Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 673–677

ISBN 978-83-60810-51-4

Folksonomy implementation based on the ART-1 neural network

Adam Sobaniec
Warsaw University of Technology,
Faculty of Mathematics and
Information Science,
pl. Politechniki 1, 00-661 Warsaw,
Poland
Email: sobanieca@gmail.com

Bohdan Macukow
Warsaw University of Technology,
Faculty of Mathematics and
Information Science,
pl. Politechniki 1, 00-661 Warsaw,
Poland
Email:
B.Macukow@mini.pw.edu.pl

***Abstract*—This document describes the sample implementation of a very popular classification method in the modern internet web applications – folksonomy. This method forces users to assign a particular keywords to the content that they are uploading. Basing on the assigned keywords it is possible to find a similar content. In this paper, there is described the method of finding similar content basing on the ART-1 neural network. Such solution allows to perform a background clustering of a content and speed up the process of retrieving the related data. In case of a heavy exploitation of an internet application, this might be a big advantage**

I. Introduction

INTERNET has significantly changed form at the beginning of the 21st century. Instead of presenting only the static content that has been updated only occasionally, it has become a completely dynamic medium where content has been created not only by a website administrators, but also by users. It was possible because of the, so called, Web 2.0. New technologies that allowed to create a dynamic pages have been introduced and have been used widely over all modern websites. Users were able to change the shape of a particular web application. There were created a new types of applications, like web forums or social networks. The Web 2.0 revolution has begun the significant increase of the amount of information stored over the internet. For this reason there has appeared a need for an arrangement of all information. It has been achieved in two ways – by introduction of search algorithms and by categorization of a content. Categorization of a content allowed to provide a new features for all modern websites. One of them is the display of the similar content to the user. Such feature allows to keep user busy for a longer period of time. This leads to the greater popularity of a website and increases it's value. One of the best examples of this feature can be found on the most popular video sharing web application – Youtube. When user begins to watch a particular video clip, there are being displayed the similar videos that might interest him.

Representation of a similar content can be achieved by assigning specific features to the given record (photo, video or other), depending on it's type. For example photos or videos might be classified basing on the key points assigned to the particular frame. Music files can be classified basing on so called signatures. All these solutions however, require complex algorithms to be applied and require many computations. The simplest and most popular method of classification of any content is the key words (tags) assignment system.

Fig 1. Sample Youtube video page with related content – in this case Honda Civic commercials

A. Tagging in practice

Assigning tags can be applied to any content available in the web. It is the most universal method of classification that quickly became widely used over the whole internet. It can be done in two ways:

- Administrators or moderators of an internet application are assigning key words to the content
- Users are assigning key words to the content that they own and upload

Fig 2. Keywords: "plant", "rose", "flower"

Fig 3. Keywords: "flower", "tulip", "plant"

First method is used in the smaller and highly specified web sites, where the key words precision is very important (for example scientific articles, news). In this case, qualified person is assigning the most appropriate tags to the particular content. This method allows more precise mapping of the content features but it requires high amount of work done by the persons responsible for tagging. It is almost impossible to use this method in the biggest web applications that contain tons of user specific data.

Second method is so called folksonomy [6], in other words – social tagging. Word folksonomy has been obtained by concatenation of words: folks (people) and taxonomy (practice of content categorization). One of the first applications that utilized folksonomy is Delicious.com. This web site allows users to upload their own bookmarks so they can access them from any place in the world and share them among each other.

In the short amount of time folksonomy has become the most popular method that has been used in the websites for user content categorization. It is widely used in the applications like Youtube.com (video sharing), Flickr.com (photos sharing) or 43things.com (social network).

This paper describes a method of folksonomy implementation that bases on the ART-1 [1] neural network. There are many methods of finding a similar content in the modern web applications. However, most of them, use complex SQL queries in order to retrieve the similar data. In case of heavily exploited website, this may become a potential bottleneck. ART-1 based solution provides the same functionality but all computations are being done by the neural network, hence reducing the database load.

B. General idea

General idea of an ART-1 based classification is to create a binary vectors, basing on the keywords assigned to the particular content. Such binary vectors may be clusterized and therefore, when user views specific content it is possible to present him similar content – the one that has been assigned to the same cluster.

