An Assessment of Case-Based Reasoning for Spam Filtering

Sarah Jane Delany[1], Pádraig Cunningham[2], Lorcan Coyle[2]

[1]Dublin Institute of Technology, Kevin St., Dublin 8, Ireland.
SarahJane.Delany@comp.dit.ie
[2]Trinity College Dublin, College Green, Dublin 2, Ireland.
{Padraig.Cunningham,Lorcan.Coyle}@tcd.ie

Abstract. Because of the changing nature of spam, a spam filtering system that uses machine learning will need to be dynamic. This suggests that a case-based (memory-based) approach may work well. Case-Based Reasoning (CBR) is a *lazy* approach to machine learning where induction is delayed to run time. This means that the case base can be updated continuously and new training data is immediately available to the induction process. In this paper we present a detailed description of such a system called ECUE. We compare its performance with an alternative system that uses Naïve Bayes (NB). We find that there is little to choose between the two alternatives in cross-validation tests on data sets. However, ECUE does appear to have some advantages in tracking concept drift over time.

1 Introduction

Spam classification is a challenging task for a number of reasons. Not least of these is the fact that something of an "arms race" has developed between spammers and the filtering systems developed to identify spam. The content and structure of spam messages is constantly changing as spammers attempt to bypass the techniques used by the filtering systems to catch the spam. This poses a difficult challenge as systems need to identify and learn new types of spam as this arms race continues.

Lazy learning is good for dynamically changing situations. With lazy learning the decision of how to generalise beyond the training data is deferred until each new unseen instance is considered. In comparison to this, eager learning systems determine their generalisation mechanism by building a model based on training data in advance of considering any new unseen instances. In this paper we present E-mail Classification Using Examples (ECUE), a lazy learning system using CBR that seamlessly incorporates new training data.

Another challenge facing effective spam filtering using machine learning is dealing with large amounts of training data. A dynamic system which integrates new training data will require some means of managing the training data. CBR research offers a number of case-base management techniques to remove noisy and redundant training data and so effectively manage the size of the training data or case base over time.

ECUE incorporates an effective case-based editing technique [1] which allows the number of training cases to remain at a manageable and efficient level.

The existing research on using a memory or case-based based approach [2,3] has a number of limitations. Firstly the evaluation is based on a restrictive data set incorporating legitimate email messages sent to a linguistics mailing list and "old-fashioned" spam emails that contain few of the obfuscations common in spam email today. Secondly all evaluations are static evaluations and do not take into account the changing nature of spam. In addition to static cross-validation tests, our evaluation of the approach presented in this paper includes dynamic evaluation of two independent datasets of over 10,000 email messages each received over the period of a year.

This paper begins with an overview of other work using machine learning techniques in spam filtering in Section 2. Section 3 presents ECUE, our case-based spam filtering approach and describes the feature selection, case retrieval and case-base management techniques we use. A preliminary evaluation of ECUE and comparison with NB is presented in Section 4. Section 5 outlines directions for future work while our conclusions are presented in Section 6.

2 Spam Filtering and Machine Learning

Existing research on using machine learning for spam filtering primarily uses NB as the technique of choice [2,4-6] with many unpublished implementations reported on the Web. In addition to NB there has been work using Support Vector Machines [7,8], Latent Semantic Indexing [9], and work using memory based classifiers [2,3,10]. Sakkis *et al.* [3] reported that their memory based classifier compared favourably to NB for spam filtering mailing lists and newsgroups while our preliminary findings [10] suggested that CBR would outperform NB.

Algorithms incorporating the NB classifier have proven to be among the most successful learners in the categorisation of text documents [11] and are good for high dimension data, hence their popularity in spam classification.

2.1 Naïve Bayes for Text Classification

NB is a probabilistic classifier that can handle a large number of features that other machine learning techniques cannot. It is 'naïve' in the sense that it assumes that the features are independent.

Consider a group of documents that are labelled as one of a set of classifications $c_i \in C$. Each document is described by a set of attributes $\{a_1,a_2,\ldots,a_n\}$ where a_i. indicates the presence of that attribute in the document. The classification returned from a NB classifier for a new document is given in Equation 1.

$$c_{NB} = \arg\max_{c_i \in C} P(c_i) \prod_j P(a_j \mid c_i) \quad (1)$$

Due to the significance of false positives (legitimate emails identified incorrectly as spam) in spam filtering, the NB classifier is not generally used in this simple *argmax* form. In practice the classification threshold is set to bias the classifier away from false positives (see Section 4.2.3).

