An Adaptive Personalized News Dissemination System

Ioannis Katakis, Grigorios Tsoumakas, Evangelos Banos, Nick Bassiliades,
Ioannis Vlahavas

Department of Informatics, Aristotle University, 54124, Thessaloniki, Greece
{ katak, greg, vmpanos, nbassili, vlahavas } @ csd.auth.gr

Abstract. With the explosive growth of the Word Wide Web, information overload became a crucial concern. In a data-rich information-poor environment like the Web, the discrimination of useful or desirable information out of tons of mostly worthless data became a tedious task. The role of Machine Learning in tackling this problem is thoroughly discussed in the literature, but few systems are available for public use. In this work, we bridge theory to practice, by implementing a web-based news reader enhanced with a specifically designed machine learning framework for dynamic content personalization. This way, we get the chance to examine applicability and implementation issues and discuss the effectiveness of machine learning methods for the classification of real-world text streams. The main features of our system named PersoNews are: a) the aggregation of many different news sources that offer an RSS version of their content, b) incremental filtering, offering dynamic personalization of the content not only per user but also per each feed a user is subscribed to, and c) the ability for every user to watch a more abstracted topic of interest by filtering through a taxonomy of topics. PersoNews is freely available for public use on the WWW (http://news.csd.auth.gr).

Keywords: Personalization, Text Classification, Concept Drift, Ontology, News Filtering, Dynamic Feature Space.

1 Introduction

The explosive growth of the World Wide Web brought fundamental changes in everyday life. Perhaps the most critical contribution was the boundless, instantaneous and costless offering of information. With the passing of time, more and more amounts of information are becoming accessible. Recently, the amount of available information became gigantic, making the discrimination of useful information out of tons of mostly worthless data a tedious task. This phenomenon is commonly named “Information Overload” and constitutes a main impediment to the user finding the needed information in time. Information overload can be observed in many domains. In a news feed for example, the user is usually deluged with lots of uninteresting articles that ruin the whole news channel. In email, the user can’t easily identify important messages and tends to spend precious time in reading email of doubtful content.

Machine Learning and especially Text Classification (also called Text Categorization) is a promising research area with a potential to contribute to the solution of the problem. Much work has been done towards this direction (Sebastiani, 2002), but unfortunately, there aren't many real-world applications that are widely used by the public. In fact, the only text classification application, where machine learning was successful and widely used is spam filtering (Androutsopoulos et al., 2000). Popular systems like SpamAssassin[1], SpamBayes[2] and Mozilla Thunderbird's[3] embedded junk filter are in daily use and have really assisted to the solution of the problem. In news classification however, similar systems never met such popularity.

We have implemented a system called PersoNews, an earlier version of which has been presented in (Banos et al., 2006), in order to bridge this gap between theory and practice. The main features of the PersoNews system are: a) the aggregation of many different news sources that offer an RSS version of their content, b) incremental filtering offering dynamic personalization of the content not only per user but also per each feed a user is subscribed to, and c) the ability for every user to watch an abstract topic of interest by keyword-based filtering through a taxonomy of topics.

Interestingly enough, the task of news classification sets numerous requirements and constrains for the machine learning component which will be used. We have designed a specialized framework for the problem of text stream classification with concept drift. A computationally undemanding yet effective implementation of the proposed framework was evaluated on two real world datasets obtaining promising results. Consequently, this instantiation was used as the classification component of PersoNews.

PersoNews is freely available for public use on the WWW (http://news.csd.auth.gr).

The rest of the document is structured as follows: In the following section (Section 2) we review and organize systems created for more efficient news reading. In Section 3 we design and in section 4 we evaluate the Machine Learning framework used for personalization in PersoNews. In the following section (Section 5) we describe the implemented system in detail. We then continue by presenting results of a system evaluation procedure (Section 6) and conclude with discussion and plans for future work (Section 7).

2 Related Systems

In this paper, we present an adaptive news (RSS) reader (PersoNews). We organize systems related to PersoNews into the following four categories a) Standard Multi-Source RSS Aggregators b) Community-Based Popularity Systems c) Single Source Adaptive News Readers and d) Adaptive RSS Aggregators. All of the above systems assist the end-user to read news articles more efficiently by either i) aggregating multiple news-sources in one application ii) filtering or low ranking uninteresting

[1] The Apache SpamAssassin Project: http://spamassassin.apache.org/

[2] SpamBayes: Bayesian Anti-Spam Classifier: http://spambayes.sourceforge.net/

[3] Mozilla Thunderbird: http://wwwmozill.com/thunderbird/

articles on behalf of the user iii) recommending interesting articles for each user. Table 1 summarizes information about the systems mentioned in this section.

Table 1. News Reading Systems and Services and their basic characteristics

System Name	Multi Source	Adaptive	Feedback	Method
Google Reader	Yes	No	No	-
BlogLines	Yes	No	No	-
SharpReader	Yes	No	No	-
Thunderbird	Yes	No	No	-
Digg	Yes	No	Explicit	-
Newscloud	Yes	No	Explicit	-
Krakatoa	No	Yes	Explicit	Content Based (Vector Similarity)
WebClipping2	No	Yes	Implicit	Content Based (Bayesian Classifier)
NewsDude	No	Yes	Explicit	Content Based (Naive Bayes, kNN)
Categorizor	No	Yes	Explicit	Content Based (SVM)
Findory	Yes	Yes	Implicit	User Based, Content Based
Spotback	Yes	Yes	Explicit	User Based
Reddit	Yes	Yes	Explicit	Content Based
Google News	Yes	Yes	Implicit	User Based
PNS	Yes	Yes	Implicit	User Based
MyFeedz	Yes	Yes	Explicit	Content Based

2.1 Standard RSS Aggregators

This category comprises systems that allow users to watch and manage multiple RSS feeds and blogs. Examples include web applications, such as *Google Reader*[4] and *Bloglines*[5], client-based software, like *SharpReader*[6] and contemporary email clients, like *Mozilla Thunderbird.* Systems of this category don't offer personalization, as they do not adapt to user interests.

2.2 Community-Based Popularity Systems

Systems of this group are based on the so-called "community wisdom" approach. The source of articles is the systems' user community. Users submit, read and review articles. Popular articles that receive positive reviews from a great number of users are recommended to all users. Examples include popular web pages like *digg*[7] and

[4] Google Reader – http://reader.google.com
[5] Bloglines - http://www.bloglines.com
[6] SharpReader - http://sharpreader.net
[7] Digg – http://digg.com

newscloud[8]. This category of systems assists users in spotting important articles of general interest but still, the element of personalization is missing.

2.3 Single-Source Adaptive News Readers

There are some older proposals in the field of personalized news reading like (Bharat et al., 1998; Billsus and Pazzani, 1999; Chan et al., 2001). Those are Machine Learning enhanced news readers but unfortunately usually are unavailable for public use.

Kamba and Bharat (1998), present a personalized on-line newspaper ("The Karakatoa Chronicles") which is automatically created for every user, based on user feedback. The approach in that paper was to convert each article into a word / feature vector. Having the user profile also as a feature vector, all articles could be ranked according to their similarity with this vector.

A special purpose news browser ("WebClipping2") for PalmOS-based PDAs is implemented in (Carreira et al., 2004). The authors use a Bayesian Classifier in order to calculate the probability that a specific article would be interesting for the user. An interesting part of this work is the fact that the system does not take direct feedback by making the user evaluate every article. Instead, the news browser takes advantage of some other characteristics like total reading time, total number of lines, number of lines read by the user, and a constant denoting the user's average line reading time. All those metrics are utilized in order to automatically infer how interesting a particular user found a news article.