Let's consider an example of a web application that allows to share photos between users. Let's assume that there are only the following keywords present in the database:

```
'Car', 'Cat', 'Dog', 'Ferrari', 'Red',
'Blue', '2012'
```

When user uploads a following photo:

Fig 4. Sample photo uploaded by a user

With a keywords `'Ferrari', 'Car', '2012'`, then the following photo may be represented by the following binary vector:

```
[1, 0, 0, 1, 0, 0, 1]
```

The following algorithm is being used for a vector generation:

Given:

$D=\{d_1, d_2, \ldots, d_N\}$ - Dictionary of all tags in the database

$T=\{t_1, t_2, \ldots, t_M\}$ - Tags assigned to the photo

$v=\{v_1, v_2, \ldots, v_N\}$ - resulting vector

Foreach d_i in D

  if $d_i \in T$

    set $v_i = 1$

  else

    set $v_i = 0$

Having such set of vectors, one may provide each of them as an input to the ART-1 neural network and obtain the cluster to which each of them has been assigned. In this case each cluster will contain a set of similar vectors i.e. with at least one "1" in common. Once the ART-1 algorithm has been finished, it is possible to display a similar photos to the user basing only on the information to which cluster it has been assigned.

Such solution doesn't require complex SQL query and allows for a quick search of the database content. In the approach presented in this paper, ART-1 neural network is supposed to be loaded into the RAM memory so all compu-

tations can be performed much faster than in the case when data is being stored in the database or directly on the disk.

II. ART-1 Neural Network

At the beginning of the research, there has been implemented a standard ART-1 neural network with attentional and orienting subsystems. There has been performed tests that were designed to show the result of the classification of 15 binary vectors of length 10.

Obtained results weren't satisfactory. First of all there appeared clusters where vectors had only one "1" in common. In case of folksonomy classification, such situation is unacceptable, because it would mean that two elements are similar if they have only single keyword in common. This would be a contradiction to the assumptions of the presented method, because such functionality might be achieved with a simple SQL query that will be faster than the neural network. The main aim of ART-1 based folksonomy was to mark two objects as similar only when they have at least two keywords in common.

To achieve mentioned above functionality there has been changed the sequence of neural network training. Input vectors have been sorted beginning from the vectors with the most number of "1"s. With this solution initial input was supposed to form base clusters that will be able to serve sparse vectors.

Unfortunately the results were still unacceptable. For this reason the orienting subsystem has been redesigned to meet the folksonomy classification requirements. Since vectors that are supposed to be provided as an input are very sparse, the standard vigilance threshold parameter has been rejected. Let's consider, a medium sized web application, that contains 5000 tags distributed among it's content. Since users usually assign between 2 and 6 keywords, created vectors will be very long and filled with a small amount of "1"s. To perform a classification of such vectors the standard vigilance threshold would need to have to be set to a very small values in order to allow proper clusterization. For this reason it has been redesigned and instead it took into account the number of common ones between two vectors.

Such simple modification allowed to specify directly how many "1"s vectors should have in common to become classified as similar, thus, it is possible to specify how many keywords should match to assign content to the same cluster.

III. Sample art-1 based content classification

To present the results of a classification there has been implemented a sample application that allows users to share the photos that they own. In order to fill the database, there has been written a crawler that downloaded samples from the most popular site of that kind – Flickr.com.

There has been downloaded 10000 photos and 9968 keywords. Unfortunately, one of the biggest drawbacks of the folksonomy implemented by Flickr is no limitation of the number of tags assigned to the single content. There is also no limit on the number of chars within the single keyword and no filter on the non-alphanumeric characters (users can provide keywords containing "&", "?" and other).

In order to construct the binary vectors that were provided as an input there has been created a dictionary of 3976 tags. Other tags out of the 9968 keywords were assigned only to a single photo, which made them useless – they stored no information and would only delay the training algorithm.

Only the photos with two or more tags assigned has been taken into account. As a result there has been created 9348 vectors that have been provided as an input. In result there has been created 3246 clusters, which gives on average about 3 photos per cluster.

