Real-World Text Clustering with Adaptive Resonance Theory Neural Networks

L. Massey
Royal Military College of Canada
Department of Mathematics and Computer Science
Kingston, Ontario, Canada, K7K 7B4
E-mail: massey@rmc.ca

Abstract – **Document clustering has been an important research area in recent years. However, most work on this subject has focused on batch processing in a static environment. For real-world applications, on-line and incremental processing of highly dynamic data is required. Adaptive Resonance Theory (ART) neural networks possess several interesting properties that make them appealing as a potential solution to this problem. In this paper, we present preliminary experimental results that examine ART text clustering under several situations characteristic of real-life applications. We also compare our present results with work we have conducted previously on the batch static case, hence determining how clustering quality is affected by incremental processing.**

I. INTRODUCTION

An enormous quantity of human knowledge exists in the form of electronic text. The difficulty is to find, within the hump of documents, the information one needs. To respond to this challenge, multiple document categorization [1] and text clustering [2] methods have been proposed. These approaches share the common objective of organizing documents by topics to facilitate information access.

Document categorization is a supervised learning task that aims at learning a classifier that will then be able to organize new documents by topics. The topics are predefined and the learning process is supported by a large labeled training set. The problem with document categorization is that it is not a plastic form of learning. Indeed, realistic document collections are usually dynamic, with new documents and new topics being added regularly. Therefore, the classifier must be re-trained on a new labeled sample of the data. This makes document categorization ill suited for the task of organizing real-life, dynamic document collections [3].

Contrary to document categorization, text clustering neither requires a labeled training set nor a known set of topics to guide learning. Clustering instead relies on the notion of similarity, usually correlation such as the inner product or dissimilarity such as distance metrics [4], to decide which documents belong to a common cluster. Particularly, incremental clustering [5, 6] has a natural ability to detect and integrate novelty. Hence, in the context of document organization in a realistic environment, incremental clustering seems better suited than categorization. However, incremental clustering suffers from one major problem: although plastic, it is not necessarily stable. That is, as new documents are added to the collection, an infinite number of new categories can be created and a document can oscillate between multiple topics indefinitely [7]. Both stability and plasticity of learning are hence essential properties of a system aiming at organizing dynamic document collections.

II. ADAPTIVE RESONANCE THEORY NEURAL NETWORKS

Adaptive Resonance Theory (ART) neural networks [8] were designed exactly to address the problem of plastic and stable learning. Furthermore, ART is known to converge in relatively few iterations compared to other artificial neural nets. Indeed, ART should be capable of efficiently clustering new documents, detecting novelty, creating the relevant new topics and integrating new knowledge while preserving stability. ART therefore seems like the candidate of choice for the task of document clustering in a high demand dynamic environment. However, up to now ART-based text clustering has focused on the static batch mode [9]-[11], including our own work [12]. One notable exception is Rajaraman & Tan [13] but their emphasis was on Topic Detection and Tracking (TDT), and particularly on topics trend analysis of time ordered text streams, i.e. news feeds. Although similar to TDT, our task is more general in that it applies to any document collection, such as intranets, document management systems databases and even users computer hard drives.

Hence, ART has not been studied at all in a realistic, dynamic text clustering environment. Its behavior and capabilities at this task are consequently unknown, although this task would seem to be an ideal domain of application for ART networks. In this paper, we propose and test the use of binary ART (ART1) neural networks to address this important problem.