Billsus and Pazzani (1999) implemented a Java Applet that uses Microsoft's Agent library to display an animated character, named "News Dude", which reads news stories to the user. The system supports various feedback options like "interesting", "not interesting", "I already know this" and "tell me more about this". After an initial training phase, the user can ask the agent to compile a personalized news program.

Finally, in a paper by Chan and Sun (2001), the user specifies his/her own categories of interest by entering keywords manually. These keywords are used in order to search for relevant articles in the World Wide Web. A classifier is used in order to filter uninteresting news. The system ("Categorizor") accepts feedback from the user who rates each article's relevancy.

These systems do indeed offer personalization but as they are relatively old, they have no RSS support and consequently are restricted to only few or even just one news source.

2.4 Adaptive Personalized RSS Aggregators

Systems most related to PersoNews are adaptive RSS aggregators, like *Findory*[9], *Spotback*[10], *Reddit*[11], *Google News*[12], *PNS*[13], and *MyFeedz*[14]. Such systems gather

[8] NewsCloud – http://newscloud.com

[9] Findory - http://www.findory.com

[10] Spotback – http://www.spotback.com

information about their users by: a) Accepting user feedback (e.g. user gives rates to articles - explicit feedback e.g. Reddit, Spotback, MyFeedz), or b) Observing user behaviour (track which articles the user is reading – implicit feedback e.g. Findory, Google News, PNS).

Having this information about a user, the system can automatically present personalized news by using two common approaches: a) Content based approach: recommend articles that fall within user interests or filter irrelevant articles by using machine learning algorithms (classifiers) or simple similarity measures (e.g. Reddit), and b) User based approach: recommend articles that have been rated high by users that have read similar articles (e.g. Google News, Findory, PNS). The obvious advantage of this category of systems is personalization, which assists user to deal with information overload.

PersoNews is categorized within this group of systems. The main advantages of PersoNews over the aforementioned systems are: a) The application of an effective and efficient Machine Learning framework specialized for text streaming applications with concept drift, b) The utilization of ontologies in order to give users the ability to watch abstracted topics of interest which consists of news from various sources, c) Personalized classifiers for every user-feed and user-topic couple, d) News Filtering in opposition to News Ranking presentation of articles which we believe is not suitable for the information overload problem.

3 Machine Learning Framework

This section initially discusses a number of issues, which we consider important for the design of a successful machine learning framework for personalized news filtering. It then presents a framework for text stream classification (Katakis et al., 2006) that deals with several of these issues. An instantiation of this framework, which is subsequently presented, is used as the personalization component of PersoNews.

3.1 Machine Learning Issues for Personalized News Filtering

A news article is described in essence by textual data. Therefore, the learning framework should preferably include: a) a learning algorithm with evident high predictive accuracy in text classification, and b) a dimensionality reduction method with eminent performance in text classification, in order to cope with the large dimensionality of text. For the first component, appropriate approaches include support vector machines (SVMs) (Joachims, 1998), instance-based learning (Yang, 1994), regression methods (Yang, 1994) and Bayesian learning (McCallum and Nigam, 1998). For the second component, suitable solutions include the use of information theoretic term

[11] Reddit – http://www.reddit.com

[12] Google News – http://news.google.com

[13] PNS - http://pns.iit.demokritos.gr/

[14] MyFeedz – http://www.myfeedz.com

selecting functions like Information Gain (Lewis and Ringuette, 1994), Chi-Squared (x^2) (Schutze et al., 1995) or Mutual Information (Dumais et al., 1998). Other approaches are Term Clustering (Lewis, 1992) and Latent Semantic Indexing (Schutze et al., 1995).

The framework should be capable of dynamically updating the acquired knowledge in response to user feedback. This update should be instant, to ensure direct perception of the personalization process by users. Consequently, the framework should include *incremental* learning (and feature selection) algorithms. From the aforementioned high performance text classification approaches, instance-based and Bayesian learning algorithms such as *k* Nearest Neighbours (*k*NN) and Naive Bayes (NB) are inherently incremental. There also exist incremental training algorithms for SVMs (Laskov et al., 2006). As for the dimensionality reduction component, the aforementioned information theoretic term selecting functions are inherently incremental too.

An issue that arises from incremental learning in textual data is the continuous expansion of the feature space. New documents will inevitably contain new features (words, phrases, n-grams, etc). The learning framework should be able to incorporate new features in its evolving model. Furthermore, it should preferably confine the feature space, to prevent it from becoming unmanageable, and at the same time include mechanisms for retaining the features with the highest predictive value. A related issue that emerges from incremental feature selection and from the aforementioned issue of expanding feature space is the following: The learning framework should be able to make recommendations based on such a *dynamic feature space,* where new features are included and existing features are removed over time.

Another issue that has to be considered in any data stream application is that of *concept drift* (Widmer and Kubat, 1996; Tsymbal, 2004), which is the gradual (*gradual concept drift*) or abrupt (*abrupt concept drift*) change of concept of a target class in a classification problem. For example, in the news filtering problem that we study, the concept of “interesting article” might change from time to time for a particular user. Wenerstrom and Ciraud-Carrier (Wenerstrom and Giraud-Carrier, 2006) defined two additional types of concept drift in terms of space: *descriptional* and *contextual.* Descriptional concept drift refers to the case where the distribution of the classes changes in relation to the values of the features but the set of features itself does not change. Contextual concept drift corresponds to situations where the set of relevant features shifts from one set to another. Much effort has been focused on tackling descriptional concept drift mainly by instance selection (Fan, 2004), instance weighting (Klinkenberg, 2004), fast incremental classifiers (Hulten et al., 2001) or even ensemble of learners (Scholz and Klinkenberg, 2007). On the contrary, only little research has been conducted towards the direction of contextual concept drift (Katakis et al., 2006; Wenerstrom and Giraud-Carrier, 2006). News classification and text stream classification in general contains all types of drift: descriptional and contextual, gradual and abrupt.

Finally, an issue that has to be considered for server-based personalized news filtering systems is that of computational cost. Server-based systems will have to maintain at least one personal classifier for every user. Therefore, the learning framework should include learning and feature selection algorithms with minimal computational complexity for training the filtering models, updating them and providing recommendations.

In Table 2, we summarize the basic characteristics of the aforementioned systems in terms of the issues discussed in this section.

Table 2. Various methods and algorithms mentioned in section 3.1 and their ability to confront effectively the issues mentioned in the same section (Incremental Updates, Performance in Text Classification Tasks, High Dimensionality, Low Computational Cost, Concept Drift, Dynamic Feature Space).

Method	**Inc**	**Text**	**H Dim**	**L Comp**	**Drift**	**Dyn FS**
Naïve Bayes	Yes	Yes	No	Yes	No	No
kNN	Yes	No	No	Yes	No	No
SVM	Yes	Yes	Yes	No	No	No
Instance Selection	Yes	No	No	Yes	Yes	No
Instance Weighting	Yes	No	No	Yes	Yes	No
Ensemble Methods	Yes	No	No	No	Yes	No
Fast Incremental Classifiers	Yes	No	No	Yes	Yes	No
Proposed Framework (see section 3.2)	Yes	Yes	Yes	Yes	Yes	Yes

3.2 The Text Stream Classification Framework

The proposed framework was designed in order to deal with all of the above issues and especially with the problem of dynamic feature space which, up to our knowledge has never met the proper attention.

The framework comprises two basic components: a) an incremental feature based classifier, and b) an incremental feature ranking method.

