On-line Clustering for Real-Time Topic Detection in Social Media Streaming Data

Robert Popovici, Andreas Weiler, and Michael Grossniklaus
Database and Information Systems Group, University of Konstanz
P.O. Box 188, 78457 Konstanz, Germany
firstname.lastname@uni-konstanz.de

Abstract

The continuous growth of social networks and the active use of social media services result in massive amounts of user-generated data. Worldwide, more and more people report and distribute up-to-date information about almost any topic. At the same time, there is an increasing interest in information that can be gathered from this data. The popularity of new services and technologies that produce and consume data streams imposes new challenges on the analysis, namely, in terms of handling high volumes of noisy data in real-time. Since social media analysis is concerned with investigating current topics and actual events around the world, there is a pronounced need to detect topics in the data and to directly display their occurrence to analysts or other users. In this paper, we present an on-line clustering approach, which builds on traditional data mining methods to address the new requirements of data stream mining: (a) fast incremental processing of incoming stream objects, (b) compactness of data representation, and (c) efficient identification of changes in evolving clustering models.

1 Introduction and Motivation

The social network platform Twitter is a main producer of large volumes of data as a continuous stream. Over 140 million registered users and about 340 million short messages, called "tweets", per day make Twitter the undisputed market leader in social microblogging today. In its initial stages, Twitter was intended to be a service where people could update their status by posting short messages. Twitter prompted users to answer a simple question "What are you doing?" and thus the users reported their actual activities, feelings, and experiences of their everyday life. As Twitter gained significance and users started exchanging matters reaching beyond one's personal status, it was decided in November 2009 to change the question to a more general one "What's happening?"[1]. The intention of the new question is to engage users in reporting and publishing current news and events happening in the world. The consequence of this change is that Twitter has developed into a vast source of information that contains a mixture of all kinds of data.

Due to the diversity of the information provided, Twitter even plays an increasingly important role as a source for news agencies. In fact, news agencies use Twitter for two important functionalities in their daily activity. First, it is used as a publication and distribution platform for current news articles with a high throughput rate. For example, any reproduction of a tweet ("retweet") reaches an average of about 1,000 users [Kwa10]. Second, news agencies, such as BBC[2], are constantly increasing the usage of Twitter as a reference in their daily news reports [Ton12].

A further characteristic of Twitter is its vibrant user community with a wide range of different personalities from all over the world. It has been shown that this whole spectrum can be sub-divided into a few categories of Twitter usage patterns, such as daily chatter, information and URL sharing, or news reporting [Jav07].

Further research undertaken has discovered that the

In: S. Papadopoulos, D. Corney, L. Aiello (eds.): Proceedings of the SNOW 2014 Data Challenge, Seoul, Korea, 08-04-2014, published at http://ceur-ws.org

[1] http://blog.twitter.com/2009/11/whats-happening.html

[2] http://www.bbc.com/

majority of users publish messages focusing on their personal concerns and matters, whereas a smaller set of users publish for information sharing [Naa10]. This variety of content in the information flow leads to the primary task of detecting significant messages in the clutter of tweets. Because of the fast broadcasting manner of Twitter, important news spread rapidly through the social network.

2 Topic Detection

Most traditional data mining methods such as *K-means*, *DBSCAN*, or *OPTICS* are not designed to be applied directly to data streams because of their infinite nature and the requirement for single pass evolutionary processing. In this paper we focus on a new stream mining method based on traditional data mining methods to address the new requirements of data stream mining: (a) fast incremental processing of incoming stream objects, (b) compactness of data representation, and (c) efficient identification of changes in evolving clustering models.

The proposed algorithmic idea relies on an extended concept of density-based clustering over an evolving data stream with noise (*DenStream* [Cao06]) with enhanced applicability for categorical data. We designed the on-line component of the extended DenStream algorithm to include the major ideas of the classical DenStream algorithm and added some new features and functionalities.

Similar to the ideas of the classical DenStream algorithm, a set of core and outlier micro-clusters is maintained incrementally with the role of outlier and core micro-clusters being often exchanged as a consequence of outdated micro-clusters fading into outliers and new micro-clusters being formed. To speed up processing, an outlier buffer is used to separate the processing of core micro-clusters and the outliers (micro-clusters attracting very few data objects for extensive time intervals). We also extended the general macro-clustering approach with a lightweight variant of the DBSCAN algorithm, which is applied on the micro-clusters as virtual points.