The conditional probabilities can be estimated by $P(a_i \mid c_j) = n_{ij} / n_j$ where n_{ij} is the number of times that attributes a_i occurs in those documents with classification c_j and n_j is the number of documents with classification c_j. This provides a good estimate of the probability in many situations but in situations where n_{ij} is very small or even equal to zero this probability will dominate, resulting in an overall zero probability. A solution to this is to incorporate a small-sample correction into all probabilities called the Laplace correction [12]. The corrected probability estimate is $P(a_i \mid c_j) = (n_{ij} + f) / (n_j + f \times n_{ki})$, where n_{ki} is the number of values for attribute a_i. Kohavi *et al.* [13] suggest a value of $f = 1/m$ where *m is* equal to the number of training documents.

3 Case-Based Spam Filtering

This section describes ECUE, the case-based system we have implemented for spam filtering. The description includes details of the feature selection, case retrieval and case-base editing techniques we used.

In a CBR learner, examples in the training data are represented as cases in a case base. For the spam filtering domain, each training example email is a case e_i represented as a vector of attributes or features $x_i = (a_1, a_2, \ldots, a_n, s)$. Features a_i are represented as binary features, i.e. if the feature exists in the email, $a_i = 1$, otherwise $a_i = 0$ whereas feature s represents the classification either spam or non-spam. It is more normal in text classification for lexical features to carry frequency information but our evaluation showed that a binary representation works better in this domain. We expect that this is due to the fact that most email messages are short and frequency information may result in overfitting. Features were identified using a variety of generic lexical features, primarily by tokenising the email into words. No domain specific feature identification was performed at this stage although work by Sahami *et al.* [5] has indicated that the effectiveness of filters will be enhanced by their inclusion.

3.1 Feature Selection

Tokenising 1000 emails results in a very large number of features, (tens of thousands of features). Feature selection is necessary to reduce the dimensionality of the feature space. Yang and Petersen's [15] evaluation of dimensionality reduction in text categorisation found that Information Gain (IG) [14] was one of the top two most effective techniques for aggressive feature removal without losing classification accuracy. We calculated the IG of each feature and the top 700 features were selected. Our cross validation experiments, varying between 100 and 1000 features across 4 datasets, indicated best performance at 700 features.

3.2 Case Retrieval

A CBR learner assigns a classification to a previously unseen example or target case by identifying and analysing the training cases that are most similar to it. Most of

these classifiers use the *k*-NN algorithm to determine the *k* most similar training cases and then use these to classify the target case. The standard *k*-NN algorithm calculates similarity on a case-by-case basis. This approach is quite inefficient in domains where there is feature-value redundancy and/or missing features in cases [16]. Because our spam cases have both of these characteristics, we use an alternative similarity retrieval algorithm based on Case Retrieval Nets (CRNs) [16].

A CRN is a memory structure which allows an efficient yet flexible retrieval of cases. They borrow ideas from neural networks and associative memory models. They are made up of a number of components.*Case nodes* represent stored cases. *Information Entity Nodes* (IEs) represent feature-value pairs within cases. *Relevance Arcs* link case nodes with the IEs that represent them. They have weights that capture the importance of the IE. Lastly, *Similarity Arcs* connect IEs that refer to the same features, and have weights relative to the similarity between connected IEs.

The idea behind the CRN architecture is that a target case is activated by connecting it to the net via a set of relevance arcs and this activation is then spread across the net. Each of the other case nodes accumulates an activation score appropriate to its similarity to the target case. The case nodes with the highest activation are the most similar cases to the target case.

We implemented a CRN for case retrieval that was configurable for different *k*-nearest neighbour classifiers. As the features in our case representation are binary, IEs are only included for features with a *true* value and similarity arcs are not needed. The relevancy arcs are all weighted with a weight of 1. We evaluated feature weighing (including a weight equal to the IG value of the feature identified during the feature selection process) but no significant improvements were found.

Fig. **1** depicts an example of our CRN for spam filtering. Our implementation of the CRN is similar in some respects to a Concept Network Graph (CNG) [17] with thresholds set so that the activations are not spread beyond the first level of nodes.