We call *feature-based classifiers*, those classifiers that can consider any subset of features for the classification of a new instance. Two inherently feature based algorithms are NB and *k*NN. In both of these algorithms each feature makes an independent contribution towards the prediction of a class. Therefore, they can be easily expanded in order to consider a subset of the feature space during prediction. Specifically, when these algorithms are used for the classification of a new instance, they should also be provided with an additional parameter denoting the selected subset of features. NB for example will only multiply the calculated probabilities of this subset, while *k*NN will measure the distance of the new instance with the stored examples based only on this subset.

Incremental feature ranking methods, such as the information theoretic term selecting functions mentioned in the previous section, can deal with a dynamic feature space, as they calculate statistics for each feature independently. Such methods evaluate each word based on cumulative statistics concerning their appearance in each different class of documents and when a new labelled document arrives, the statistics are updated and the evaluation can be immediately calculated without the need of reprocessing past data.

The most important functionality of any machine learning framework designed for streaming data is the adaptation of the model to new instances. Fig. 1 describes the cooperation of the two basic components of the framework for the update and the classification functionality.

```
-------------------------------------------------------
IFE: incremental feature ranking method
IFBC: incremental feature based classifier
-------------------------------------------------------
Functionality: Update
Begin
     foreach feature ∈ Document
            if feature ∉ Vocabulary
                   Vocabulary.expand(feature)
     IFE.update(Document, DocClass)
     IFBC.update(Document, DocClass)
End
--------------------------------------------------------
Functionality: classify
Begin
     TopFeatures ← IFE.getTop(N)
     Decision ← IFBC.classify(Document,TopFeatures)
End
---------------------------------------------------------
```

Fig. 1. The two main functionalities of the framework

3.3 Framework Instantiation

The x^2 method was selected as the feature ranking component of the proposed framework, due to its simplicity and effectiveness in text categorization problems (Yang and Pedersn, 1997). Following the most common approach in text classification, we used single words as features. The NB algorithm was selected for instantiating the learning module of the proposed framework. The *k*NN algorithm is inefficient for data streams, as it requires the storage of training examples. NB on the other hand stores only the necessary statistics and is also widely used in text classification applications[15]. In addition, NB can take advantage of the already stored feature statistics for the purpose of feature ranking and thus integrates easier in the proposed approach.

We extended the implementation of the x^2 feature evaluation method of the Weka library (Witten and Frank, 2005) with a function that allows incremental updates. We also extended Weka's implementation of NB with a function that accepts a feature subset along with a new instance and uses only the features of this subset for the classification of the instance.

Figure 2 depicts the instantiation of the framework using the Naive Bayes classifier and the x^2 measure as a feature reduction component. As we can see there are four basic data tables in use. a) the vocabulary b) a table containing statistical information

[15] http://spambayes.sourceforge.net/, http://popfile.sourceforge.net/

about words c) a table containing the information value for every word d) a table containing probabilities used by the Naive Bayes classifier for every word. All tables have the same size and expand in parallel when necessary (e.g. when a new word appears in the stream). The vocabulary table is used as reference for all other tables. As can also be observed from the figure, every new document not only is automatically categorized by the classifier but also used to update the statistics and/ or expand the vocabulary (if new words are contained). Of course, a new document can be used as training document only if user feedback has been given. This is a general assumption in stream classification. Moreover, as we discuss later, in PersoNews the user has the ability to mark news articles as *interesting* (I) or *junk* (J). The table of statistics is used in order to calculate the x^2 measure and probabilities needed for classification and feature selection.

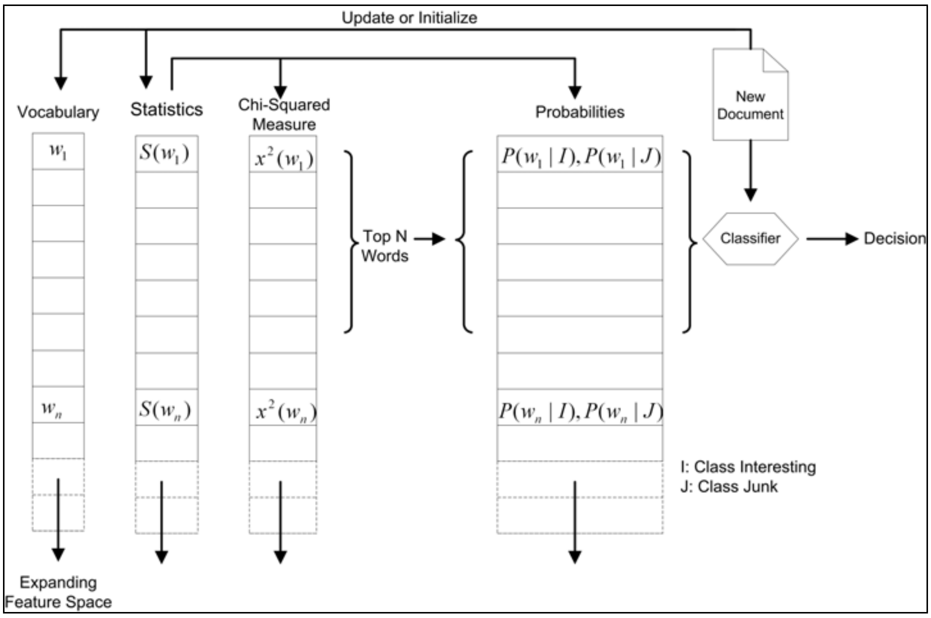


Fig. 2. An instantiation of the proposed framework with the Naive Bayes classifier and an information theoretic feature ranking method.

4 Framework Evaluation

To evaluate the proposed framework we incorporate the aforementioned instantiation as the learning component of three basic stream classification techniques: a) simple incremental learner, where a classifier updates with every new instance b) time window, where a classifier makes predictions based only on the latest batch of instances and c) weighted examples, where a classifier updates incrementally but also focuses on recent instances by using instance weighting. With experimentation we plan to provide evidence on the effectiveness of the implementation discussed in the previous section (and consequently of the proposed framework) when applied to these three baseline streaming methods.

4.1 Experimental Setup

4.1.1 Datasets

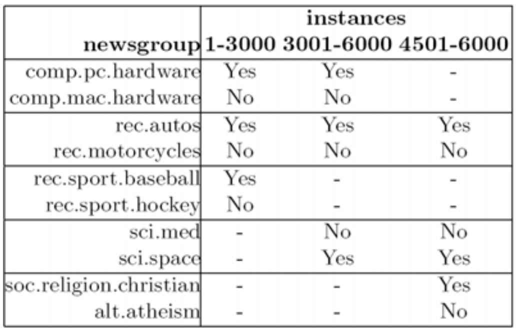
The first requirement of an empirical study of the proposed approach is a data set with documents obtained from a real word textual data stream. We actually experimented with two content filtering domains, spam and news.

For the domain of spam filtering, we ideally need real-world spam and legitimate emails chronologically ordered according to their date and time of arrival. In this way we can approximate the time-evolving nature of the problem and consequently evaluate more properly the proposed approach. For that reasons, we used the SpamAssassin (http://spamassassin.apache.org/) data collection because a) Every email of the collection is available with the headers, thus we were are able to extract the exact date and time that the mail was sent or received, and b) It contains both spam and legitimate (ham) messages with a decent spam ratio (about 20 %). This dataset consists of 9324 instances and initially 40000 features. This datasets represents the so-called gradual concept drift.