With a view toward achieving a both efficient and accurate estimate of the centroid of the clustering we propose a new approach that uses cluster feature vectors with sufficient summary statistics as components. We use POS tagging to extract a number of relevant features per cluster, the set of selected features consisting mainly of common and proper noun structures.In order to be able to detect new trends in a steadily evolving stream, an incoming data object is assigned to the nearest cluster based on the average of the closest similarity values to the cluster summaries attained by previous objects in the stream. Since the tweet

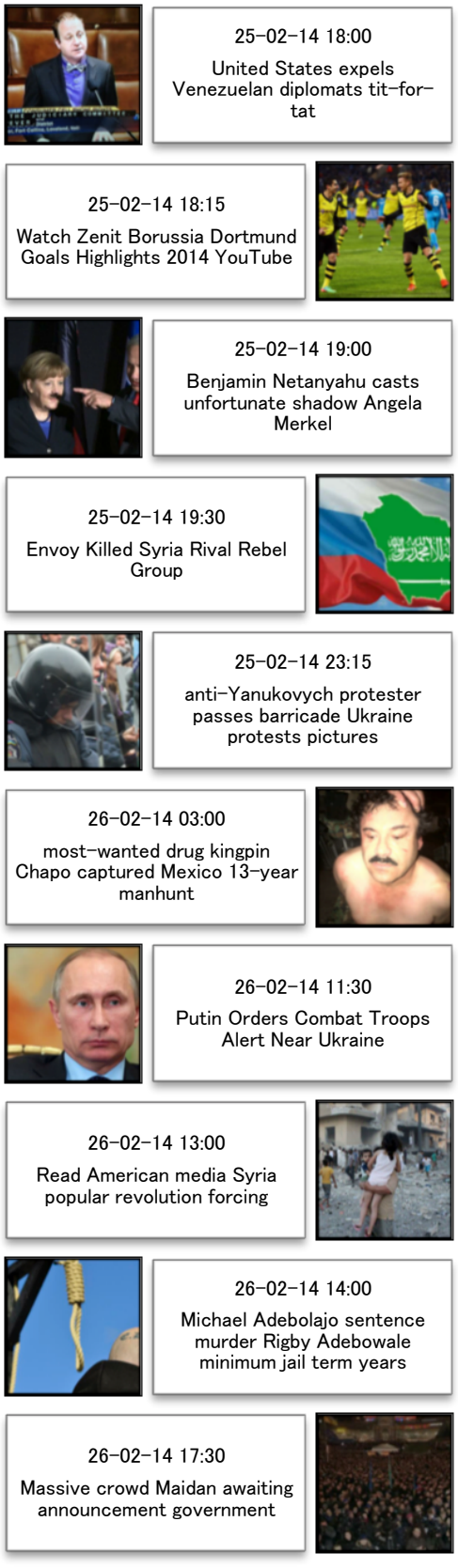

Figure 1: Timeline of 10 sample topics

objects can be very small in size, vector components consist of the inverse cluster frequency of a selected feature combined with the cluster frequency of that same feature. Effectively, the frequency of a selected feature in the cluster is offset by the frequency of that same feature across all documents in the cluster.We refer to these vectors as *CF-ICF vectors*.

The number of selected vector components to be monitored in a given cluster turns out to be exponential to the number of selected unique features, and therefore only a small subset which represents the frequent features needs to be kept. Infrequent features are removed from the vector representation by means of dimensionality reduction to speed up the processing. This also avoids excessive storage and, at the same time, simplifies and summarizes the incoming data, achieving a convergence effect that contributes to reaching a steady distribution of topics. Computation of similarity is done using the cosine similarity metric.

Micro-clusters are maintained incrementally. Effectively, the number of points and the linear sum of term frequencies of the micro-clusters are continuously updated.

We consider the problem of clustering a data stream in the damped window model, in which the weight of each stream object decreases exponentially with time t via an exponential fading function $f(t) = 2^{-\alpha \cdot t}$, where α is a constant called the decay factor and $\alpha > 0$. The fading function controls the importance of the historical data compared to the most recent data by taking into account the timestamp of the last update to the clustering. The higher the value of α, the greater emphasis is placed on the more recent data.

The overall weight W of all stream objects is nearly constant, verified by applying a geometric series to it

$$W = v \cdot \sum_{t=0}^{t_n} \xrightarrow{t \to \infty} \frac{v}{1 - 2^{-\alpha}},$$

where v is the speed of the stream.

