Scalable Detection of Emerging Topics and Geo-spatial Events in Large Textual Streams

Erich Schubert1,2 Michael Weiler1 Hans-Peter Kriegel1

1 Lehr- und Forschungsstelle Datenbanksysteme, Ludwig-Maximilians-Universität München
2 Lehrstuhl für Datenbanksysteme, Ruprecht-Karls-Universität Heidelberg

{schubert,weiler,kriegel}@dbis.f4.lmu.de

Objective and Summary

• Scalable: able to process years of news and Twitter
• Detection: topics and keywords should not need to be defined beforehand
• Emerging: significant increase (cf. “Trending Topics”)
• Topics: not every single message, but groups of related messages are of interest
• Geo-spatial Events: observe locality and able to detect geographic change and differences

Change Model and Implementation

For every word, word-word-, or word-location-pair \((w,l)\) we use a 2-score-like significance:

\[
\zeta(w,l) = \max \left\{ \frac{\text{EWMA}[f(w,l), t]}{\sqrt{\text{EWMA}[f(w,l), t] + \beta}} \right\}
\]

where \(f(w,l)\) is the observed frequency of pair \((w,l)\), \(\text{EWMA}\) Exponentially-weighted moving average, \(\beta\) Laplace-style smoothing term (for rare words)

Because we cannot afford to store and maintain all \(\text{EWMA}[f(w,l), t]\) values, we employ a Bloom-filter-like hashing strategy to estimate them efficiently.

Bloom-filter / Heavy Hitters

Counting Bloom filters increment each hash bucket. When estimating counts, the minimum found in the buckets is used as estimate. Here, \(h = 3\) buckets are used:

<table>
<thead>
<tr>
<th>Word</th>
<th>Apple</th>
<th>Banana</th>
<th>Cherry</th>
</tr>
</thead>
<tbody>
<tr>
<td>min(2, 0, 1) = 0</td>
<td>min(2, 3, 2) = 2</td>
<td>min(2, 1, 4) = 1</td>
<td></td>
</tr>
</tbody>
</table>

Counting Bloom filters never underestimate, but if a term has \(h\) hash collisions with more frequent terms, it may overestimate the true frequency.

Table Hash Maintenance

<table>
<thead>
<tr>
<th>Old counts</th>
<th>Read minimum</th>
<th>Increment</th>
<th>Write if new &lt; c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old counts</td>
<td>Read minimum</td>
<td>Increment</td>
<td>Write if new &lt; c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word</th>
<th>Read minimum</th>
<th>Increment</th>
<th>Write if new &lt; c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>0</td>
<td>2</td>
<td>0&lt;br&gt;&lt;br&gt;Banana</td>
</tr>
</tbody>
</table>

(c) Vectors statistics table update

<table>
<thead>
<tr>
<th>Counts</th>
<th>Old EWMa</th>
<th>Old EWMaVar</th>
<th>New EWMa</th>
<th>New EWMaVar</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

(c) Check thresholds for new events

<table>
<thead>
<tr>
<th>Old EWMa</th>
<th>Old EWMaVar</th>
<th>New EWMa</th>
<th>New EWMaVar</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Experiment

Data Set for SigniTrend Experiments (2014)

News: articles from 2013 of Reuters and Bloomberg news. Twitter: 114 days of the 1% Twitter sample, originally 279 million tweets before filtering duplicates, retweets, and non-English tweets.

StackOverflow: dump of the main programming Q&A site for years 2010 to 2013.

Experiment: New Year Around the World

Top Events in News 2014 (Chronological)


Key Ideas of our Solution

• From statistics: control charts for change detection.
• From computational linguistics: Analyze word cooccurrences for more meaningful results.
• From mathematics: Exponentially weighted moving averages for streaming operation.
• From databases: Hashing and Count-Min sketches for scalability to large data.
• From data mining: Clustering of word pairs into simple “topics” based on cooccurrences.
• From visualization: Word-cloud like visualization, but incorporating the relationships of words.
• Integrate geographic information by mapping coordinates to tokens similar to text.

The big challenge is scalability to millions of word pairs, at thousands of Tweets per second!

Tracking all Cooccurrences

Word combinations are interesting:

• “Facebook” bought “WhatsApp”
• Edward “Snowden” traveled to “Moscow”
• “Putin”, “Obama” and “Merkel” — their interactions are more interesting than their frequency

Why is the most popular term?

• “@justinbieber” is always popular on Twitter
• Domain specific stopwords (e.g. “follow me”, “BT”)
• Cultural, language- and geographic differences

Word why pairs and not just words?

Pairs allow the discovery of interactions and structure.

Integrating Geographic Information

We map geographic data to tokens

<table>
<thead>
<tr>
<th>(longitude, latitude)</th>
<th>Symbol</th>
<th>...</th>
</tr>
</thead>
</table>

such that nearby locations produce the same symbol.

Example tokenization of a tweet:

“Peaches Geldof died of heroin overdose...”

Riverina / Krajina / ...