For the domain of news filtering we needed a collection of news documents corresponding to the interests of a user over time. As such a collection was not available; we tried to simulate it using Usenet articles from the 20 Newsgroups collection[16]. The data set was created to simulate concept drift. The scenario involves a user that over time subscribes to and removes from different general mailing lists (or news feeds) (e.g. sports, science etc) but is interested only on certain subcategories of these mailing lists. Table 3 shows, the particular interests of the user and how his/her general interests change over time. For example the user is initially interested in sports, but then loses this interest and subscribes in a science mailing list. The user is perpetually interested in driving, while in the last part he/she also gets into religion issues and at the same time unsubscribes from the hardware list. This dataset represents the instant concept drift[17] and consists of 6000 instances and initially 28000 features.

[16] http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

[17] Both datasets are available at http://mlkd.csd.auth.gr/datasets.html



	instances		
newsgroup	**1-3000**	**3001-6000**	**4501-6000**
comp.pc.hardware	Yes	Yes	-
comp.mac.hardware	No	No	-
rec.autos	Yes	Yes	Yes
rec.motorcycles	No	No	No
rec.sport.baseball	Yes	-	-
rec.sport.hockey	No	-	-
sci.med	-	No	No
sci.space	-	Yes	Yes
soc.religion.christian	-	-	Yes
alt.atheism	-	-	No

Table 3. Interest of user in the various newsgroups over time

4.1.2 Methods

The methods that participate in the experimentation are the following:

INB (Incremental Naive Bayes Classifier): A Naive Bayes classifier that updates at each new instance.
TW (Time Windows): A Naive Bayes Classifier that is always trained from the last batch (window) of instances.
WE: (*Weighed Examples*): Like the INB method but here a weighting scheme is used in order for the classifier to focus on recent instances.

We then incorporate the implementation discussed in section 3.3. and generate three additional methods. What the following methods have in common are the two components of our framework: a) continuous Incremental Feature Selection (IFS) using the x^2 measure and b) the necessary feature based classifier (Naive bayes in our case). These methods are:

INB+IFS: A feature based incremental Naive Bayes classifier with incremental feature selection using the chi-squard measure.
TW+IFS: An incremental feature based Naive Bayes classifier that has been trained from the last window of instances. Again, feature selection is executed in every step.
WE+IFS: Like the INB+IFS method but here a weighting scheme is in use in order for the classifier to focus on recent instances.

We conduct a comparison of all these six methods on the two datasets in order to a) show the effectiveness of the implementation proposed in section 3.3 (and consequently of the framework described in 3.2). b) select the most suitable implementation to use in PersoNews.

4.2 Results and Discussion

All methods are executed on the two document collections (spam, news), using as initial training set the first 10, 20 and 30% of the documents[18]. The rest of the documents were used for testing: all methods first predict the class for each new document and then update their models based on the actual class of it. We fixed the number of features to select to 500, as past results have shown that a few hundreds of words are an appropriate size of features. After preliminary experimentation, we concluded that 300 instances was a good window size for the TW method and that a decent way to update the weights in the WE method was $w(n)=w(n-1)+n^2$, where $w(n)$ is the weight of the *n-th* instance. Note that we are not focusing on the accuracy of the aforementioned methodologies, but rather on the effect of our framework and especially of IFS on them.

Table 4 contains the results of the experiments.We first notice that all methods when enhanced with IFS have better predictive performance than the classical approaches for both data sets, all percentages of initial training documents and both metrics. This shows that incremental feature selection manages to catch up with the new predictive words that are introduced over time. In the spam domain, the inclusion of more training data increases the predictive performance of all methodologies due to the inclusion of the important features that appear early in the data set from the beginning. In the news domain on the other hand, the inclusion of more training data does not increase performance significantly as it becomes harder for the classifiers to forget the initial knowledge and adapt to the new predictive features that appear later on. Fig. 3a to Fig. 3c show the moving average (over 200 instances) of the prediction accuracy of all methods (with and without IFS) using the first 20% of all messages of the news collection for training[19]. We notice that the performance is comparable for the first instances, but from then on the performance of IFS-enhanced methods becomes and remains much better than simple methods. This happens because at that time-point the user subscribed to new lists and new predictive words appeared. The same thing occurred after the first 3200 examples when the user changed interests for the second time. Non-IFS methods failed to keep up with the new user interests, while IFS-enhanced methods managed to maintain their initial predictive performance. The TW method is the only one that is not significantly affected from IFS, and that can be seen in

Table 4 (see accuracy TW versus TW+IFS in both datasets) and also from Fig. 3c. This is mainly because of the small window size that the IFS is applied to. Fig. 3e and Fig. 3d show the moving average (over 200 instances) of the number of words promoted to/demoted from the top 500 words in both datasets for the INB+IFS method. Note that in the news domain, in the beginning more words are promoted to/demoted from the top 500 words as the evaluation scores of already included words continue to change with more training examples, while towards the end they stabilize. The peak

[18] Note that all IFS enhanced methods can be applied with no initial training set. Unfortunately the three baseline methods described in the section need a set of training documents in order to construct the feature space that they use.

[19] The respective figures for the spam corpus are similar

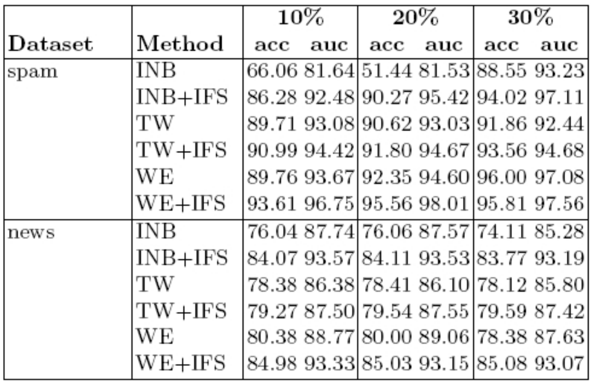
in the spam domain is due to the skewness of the collection (a large number of new spam messages arrived at that time point).

SVMs are well known accurate text classifiers and independent of feature selection. Indicatively, we applied an SVM (Weka implementation (SMO), default parameter setting (Polynomial Kernel of degree *1, C=1, L=0,001*), no initial training) in the news dataset with retraining for every 300 instances and obtained an average accuracy of 70.02%. With TW+IFS method (no retraining) we obtained 77.95% accuracy. Naturally the time needed for the execution of the SVM was much larger (approx. 4 times).

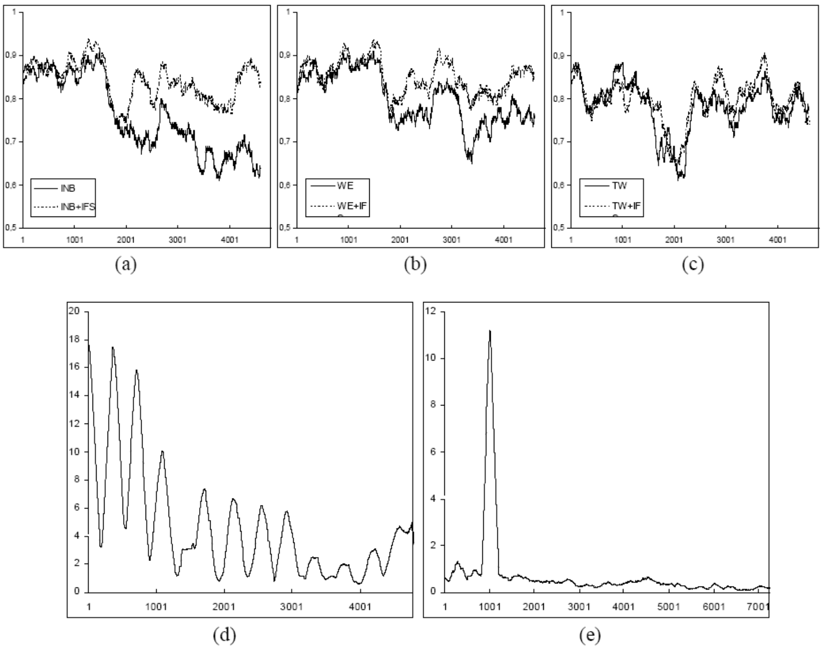
Comparing the IFS methods we could note that in the spam domain, WE+IFS has a noticeable advantage for both metrics. That is may be happening because the continuous increase of instance weights matches the gradual concept drift of the problem. On the other hand in the news domain, where changes are more abrupt WE+IFS seems to outmatch INB+IFS in terms of accuracy but fails in terms of AUC. Considering that the AUC metric is independent of the decision threshold and invariant to a-priori class probabilities, we consider it as a more reliable metric for the text filtering tasks that are studied here. Consequently the INB+IFS method is the one we select as the learning component of PersoNews.

