Topic Detection, Tracking, and Trend Analysis Using Self-Organizing Neural Networks

Kanagasabi Rajaraman and Ah-Hwee Tan

Kent Ridge Digital Labs
21 Heng Mui Keng Terrace
Singapore
{kanagasa,ahhwee}@krdl.org.sg
http://textmining.krdl.org.sg

Abstract. We address the problem of Topic Detection and Tracking (TDT) and subsequently detecting trends from a stream of text documents. Formulating TDT as a clustering problem in a class of self-organizing neural networks, we propose an incremental clustering algorithm. On this setup we show how trends can be identified. Through experimental studies, we observe that our method enables discovering interesting trends that are deducible only from reading all relevant documents.

Keywords: Topic detection, topic tracking, trend analysis, text mining, document clustering

1 Introduction

In this paper, we address the problem of analyzing trends from a stream of text documents, using an approach based on the Topic Detection and Tracking initiative. Topic Detection and Tracking (TDT) [1] research is a DARPA-sponsored effort that has been pursued since 1997. TDT refers to tasks on analyzing time-ordered information sources, e.g news wires. Topic detection is the task of detecting topics that are previously unknown to the system[8]. Topic here is an abstraction of a cluster of stories that discuss the same event. Tracking refers to associating incoming stories with topics (i.e. respective clusters) known to the system[8]. The topic detection and tracking formalism together with the time ordering of the documents provides a nice setup for tracing the evloution of a topic. In this paper, we show how this setup can be exploited for analyzing trends.

Topic detection, tracking and trend analysis, the three tasks being performed on incoming stream of documents, necessitate solutions based on incremental algorithms. A class of models that enable incremental solutions are the Adaptive Resonance Theory (ART) networks[2], which we shall adopt in this paper.

D. Cheung, G.J. Williams, and Q. Li (Eds.): PAKDD 2001, LNAI 2035, pp. 102–107, 2001.

2 Document Representation

We adopt the traditional vector space model[6] for representing the documents, i.e. each document is represented by a set of keyword features. We employ a simple feature selection method whereby all words appearing in less than 5% of the collection are removed and, from each document, only the top n number of features based on $tf.idf$ ranking are picked. Let M be the number of keyword features selected through this process. With these features, each document is converted into a keyword weight vector

$$\mathbf{a} = (a_1, a_2, \ldots, a_M) \tag{1}$$

where a_j is the normalized word frequency of the keyword w_j in the keyword feature list. The normalization is done by dividing each word frequency with the maximum word frequency.

We assume that text streams are provided as document collections ordered over time. The collections must be disjoint sets but could have been collected over unequal time periods. We shall call these time-ordered collections as *segments*.

3 ART Networks

Adaptive Resonance Theory (ART) networks are a class of self-organizing neural networks. Of the several varieties of ART networks proposed in the literature, we shall adopt the fuzzy ART networks[2] .

Fuzzy ART incorporates computations from fuzzy set theory into ART networks. The crisp (nonfuzzy) intersection operator ($\cap$) that describes ART 1 dynamics is replaced by the fuzzy AND operator ($\wedge$) of fuzzy set theory in the choice, search, and learning laws of ART 1. By replacing the crisp logical operations of ART 1 with their fuzzy counterparts, fuzzy ART can learn stable categories in response to either analog or binary patterns.

Each fuzzy ART system includes a field, F_0, of nodes that represents a current input vector; a field F_1 that receives both bottom-up input from F_0 and top-down input from a field, F_2, that represents the active code or category. The F_0 activity vector is denoted $\mathbf{I}$. The F_1 activity vector is denoted $\mathbf{x}$. The F_2 activity vector is denoted $\mathbf{y}$.

Due to space constraints, we skip the description of fuzzy ART learning algorithm. The interested reader may refer to [2] for details.

4 Topic Detection, Tracking, and Trend Analysis

In this section we present our topic detection, tracking and trend analysis methods.

