologies of Web 2.0 and different thesauri is an appropriate method to perform this annual Blog Track task 'Feed Distillation'. Apart from the infinite definition of diversity, the diversity of classifiers could be fulfilled by combining the sources considering both the semantic and statistical aspects related to a certain topic. However, the title- and content-based classifiers refer to the terms used in the blogs within a feed. In fact, the diversity required by ensemble-based systems has not only to base on the content-based facets, but also has to consider the structural properties of blogs such as incoming and outgoing hyperlinks, as proposed by [7, 39] for topic distillation of web sites.

Additionally, the performance of the implemented algorithm varies from topic to topic. This issue can be caused by the diversity of the topics suggested by the participating groups. In this context, the 45 topics of TREC do not only contain general concepts such as music, food, etc., but also specific concepts such as 'Buffy', 'Lost TV', which can only be extracted by search engines such as Yahoo, Dmoz or Google Suggest. However, extracting terms of specific concepts leads to the problem that the semantic resources such as Wordnet and Wortschatz cannot provide queries based on the fusion of terms. Separate extracting terms from parts of the topic generates an alienation of origin concept.

Moreover, blogs deal with individual diaries and stories using narrative terms, from the individual perspective of bloggers. This prompts to the question, how high is the similarity between the commonly used keyword sources and the emerging technologies of Web 2.0, which blogs and feeds belong to. However, relating to a general applicability of topic-related concepts and terms for the feed distillation, further research should not only focus on evaluating the quality of commonly used keyword sources such as Wortschatz, Yahoo, Google and Dmoz, but also blog search engines such as Technorati, Blogger, Bloglines, etc, which could provide keywords discovering the narrative aspects of bloggers.

Another one of the most important findings refers to the diversity measures and performance results provided by a set of dependent and independent predictors. In this context, pairwise and non-pairwise diversity measures such as Q-statistic and Entropy Measure have contradictory results in terms of the diversity of the overall system. Additionally, the correlation between the prediction success and diversity of classifiers cannot be consistently reproduced for all topics. While some hard-to-classify topics are better predicted by dependent predictors, other topics are better classified by independent ones. In addition, splogs impede the determining of relevant blogs. This non-straightness of prediction success over all topics concludes that a pre-clustering of blogs, which are more or less vulnerable to splog, is necessary, e.g. while the topic 'machine learning' is less vulnerable to this issue, the topic 'music' or 'mobile phone' are highly infected by splogs using terms such as 'download', 'free', etc. However, the set of dependent classifiers results a better performance than the committee of independent classifiers on an average. Therefore, in the context of blog distillation, designing a set of dependent, content-based classifiers with low similarity is recommended. To do so, classifiers with similar accuracy characteristics can be used as an initial prototype system before scrutinizing the diversity of classifiers extensively.

Finally, the implemented AdaBoost.M1 cannot adequately consider the proportionality of relevant and non-relevant blogs as well as splogs and non-splogs within a feed. Since the final score of AdaBoost applying to a feed is limited to the blog's level and is the result of averaging summarized scores of blogs within a feed, it is also interesting to know whether there is another scoring method, which alternatively achieves the same or better accuracy. Based on the findings by [12], that the boosting effects of AdaBoost can be defeated by classification noise and the error rates of the individual classifiers become very high, experiments considering the diversity measures can be used for adjusting this noise and ranking the feeds. Therefore, further work on improving the proposed approach has to consider an appropriate normalization factor for stabilizing the error variances biased by the various facets of a topic and its underlying splog-specific characteristics, which could reduce the boosting effects a la major votes within a feed.

References

[1] B. Bartell. *Optimizing Ranking Functions a Connectionist Approach to Adaptive Information Retrieval*. PhD thesis, University of California, San Diego, 1994.

[2] M. Biezunski, M. Bryan, and S. Newcomb. ISO/IEC 13250: 2000 Topic Maps: Information Technology–Document Description and Markup Language. *International Organization for Standarization (ISO) and International Electrotechnical Commission (IEC), Dec*, 1, 1999.

[3] E. Bingham, H. Mannila, and J. Seppänen. Topics in 0–1 data. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 450–455, 2002.

