# Research project proposal

Your Name

Date

## Event Detection in Twitter Streams

Twitter is a social awareness stream, where users post tweets about events in the real world in real-time. This allows the Twiiter data stream to be a source of real-time event detection. Models and algorithms for process this stream to: a) identify the event and b) use the traffic of discussion of the event to rank its importance.

This basis of my project is to use frequent item counting in data streams to detect the events and the volume of their trafffic.

## Background

Frequent item mining in data streams is a classic problem in the field of data mining with data streams. The most basic approach centers around counting the frequency of keywords in a rolling window. A central issue related to this approach is accurately counting keywords so that synonyms, hashtags and misspellings that mean the same thing are counted correctly.

## Data sources

I plan to use Twitter data using the Twitter public REST API
https://dev.twitter.com/rest/public

I plan to use Twitter's REST API using R.

http://www.r-bloggers.com/talking-to-twitters-rest-api-v1-1-with-r/

WordNet: An Electronic Lexical Database https://wordnet.princeton.edu/

## Algorithms

I will also be using the UCS Toolkit in R (UCS Toolkit Documentation)

- Download UCS Toolkit in R: UCS-0.6.tar.gz

I plan to use several natural language processing algorithms such as:

-Viterbi Algorithm: https://en.wikipedia.org/wiki/Viterbi_algorithm

-Forward-Backward Algorithm: https://en.wikipedia.org/wiki/Forward%E2%80%93backward_algorithm

-CYK Algorithm: https://en.wikipedia.org/wiki/CYK_algorithm

tf–idf (term frequency–inverse document frequency) algorithm https://en.wikipedia.org/wiki/Tf%E2%80%93idf

Mutual information https://en.wikipedia.org/wiki/Mutual_information

The Mutual information quantifies the mutual dependence of the two random variables. It is a measure of the “stickiness” between two items. It measures how much knowing one of these variables reduces uncertainty about the other. We can use mutual information to quantify the association between two tags (words). Mutual information is given by:

$$MI(X,Y) = \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} \ln \left( \frac{p_{ij}}{p_{i.} \cdot p_{j}} \right)$$

where $p_{ij}$ is the joint probability distribution function of X and Y, and $p_i$ and $p_i$ are the marginal probability distribution functions of X and Y respectively.

## Resources

There are a couple resources and tutorials that already exist for my data sources, such as:

Codeacademy - Twitter API Tutorial

UCS R Tutorial

## UCS Toolkit in R

The UCS Toolkit in R was created for Stefan Evert's PhD dissertation "The Statistics of Word Co-occurrences Word Pairs and Collocations." (Evert 2004, 2013) for efficiently calculating measures of association in large text data,

Small term-document matrices are created using the R text mining package tm. For processing larger amounts of text, tag counts are generated as described in the section "Counting Tags in Text." However rather than summing the tag counts over the individual entities of text (e.g. tweets, abstracts, webpages, etc.) the tag-count data for each entity is first stored in associative array, denoted A. The mapper emits key-value pairs with an entity id as the key and the corresponding associative array of tag counts as values. The source already has an entity id like a tweet id or a PubMed id then that id is used otherwise a unique id is created.

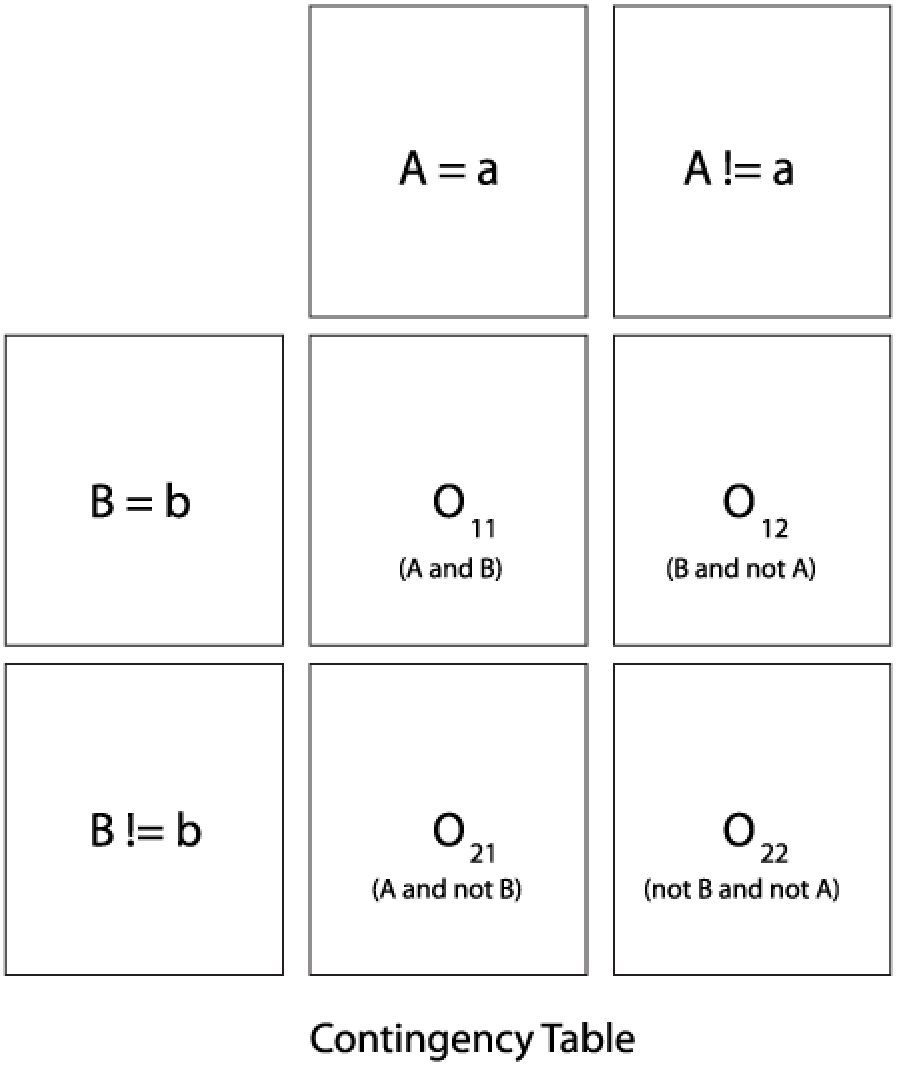
Most of our processing uses the Bag-of-words model in which the text of a document is represented as an unordered collection of words. As such, the reducer can take the output of the mapper without further processing as we already have an entity id as the key and the corresponding associative array (Bag-of-words) as its model. When a full n-tag by n-tag term-document matrix is needed the reducer fills in zeros for all missing tags. For large Hadoop jobs both the "pairs" and "stripes" MapReduce algorithms to count co-occurrence as described in "Data-Intensive Text Processing with MapReduce" by Jimmy Lin, Chris Dyer (Lin et al 2010) was used.

