The need for comparative evaluation of online clustering approaches for topic and event detection on Twitter

drd. Robert Popovici
Universitatea Stefan cel Mare Suceava

1 Introduction

With the advent of Twitter as the undisputed market leader in social micro-blogging and the steady growth of user-generated content (over 500 million tweets daily), the use of Twitter as news reporting mechanism is constantly increasing. As a response to this developing trend a number of topic and event detection techniques have been proposed in order to meet the challenging task of detecting valid interesting information in a continuous stream of short natural language text.

According to a broad classification based on their underlying methods, existing topic and event detection approaches can be grouped in two main categories. The former discovers events by clustering documents if the semantic distance between documents and clusters of documents is within a similarity threshold [Yiming98], while the latter analyzes word distributions and detects events by clustering words together [Kleinberg02].

Although some form of evaluation as to the quality of the underlying techniques has been conducted for event detection approaches in general, no sufficient comparative evaluation effort can be documented for online clustering approaches falling in the first category.

2 Online density-based clustering methods

Online density-based clustering approaches, though suitable candidates for the data stream processing task primarily because of their ability to keep track of an evolving stream without requiring prior knowledge about the number of clusters or performing costly reorganizations of the clustering structure, cannot, however, be directly applied to the clustering of text data streams.

This is due to a number of reasons. First, in a text stream setting the dimensionality of the feature space may change, with the number of vector components in a given cluster growing exponentially with the number of unique features. A direct consequence of the increase in the number of vector components as well as of the changing dimensionality of the vector representations of the cluster summaries is the decrease in run-time performance. In this context, a dimensionality reduction of the text feature space to a *k*-dimensional semantic space needs to be performed by way of removal of infrequent features at pre-set intervals.

The second reason refers to a trait pertaining to incremental clustering approaches in general: the assignment of incoming stream objects to clusters based on a similarity threshold. Since the latter cannot be determined experimentally due to the unpredictable diversity of the stream data, a similarity threshold adjusting itself dynamically to the characteristics of the stream yields more promising results. To obtain an accurate estimate of the cluster centroids for text data, a synopsis data structure is maintained that contains sufficient cluster and summary information about the set of processed text documents in the text data stream.

An example of such an adjustment of an online density-based clustering algorithm for Twitter has already been described in [Popovici14]. In this particular case, the proposed algorithm extends the concept of density-based clustering over an evolving data stream with

noise [(DenStream [Cao06]) with enhanced applicability for Twitter. The similarity threshold parameter used for cluster assignments of new data objects in the classical version of the DenStream algorithm is defined as a minimum density requirement for a given cluster and is calculated as a factor of the standard deviation of the vector components. The standard deviation measures the internal variance of the cluster (the radius of the clustering) and is calculated as a function of the weighted linear (WLS) and squared sums (WSS) of the points. The problem with the original radius equation is that the Euclidean norm of the *WSS* and *WLS* vectors are used. Using the absolute value inside the square root the radius can produce negative arguments to the square root, leading to inaccurate results. The alternative equation suggested by the MOA-Framework, implemented as the maximum of the component-wise standard deviation, however, does not work as expected in a scenario where feature transformations of the text data are stored in cluster summary vectors. This problem is twofold. First, during the assignment phase the original implementation of the DenStream algorithm produces a temporary copy of the cluster that contains the newly inserted stream object and then calculates the radius of the temporary copy as a function of the linear and squared sums of the vector components. Since vector representations of incoming tweets always contain 1-entries (and therefore the squared sums always equal the linear sums of the points) they cannot be used to compute the radius parameter in a text stream setting. Secondly, the computation of the standard deviation involves several iterations over the complete set of vector components of the cluster summaries (two separate lists containing the linear and squared sums of the vector components are maintained). Due to the varying dimensionality of the vectors in the text stream setting, the costs associated with this extra computation combined with the costs of creating and deleting the temporary copy of the cluster as well as updating the lists containing the linear and squared sums of the points cause additional delays in execution time and unnecessary filtering of possibly relevant tweets.

Additionally, the epsilon parameter is hard to determine experimentally and cannot be adjusted as a function of the evolving characteristics of the stream. Yet another drawback of the classical DenStream algorithm is directly related to its inability to distinguish between clusters with variable levels of density (a problem also encountered with density-based DBSCAN).

To overcome these problems, the solution proposed in [Popovici14] uses the dynamic estimation of the similarity threshold parameter based on the average of the last *n* closest similarity values to the cluster summaries. A tweet object is assigned to the nearest cluster based on the average of the closest similarity values to the cluster summaries attained by previous objects encountered in the stream. The assignment is successful unless the closest similarity value is considerably lower than the values attained by previous stream objects. In order to be able to determine whether the closest similarity value is considerably below the ones previously attained, the similarity threshold can be dynamically estimated as the mean of the last three closest similarity values to the cluster summaries added to a tolerance threshold which can be determined experimentally.