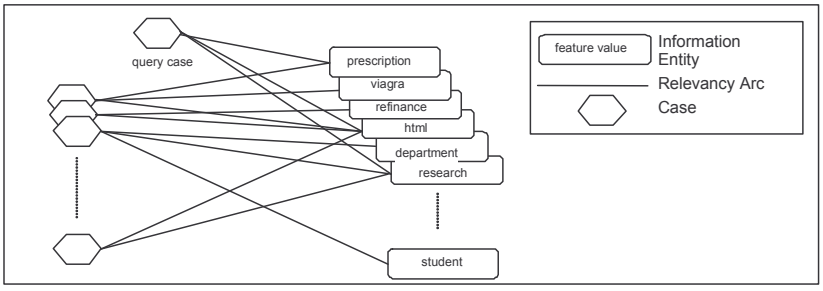

Fig. 1. Case Retrieval Net

3.3 Case Base Management

Research to date on machine learning for spam filtering has focused on static evaluations on datasets of manageable size. For instance, the LingSpam corpus [3,9] contains 481 spam emails. Since a working spam filter could face this number of spam messages in a week there is a need to actively manage the training data. A key

step in managing the training data is the case base editing process that deletes noisy examples and removes redundant cases from the case base.

Case base editing techniques involve reducing a case base or training set to a smaller number of cases while endeavouring to maintain or even improve the generalisation accuracy. There is significant research in this area [18-20]. The case base editing technique that we used is Competence Based Editing [1] that builds a competence model of the case base and uses it to determine the cases to remove.

4 Evaluation

This section presents our evaluation of ECUE. Two types of evaluation were performed. Firstly the generalisation accuracy achieved by ECUE was evaluated against that achieved using NB on four static datasets of 1000 emails each. Secondly the performance of ECUE in a dynamic situation was evaluated over a period of a year using 2 datasets of over 10,000 emails each. This second evaluation was not an online evaluation but an offline one using data collected over a year.

It is worth noting that rudimentary feature extraction techniques, described in Section 3, were used for these evaluations. To achieve a high performance comparable with existing commercial spam filtering systems, such as Spamassassin, "commercial grade" feature extraction techniques need to be implemented.

4.1 Static Classifier Comparison

A key objective was to evaluate the generalisation accuracy of ECUE using a *k*-NN classifier with different values of *k*. Four datasets were used. The datasets were derived from two corpora of spam and legitimate email collected by two individuals over a period of approximately eighteen months up to and including December 2003 for Dataset 1 and up to and including January 2004 for Dataset 2. The legitimate emails in each corpus include a variety of personal, business and mailing list emails.

Two datasets of one thousand cases were extracted from each corpus. Each included five hundred spam emails and five hundred non-spam or legitimate emails. Datasets Feb-1 and Feb-2 consisted of 500 consecutive spam and legitimate emails received up to and including February 2003 while Datasets Nov-1 and Nov-2 consisted of 500 spam and legitimate consecutive emails received between February 2003 and November 2003. Given the evolving nature of spam it was felt that these datasets gave a representative collection of spam.

In both datasets the emails were not altered to remove HTML tags and no stop word removal, stemming or lemmatising was performed. Since the datasets were personal it was felt that certain headers may contain useful information, so a subset of the header information was included in the tokenisation. For each dataset we used 20 fold cross-validation, dividing the dataset into 20 stratified divisions or folds. Each fold in turn is considered as a test set with the remaining 19 folds acting as the training set.

A number of *k*-NN classifiers were evaluated. Once the *k*-NN classifier returns the cases that are determined to be closest to the query case, a voting algorithm is implemented to determine the classification of the query case. For this evaluation we

used a distance weighted voting algorithm. The vote returned for classification c_i for query case x_q, over the k nearest neighbours $x_1,\ldots,x_k$ using distance weighted voting is given in Equation 2 where $1(a,b)=1$ if $a=b$, $1(a,b)=0$ if $a\neq b$, w_j is given in Equation 3 and c_j is the classification of neighbour x_j. The classification with the highest vote is deemed to be the classification of the query case.

$$Vote(c_i)=\sum_{j=1}^{k} w_j 1(c_j,c_i) \tag{2}$$

$$w_j=\left(\sum_{m=1}^{n}\left|a_m(x_q)-a_m(x_j)\right|\right)^2 \tag{3}$$

The votes for spam and non spam are normalised and the spam normalised vote is compared with a set threshold. If the spam vote is greater than the threshold the query case is considered to be spam. By varying the threshold from 0 to 1 and plotting the resulting rate of False Positive (FP) classifications (legitimate emails classified incorrectly as spam), against 1 minus the rate of False Negative (FN) classifications (spam emails classified incorrectly as legitimate), an ROC curve can be plotted [21].