During the on-line part we distinguish between potential core-micro-clusters, if $w \geq \beta\mu$ and outlier micro-clusters, with $w \leq \beta\mu$, where β is the outlier threshold and μ the minimum overall weight for a core micro-cluster.

Effectively, for time interval t, if no points are merged into a micro-cluster the weight decreases

$$MC = (\ 2^{-\alpha \cdot t} w, LS, t_c),$$

where LS is the linear sum of the term frequencies and t_c is the creation timestamp of the micro-cluster.

If a data point p is merged the updated micro-cluster is defined as

$$MC = (\ w + 1, LS + 1, t_c).$$

In order to be able to keep track of the evolution of interesting sub-topics as part of a major topic, we introduce the notion of sub-clusters that are incrementally maintained within core micro-clusters in a way similar to which micro-clusters are maintained. Specifically, incoming stream objects are reassigned to the closest sub-cluster by comparing them to vector representations of the sub-cluster summaries.

The sub-cluster summaries consist of the number of data points contained, the linear sums of a feature (LS), the linear sums of occurrences of a feature per window (LSW) and the linear sums of co-occurrences per feature (LSC).

We distinguish between potential core-sub-clusters (*p-sub-cluster*), if $w \geq \beta\mu$ and $N \geq min$ and outlier sub-clusters (*o-sub-cluster*), if $w \leq \beta\mu$, where N is the number of data objects in the sub-cluster, min the minimum number of objects required for a core sub-cluster,β is the outlier threshold and μ the minimum overall weight for a core sub-cluster.

For time interval t, if no points are merged into a sub-cluster, the weight decreases

$$SC = (\ 2^{-\alpha \cdot t} w, LS, LSW, LSC, t_c).$$

If a data point p is merged the updated sub-cluster is defined as

$$SC = (\ w + 1, LS + 1, LSW + 1, LSC + 1, t_c).$$

If an outlier micro-cluster has attracted sufficient data to be converted into a core micro-cluster, data objects that have been assigned to the latter are redistributed to underlying sub-clusters. This effectively means that an incoming stream object that has been assigned to a nearest micro-cluster is reassigned to its nearest sub-cluster, 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 one previously attained, the mean of the last three closest similarity values to the sub-cluster summaries is maintained. The similarity values are additively maintained, to increase efficiency. Algorithm 1 defines the extended merging procedure, which is also visualized in Figure 3.

The potentially unbounded nature and uncertain arriving speed of data streams along with the requirement of single pass scanning imposes a limited space (memory) and a strict time constraint to the implementation of the data stream processing. Therefore, a checking strategy is performed every Tp time steps, where Tp is defined as the minimal timespan for a cluster fading into an outlier. This ensures that outdated clusters that have either received few data or have had their weight reduced by the decay factor α are pruned.

Algorithm 1 Extended `DenStream`: Merging technique

Require: $\epsilon_1 \geq 0 \leq 1, \epsilon_2 \geq 0 \leq 1$;
  {1}: Try to merge p into its nearest p-micro-cluster *cp*;
  if dp (the closest similarity value) ¡ ϵ_1 **then**
    Merge p into cp;
    {2}: Try to merge p into the nearest p-sub-cluster *csp* of p-micro-cluster *cp*;
    if dsp (the closest similarity value) ¡ ϵ_2 **then**
      Merge p into *csp*;
    else
      {3}:Try to merge p into the nearest o-sub-cluster *cso* of p-micro-cluster *cp*;
      if dso (the closest similarity value) ¡ ϵ_2 **then**
        Merge p into *cso*;
      else
        Create a new o-sub-cluster containing p
      end if
    end if
  else
    {4}:Try to merge p into its nearest o-micro-cluster *co*;
    if do (the closest similarity value) ¡ ϵ_1 **then**
      Merge p into *co*;
      if w (the new weight of *co*) ¿ $\beta\mu$ **then**
        Convert *co* into a p-micro-cluster and create a new o-sub-cluster with all stream objects of the converted o-micro-cluster;
      else
        Create a new o-micro-cluster containing p
      end if
    end if
  end if

Otherwise, they will take up a lot of memory space, and either the clustering result may contain outdated data with the immediate effect of lessening the evolving character of the data stream or clusters consisting of outliers will combine data that should not be in the same cluster into a same cluster in subsequently merging micro-clusters, thus decreasing clustering efficiency.