ART1 networks consist of two layers of neurons: N input neurons and M output neurons, where N is the input size and M the number of categories (or topics). Neurons are fully connected with both feed-forward and feedback weighted links. The feedforward links connecting to the output neuron j are represented by the real vector W_j while the feedback links from that same neuron are represented by the binary vector T_j. The latter stores the prototype for category j. The algorithm we used to simulate an ART1 network follows:

```
1. Initialize network weights and provide
   parameter values:
       $0 < \rho \leq 1$ (the vigilance parameter)
       $L > 1$
       $W_j = 1/(1+N)$ for all connections
       $T_j = 1$ for all connections
2. $u_j = 0$ for j=1..M and present a document $d_k$ to the
   network
3. Compute output activations, for j=1..M:
       $u_j = d_k \wedge W_j$ (" $\wedge$ " is the logical AND)
4. Competition: select output neuron j* = $\max_j (u_j)$
   as winner; if all neurons $u_j = -1$, create and
   initialize a new neuron, and set M to M+1.
5. Vigilance test: determine if j* is close enough
   to $d_k$ :
       $\| d_k \wedge T_j\| / \| d_k\| \geq \rho$
   If true, step 6 (resonance mode); otherwise,
   step 8 (search mode).
6. Update weights:
       $T_j = T_j \wedge d_k$
       $W_j = L(d_k \wedge T_j)/(L - 1 + \| d_k \wedge T_j \|)$
7. Return to step 2 with a new document.
8. $u_{j*} = -1$ (remove category j* from current search)
   and return to step 4
```

III. Experimental setup

As for our previous work on the static case [12], we use the F_1 [14] clustering quality value to allow comparison with those previous results. F_1 is a well-known measure of quality in supervised text categorization [1] and in text clustering [6]. It can be computed with the same pair-wise counting procedure employed by traditional clustering validation methods [15] to establish a count of false negatives and false positives, but combines these values following the F_1 formulae:

$$F_1 = 2pr / (p + r)$$

where:

$$p = a / (a + b) \text{ is the precision}$$
$$r = a / (a + c) \text{ is the recall;}$$

a is the pair-wise number of true positives, i.e. the total number of document pairs grouped together in the desired solution and that are clustered together by the clustering algorithm; b the pair-wise number of false positives and c is the pair-wise number of false negatives. A F_1 value of 1 indicates maximal quality and 0 worst quality. We also use the k-means [16] clustering algorithm in incremental mode to establish a reference quality level. The parameter k is set to the number of topics (93) specified by the domain experts who manually organized the Reuter text collection. K-means initial cluster centroids are determined randomly and clustering results are averaged over 10 trials to smooth out extreme values obtained from good and bad random initialization.

We perform our experiments on the "ModApté" split [17] of the benchmark Reuter-21578 corpus. The corpus provides the following pre-established document sets for supervised classification: training set, test set and discarded documents. For clustering, the training set is not required, so we only cluster the 3,299 documents in the test set. Using this specific data set and the F_1 quality measures makes our work comparable to supervised categorization, something we have done successfully in previous work [12]. We represent documents numerically according to the vector-space model [18]. In this model, a document is characterized by a feature set corresponding to the words present in the documents collection. Stop words such as articles and prepositions are first removed. If N is the number of words left in the collection, then each document d is translated into an N-dimensional binary vector. The vector's i^{th} component corresponds to the i^{th} word in the collection vocabulary. A value of 1 indicates the presence of this word in d while a value of 0 signifies its absence. Component i is fed in the ART network at input neuron i.

Contrary to most clustering algorithms, ART does not assume prior knowledge of the number of clusters M. Instead, the vigilance parameter $\rho \in (0,1]$ determines the level of abstraction at which ART discovers clusters, and thus their quantity. We will not use vigilance values that result in more than 200 clusters because such a large number of clusters (compared to the 93 expected) would result in information overload for a user, which is contrary to the objective of text clustering.