We have of course considered other methodologies proposed in the literature for streaming and concept drifting data like the ones described in the end of section 3.1, but not only they are computationally demanding but most of them could not be applied in a dynamic feature space.

Table 4. Accuracy (acc) and area under the ROC curve (auc) for the two data sets and the three different percentages of training documents for three learning methodologies (with and without IFS)



		10%		20%		30%	
Dataset	**Method**	**acc**	**auc**	**acc**	**auc**	**acc**	**auc**
spam	INB	66.06	81.64	51.44	81.53	88.55	93.23
	INB+IFS	86.28	92.48	90.27	95.42	94.02	97.11
	TW	89.71	93.08	90.62	93.03	91.86	92.44
	TW+IFS	90.99	94.42	91.80	94.67	93.56	94.68
	WE	89.76	93.67	92.35	94.60	96.00	97.08
	WE+IFS	93.61	96.75	95.56	98.01	95.81	97.56
news	INB	76.04	87.74	76.06	87.57	74.11	85.28
	INB+IFS	84.07	93.57	84.11	93.53	83.77	93.19
	TW	78.38	86.38	78.41	86.10	78.12	85.80
	TW+IFS	79.27	87.50	79.54	87.55	79.59	87.42
	WE	80.38	88.77	80.00	89.06	78.38	87.63
	WE+IFS	84.98	93.33	85.03	93.15	85.08	93.07



(a) (b) (c)

(d) (e)

Fig. 3. (a),(b),(c) Moving average of the accuracy for the news domain, and (d),(e) Moving average of the number of words promoted to / demoted from the top 500 words, using the first 20% of all messages of the news collection (d) and spam corpus (e) for training (INB+FS method).

5 System Implementation

PersoNews is a web-based application with the ability to monitor a large set of RSS and Atom feeds and notify users about new publications regarding topics of their interest. The system is publicly available via the URL address http://news.csd.auth.gr. The architecture of PersoNews is shown in Fig. 4. It consists of three modules which function in parallel using a common database to store information: PersoNews.aggregator is a server-side process that handles feed monitoring and content personalization, PersoNews.email is the user notification service and PersoNews.portal is the web application responsible for user interaction and the incremental update of the classifier.

5.1 The PersoNews.aggregator Module

PersoNews.aggregator performs periodical polling of all feeds that it monitors in order to retrieve new publications and store meta-data like title, description, date and URL in the system’s database. At the same time, it performs content filtering by classifying new publications into interesting/junk for each user according to their interests. PersoNews.aggregator offers two channels of personalization: Feed filtering and topic filtering (see Fig. 5). In each case, filtering is achieved by using INB+IFS classifiers described in the previous section.

5.1.1 Feed Filtering

Feed filtering personalizes the dissemination of news of a specific feed to each specific user that chooses to monitor it. PersoNews.aggregator accomplishes the personalization using for each user-feed pair a dynamic filtering mechanism based on the machine learning framework discussed in the previous section (see Fig. 5) The main goal of the filtering mechanism is the automatic discrimination of incoming publications into interesting and uninteresting ones according to user preferences.

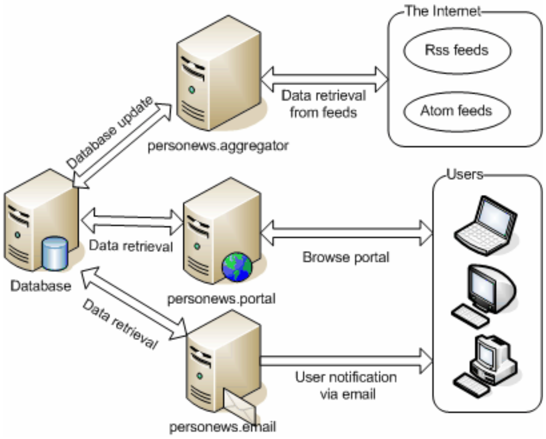


Fig. 4. The architecture of PersoNews

Initially, PersoNews.aggregator retrieves the actual text of the new publication and applies stemming using an implementation of the Porter Stemmer (Porter, 1980) Stemming is an essential preprocessing step for any text mining related task (text categorization, document clustering, information retrieval), because morphological variants of words usually have similar semantic interpretations and can be considered as equivalent.

Subsequently, for every user that monitors this feed, the corresponding classifier that is responsible for this user-feed pair calculates the probability of the new publication to be interesting / junk to the user.

5.1.2 Topic Filtering

Apart from monitoring specific RSS feeds, PersoNews offers to its users the ability to monitor publications regarding a special topic of their interest, such as “Database Management”, that belongs to the system's domain specific topic hierarchy;

regardless the source feed of the publications. To accomplish this, PersoNews checks all the feeds it is monitoring in order to locate new publications relevant to the topic.

The topic hierarchy is a tree structure featuring multiple levels of topic abstraction and currently forms a variant of the ACM Computing Classification System[20]. The top level contains general topics such as “Hardware”, “Computer Systems Organization”, and “Software” while lower levels contain more specific topics such as “Process management” and “Object oriented programming”. The topic hierarchy is implemented using an XML file to store topic descriptions as well as the hierarchical relationships between them. Fig. 6 depicts a sample of the system’s topic hierarchy XML file. The top level nodes correspond to general topics, while sub-nodes correspond to more specific topics. Notice that the modular design of PersoNews allows the effortless replacement of its domain-dependent topic hierarchy with any topic hierarchy that obeys the structure shown in Fig. 6.