In order to compare ECUE with the current spam filtering technique of choice, a NB classifier was implemented using the algorithm described in Section 2.1. Normalising the probabilities returned by the NB algorithm and varying the threshold for a spam classification as described above allowed an ROC curve to be plotted for the NB classifier. The larger the area under the curve for an ROC curve, the better the classifier. The results of the best *k*-NN classifier, for an edited and non edited case base and the NB classifier are presented in Fig. 2. To show the detail of the curve more clearly, only the top left hand corner of the graphs are presented.

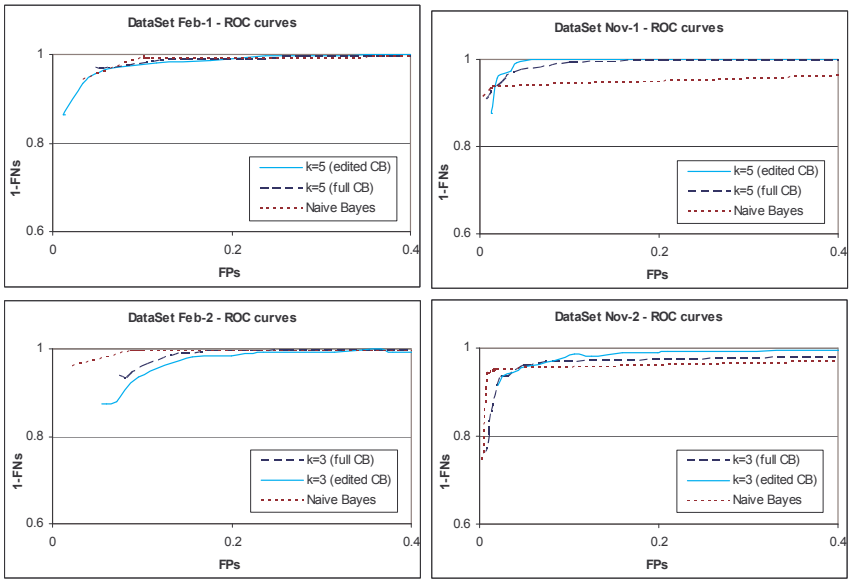

Fig. 2. Results of comparing different classifiers

The results presented above do not show that one classifier outperforms in all cases. NB seems to perform best in the February datasets while the *k*-NN classifier on an edited case base performs best in the November datasets.

4.2 Dynamic Evaluation

We also evaluate how ECUE performs over a period of time, allowing the system to dynamically update its training data with examples of spam and legitimate email that were incorrectly classified.

4.2.1 Experimental Setup

Two datasets were used. The datasets were derived from the same two corpora of email as described above. A case base of 1000 cases, 500 spam emails and 500 legitimate emails were set up in each case. This training data included the last 500 spam and non spam emails received up to and including February 2003 in the case of Dataset 1 and up to and including January 2003 in the case of Dataset 2. This left the remainder of the data for testing. Table 1 shows the number of spam and legitimate emails received each month for both datasets.

A case base was set up for each training dataset. The classifier selected was *k*-nearest neighbour with $k = 3$. Due to the fact that an FP is much more serious than an FN, the classifier used unanimous voting to determine whether the target case was spam or not. All neighbours returned had to have a classification of spam in order for the target case to be classified as spam. This corresponds to the leftmost point on the ROC curve in Fig. 2. This strongly biases the classifier away from false positives.

Each case base was edited using the *k*-NN classifier with $k = 3$ and the CBE editing technique. Our previous experiments with case editing using CBE and a unanimous voting classifier indicated that generalisation accuracy increased using an edited case base [1]. Each email in the testing datasets, documented in Table 1, was presented for classification in date order to closely simulate what would happen in a real-time situation.

Table 1: Profile of the testing data

		Feb '03	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan '04	Tot
Data Set 1	spam		629	314	216	925	917	1065	1225	1205	1830	576		**8902**
	non spam		93	228	102	89	50	71	145	103	85	105		**1076**
Data Set 2	spam	142	391	405	459	406	476	582	1849	1746	1300	954	746	**9456**
	non spam	151	56	144	234	128	19	30	182	123	113	99	130	**1409**

4.2.2 Evaluation Metrics

Since FPs are much more serious that FNs, accuracy (or error) as a measure of performance does not present the full picture. Two filters with similar accuracy may have very different FP and FN rates.