[4] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1995.

[5] R. Blood. How blogging software reshapes the online community. *Communications of the ACM*, 47(12):53–55, 2004.

[6] A. Bookstein. Relevance. *Journal of the American Society for Information Science*, 30(5):269–273, 1979.

[7] S. Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. *Proceedings of the tenth international conference on World Wide Web*, pages 211–220, 2001.

[8] W. Cooper. On selecting a measure of retrieval effectiveness. *Journal of the American Society for Information Science*, 24(2):87–100, 1973.

[9] N. Cristianini and J. Shawe-Taylor. An introduction to Support Vector Machines. 2000.

[10] P. Cunningham and J. Carney. Diversity versus Quality in Classification Ensembles Based on Feature Selection. *Machine Learning: ECML 2000: 11th European Conference on Machine Learning, Barcelona, Catalonia, Spain, May 31-June 2, 2000*, 2000.

[11] W. de Winter and M. de Rijke. Identifying Facets in Query-Biased Sets of Blog Posts. *ICWSM 2007*, Boulder, CO USA, 2007.

[12] T. Dietterich. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, 40(2):139–157, 2000.

[13] J. Dong, A. Krzyzak, and C. Suen. Fast SVM training algorithm with decomposition on very large data sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):603–618, 2005.

[14] D. Drezner and H. Farrell. The power and politics of blogs. *Download at 'http://www. danieldrezner. com/research/blogpaperfinal. pdf'*, 2004.

[15] Y. Freund and R. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[16] Y. Freund and R. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.

[17] Y. Guo, H. Harkema, and R. Gaizauskas. Sheffield University and the TREC 2004 Genomics Track: Query Expansion Using Synonymous Terms. *Proceedings of the Thirteenth Text REtrieval Conference (TREC-13)*, pages 753–757, 2004.

[18] S. Harter. Psychological relevance and information science. *Journal of the American Society for Information Science*, 43(9):602–615, 1992.

[19] K. Hornik, M. Stinchcombe, and H. White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5):551–560, 1990.

[20] D. Hull. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1):70–84, 1996.

[21] R. Iyer, D. Lewis, R. Schapire, Y. Singer, and A. Singhal. Boosting for document routing. *Proceedings of the ninth international conference on Information and knowledge management*, pages 70–77, 2000.

[22] B. Katz, J. Lin, D. Loreto, W. Hildebrandt, M. Bilotti, S. Felshin, A. Fernandes, G. Marton, and F. Mora. Integrating web-based and corpus-based techniques for question answering. *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, 2003.

[23] L. Kuncheva and C. Whitaker. Measures of diversity in classifier ensembles. *Machine Learning*, 51:181–207, 2003.

[24] J. Lee. Analyses of multiple evidence combination. *ACM SIGIR Forum*, 31:267–276, 1997.

[25] M. Lennon, D. Peirce, B. Tarry, and P. Willett. An evaluation of some conflation algorithms for information retrieval. *Journal of Information Science*, 3(4):177, 1981.

[26] Y. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. Tseng. Splog detection using self-similarity analysis on blog temporal dynamics. *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 1–8, 2007.

[27] C. Macdonald and I. Ounis. The trec blogs06 collection: Creating and analysing a blog test collection. *Department of Computer Science, University of Glasgow Tech Report TR-2006-224*, 2006.

[28] L. Maicher and J. Park. Charting the Topic Maps Research and Applications Landscape: First International Workshop on Topic Map Research and Applications, TMRA 2005, Leipzig, Germany,(Lecture Notes in Artificial Intelligence). 2006.

[29] R. Mandala, T. Tokunaga, and H. Tanaka. Combining multiple evidence from different types of thesaurus for query expansion. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–197, 1999.

[30] M. Maron. On indexing, retrieval and the meaning of about. *Journal of the American Society for Information Science*, 28(1):38–43, 1977.

[31] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. *Proceedings of the 15th international conference on World Wide Web*, pages 533–542, 2006.

[32] D. Michie, D. Spiegelhalter, C. Taylor, and J. Campbell. *Machine learning, neural and statistical classification*. Ellis Horwood Upper Saddle River, NJ, USA, 1995.