IV. Folksonomy Implementation Methods Comparison

The main purpose of the ART-1 based implementation was to achieve a much higher speed of similar content finding than in the standard SQL query approach. It was supposed to be especially visible for the bigger amount of content available in the web application. Instead of performing one complex SQL query, there is being executed only a single simple query to retrieve the content assigned to the same cluster. There has been prepared a benchmark that simulated the web application that facilitates news articles. Created application allowed to perform the following activities:

- Generating sample database
- Browsing data
- Train ART-1 neural network
- Compare the execution time of ART-1 based approach and standard SQL approach

Generation of the sample database was achieved by providing the number of articles that were supposed to be created, number of clusters that are supposed to be generated and number of keywords specific for each cluster. Algorithm responsible for generation of a single article, was designed in such way, that it picked up a random cluster and basing on this information it picked up a random keywords related with a given cluster. Within this approach, there has been created many elements that contain common features and might be successfully classified.

Below are presented results of similar contents finding comparison. For each testing sample there has been performed two tests. The results show that the difference grows proportionally to the data set size.

Tests have been performed basing on the following data sets (generated via the application interface):

- Set 1:
  - Articles count: 10 000
  - Clusters count: 100
  - Tags per cluster: 10
- Set 2:
  - Articles count: 100 000
  - Clusters count: 10
  - Tags per cluster: 10
- Set 3:
  - Articles count: 1 000 000
  - Clusters count: 10
  - Tags per cluster: 10

The following charts present results for each set:

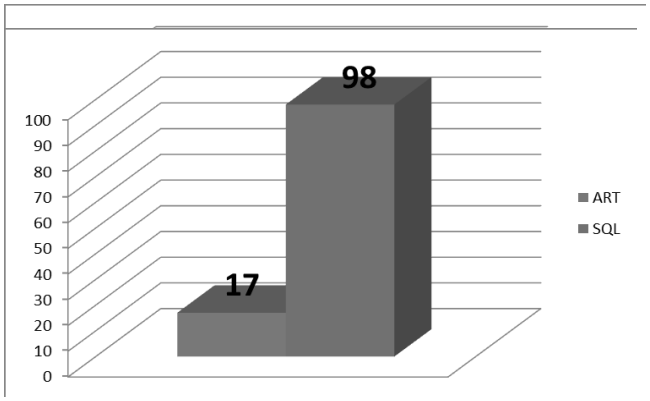

Fig 6. Set 2 classification speed

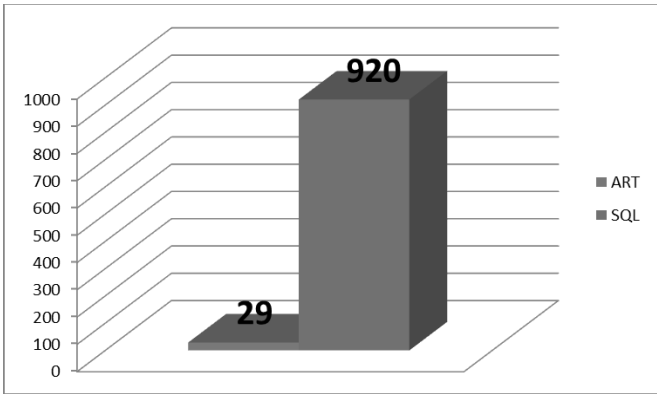

Fig 7. Set 3 classification speed

- Set 3:

When using the ART-1 based classification the number of rows in the table doesn't affect the speed of the classification in the same way as within the SQL approach. For 1 million rows, the query takes almost 1000 ms. For most modern, big web applications such amount of computations is unacceptable. However, many modern websites have implemented the mechanisms that are able to serve the similar content even under the heavy exploitation from the users. One these mechanisms is queries caching. Namely, only the first run of a query is being performed directly on the database. During the next call of a similar content finding, instead of querying the database, there are being returned the results from the cache. Such solution is very fast and even for a large data sets, the query execution takes only a couple of milliseconds.

Implemented tests were supposed to simulate the situation when user uploads his own article, so there can be displayed similar content, immediately. The ART-1 neural network solution has proved to be faster at each such demonstration. However, the tests have shown the biggest drawback of this solution – huge memory requirements.