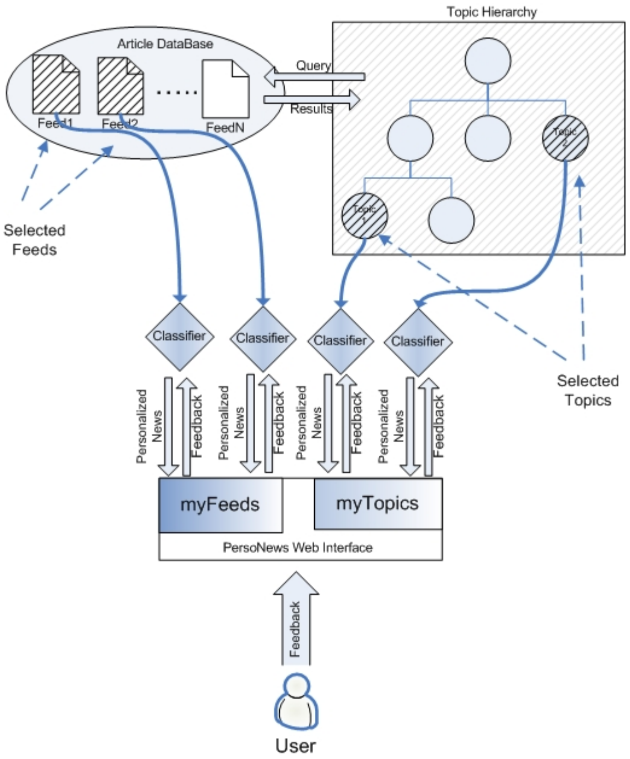
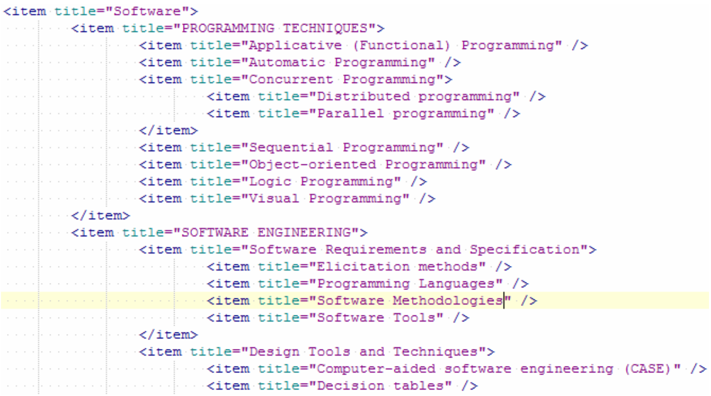


Fig. 5. Personalization in PersoNews

[20] http://www.acm.org/class/



```
<item title="Software">
        <item title="PROGRAMMING TECHNIQUES">
                <item title="Applicative (Functional) Programming" />
                <item title="Automatic Programming" />
                <item title="Concurrent Programming">
                        <item title="Distributed programming" />
                        <item title="Parallel programming" />
                </item>
                <item title="Sequential Programming" />
                <item title="Object-oriented Programming" />
                <item title="Logic Programming" />
                <item title="Visual Programming" />
        </item>
        <item title="SOFTWARE ENGINEERING">
                <item title="Software Requirements and Specification">
                        <item title="Elicitation methods" />
                        <item title="Programming Languages" />
                        <item title="Software Methodologies" />
                        <item title="Software Tools" />
                </item>
                <item title="Design Tools and Techniques">
                        <item title="Computer-aided software engineering (CASE)" />
                        <item title="Decision tables" />
```

Fig. 6. Topic Hierarchy sample XML file

For each topic-user pair there are two layers of filtering (see Fig. 5). The first layer is keyword-based. Each topic is associated with a set of keywords that initially comprise the union of its subtopics in the hierarchy. For example, when choosing to monitor the topic “Database Systems”, PersoNews automatically aggregates every subtopic keywords such as “Database Concurrency”, “Distributed databases”, “Multimedia databases”, “Object oriented databases”, “Rule based databases”, “Textual databases” and “Transaction processing”. Users can also add their own custom keywords and phrases or remove some keywords if they consider it appropriate. If a new publication from any feed contains any of the keywords of a topic, then it is considered relevant to that topic, otherwise it is filtered out.

We argue that this keyword-based filtering is a primitive form of semantic filtering, since each topic is accompanied by a number of user-defined keywords that supposedly describe the topic and can be considered as topic synonyms. Furthermore, the hierarchy of topics is also taken into account, since the keywords of all sub-topics are also considered to describe all their super-topics.

The second filtering layer is performed by a classifier responsible for each user-topic pair. Only publications that pass through the keyword-based filter are subsequently forwarded to the classifier of the user-topic pair, which functions in exactly the same manner as in the case of the user-feed pair.

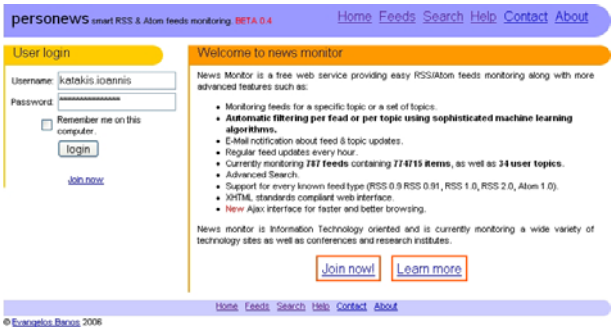
5.2 The PersoNews.email Module

PersoNews.email is responsible for notifying users via email about updates on feeds and topics they monitor. Most importantly, users can interact with personews by giving feedback straight from their email clients. More specifically, there are two links in each email that will allow user to label the article as positive or negative example. It is executed on a daily basis in order to check if there are any new publications in the feeds and topics monitored by each user. In that case, users are notified via email. It is fully customizable giving users the option to have email notifications for specific feeds and topics, change the email format or modify their email address.

PersoNews.email is quite an important part of the PersoNews functionality. Thanks to email notifications, users do not have to bother visiting PersoNews to check for updates in their feeds and topics. Instead, PersoNews sends an email report whenever an update occurs.

5.3 The PersoNews.portal Module

The PersoNews.portal module is a dynamic web application which is accessible via any web browser at the URL http://news.csd.auth.gr. It is implemented using W3C standards such as XHTML 1.0[21], Cascading Style Sheets[22], JavaScript[23] and Dynamic HTML[24]. Site content is divided into public pages, which contain general information about the system, and personal pages, which contain user defined information such as the feeds and topics he/she is currently monitoring. After logging in, the user can check his/her feeds or topics for updates, view a publication or change his/her personal settings. The home page of PersoNews is shown in Fig. 7.


personews smart RSS & Atom feeds monitoring BETA 0.4
Home Feeds Search Help Contact About
User login
Welcome to news monitor
□ Remember me on this computer.
login
Join now
Join now!
Learn more
Home Feeds Search Help Contact About
© Evangelos Banos 2006

Username: katakis.ioannis

Password: ***************

News Monitor is a free web service providing easy RSS/Atom feeds monitoring along with more advanced features such as:

- Monitoring feeds for a specific topic or a set of topics.
- **Automatic filtering per fead or per topic using sophisticated machine learning algorithms.**
- E-Mail notification about feed & topic updates.
- Regular feed updates every hour.
- Currently monitoring ***787* feeds** containing ***774715* items**, as well as **34 user topics**.
- Advanced Search.
- Support for every known feed type (RSS 0.9 RSS 0.91, RSS 1.0, RSS 2.0, Atom 1.0).
- XHTML standards compliant web interface.
- New Ajax interface for faster and better browsing.

News monitor is Information Technology oriented and is currently monitoring a wide variety of technology sites as well as conferences and research institutes.

Fig. 7. PersoNews home page

5.4 Feed Manipulation and Monitoring

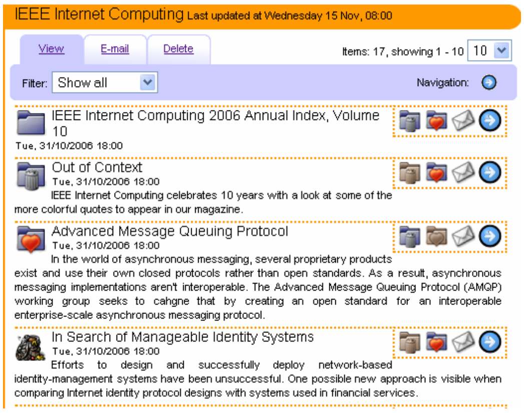
Users can start monitoring a feed by selecting it from the list of feeds that are already registered in PersoNews. Currently, PersoNews aggregates 1636 feeds which are organized into 8 categories according to their topics, in order to make feed browsing and selection easier for users. In addition, users can add a new feed by entering its URL, or batch import any number of feeds using the OPML protocol. New feeds are added into the "various" category and are immediately available for selection by other users.