[33] G. Mishne. Using Blog Properties to Improve Retrieval. *ICWSM'2007; colorado, USA*, Boulder, 2007.

[34] Y. Murphey, Z. Chen, and H. Guo. Neural learning using AdaBoost. *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, 2, 2001.

[35] B. Niu, Y. Cai, W. Lu, G. Li, and K. Chou. Predicting protein structural class with AdaBoost learner. *Protein Peptide Lett*, 13:489–492, 2006.

[36] F. Paradis. Using linguistic and discourse structures to derive topics. *Proceedings of the fourth international conference on Information and knowledge management*, pages 44–49, 1995.

[37] J. Park and I. Sandberg. Approximation and radial-basis-function networks. *Neural Computation*, 5(2):305–316, 1993.

[38] S. Pepper. The TAO of Topic Maps. *Proceedings of XML Europe*, 2000.

[39] V. Plachouras and I. Ounis. Distribution of relevant documents in domain-level aggregates for topic distillation. *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 372–373, 2004.

[40] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, Third Quarter:21 – 45, 2006.

[41] R. Polikar, L. Upda, S. Upda, and V. Honavar. Learn++: an incremental learning algorithm for supervised neuralnetworks. *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, 31(4):497–508, 2001.

[42] M. Porter. An algorithm for suffix stripping. *information systems*, 40(3):211–218, 2006.

[43] U. Quasthoff, M. Richter, and C. Biemann. Corpus Portal for Search in Monolingual Corpora. *Proceedings of the fifth international conference on Language Resources and Evaluation, LREC 2006*, pages 1799–1802, 2006.

[44] H. Rath. The Topic Maps Handbook, empolis GmbH. Download at 'http://www.empolis.com/downloads/empolis_TopicMaps_Whitepaper20030206.pdf', 2003.

[45] G. Ratsch, S. Mika, B. Scholkopf, and K. Muller. Constructing boosting algorithms from SVMs: an application toone-class classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(9):1184–1199, 2002.

[46] E. Romero, L. Màrquez, and X. Carreras. Margin maximization with feed-forward neural networks: a comparative study with SVM and AdaBoost. *Neurocomputing*, 57:313–344, 2004.

[47] R. Schapire. Theoretical Views of Boosting and Applications. *Algorithmic Learning Theory: 10th International Conference, ALT'99, Tokyo, Japan, December 6-8, 1999: Proceedings*, 1999.

[48] R. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, 37(3):297–336, 1999.

[49] C. Scheel, N. Neubauer, A. Lommatzsch, K. Obermayer, and S. Albayarak. Effiencient query delegation by detecting redundant retrieval strategies. *SIGIR 2007, Workshop on Learning to rank for Information Retrieval*, 2007.

[50] F. Sebastiani, A. Sperduti, and N. Valdambrini. An improved boosting algorithm and its application to text categorization. *Proceedings of the ninth international conference on Information and knowledge management*, pages 78–85, 2000.

[51] J. G. Shanahan and N. Roma. Boosting support vector machines for text classification through parameter-free threshold relaxation. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 247–254, New York, NY, USA, 2003. ACM Press.

[52] D. Shen, R. Pan, J. Sun, J. Pan, K. Wu, J. Yin, and Q. Yang. Query enrichment for web-query classification. *ACM Transactions on Information Systems (TOIS)*, 24(3):320–352, 2006.

[53] P. Sneath and R. Sokal. *Numerical taxonomy*. Springer, 1973.

[54] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[55] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2000.

[56] E. Voorhees and W. Croft. Query expansion using lexical-semantic relations. *Proceedings of the 17th Annual International ACM-SIGIR Conference*, pages 61–69, 1994.

[57] O. Yilmaz and W. Frakes. A Case Study of Using Domain Analysis for the Conflation Algorithms Domain. *IEEE Transactions on Software Engineering*, 2007.

[58] S. M. zu Eißen. *On Information Need and Categorizing Search*. PhD thesis, Faculty of Electrical, Computer Science, and Mathematics Department of Computer Science of the University of Paderborn, Germany, 2007.