V. Possibilities of Implementation in a Real System

There are a couple of problems that one has to solve when implementing ART-1 based folksonomy in the real system. First of all, each time when the user uploads a new content and assigns a new keywords that hasn't been earlier used to describe content, there is need to retrain the neural network. It is required due to the fact that dictionary of tags, basing on which, binary vectors are generated, is being modified. Since this process is computationally expensive and requires big amount of time, it should be implemented as a background process that reads the data from the database and performs the ART-1 training algorithm. At some point, the frequency of dictionary size change will be systematically decreasing and there will be no need to repeat the process very often.

Another important point that requires special solution is a process of saving into the servers memory ART-1 neural network. In most cases there is a significant memory limit for most web applications. Especially when using virtual hosting. In the sample photo sharing application where has been classified 10000 photos, neural network occupied only 110 MB. It is relatively small amount of memory so the process of reading/writing network to the server memory has been solved by simple operations on a binary file that contained serialized ART-1 network. Due to the small size, the whole operation was performed relatively fast. However there has to be handled many cases when the application is being restarted due to the inactivity or maintenance. In such case, when the ART-1 network grows in size, this might become a potential bottleneck.

The solution might be to use the most recent technologies available in the cloud hosting solutions. For instance, Microsoft Windows Azure cloud offers, so called, AppFabric Cache service, that keeps the cache among the web servers and is supposed to be quickly accessible whenever it is required. It can be used to store the ART-1 neural network. In such case the problems with reading/writing trained network on the disk would be solved.

A. The biggest disadvantage

Presented in this paper solution has one crucial disadvantage – memory occupancy. For a big websites, that contain a lot of user specific content, the size of a network might exceed the physical RAM limits of most modern web servers. For instance, when the web application contains a dictionary of 100 000 tags, and there is being set a limit of maximum 10 000 clusters. In the worst case, in the Microsoft .NET environment, the memory required to store such neural network will be equal to about 10 GB.

Such huge memory requirement is related to the fact that to each neuron in the input layer, that is being assigned a vector of floating point weights, that require significant amount of memory allocation. For this reason the size of an object that represents single neuron can be especially visible when dealing with bigger number of clusters and vectors.

VI. Summary

Presented in this paper method of folksonomy implementation allows to automate most of the functionalities available to obtain by having keywords assigned to the particular content. There is no need to hire additional moderators that

will filter records in the database in such way that there will be a possibility of their classification. There might be introduced a feature that will automatically perform an ART-1 training at the time when website is not being heavily exploited.

During the work on this paper there has been revealed that a huge memory requirements eliminate this solution from the production applications. Especially when they contain large amount of user specific data that is heavily exploited. However, medium sized web services, that contain only thousands of records might successfully integrate this solution and enhance the process of content classification.

The DLL library that has been implemented during research, may be used not only in the web applications, but also in other types of applications. It is ready to perform a classification of any binary vectors. Because the whole application has been created in the modern Microsoft .NET environment, there could be introduced an enhancements to the ART-1 neural network algorithm, like dynamic allocation of output neurons.

Presented in this paper implementation of folksonomy hasn't been published and serves as a first step toward the improvement of the folksonomy based classification process. The main aim is to reduce the amount of work performed by moderators and amount of computation performed by database. Once it is done, all computations can be focused on the other parts of an application, where they can be performed faster.

References

[1] G. A. Carpenter, S. Grossberg "Adaptive Resonance Theory", Cambridge, MIT Press, 2003.
[2] F. Hao, S. Zhong "ECKDF: Extended Conceptual Knowledge Discovery in Folksonomy", ICCP Proceedings, 2010.
[3] H. Kawakubo, Y. Akima, K. Yanai "Automatic Construction of A Folksonomy-based Visual Ontology", IEEE International Symposium on Multimedia, 2010.
[4] L. Massey "On the quality of ART1 text clustering", Elsevier Science Ltd., 2003.
[5] L. Massey "Determination of Clustering Tendency With ART Neural Networks", Intl. Conf. on Recent Advances in Soft Computing, 2002.
[6] I. Peters "Folksonomies: Indexing and retrieval in Web 2.0", Saur K. G. Verlag GmbH, 2009.
[7] J. Pipes „Tagging and Folksonomy Schema Design for Scalability and Performance", MySQL, Inc., 2006.