IV. Experimental results and discussion

1. *Feature selection and vocabulary growth*

When building the vector-space representation of the documents, the whole document collection is scanned and keywords extracted. This works in the static case since the collection is available prior to clustering. However, in the on-line incremental case, the documents are received sequentially. Since documents in an on-line system should normally be processed immediately (for instance, for a military intelligence application), we seemingly have no choice but to accumulate the vocabulary incrementally. This is caused by one's a priori ignorance of terms usefulness, which could be based for instance on term frequency information if one had access to the whole document set initially. Incrementally computing term statistics and incrementally selecting features is not a good idea since the usefulness value of a term may change over time as documents are processed. This means that a decision on the usefulness of a feature made at time t and all subsequent clustering may need to be undone at time t' > t based on the accumulation of new evidence. This means that no word removal based on collection term statistics is possible: all the words minus stop words must be kept. The direct consequence is that the vector space representation will continually grow and the dimensionality will rapidly become very large. High dimensionality in turn means slower processing time. Furthermore and possibly worse, since the low frequency noisy or un-informative words are not removed, clustering will likely be of lower quality. Consequently, it is highly desirable to obtain term statistics to allow for the removal of some words. We are back to square one, but there is a possible solution to this problem.

Generally, the motivating factor to implement an automated document classification system in an organization arises from an existing large amount of electronic text data accumulated over the years. In such cases, a solution to the problem stated above is to use the legacy text as a *surrogate for the yet unseen documents*. Feature reduction is then based on the word statistics of the legacy text. In this work, we use the training and discarded documents of the ModApte split for that purpose. We tested this approach in the following manner. First, discarded and training documents were used to build the collection vocabulary and collect frequency information for each word. This amounts to a total of 18279 documents and 39853 terms once stop words are removed. Then, the least frequent terms (appearing in 3 documents or less) were removed from the vocabulary. This left N= 16888 words to be used as features. Yang & Pedersen [19] showed that this is a simple yet effective feature space dimensionality reduction method that also helps improve classification accuracy.

The next issue that must be addressed is how to handle vocabulary growth. That is, once the initial vocabulary has been built and a feature set extracted, what do we do with new words when they are encountered during operational use (in our case, this is when we process the test set)? We experiment with two strategies based on the datasets described below :

a) Data set NONE is built by using only the terms present in the initial vocabulary (obtained from the discarded and training documents). This amounts to ignoring new vocabulary items showing up during system operation (here, when processing the test set).

b) Data set ALL on the contrary is constructed based on the vocabulary of the discarded and training documents to which were added *all new terms* introduced by the test set (but only those not previously removed during feature reduction).

We also create data set TST which consists of terms extracted from the Reuter test set only. Feature selection is performed based solely on term statistics of the test set. This is the control data set, representing the static case. We then simulate a real-life operational use and compare the two vocabulary growth strategies by processing the 3299 document-vectors in datasets NONE and ALL. Each dataset is processed individually and at various values of vigilance, and for each case the F_1 quality is computed. We repeat the same process with dataset TEST. Figure 1 shows the resulting quality for each dataset at various vigilance values.

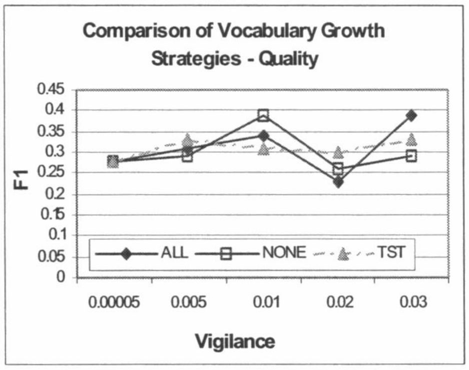

Fig. 1: F_1 quality at various vigilance levels for the two different vocabulary growth strategies ALL and NONE, compared to the static case TST.