The weight of outlier micro-clusters and outlier sub-clusters is compared against

$$\theta(t) = \frac{2^{-\alpha(t-t_c+T_p)} - 1}{1 - 2^{-\alpha}},$$

where t_c is the creation timestamp of the outlier micro-cluster and Tp is the minimal timespan for a micro-cluster/sub-cluster fading into an outlier. Outlier sub-clusters that have been turned into core sub-clusters will have a lifespan at least as long as the core micro-cluster to which they belong.

The longer an outlier micro-cluster or outlier sub-cluster exists, the higher its expected weight

$$\lim_{t_c \to \infty} \theta(t_c) = \frac{1}{1 - 2^{-\alpha \cdot T_p}} = \beta\mu.$$

Based on this assumption, the cumulated maximal number of micro-clusters and sub-clusters in memory is $\frac{W}{\beta \cdot \mu}$, where W is the overall weight of the data streams and $\beta \cdot \mu$ acts as the filtering parameter. Therefore, the runtime complexity of the extended Denstream algorithm is $O(\frac{W}{\beta \cdot \mu} + x)$, where x is the length of the stream and $\frac{W}{\beta \cdot \mu}$ is the maximal cumulated number of core micro-clusters and core sub-clusters in memory. As a consequence of the pruning strategy and the dimensionality reduction, memory increases only logarithmically with stream length. The pruning technique used by our algorithm is shown in Figure 2, where N is the number of objects and min the minimum number of objects.

To handle the case where micro-clusters created independently might at some point during the clustering turn out to contain topics that are semantically related we implemented a modified lightweight variant of the DBSCAN algorithm that runs periodically (every N stream objects, with N typically set to 10,000) on the live set of micro-clusters as virtual points. The intuition behind this is based on the symmetric property of density-connectedness of the DBSCAN algorithm.

Apart from the immediate effect of semantic compaction of the topic distribution, the macro-clustering phase (see Figure 2) also effectively reduces the number of micro-clusters to be processed. While the core sub-clusters of the merged micro-clusters are added to the set of the existing core sub-clusters of the merging micro-cluster, the set of outlier sub-clusters effectively

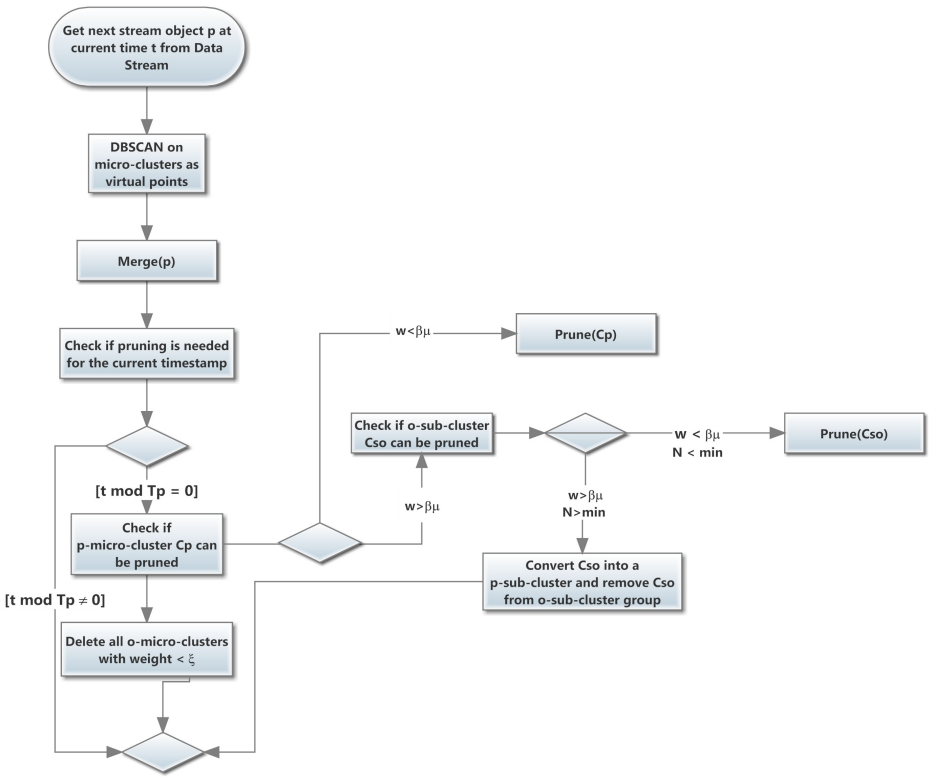

Figure 2: Extended `DenStream`: pruning technique, DBSCAN macroclustering phase

contribute only their summary statistics to the merging micro-cluster. The notion of distance translates to the number of common top keywords defining the stable distribution of the micro-clusters at some point during the clustering, thus effectively replacing the epsilon parameter required for the DBSCAN algorithm.

