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
massey-l@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-learned 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. 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. 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. 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. However, this would seem to be its major strength.

In this paper, we propose and test the use of binary ART (ART1) neural networks to address this 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 ART1 algorithm we used to simulate an ART1 network follows:

1. Initialize network weights and provide parameter values:

   \[0 < \rho \leq 1\] (the vigilance parameter)
   \[1 > \lambda\]
   \[W_j = 1/(1+\rho)\] for all connections
   \[T_j = 1\] for all connections

2. \(u_j = 0\) for \(j = 1..M\) and present a document \(d_i\) to the network.

3. Compute output activations, for \(j = 1..M\):

   \[u_j = d_i \cdot W_j\] ("\cdot" is the logical AND)

4. Competition: select output neuron \(j^* = \max\)

   \[a_j = d_i \cdot T_j\]

5. Vigilance test: determine if \(j^*\) is close enough to \(d_i\):

   \[||d_i - T_j|| / ||d_i|| \geq \rho\]

   If true, step 6 (resonance mode); otherwise, step 8 (search mode).

6. Update weights:

   \[T_j = T_j + d_i\]
   \[W_j = 1/(1+d_i \cdot T_j)(1+1/||d_i||)\]

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 \(F_1\) formulæ:

\[F_1 = pr / (p + r)\]

where:

\[p = a / (a + b)\] is the precision
\[r = a / (a + c)\] 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. IFN 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 \(i\).

ART, contrary to most clustering algorithms, 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

**Vocabulary selection and 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 means that no word removal based on collection term statistics is possible, i.e. 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.

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 in
that collection. 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. 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, when 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 data sets described below. We simulate a real-life operational use by processing the 3299 test documents, transforming them into a vector space representation in the following different manners:

a) Data set NONE was built by using only the terms present in the initial vocabulary (obtained from the discarded and training documents); that is, none of the new terms present in the test set were used as features.

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

c) Data set TST consists of terms extracted from the Reuter test set only. It is the control data set, representing the static case.

Figure 1 shows that there is no clear winner between both vocabulary growth strategies with data set ALL or NONE each taking turns in giving the best F1 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 F1=0.31). These results indicate that using legacy data for initial vocabulary creation and feature selection is a viable solution and that new words introduced during operational use (here simulated by the test set documents) are not necessary to improve clustering quality. In fact, keeping new words has the undesirable effect of increasing dimensionality by 3000, which penalizes computation speed with no significant gain in quality. However, one must not hastily draw general conclusions from this result. Given that the test set introduces nearly 3000 new terms, 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.

**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: preliminary experiments on legacy data could be conducted if labeled data is available to evaluate quality, or otherwise one could rely on costly, time consuming and possibly subjective users evaluation. A possibly more realistic option, if customers need quite general clusters, is to use minimal vigilance [20], that is \( \rho_{\text{min}} \leq 1/N \) where \( N \) is the number of feature-words. However, the question is: are there other, better values for vigilance? 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. Another important question is by what value should vigilance be incremented? Up to now we have interpolated quality between rather large vigilance intervals, but are there other, possibly better clusters resulting from intermediary vigilance values? Based on the vigilance test of ART1 algorithm (step 5):

\[
\|d_k^T_j\| / \|d_k\| \geq \rho
\]

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, \( \|d_k^T_j\| \) 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 \( \|d_k\| = b \)
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.

**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 documents, which can include duplicates or revised versions, can exist in important numbers and that hence the problem must not be merely 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 $\rho = 0.001$ and $\rho = 0.0435$ were required to attain a stable representation, which is much less than the theoretical upper bound of $N-1$. This still translates into 3-4 more times to process the set of 3299 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. The issue of how frequent stabilization operations should be must be 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. It rather seems uncertain how the stability advantage of ART networks can be harnessed in a real-world, dynamic environment.

**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-
We have also 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 F1 quality obtained with the best supervised text classification approaches is reached by using ART1 with no labeled training data at all.

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. Furthermore, the preliminary experiments presented here have clearly shown the potential problems with using ART in a real-life, dynamic environment: vocabulary selection, vigilance value determination and stabilization are issues that we are currently studying in greater detail.

V. CONCLUSIONS

We have investigated the applicability of ART neural networks to text clustering in a realistic 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.

Our hypothesis was that ART fits well with these requirements. Based on the experiments presented in this paper, our main conclusions are: 1) Term selection is an area that needs further investigation. 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 words set is nearly complete. Otherwise new terms only add to the dimensionality of the vector space model and slow down processing with no gain in quality; 2) The choice of a vigilance value is difficult in a real-life environment. A possible solution is to use 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; 3) Stabilization improves quality but how and when to perform stabilization needs further investigation. Although stabilization, in theory, allows avoiding the problems of documents being re-assigned between topics, in a real-world environment it is not clear if this can be achieved.

REFERENCES