We observe that there is no clear winner between the two vocabulary growth strategies, with data set ALL or NONE each taking turns in giving the best F_1 clustering quality. The same observation applies to a comparison with the quality obtained on the static control data set TST, which also yields no better results on average (average $F_1 = 0.31$). These results indicate that using legacy data for initial vocabulary creation and feature selection is a viable solution since it gives no worse results than the usual approach in the static case. Furthermore, it seems that new words introduced during operational use are not necessary to improve clustering quality. In fact, keeping new words has the undesirable effect of increasing dimensionality to N= 19932 (by over 3000 compared to dataset NONE), which penalizes computation speed with no gain in quality. However, one must not hastily draw general conclusions from this result. Given that the test set introduces over 3000 new terms, we suspect there may be a problem with vocabulary completeness in the training and discarded sets. Hence, if a more comprehensive corpus to build the initial feature set were used, new words would likely be more infrequent and may indeed become indicator of new content. Indeed, words like Internet, Web, ".com" might have been totally absent from an economic domain corpus like Reuter in the 1980's, but definitely important features since the mid-1990's with the rapid development of an Internet-related industry. Therefore, one cannot merely reject the strategy that consists in ignoring all new words.

2. Determination of the vigilance parameter value

Another issue for incremental clustering is the determination of the vigilance parameter values. In practice, finding an optimal or at least an acceptable vigilance level may be difficult. A possible approach is to conduct preliminary experiments on legacy data if labeled data is available to evaluate quality. Otherwise, one could rely on costly, time consuming and possibly subjective users evaluation. A possibly more realistic option is to use *minimal vigilance* [20], that is $\rho_{min} \leq 1/N$ where N is the number of features. Minimal vigilance is the smallest *effective* vigilance value; that is the smallest vigilance value that will affect the number of clusters. Hence, no further generalization is possible below that value, which means that the most general clusters are created at $\rho = \rho_{min}$. Therefore, this is an interesting solution to determining the vigilance value if the application requires few highly general clusters.

However, one question remains: are there other, better values for vigilance? To answer this question, we proceed as follows: we evaluate clustering results at various vigilance values starting at minimal vigilance and incrementing the vigilance value until more than 200 clusters are formed. This raises another important question: by what value should vigilance be incremented? Up to now we have interpolated quality between rather large vigilance intervals (see figure 1), but are there other, possibly better clusters resulting from intermediary vigilance values? Based on the vigilance test of ART1 algorithm (step 5):

$$\| d_k \wedge T_j\| / \| d_k\| \geq \rho$$

we observe that the left side is the ratio between the number of active binary features (i.e. set to 1) common between a document d_k and a cluster prototype T_j and between the number of active features in the document ($\|d_k\|$). The vigilance test then tells us that this ratio must be greater or equal to the vigilance value for the document to be assigned to cluster j represented by T_j. Hence, any change in vigilance that toggles the truth-value of the inequality will also change the resulting clustering solution. Since we are dealing with binary features, the left side can only vary in a discrete fashion. For example, letting $\| d_k\| = b$, some positive non-zero integer, $\|d_k \wedge T_j\|$ can only take a positive integer value a= [0,b]. Then, for a given document, a/b can only be one of the following "quantum levels": 0, 1/b, 2/b, …, (b-1)/b, 1, each level separated by a "gap" of 1/b. The maximum value b = $\| d_k\|$ can take is N, therefore changing vigilance by increments of less than 1/N is required to visit all possible clustering solutions. Hence, when looking for an optimal vigilance level, one must start at minimum vigilance and use increments smaller than 1/N. Figure 2 shows how interpolation with large vigilance increments (0.005) is missing much better clustering solutions than with smaller increments (0.001, 0.0005, and 0.00025). Another interesting and quite surprising observation is the very high variability in quality, even for "neighboring" vigilance levels. Only at low vigilance is the quality relatively stable.

This experiment indicates that a comprehensive search for a good vigilance value can make a huge difference in clustering quality. However, in a real-life environment it might be difficult to perform such a search. One may then have to rely on minimal vigilance, which corresponds to an almost average quality over all vigilance levels.