3 Evaluation

The SNOW challenge [Pap14] data set was collected for a 24-hour time frame from Tue 25 Feb 2014, 18:00 GMT to Wed 26 Feb 2014, 18:00 GMT. The collection contains a total of 1,041,227 tweets with an average of about 723 tweets per minute and about 10,846 tweets per 15-min window.

The evaluation was performed on a computer with Intel i7 CPU and 6 GB main memory running the 64-bit Eclipse Indigo Platform on 64-bit Windows 8. We implemented our algorithm[3] in Java. We used a well-known language detector [Shu10], and a tokenizer and part-of-speech tagger for Twitter [Gim11], with training data of manually labeled annotated tweets and hierarchical word clusters from unlabeled tweets. Furthermore, we used a standard English stop word list to remove repeating terms and and simple plural stemming to match the different forms of terms with each other. The n-gram signature of a topic consisted of the top keywords within the range of at least 1.5 standard deviations away from the mean derived from the list of term frequencies per cluster.

The tweet with the highest degree of semantic relevance within a cluster (that is, having the maximal similarity value to the cluster summaries) was selected as the topic headline. The selected tweet was parsed using a less restrictive configuration of the POS tagger, with the extracted tokens reassembled in the same syntactic order in which they were originally processed to ensure minimum semantic coherence.

Micro-clusters pruned away by the exponential fading function were written to a separate file and then joined with the list of active micro-clusters to produce the final result.

The final results consist of a total number of 210 topics (see Figure 1 for a sample) over the whole 24-hour time frame with a significance factor of at least 200 tweets per cluster. A detected topic had to be at least 150–200 tweets in length or span over a 15 minute interval. The top keywords were derived from the term frequency lists maintained per window interval. The top tweets were selected based on the closest similarity values between incoming tweets to the cluster vectors within the first window interval until convergence was attained. The same procedure was applied for finding the pictures, which are associated to a topic.

