SigniTrend: Scalable Detection of Emerging Topics in Textual Streams by Hashed Significance Thresholds

Erich Schubert, Michael Weiler, Hans-Peter Kriegel

Institute of Informatics
Database Systems Group
Ludwig-Maximilians-Universität München
Trend detection on streams should be early and accurate

Twitter Streaming API on Feb. 19th 2014
Trend detection on streams should be early and accurate.

Twitter Streaming API on Feb. 19th 2014
Trend detection on streams should be early and accurate

Twitter Streaming API on Feb. 19th 2014
Trend detection on streams should be early and accurate
Trend detection on streams should be early and accurate

Facebook bought WhatsApp

Term frequency

0
14
28
42
56
70
10:47
10:49
10:51
10:53
10:55
10:57
10:59
11:01
11:03
11:05
11:07
11:09
11:11
11:13

Facebook bought WhatsApp

Michael Weiler | SigniTrend: Scalable Detection of Emerging Topics in Textual Streams by Hashed Significance Thresholds | Page 1/10
Problem description

1. **Statistical significance score**
   Popular topics ≠ trending topics (e.g. Obama)

2. **Track interacting terms**
   - Facebook bought WhatsApp
   - Edward Snowden traveled to Moscow

3. **Scalability**
   Efficient calculation for all terms and pairs
SigniTrend on textual streams

tracking both: single terms and pairs

A. Preprocessing (stopwords, stemming, duplicates)

B. Trend detection cycle

• Temporal slicing for statistical aggregation
• Score all terms and pairs based on expectations from past slices

C. Refinement with clustering
Trend detection cycle

1. **Terms and pairs**
2. **Count frequency**
   - **exceeds threshold?**
   - **Trend candidates**
3. **Update statistics**
   - **at the end of each time slice**
   - **new alerting thresholds**
Update statistics
for time slice \( t \) and term or pair \( e \)

• How many standard deviations is the current frequency \( x \) higher than its mean

\[
z(x_{t,e}) := \frac{x_{t,e} - \mu_{t-1,e}}{\sigma_{t-1,e}}
\]
Update statistics
for time slice t and term or pair e

• How many standard deviations is the current frequency \( x \) higher than its mean

\[
z(t,e) := \frac{x(t,e) - \text{EWMA}_{t-1,e}}{\sqrt{\text{EWMVar}_{t-1,e}}}\]

• Exponential weighted moving average/variance for continuous estimation on a stream [Finch09]

\[
\begin{align*}
\Delta_{t,e} & \leftarrow x(t,e) - \text{EWMA}_{t-1,e} \\
\text{EWMA}_{t,e} & \leftarrow \text{EWMA}_{t-1,e} + \alpha \cdot \Delta_{t,e} \\
\text{EWMVar}_{t,e} & \leftarrow (1 - \alpha) \cdot (\text{EWMVar}_{t-1,e} + \alpha \cdot \Delta_{t,e}^2)
\end{align*}
\]

Significance and frequency for term “Facebook”
How to track statistics of all pairs efficiently?

Problem: Too many terms and pairs to track everything

2013 News Dataset

<table>
<thead>
<tr>
<th>STEMMED TERMS</th>
<th>OBSERVED PAIRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL</td>
<td>56,661,782</td>
</tr>
<tr>
<td>UNIQUES</td>
<td>300,141</td>
</tr>
</tbody>
</table>

Therefore, we designed an efficient hashing scheme (based on Bloom Filters and Heavy Hitters) for probabilistic upper-bound statistics.
Hashing scheme for efficient tracking
L=7 buckets, K=2 hash functions
Hashing scheme for efficient tracking

L=7 buckets, K=2 hash functions

{WhatsApp}: 60

h₁ h₂

45 ± 30 45 ± 30
Hashing scheme for efficient tracking

L=7 buckets, K=2 hash functions

{WhatsApp}: 60

{Facebook, WhatsApp}: 2

h₁  h₂

45 ± 30  2 ± 1  45 ± 30  2 ± 1
Hashing scheme for efficient tracking
L=7 buckets, K=2 hash functions

{WhatsApp}: 60

{Facebook, WhatsApp}: 2

{Obama, US}: 25

45 ± 30
2 ± 1
45 ± 30
2 ± 1
20 ± 10
20 ± 10
Hashing scheme for efficient tracking

$L=7$ buckets, $K=2$ hash functions

{WhatsApp}: 60

{Facebook, WhatsApp}: 2

{Obama, US}: 25

write MAX (upper bound)

45 ± 30  2 ± 1  45 ± 30  2 ± 1  20 ± 10
Hashing scheme for efficient tracking

$L=7$ buckets, $K=2$ hash functions

Upper-bound estimate for mean and its variance

\begin{align*}
\text{read } \{\text{Obama, US}\}: & \quad \min(45 \pm 30, 20 \pm 10) = 20 \pm 10 \\
\text{write } \text{MAX} & \quad \text{(upper bound)}
\end{align*}
Hashing scheme for efficient tracking

L=7 buckets, K=2 hash functions

write MAX
(upper bound)

read MIN
(lowest collision)

Upper-bound estimate for mean and its variance

Performance on news dataset: 104s/day with a Raspberry-Pi
Artificial trends evaluation

Inject artificial words with frequency $\alpha$
e.g. “Obama meets $<X123>$ Netanyahu”

Hash table size large enough $\rightarrow$ recall saturation
Refinement & clustering

• Inverted index (Apache Lucene) to verify trend candidates and measure exactly (without hashing) for precise reporting (false-positives can be eliminated)

• Single Link clustering with Ward of remaining trends (similarity matrix is built with the exact significance of all pairs)

• Future work: include topic modeling techniques (e.g. pLSI, LDA)
Thank You!

Questions?