3. *Stabilization*

ART converges to a stable representation after at most N-1 presentations of the data [21]. By stable, it is meant that if an identical document is presented several times to the network, it will be assigned to the same topic, and the cluster prototypes will neither change nor needlessly proliferate. Although some may consider that identical documents are a rarity and that this problem is therefore minor, we on the contrary claim that in some environments (such as an enterprise document management system), identical or nearly identical documents, which can include duplicates or revised versions, can exist in important numbers and that hence the problem must not merely be discarded. Furthermore, although actual documents may be different, they may have the same vector-space representation, thus making them identical to the system. Unstable clusters are problematic because they result in changing document-topic assignments and thus defeat the purpose of organizing the documents and helping information access.

Figure 3 shows that cluster quality increases for both vigilance levels tested after ART has stabilized, which is a further reason to seek a stable representation. Only 3 and 4 iterations respectively for ρ = 0.001 and ρ = 0.0435 were required to attain a stable representation, which is much less than the theoretical upper bound of N-1. This is to be expected since the N-1 upper bound is reached when all documents are presented in decreasing order of magnitude $\| d_k\|$, an unlikely situation. However, this still translates into 3-4 more times to process documents, which can be problematic in a real-time environment. To address this issue, stabilization could be scheduled to take place during system low activity periods. This would be analogous to "sleep time" during which the neural network consolidates acquired knowledge by "replaying" previous document submission events. The issue of how frequent stabilization operations should be must be

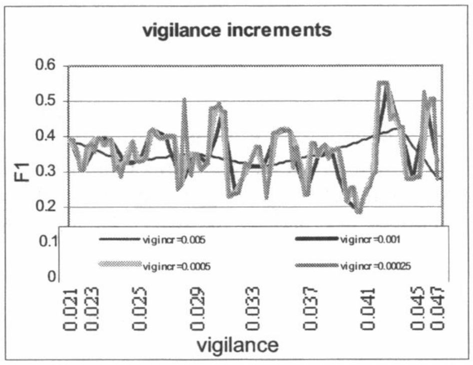

Fig. 2: Interpolation with large vigilance increments (0.005) is missing much better clustering solutions than with smaller increments (0.001, 0.0005 and 0.00025).

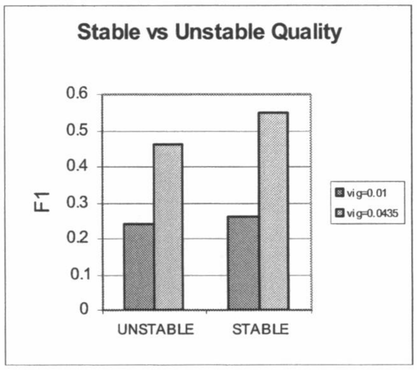

Fig. 3: Stabilization improves clustering quality.

addressed.

Furthermore, in real world, high-volume, 24/7 operations this could still be a problem as little idle time in the system operation may be available to stabilize. Finally, one must question what happens in between stabilization operations with new documents awaiting stabilization. Documents will be assigned to some clusters, then once stabilization takes place, documents may be moved, defeating the whole purpose of providing a stable and consistent environment to users. In fact, the whole idea of stabilization rests on the premise that convergence to the so-called stable representation is achieved after the ART network has been able to iterate through the whole document collection several times. In an incremental setting, there is never a state of "complete" document collection. A possible approach would be to treat the stabilization process not as "conceptual shifts" (i.e. documents moving between categories) but rather as conceptual copying. In the latter case, all associations between a document and categories are remembered by the network, even those invalidated by stabilization.

4. Level of quality achieved

Figure 4 shows that ART does better than k-means at both minimal vigilance and at the vigilance at which the best clustering was found (0.0435). However, the difference in quality is quite small at minimal vigilance. Furthermore, using k-means to generate the same number of clusters (43) as obtained at minimal vigilance with ART, we get $F_1 = 0.23$, which is even closer to ART's 0.27. We also compare the best F_1 quality obtained with ART1 to the best published supervised text classification results on Reuter's ModApté test set [22] (the best results were obtained with Support Vector Machines (SVM) and k-Nearest Neighbors (kNN)). ART1 at vigilance 0.0435 achieves 63% of the text classification quality. In previous work [12] we have conducted with the static batch case we demonstrated that text clusters formed by ART1 achieve between 51% and 65% of text classification quality. In the incremental case, we obtain 31% to 63%. Therefore, on-line incremental processing negatively affects the quality of clusters at minimal vigilance compared to static batch clustering. However, the quality is little affected if the best vigilance value can be determined.