4.1 Topic Detection Algorithm

As described in Section 3, ART formulates recognition categories of input patterns by encoding each input pattern into a category node in an unsupervised manner. Thus each category node in F_2 field encodes a cluster of patterns. In other words, each node represents a topic. Hence, identification of new topics translates to the method of creation of new categories in the F_2 field as more patterns are presented. Using this idea, we derive the topic detection algorithm in Table I.

Table 1. Topic Detection Algorithm.

Step 1. Initialize network and parameters.
Step 2. Load previous network and cluster structure, if any.
Step 3. Repeat - present the document vectors - train the net using fuzzy ART Learning Algorithm until convergence
Step 4. Prune the network to remove low confidence category nodes
Step 5. Save the net and cluster structure.

4.2 Topic Tracking Algorithm

For tracking new documents, the latest topic structure is loaded before processing the documents. For an incoming document, the activities at the F_2 field are checked to select the winning node, i.e. the one receiving maximum input. The document is then assigned to the corresponding topic. This is the idea behind the tracking algorithm presented in Table II.

Table 2. Topic Tracking Algorithm.

Step 1. Initialize network and parameters.
Step 2. Load previous network and cluster structure, if any.
Step 3. Present the document to be tracked, to the net
Step 4. Assign the document to the topic corresponding to the winning category node, i.e. category node that receives maximum input.

4.3 Trend Analysis

The topic detection and tracking setup together with the time ordering of the documents provides a natural way for topic-wise focussed trend analysis. In particular, for every topic, suppose we plot the number of documents per segment versus time. This plot can be thought of as a trace of the evolution of a topic. The 'ups' and 'downs' in the graph can be used to deduce the trends for this topic. For more specific details on the trends, one can zoom in and view documents on this topic segment-wise. This process is illustrated in the following section.

5 Experiments

For our experiments, we have grabbed daily news articles from CNET and ZD-Net and grouped the articles into weekly segments. Starting from 1st week of September 2000 up till 4th week of October 2000, we collected 8 segments in all. Totally there were 1468 documents at an average of about 180 documents in each segment. Documents in each segment are converted into weight vectors as described in Section 2. We then applied our topic detection and tracking and performed trend analysis. Some qualitative results are presented below:

5.1 Topic Detection and Tracking

Typically we observed 10 to 15 new topics being identified per segment when choice parameter $\alpha = 0.1$ and vigilance parameter $\rho = 0.01$ (ignoring small clusters with 1 or 2 documents only).

A list of some of the hot topics that have been identified by the topic detection algorithm can be viewed at http://textmining.krdl.org.sg/people/kanagasa/tdt. The tracking results can also be viewed at the same URL. We skip the details due to space constraints.

5.2 Trend Analysis

The evolution graphs for some selected topics are shown below. Time is represented through the segment ID which takes values $1,\cdots,8$. ID=1 corresponds to Sep 1st week, ID=2 corresponds to Sep 2nd week and so on.

The topics 'MS Case' (i.e. Microsoft Case), 'Linux' and 'Windows ME' have been plotted in Fig 1. The 'MS Case' topic shows an initial up trend early September. An examination of the documents under this topic reveals the reason to be Bristol Technology ruling against Microsoft. Similarly the topic on 'Linux' shows a peak for early October when the Open source conference was held. 'Windows ME' graph peaks during September 2nd week coinciding with Win ME release.

The topics 'Apple' and 'Hackings' have been plotted in Fig 2. The 'Apple' topic shows an up trend during mid September when Apple Expo was on. The Microsoft hack-in can be seen to have lead to the sudden peak in 'Hackings' topic around late October.

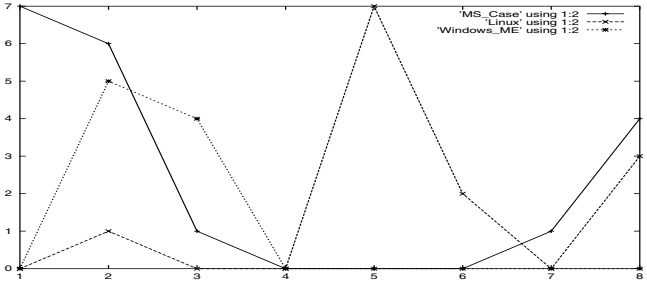

Fig. 1. Trends for 'MS Case', 'Linux' and 'Windows ME'.