A common approach in adaptive news readers is to present articles to the user by ranking them according to how close they are with user interests. So eventually all

[21] http://www.w3.org/MarkUp/
[22] http://www.w3.org/Style/CSS/
[23] http://www.mozilla.org/js/
[24] http://www.dhtmlcentral.com/

articles are shown to the user. We preferred a "spam-filtering" approach for PersoNews. We filter the uninteresting messages out. We believe that this is a more useful approach for an information overload problem.

Fig. 8 shows a list of publications in a feed. Publications are presented in reverse chronological order and unread publications are indicated with a starred folder icon. On the top right of each publication there are four icons which correspond to the available user actions. Clicking on the first icon from the left marks the publication as uninteresting, clicking on the second adds the publication to a virtual folder containing the favorite publications of the user, clicking on the third icon allows the user to send the information of this publication to a friend by email and finally, clicking on the last icon the user can visit the publication's source URL.


IEEE Internet Computing Last updated at Wednesday 15 Nov, 08:00
View
E-mail
Delete
Items: 17, showing 1 - 10
10
Filter:
Show all
Navigation:

IEEE Internet Computing 2006 Annual Index, Volume 10
Tue, 31/10/2006 18:00

Out of Context
Tue, 31/10/2006 18:00
IEEE Internet Computing celebrates 10 years with a look at some of the more colorful quotes to appear in our magazine.

Advanced Message Queuing Protocol
Tue, 31/10/2006 18:00
In the world of asynchronous messaging, several proprietary products exist and use their own closed protocols rather than open standards. As a result, asynchronous messaging implementations aren't interoperable. The Advanced Message Queuing Protocol (AMQP) working group seeks to cahgne that by creating an open standard for an interoperable enterprise-scale asynchronous messaging protocol.

In Search of Manageable Identity Systems
Tue, 31/10/2006 18:00
Efforts to design and successfully deploy network-based identity-management systems have been unsuccessful. One possible new approach is visible when comparing Internet identity protocol designs with systems used in financial services.

Fig. 8. Sample feed view

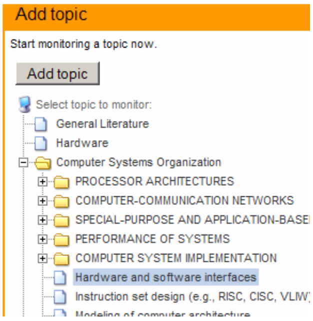
In many systems, user feedback can be given implicitly (by observing which links are followed) or explicitly (by letting the user rate every article). We considered an intermediate solution: In case a publication is marked by the user as junk, the document is forwarded to the classifier as a negative example in order to update its knowledge (explicit feedback). When the user follows a URL, the system assumes that the user was interested in this publication and forwards the document to the classifier as a positive example (implicit feedback). If after reading the publication, the user finds out that it was not interesting, he/she can still mark the article as junk (explicit feedback). In that case, the classifier removes from its knowledge base the previously added positive example and adds the publication as a negative example. The user can also browse the junk folder and mark any misclassified publications as interesting (explicit feedback). This approach gives users the ability to correct their mistakes and does not require the extra burden of giving an exact grade to every article.

5.4.1 Topic Manipulation and Monitoring

Users can start monitoring a topic by selecting it from the system's domain dependent hierarchy. **Fig. 9** shows the form used for adding topics, whereas Fig. 10

shows a form that displays the keywords of a selected topic and allows users to edit them.

Topic publications are displayed in the same way feed publications are presented to the user with the exception of three new user actions. Users can visit the source feed of the publication, add the source feed to the topic blacklist or can start monitoring the source feed. Publications coming from blacklisted feeds are not inserted in topics regardless of their content.

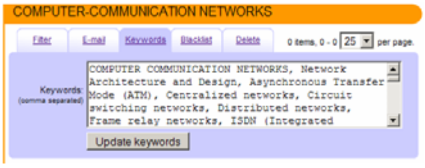


Add topic

Start monitoring a topic now.

Add topic
Select topic to monitor:
General Literature
Hardware
Computer Systems Organization
PROCESSOR ARCHITECTURES
COMPUTER-COMMUNICATION NETWORKS
SPECIAL-PURPOSE AND APPLICATION-BASE
PERFORMANCE OF SYSTEMS
COMPUTER SYSTEM IMPLEMENTATION
Hardware and software interfaces
Instruction set design (e.g., RISC, CISC, VLIW
Modeling of computer architecture

Fig. 9. Form used to start monitoring a topic by selecting it from the system's topic hierarchy



COMPUTER-COMMUNICATION NETWORKS

0 items, 0 - 0 25 per page.

Keywords: (comma separated) COMPUTER COMMUNICATION NETWORKS, Network Architecture and Design, Asynchronous Transfer Mode (ATM), Centralized networks, Circuit switching networks, Distributed networks, Frame relay networks, ISDN (Integrated

Fig. 10. Topic Keywords

5.5 The PersoNews Database

The database is crucial to the functionality of PersoNews as all the other modules are constantly connected to it in order to insert or process data. Furthermore, the database stores the knowledge used by the machine learning algorithms in an efficient way. Currently the size of the database is approximately 1,4 GB, with 66 users and 1636 feeds.

Database tables can be divided in two large groups, those relevant with feeds and those relevant with topics. Tables of the first group contain information such as the feeds monitored by each user, which publications were classified as interesting or not interesting by users, the word stems contained in each publication, etc. Tables of the second group contain information such as the topics monitored by each user, topic publications considered as interesting or not, word stems contained in each topic publication, topic blacklists, etc.

5.6 Implementation Details

PersoNews is implemented using open source tools and software components. The operating system of choice for development and server deployment is Debian Linux 3.1[25]. Apache web server[26] is used to serve static and dynamic web content while the

[25] http://www.debian.org

MySQL[27] database management system is used for data storage. Application logic is implemented using PHP5[28] in conjunction with some open source software libraries such as PEAR[29], MagpieRSS[30] for feed manipulation and the Smarty Template System[31] for web content generation.

5.7 Hardware Requirements

PersoNews hardware requirements are quite modest and depend mainly on the number of active users as well as the number of feeds and topics monitored by the system. Minimum requirements are an Intel PIII CPU or AMD equivalent, 256 Mbytes of RAM and 5GB hard drive for the operating system, database and application files. Currently, PersoNews is set up in an Intel P4 1GB RAM PC. Resource usage varies and depends mainly on the current state of PersoNews modules. PersoNews.portal must operate constantly in order to cope with user browsing. On the other hand, PersoNews.aggregator operates periodically in order to aggregate data from feeds and update the system's database. PersoNews.email also operates periodically in order to notify users about new updates on the feeds and topics they monitor.

6. Preliminary System Evaluation

Unfortunately the system was up until recently in beta version and thus, although PersoNews is open to public[32], we did not encourage users to register. Although an exhaustive evaluation is in our immediate plans, we had a crude estimation of the system's performance taken from a small amount of registered users (66).

Over six months of usage we collected a false positive rate (percentage of messages that the classifier marked as junk but the users moved to the interesting folder[33]) nearly up to 6% for feeds and 11% for topics. These numbers however are misleading because of the existence of users that did not regularly gave user feedback (and therefore did not correct system's mistakes). By taking under consideration only the group of most active users the corresponding numbers are 29% and 30%. As we can see FP rate is higher in topics in both groups mainly because of diversity of content in the beginning. Numbers are of course getting better as time goes by if feedback is given. Although we do not claim statistical adequacy of this evaluation, we believe that these numbers are indeed encouraging. Moreover, the machine learning component used in personews is already evaluated in similar data (see section 4) so we assume that if

[26] http://www.apache.org
[27] http://www.mysql.com
[28] http://www.php.net
[29] http://pear.php.net
[30] http://magpierss.sourceforge.net/
[31] http://smarty.php.net
[32] http://news.csd.auth.gr
[33] As positive, we consider the characterization of a message as uninteresting

consistent feedback is given, accuracy will approximate levels that are presented in Section 4.