In previous work on spam filtering a variety of measures have been used to report performance. The most common performance metrics are precision and recall [9]. Sakkis *et al.* [3] introduce a weighted accuracy measure which incorporates a measure

of how much more costly an FP is than an FN. Although these measures are useful for comparison purposes, the FP and FN rate are not clear so the true effectiveness of the classifier is not evident. For these reasons we will use the rate of FPs, the rate of FNs, and the average within class error rate, *AvgError* = (*FPRate*+*FNRate*)/2 as our evaluation metrics. A final justification for using this set of metrics is that this is in line with how commercial spam filtering systems are evaluated on the web and in the technical press.

4.2.3 Evaluation Methods

A number of experiments were performed, varying from making no updates to the original case base to updating the case base on a monthly, weekly and daily basis with those emails that were misclassified over the specified period. Our evaluation showed the best performance occurred when updating the case base on a daily basis with any emails misclassified that day. These results are presented in Fig. 3.

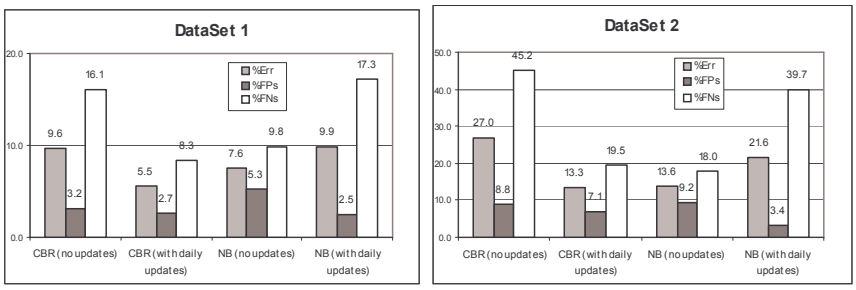

Fig. 3. Results of evaluations over a period of time

The same experiments were performed using the NB classifier on unedited training data. The training data could not be edited for the NB classifier as the editing technique is a competence-based editing technique which uses a *k*-NN classifier to determine the competence of each case in the case base and analyses the competence properties of the cases to determine which cases should be removed. Due to the significance of FPs, the NB classifier was configured to be biased away from false positives by setting the threshold equal to 1.0. Fig. 3 includes the results of using NB.

4.2.4 Results

Although NB has a lower overall error rate over the datasets with no updating, the CBR system performs better in both datasets when dynamically updating the data to learn from incorrectly classified emails. It can be seen that daily updating of the training data with misclassified emails improves performance of the CBR system but has an overall detrimental effect on the NB classifier. NB with daily updates does improve the FP rate more than ECUE but the degradation of the FN rate has an overall negative effect on performance.

CBR only needs individual marker cases to construct its model whereas NB requires a full concept description. This may affect the performance of the NB classifier however the need to train the NB classifier on a full set of data presents its own set of data management problems.

It is worth noting that updating a system using NB with any new training data requires a separate learning process to recalculate the probabilities for all features. Updating a CBR system, such as ECUE, with new training data simply requires new cases to be added to the case base.

5 Future Work

The focus of the research presented in this paper is on the case base classifier's ability to dynamically update the training data as new examples of spam and non spam are encountered. However, we envisage a hierarchy of learning within this domain where this continuous updating with misclassified emails is only the first level within three levels of learning.

As time passes and spam changes, the features selected for earlier training data may not be as predictive for new training examples. The second level of learning is to re-train the classifier by performing the feature selection process on the updated training data. This level of retraining may need to be performed infrequently, e.g. every month or every other month. The highest level of learning, performed even more infrequently than feature selection, is to allow new feature extraction techniques to be added to the system. For instance, when domain specific features are used in the system, new feature extraction techniques will allow new features to be included. The benefit of using a CRN for implementing the second and third levels of learning is that it can easily handle cases with new features. The fact that these features may be missing in old cases is not a problem.

Future work on our CRN will also incorporate CNG-type activation spreading to allow cases that do not include the actual selected features to influence the classification process.

6 Conclusions

Using CBR for spam filtering is certainly no worse than using NB. As techniques which utilise a probabilistic classifier to detect spam e-mail are already patented [23], it is necessary to find other techniques which offer at least comparable results. In fact, our research suggests that CBR demonstrates better performance for learning over time than NB. CBR as a lazy learner offers significant advantages; it provides capabilities to learn seamlessly without the need for a separate learning process and facilitates extending the learning process over different levels of learning.