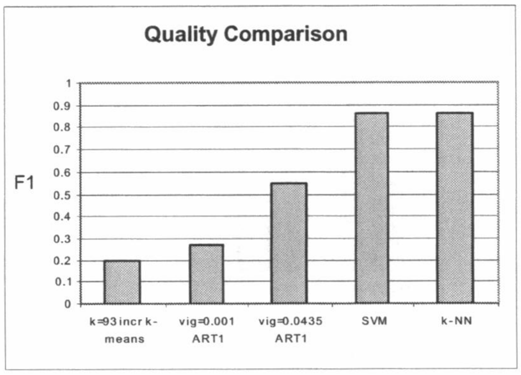

Figure 4 - ART1 clustering quality compared with k-means and text classification (SVM and k-NN).

V. CONCLUSIONS

We have investigated the applicability of ART neural networks to text clustering in a realistic, dynamic environment. This environment is characterized by frequent addition of documents and creation of new topics. Such environment makes periodic retraining with supervised methods impractical due to cost and time constraints of labeling and training. Learning that is both plastic and stable as well as autonomous is deemed essential to meet the requirements of real-life organization of large dynamic text collections.

Based on the experimental results presented in this paper, our main conclusions and contributions are:

1) Large existing legacy text data can be used as a surrogate for yet unseen documents when performing feature selection;
2) New terms should be added to the vocabulary only if the initial vocabulary was based on a large, representative selection of text such that the initial vocabulary is comprehensive enough. Unnecessarily adding new terms only increases the dimensionality of the document vectors and slows down processing with no gain in quality;
3) The choice of a vigilance value is a difficult problem. We propose the use of minimum vigilance. This will result in average quality with highly general clusters. However, much better quality is possible with other vigilance levels. It is important to set the increments to < 1/N when searching vigilance due to a high variability in quality between vigilance values;
4) Stabilization improves quality but how and when to perform stabilization needs further investigation. It seems rather uncertain how the stability advantage of ART networks can be harnessed in a real-world, dynamic environment. We have proposed but not tested a modification to the ART framework in which stabilization remembers previous category assignments; and
5) We have shown that on-line incremental clustering with ART negatively affects the quality of clusters at minimal vigilance compared to static batch clustering. K-means, a simple but unstable incremental clustering algorithm, achieved slightly lower quality than ART1. However, ART quality in the incremental case is similar to the static case if the best vigilance value can be determined. In this case, over 60% of the F_1 quality obtained with the best supervised text classification approaches is reached by using ART1 with no labeled training data.

Our hypothesis was that the rapid convergence, plasticity and stability properties of ART networks make this type of neural network perfectly suited for clustering documents in a high-demand, on-line dynamic environment. Stabilization raised important issues that need to be fully investigated. On other aspects, ART seems to work well in a dynamic environment.

Overall, although the clustering quality of ART1 is still far from what can be achieved with supervised methods (in particular at minimal vigilance), we must reiterate that ART-based text clustering is totally autonomous and that it requires neither a training set nor prior knowledge of topics. Therefore, text clustering can be seen as a low cost approach to organizing vast amount of dynamic textual information. More sophisticated ART architectures with non-binary representation and more advanced feature selection may result in improved clustering. We are currently investigating these possibilities.

REFERENCES

[1] F. Sebastiani, "Machine Learning in Automated Text Categorization", *ACM Computing Surveys*, Vol. 34, No. 1, March 2002, pp. 1–47, 2002.

[2] M. Steinbach., G. Karypis, and V. Kumar, "A Comparison of Document Clustering Techniques", In: *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 20-23, 2000, Boston, MA, USA.