To compute similarity between the cluster vectors and the vector representations of tweets that differ in the number of their respective components, the length of the vector representation of the tweets needs to be adjusted to the length of a subset of the vector components of the cluster summaries. For efficiency reasons, this subset consists of those vector components for which there exists a matching component in the document vector, plus a set of components of the cluster summaries (defined by the top *n* most frequent non-matching entries of the cluster vector, with *n* typically set to 10) for which there is no equivalent component in the document vector. This ensures that a sufficiently accurate approximation of the characteristics distinguishing the cluster summaries from the tweet vector can be obtained without having to use the complete set of components of the

respective cluster representation. The vector representations to be compared are equally required to have the same ordering of their vector components.

With regard to the run-time aspect of the evaluation, there is a perceived need to reduce the overhead associated with the cluster assignments. Adjusting the pruning parameter used by incremental clustering approaches to react to changes in the stream (for example, when the number of clusters has grown above a certain threshold) can be alternatively used, yet a trade-off between performance and quality must be taken into account. On the other hand, the extensive use of pruning is likely to cause topic fragmentation, thus causing a decrease in precision and in the quality of the output topic distribution. Although a duplicate merging technique can solve the problem of duplicate topics in a scenario in which extensive pruning is used, more significant optimization potential method lies in clustering streaming data to the same set of data nodes using the technique of Locality Sensitive Hashing. This method utilizes the compact bitwise representation of document vectors called fingerprints to increase the data processing speed and performance. We consider investigating the effects of this improvement to our current clustering technique as baseline approach in our future work and compare its task and run-time performance to a series of other state-of-the-art event and topic detection techniques.

3 Challenges of comparative evaluation

In this section of the paper, we argue that the need for a more comprehensive comparative evaluation effort in the case of online density-based clustering algorithms also arises as a consequence of a number of challenges with which the evaluation of event detection techniques in general is confronted.

First, because manually creating a ground truth standard for the large volumes of data generated by Twitter users does not scale (in particular, complex sub-structures in a two-level clustering scenario can consist of a large number of fine-grained sub-topics or sub-events), an automatic labeling of reference data sets in the form of a self-built ground truth (for example, prepared by the latent dirichlet allocation (LDA) method or the EDCoW method) would yield a more streamlined and efficient approach to comparative evaluation than manual evaluation.

An automatic labeling method should be able to generate a clean set of topics that are representative for the time slot of interest and offer a guarantee that the generated set of ground truth topics across time slots are duplicate-free or contain very few duplicates. Since the LDA-generated ground truth topics tend to be biased towards the very method that produces them (due to the inexact nature of the approximation of the posterior distribution during the online inference, topic collation and duplicate topics occurring across different time slots are quite common) the EDCoW method [Weng11] would be more promising for the task (due to the small number of duplicate events or topics generated across time slots).

Yet another challenge posed by the comparative evaluation effort is the type of topic distribution considered for evaluation. For each type of topic distribution (newsworthy topics, trending topics) a classifier needs to be trained based on the characteristics found in the Twitter data set. Training data is very hard to find or to generate. Becker *et al.* [Becker11] implement an approach that uses an online clustering method in combination with a support vector machine classifier. They assemble training data by checking hashtags with special capitalization as well as retweets, replies, and mentions.

Additionally, an automatic evaluation effort should concentrate mainly on precision and recall since these are two relevance measures that can be derived without the help of human evaluators.

A comparative evaluation of online density-based clustering approaches for topic and event detection on Twitter would provide a more objective insight into the performance of different approaches and help investigate to what extent the different parameters of a technique influencing the trade-off between run-time and task-based performance could give rise to event or topic detection techniques adjusting the parameters as a response to changes in the stream.

References

1. [Cao06] Cao F., Ester M., Qian W., Zhou A.: Density-based Clustering over an Evolving Data Stream with Noise. In Proc. Intl. SIAM Conf. on Data Mining (SDM) (2006), pp. 328–339.
2. [Blei03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993–1022, 2003.
3. [Kleinberg02] Jon Kleinberg. Bursty and hierarchical structure in streams. In KDD ’02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 91–101, New York, NY, USA, 2002. ACM.
4. [Yiming98] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and online event detection. In SIGIR ’98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pages 28–36, New York, NY, USA, 1998. ACM.
5. [Weng11] Weng, Jianshu, Yao, Yuxia, Leonardi, Erwin and Lee, Francis Event Detection in Twitter HP Laboratories (2011)
6. [Popovici14] Popovici Robert, Weiler Andreas and Grossniklaus Michael. Online Clustering for Real Time Topic Detection in Social Media Streaming Data. In Proceedings of the SNOW 2014 Data Challenge (2014)
7. [Becker11] Becker, H., Naaman, M., Gravano, L. Beyond trending topics: real-world event identification on Twitter, pp.438-441 (2011)