In particular, topics referring to the political up-

[3]Available here: `http://bit.ly/1qeGNry`

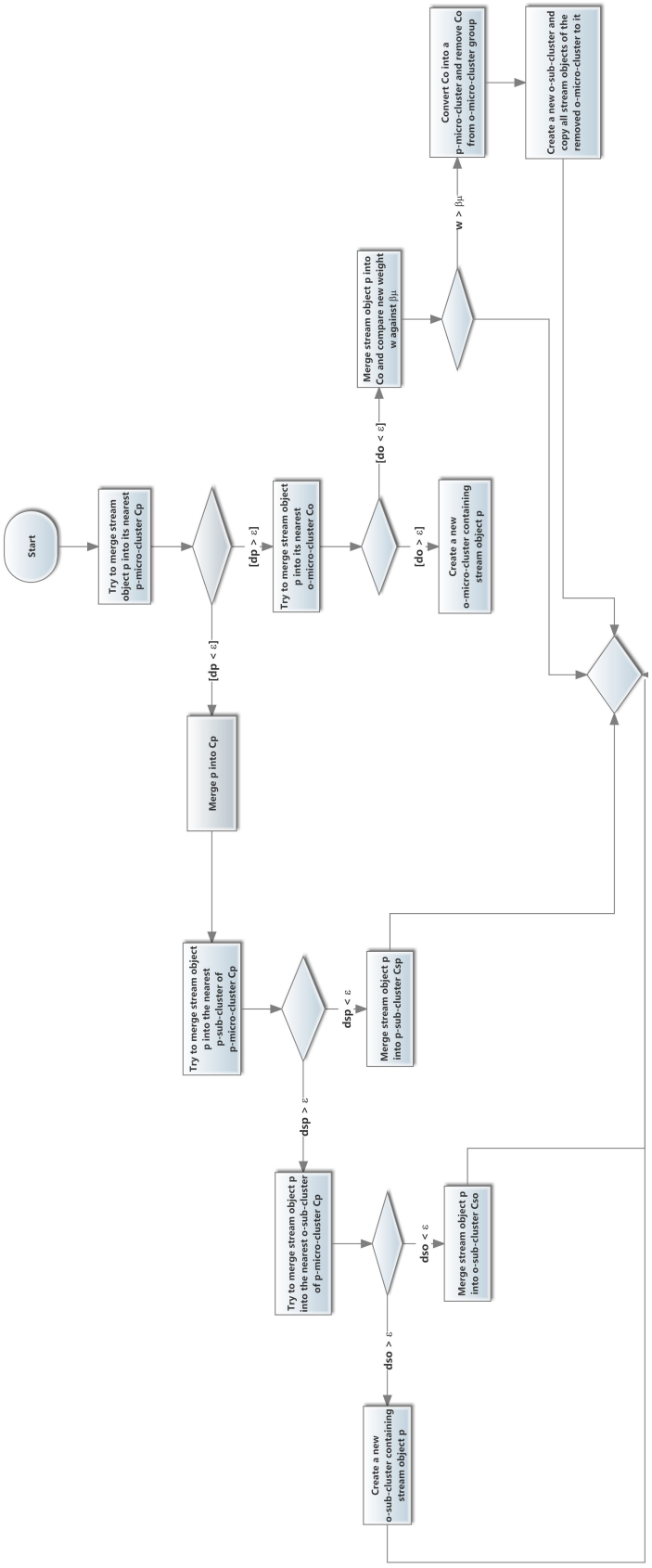

Figure 3: Extended `DenStream`: merging technique

heaval in Ukraine, the *Bitcoin* exchange shutdowns due to alleged hacker theft, the clashes between rebels and the Syrian government forces in Syria and the *Champions League* results were most prominent in the topic distribution, containing more than 12,000 tweets. Since the major topics (mostly macro-clusters found by the DBSCAN algorithm) spanned over large time intervals yet contained a large diversity of sub-topics that were sufficiently different from each other, only the contained sub-topics were written to the result file to meet the interval requirement. For each sub-cluster, 15-minute intervals were output for which there was a significant difference in the n-gram signature between two successive windows of the respective sub-cluster (to avoid duplicates).

An example of a major topic with component sub-topics are the events revolving around the Syrian conflict in general, e.g., the major ambush involving rebels in Damascus, Germany monitoring jihadis in battle-hardened Syria, the photos of the Yarmouk refugee camp in Syria, Syrian al Qaeda giving rival rebel group an ultimatum. This kind of approach might prove useful in helping journalists gain more insight into ongoing events and perhaps acquire a better understanding of the significance of more complex events by assessing their impact on a more global scale while at the same time allowing them to maintain the focus on the more detailed aspects of those events.

The official evaluation results of our method in the Data Challenge are included in Papadopoulos *et al.* [Pap14].

References

[Cao06] Cao F., Ester M., Qian W., Zhou A.: Densitybased Clustering over an Evolving Data Stream with Noise. In *Proc. Intl. SIAM Conf. on Data Mining (SDM)* (2006), pp. 328–339.

[Gim11] Gimpel K., Schneider N., O'Connor B., Das D., Mills D., Eisenstein J., Heilman M., Yogatama D., Flanigan J., Smith N. A.: Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proc. Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers (HLT)* (2011), pp. 42–47.

[Jav07] Java A., Song X., Finin T., Tseng B.: Why We Twitter: Understanding Microblogging Usage and Communities. In *Proc. Intl. Workshop on Web Mining and Social Network Analysis* (2007), pp. 56–65.

[Kwa10] Kwak H., Lee C., Park H., Moon S.: What is Twitter, a Social Network or a News Media? In *Proc. Intl. Conf. on World Wide Web* (2010), pp. 591–600.

[Naa10] Naaman M., Boase J., Lai C.-H.: Is It Really About Me?: Message Content in Social Awareness Streams. In *Proc. Intl. Conf. on Computer Supported Cooperative Work (CSCW)* (2010), pp. 189–192.

[Pap14] Papadopoulos S., Corney D., Aiello L. M.: SNOW 2014 Data Challenge: Assessing the Performance of News Topic Detection Methods in Social Media. In *Proc. SNOW 2014 Data Challenge* (2014).

[Shu10] Shuyo N.: Language Detection Library for Java. `http://code.google.com/p/language-detection/`, 2010.

[Ton12] Tonkin E., Pfeiffer H. D., Tourte G.: Twitter, Information Sharing and the London Riots? *Bulletin of the American Society for Information Science and Technology 38*, 2 (2012), 49–57.