[3] D. Merkl, "Text data mining". In: *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*. Marcel Dekker, New York, 1998.

[4] A.K. Jain., M.N. Murty and P.J. Flynn, "Data Clustering: A Review", *ACM Computing Surveys*, Vol. 31, No. 3, Sept 1999.

[5] J. Aslam, K. Pelekhov and D. Rus, "A practical clustering algorithm for static and dynamic information organization", In: *Proc. of the 1999 Symposium on Discrete Algorithms*, 1999.

[6] W-C Wong and A.W-C Fu, "Incremental document clustering for web page classification", In: *Proceedings of IEEE 2000 International Conference on Information Society in the 21st Century: Emerging Technologies and New Challenges*, Japan, 2000.

[7] B. Moore, "ART and Pattern Clustering", In: *Proceedings of the 1988 Connectionist Models Summer School*, p. 174-183, 1988.

[8] G.A. Carpenter and S. Grossberg, "Invariant pattern recognition and recall by an attentive self-organizing art architecture in a nonstationary world". In: *Proceedings of the IEEE First International Conference on Neural Networks*, pages II-737-II-745, June 1987.

[9] R. Kondadadi and R. Kozma, "A Modified Fuzzy ART for Soft Document Clustering", In: *Proc. International Joint Conference on Neural Networks*, Honolulu, HA, May 2002.

[10] K. J. MacLeod and W. Robertson, "A neural algorithm for document clustering". *Information Processing & Management*, 27(4):337-346, 1991.

[11] N. Vlajic and H.-C. Card, "Categorizing Web Pages using modified ART". In: *Proceedings of IEEE 1998 Canadian Conference on Electrical and Computer Engineering*, 1998.

[12] L. Massey, "On the quality of ART1 text clustering". *Neural Networks* (16)5-6 pp. 771-778, 2003.

[13] K. Rajaraman and A-H. Tan, "Topic detection, Tracking and Trend Analysis using Self-organizing Neural Networks". In: *Proc. of the Fifth Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'01)*, Hong Kong, pp. 102-107, Apr. 16-18, 2001.

[14] C. J. Van Rijsbergen, *Information Retrieval*. London: Butterworths, 1979.

[15] G.W. Milligan, "A Monte Carlo Study of Thirty Internal Criterion Measures for Cluster Analysis", *Psychometrika* 46, pp. 187-199, 1981.

[16] J. MacQueen, "Some methods for classification and analysis of multivariate observations", In: *Proceedings of the 5th Berkeley symposium on Mathematical Statistics and Probability*. Vol 1, Statistics. Ed. by L.M. Le Cam and J. Neyman. Univ of California Press, 1967.

[17] C. Apte, F. Damerau and S.M. Weiss, "Automated learning of decision rules for text categorization", *ACM Transactions on Information Systems*, 12(2):233-251, 1994.

[18] G. Salton and M.E. Lesk, "Computer evaluation of indexing and text processing". *Journal of the ACM*, Vol 15, no.1, pp 8-36, January 1968.

[19] Y. Yang and J.O. Pedersen, "A comparative study on feature selection in text categorization". In: *Proc. Of ICML-97, 14th International Conf. on Machine Learning* (Nashville, USA), pp. 412-420, 1997.

[20] L. Massey, "Determination of Clustering Tendency With ART Neural Networks", In : *Proc of 4th Intl. Conf. on Recent Advances in Soft Computing*, Nottingham, U.K., Dec. 2002.

[21] M. Georgiopoulos, G.L. Heileman and J. Huang, "Convergence properties of learning in ART1". *Neural Computation*, 2(4):502--509, 1990.

[22] Y. Yang and X. Liu, "A re-examination of text categorization methods". In: *Proceedings of Int'l ACM Conference on Research and Development in Information Retrieval (SIGIR-99)*, 42-49, 1999.