We also asked our beta-users to fill a questionnaire in order to measure user satisfaction. The questionnaire consists of 5 questions concerning personal data and news reading habits, 10 questions from the questionnaire described in (Chin et al., 1988) concerning easiness of learning and system capabilities, and 10 more questions about the features and functionalities of PersoNews. In general, the users were satisfied with the system capabilities and they considered it quite easy to learn and use. Most of them declared that although they spent more time reading news with PersoNews than before, they usually find more interesting articles. Many of them admitted that they did not always give feedback to the system and consequently they noted that the system adapts to their interests with mediocre speed. Users that gave feedback more frequently reported that the system adapts fast enough.

7. Conclusions and Future Work

We made an effort to reconstitute the conception of intelligent news readers. We implemented a Personalized News Reader, enhanced by a specially designed Machine Learning Framework and a primitive form of Semantic Filtering. The main features of PersoNews are:

a) The straightforward aggregation of numerous different news sources. We accomplish this by utilizing the RSS and OPML protocols.

b) The filtering of uninteresting articles on behalf of the users alleviating information overload. This is achieved by embedding an especially designed framework for text stream classification that can execute in a dynamic feature space. Note that intelligent filtering takes places, separately, in every feed the user is subscribed to.

c) The option we give to the user to subscribe and watch a more general topic of interest. We attain this goal by employing a simple form of semantic filtering. Filtering is also aided by machine learning in order to achieve further personalization.

Currently we are working on adding content in PersoNews from web pages that do not offer an RSS Feed. Our efforts are focused in web content extraction technology and the modeling of the extraction rules with the use of conceptual graphs (Kokkoras et al., 2007).

Aside from an extensive evaluation of PersoNews, it is in our future plans to make the hierarchy offered by the system fully customizable, meaning that the user could add certain concepts in any level of the hierarchy. We also plan to investigate alternative machine learning frameworks which combine good scalability and performance. Another interesting extension of the system would be to incorporate collaborative filtering methodologies and combine them with the content-based recommendation already made by the classification algorithms (Kim et al., 2006).

8. References

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Paliouras, G. and Spyropoulos, C. D. (2000). An Evaluation of Naive Bayesian Anti-Spam Filtering. Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), Barcelona, Spain.

Banos, E., Katakis, I., Bassiliades, N., Tsoumakas, G. and Vlahavas, I. (2006). PersoNews: A Personalized News Reader Enhanced by Machine Learning and Semantic Filtering. 5th International Conference on Ontologies, DataBases and Applications of Semantics (ODBASE 2006), Montpellier, France, Springer-Verlag.

Bharat, K., Kamba, T. and Albers, M. (1998). "Personalized, interactive news on the web." Multimedia Systems **6**(5): 349-358.

Billsus, D. and Pazzani, M. (1999). A Hybrid User Model for News Story Classification. Seventh International Conference on User Modeling, Banff, Canada, Springer-Verlag.

Carreira, R., Crato, J. M., Goncalves, D. and Jorge, J. A. (2004). Evaluating adaptive user profiles for news classification. 9th International Conference on Intelligent user Interface, Funchal. Madeira, Portugal, ACM Press.

Chan, C.-H., Sun, A. and Lim, E.-P. (2001). Automated Online News Classification with Personalization. 4th International Conference of Asian Digital Library (ICADL2001), Bangalore, India.

Chin, J. P., Diehl, V. A. and Norman, K. L. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. SIGCHI Conference on Human factors in computing systems, Washington, D.C., United States, ACM Press.

Dumais, S., Platt, J., Heckerman, D. and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. Proceedings of the seventh international conference on Information and knowledge management. Bethesda, Maryland, United States, ACM Press.

Fan, W. (2004). Systematic data selection to mine concept-drifting data streams. Tenth ACM SIGKDD international conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, ACM Press.

Hulten, G., Spencer, L. and Domingos, P. (2001). Mining time-changing data streams. Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, ACM Press.

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant Features. ECML-98, 10th European Conference on Machine Learning, Springer Verlag.

Katakis, I., Tsoumakas, G. and Vlahavas, I. (2006). Dynamic Feature Space and Incremental Feature Selection for the Classification of Textual Data Streams. ECML/PKDD-2006 International Workshop on Knowledge Discovery from Data Streams, Berlin, Germany, Springer Verlag.

Kim, B. M., Li, Q., Park, C. S., Kim, S. G. and Kim, J. Y. (2006). "A new approach for combining content-based and collaborative filters." Journal of Intelligent Information Systems **27**(1): 79-91.

Klinkenberg, R. (2004). "Learning Drifting Concepts: Example Selection vs. Example Weighting " Intelligent Data Analysis, Special Issue on Incremental Learning Systems Capable of Dealing with Concept Drift **8**(3): 281-200.

Kokkoras, F., Bassiliades, N. and Vlahavas, I. (2007). Cooperative CG-Wrappers for Web Content Extraction. 15th International Conference on Conceptual Structures, ICCS'07, Sheffield, UK.

Laskov, P., Gehl, C., Kruger, S. and Muller, K.-R. (2006). "Incremental Support Vector Learning: Analysis, Implementation and Applications." Journal of Machine Learning Research **7**: 1909-1936.

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval. Copenhagen, Denmark, ACM Press.

Lewis, D. D. and Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US.

McCallum, A. and Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. AAAI-98 Workshop on Learning for Text Categorization.

Porter, M. F. (1980). "An algorithm for suffix stripping." Program **14**(3): 130-137.

Scholz, M. and Klinkenberg, R. (2007). "Boosting classifiers for drifting concepts." Intelligent Data Analysis **11**(1): 3-28.

Schutze, H., Hull, D. A. and Pedersen, J. O. (1995). A Comparison of Classifiers and Document Representations for the Routing Problem. SIGIR '95, 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, ACM Press.

Sebastiani, F. (2002). "Machine Learning in Automated Text Categorization." ACM Computing Surveys **34**(1): 1-47.

Tsymbal, A. (2004). The problem of concept drift: Definitions and related work. Technical Report. Dublin, Ireland, Department of Computer Science, Trinity College.

Wenerstrom, B. and Giraud-Carrier, C. (2006). Temporal Data Mining in Dynamic Feature Spaces. Sixth International Conference on Data Mining.

Widmer, G. and Kubat, M. (1996). "Learning in the Presense of Concept Drift and Hidden Contexts." Machine Learning **23**(1): 69-101.

Witten, I. and Frank, E. (2005). Data Mining: Practical Machine Learning tools and techniques", 2nd Edition. San Francisco.

Yang, Y. (1994). "An example-based mapping method for text categorization and retrieval." ACM Transactions on Information Systems **12**(3): 252-277.

Yang, Y. (1994). Expert Network: Effective and efficient learning from human decisions in text categorization and retrieval. 17th Annual International ACM SIGIR conference on research and development in information retrieval, Dublin, Ireland, Springer-Verlag.

Yang, Y. and Pedersn, J. O. (1997). A Comparative Study on Feature Selection in text Categorization. ICML-97, 14th International Conference on Machine Learning, Morgan Kaufmann.