References

1. Delany SJ., Cunningham P.: An Analysis of Case-Based Editing in a Spam Filtering System, to be presented at 7th European Conference in Case-Based Reasoning (2004)
2. Androutsopoulos, I., Koutsias, J., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., Stamatopoulos, P.: Learning to Filter Spam E-mail: A Comparison of a Naive Bayesian and a Memory-Based Approach. 4th PKDD Workshop on Machine Learning and Textual Information Access. (2000)

3. Sakkis G., Androutsopoulos I., Paliouras G., Karkaletsis V., Spyropoulos C.D., Stamatopoulos P.: A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists. Information Retrieval, 6 (1), Kluwer Academic Publishers (2000) 49-73
4. Pantel P. Lin D.: SpamCop: A Spam Classification and Organization Program. Learning for Text Categorization—Papers from the AAAI Workshop, Madison Wisconsin. AAAI Technical Report WS-98-05 (1998) 95–98
5. Sahami M., Dumais S., Heckerman D., Horvitz E.: A Bayesian Approach to Filtering Junk Email. AAAI-98 Workshop on Learning for Text Categorization. AAAI Technical Report WS-98-05 (1998) 55-62
6. Androutsopoulos I., Koutsias J., Konstantinos V., Chandrinos V., Paliouras G., Spyropoulos C.: An evaluation of Naive Bayesian Anti-Spam Filtering. Proc. of the Workshop on Machine Learning in the New Information Age. G. Potamias, V. Moustakis and M. van Someren (eds.), 11th European Conference on Machine Learning, Barcelona, Spain (2000) 9-17
7. Androutsopoulos I., Paliouras G., Michelakis E.: Learning to Filter Unsolicited Commercial E-Mail, Tech rpt 2004/2, NCSR "Demokritos", (2004) http://www.iit.demokritos.gr/skel/i-config/publications/
8. Drucker HD., Wu D., Vapnik V.: Support Vector Machines for Spam Categorization. IEEE Transactions On Neural Networks, 10(5) (1999) 1048–1054
9. Gee K.R.: Using Latent Semantic Indexing to Filter Spam. Proceedings of the 2003 ACM Symposium on Applied Computing (SAC) Melbourne, FL, USA. ACM (2003) 460-464
10. Cunningham P., Nowlan N., Delany S.J., Haahr M.: A Case-Based Approach to Spam Filtering that Can Track Concept Drift, The ICCBR'03 Workshop on Long-Lived CBR Systems, Trondheim, Norway,. (2003)
11. Lewis D., Ringuette M.: Comparison of Two Learning Algorithms for Text Categorization. SDAIR (1994) 81-93
12. Niblett: Constructing Decision Trees in Noisy Domains. Proceedings of the Second European Working Session on Learning. Bled Yugoslavia Sigma (1987) 67-78
13. Kohavi R., Becker B., Sommerfield D.: Improving Simple Bayes. ECML-97 Proceedings of the Ninth European Conference on Machine Learning (1997)
14. Quinlan J. R.: C4.5 Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA. (1997)
15. Yang Y., Pedersen J.O.: A Comparative Study on Feature Selection in Text Categorization. Proc of 14th Int Conf on Machine Learning. Nashville US (1997) 412–420.
16. Lenz M., Auriol E., Manago M.: Diagnosis and Decision Support. M. Bartsch-Sporl, H. D. B., and Wess, S., eds., Case-Based Reasoning Technology: From Foundations to Applications, LNCS Vol 1400. Springer-Verlag (1998)
17. Ceglowski M., Coburn A., Cuadrado J.: Semantic Search of Unstructured Data using Contextual Network Graphs. http://www.nitle.org/semantic_search.php
18. McKenna E., Smyth B.: Competence-Guided Editing Methods for Lazy Learning. Proceedings of the 14th European Conference on Artificial Intelligence, Berlin (2000)
19. Wilson D, Martinez T.: Instance Pruning Techniques. Proc. of 14th Int. Conf on Machine Learning, Fisher D. (ed.) Morgan Kaufmann, San Francisco, C.A. (1997) 404-411
20. Brighton H., Mellish C.: Advances in Instance Selection for Instance-Based Learning Algorithms. Data Mining and Knowledge Discovery, Vol 6. Kluwer Academic Publishers (2002) 153-172
21. Bradley A.P.: The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition, Vol 30. (1997) 1145-1150.
22. Baeze-Yates R., Ribeiro-Neto B.: Modern Information Retrieval. Addison-Wesley, ACM Press New York (1999)
23. United States Patent 6,161,130. (2000)