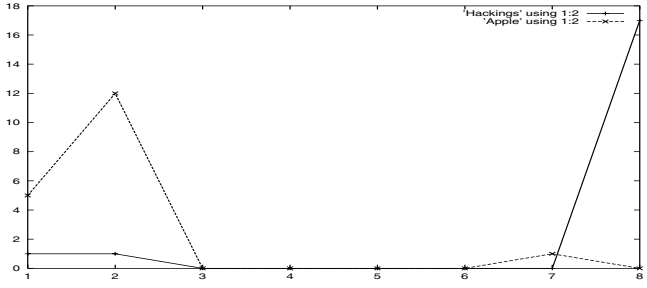

Fig. 2. Trends for 'Apple' and 'Hackings'.

The above study thus shows that our method can be used to detect hot topics automatically and track the evolution of detected topics. The method also serves to spot emerging trends (with respect to the timescale defined) on topics of interest.

6 Related Work

TDT research has been predominantly 'pure IR' based and can be categorized as based on either incremental clustering (e.g. [7]) or routing-queries (e.g. [5]). (One notable exception is the tracking method by Dragon Systems which is based on language-modelling techniques.) Incremental clustering based methods come the closest to our work, but we use ART networks for document processing in contrast to the traditional document similarity measures. Our main motivation is that ART networks enable truly incremental algorithms.

Trend analysis for numerical data has been well investigated. For free-text, where the challenge is tough, we are aware of only very few papers. [3] defines concept distributions and propose a trend analysis method by comparing distributions of old and new data. Typically, the trends discovered are of the type "keyword 'napster' appeared x% more now than in old data", "keyword 'divx' appeared y% less now than in old data", etc. [4] uses the popular a-priori

algorithm employed in association-rule learning, for finding interesting phrases. Trend analysis is done by applying a shape based query language on the identified phrases. Queries like 'Up' or 'BigDown' could be used to identify upward and strong downward trends respectively, in terms of phrases. However, there could be potentially large number of candidate phrases that could make this method inefficient.

In contrast, our trend analysis method being based on topic detection and tracking enables finding specific, topic-wise trends. The TDT formulation offers several advantages. The topic detection and tracking step enables the trend analysis be focussed and more meaningful. Since the documents under each topic are relatively small, the analysis can be done efficiently. (On a related note, the ART learning algorithm can be implemented parallelly and this implies potential further speedup.)

7 Conclusion

We have addressed the problem of analyzing trends from a stream of text using the TDT approach. First we have formulated TDT as a clustering problem in ART networks and proposed an incremental clustering algorithm. On this setup we have shown how trends can be identified. Through experimental studies, we have found our method enables discovering interesting trends that are not directly mentioned in the documents but deducible only from reading all relevant documents.

References

1. TDT homepage. `http://www.itl.nist.gov/iad/894.01/tests/tdt/index.htm`, 2000.
2. G. A. Carpenter, S. Grossberg, and D. B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.
3. R. Feldman and I. Dagan. Knowledge discovery in textual databases (KDT). In *Proceedings of KDD-95*, 1995.
4. Brian Lent, Rakesh Agrawal, and R. Srikant. Discovering trends in text databases. In *Proceedings of KDD-97*, 1997.
5. Ron Papka, James Allan, and Victor Lavrenko. UMASS approaches to detection and tracking at TDT2. In *Proceedings of the TDT-99 workshop*. NIST, 1999.
6. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.
7. Fredrick Walls, Hubert Jin, Sreenivasa Sista, and Richard Schwartz. Topic detection in broadcast news. In *Proceedings of the TDT-99 workshop*. NIST, 1999.
8. Charles Wayne. Overview of TDT. `http://www.itl.nist.gov/iaui/894.01/tdt98/doc/tdtslides/sld001.htm`, 1998.