*Creating Tag Co-occurrence Matrices*

For small amounts of text the tag co-occurrence matrix was created using the same Bag-of-words mapper that emits key-value pairs with an entity id as the key and the corresponding associative array of tag counts as values as described in the section "Creating Term-Document Matrices from Text."

For larger data we used a "frequency signature" approach to convert the Bag-of-words output to a format that we can use to calculate tag co-occurrence associations and mutual information. Frequency signatures are described in detail in Stefan Evert's PhD dissertation "The Statistics of Word Cooccurrences Word Pairs and Collocations." (Evert 2004)

To calculate tag co-occurrence associations and mutual information for two tags, A and B, we need four items of data. The co-occurrence count of A and B, the count of A but not B, the count of B but not A, and the total number of tags in a corpus. This co-occurrence frequency data for a word pair (A,B) are usually organized in a contingency table show below. The contingency table stores the observed frequencies$O_{11}$ … $O_{22}$. The table below (adapted from Evert's dissertation) shows an observed contingency table.



|  | A = a | A != a |
|---|---|---|
| B = b | $O_{11}$<br>(A and B) | $O_{12}$<br>(B and not A) |
| B != b | $O_{21}$<br>(A and not B) | $O_{22}$<br>(not B and not A) |

Contingency Table

*Contingency table : $O_{11}$ is co-occurrence count of A and B, $O_{12}$ is the count of A but not B, $O_{21}$ is the count of B but not A, and $O_{22}$ is the count of not B and not A.*

However, while the co-occurrence count of A and B, and the total number of tags in a corpus are efficiently and easily counted the count of A but not B, the count of B but not A are tricky and computationally expensive. The insight and advantage of frequency signatures is that they calculate the count of A but not B, the count of B but not A by just counting A and B and the co-occurrence count of A and B. That is, the count of A but not B is equal to count of A minus the co-occurrence count of A and B. Likewise, the count of B but not A is equal to count of B minus the co-occurrence count of A and B.

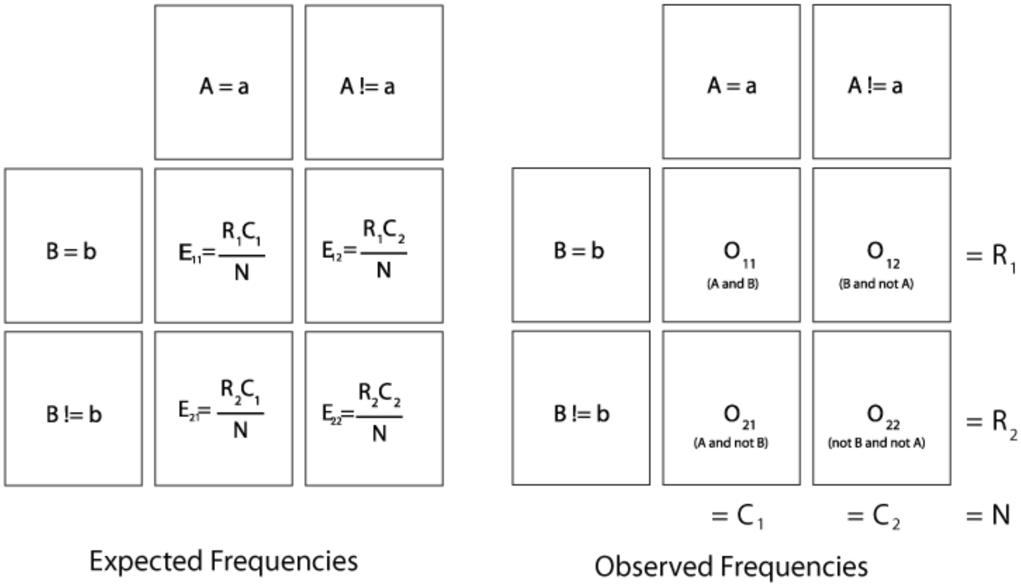
The frequency signature of a tag pair (A, B) is usually written as (f, f1, f2,N). Where f is the co-occurrence count of A and B, f1 is the count of A but not B, f2 is the count of B but not A, and N is the total counts. Notice that the observed frequencies $O_{11}$, ..., $O_{22}$ can be directly calculated from the frequency signature by the equations below:

1. $O_{11} = f$
2. $O_{12} = f1 - f$
3. $O_{21} = f2 - f$
4. $O_{22} = N - f1 - f2 + f$

Generating all of the data tag co-occurrence association and mutual information calculations using this approach can be generated using a single pass of the data and two associative arrays; one of the tag counts and another for the tag co-occurrence counts.

*Calculating Associations and Mutual Information from Frequency Signatures*

Evert shows the many association and mutual information statistics can be calculated from the observed frequencies $O_{11}$, ..., $O_{22}$ if we can generate the expected frequencies $E_{11}$, ..., $E_{22}$. [99] The table below (adapted from Evert's dissertation) shows the expected versus observed contingency tables.



|  | A = a | A != a |
|---|---|---|
| B = b | $E_{11}=\frac{R_1C_1}{N}$ | $E_{12}=\frac{R_1C_2}{N}$ |
| B != b | $E_{21}=\frac{R_2C_1}{N}$ | $E_{22}=\frac{R_2C_2}{N}$ |

Expected Frequencies

|  | A = a | A != a |  |
|---|---|---|---|
| B = b | $O_{11}$<br>(A and B) | $O_{12}$<br>(B and not A) | $= R_1$ |
| B != b | $O_{21}$<br>(A and not B) | $O_{22}$<br>(not B and not A) | $= R_2$ |
|  | $= C_1$ | $= C_2$ | $= N$ |

Observed Frequencies

The sum of all four observed frequencies (called the sample size N) is equal to the total number of pair tokens extracted from the corpus. R1 and R2 are the row totals of the observed contingency table, while C1 and C2 are the corresponding column totals. The expected frequencies can be directly calculated from observed frequencies $O_{11}$, ..., $O_{22}$ by the equations below:

a. $R1 = O_{11}+O_{12}$
b. $R2 = O_{21}+O_{22}$
c. $C1 = O_{11}+O_{21}$
d. $C2 = O_{12} + O_{22}$
e. $N = O_{11}+O_{12}+O_{12} + O_{22}$

Evert went on to show that several association measures can be easily calculated once one has the expected and observed contingency tables. For example, the pointwise mutual information (MI) is calculated by below.

*pointwise mutual information*

$$MI = \ln(\frac{O_{11}}{E_{11}})$$

The Likelihood measures that can be calculated using the expected and observed contingency tables are: multinomial-likelihood, binomial-likelihood, Poisson-likelihood, the Poisson-Stirling approximation, and hypergeometric-likelihood. The exact hypothesis tests that can be calculated using the expected and observed contingency tables are: binomial test, Poisson test, and Fisher's exact test. The asymptotic hypothesis tests that can be calculated using the expected and observed contingency tables are: z-score, Yates' continuity correction, t-score (which compares $O_{11}$ and $E_{11}$ as random variates), Pearson's chi-squared test, and Dunning's log-likelihood (a likelihood ratio test). The measures from information theory that can be calculated using the expected and observed contingency tables are: MI (mutual information, mu-value), logarithmic odds-ratio logarithmic relative-risk, Liddell's difference of proportions, MS (minimum sensitivity), gmean (geometric mean) coefficient, Dice coefficient (aka. "mutual expectation"), Jaccard coefficient, ,MIconf (a confidence-interval estimate for the mu-value), MI (pointwise mutual information), local-MI (contribution to average MI of all co-occurrences), average-MI (average MI between indicator variables).

Stefan Evert also developed a R library called UCS toolkit (Evert 2013) for the statistical analysis of co-occurrence data with association measures and their evaluation in a collocation extraction task.

## References

Alon, N.; Matias, Y.; Szegedy, M. The space complexity of approximating the frequency moments. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 20{29. ACM, 1996.

Arasu, A. and Manku, G. Approximate counts and quantiles over sliding windows. In Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 286{296. ACM, 2004.

Becker, H.; Naaman, M.; Gravano, L. Beyond trending topics: Real-world event identification on twitter, 2011.

Cataldi, M.; Di Caro, L.; Schifanella, C. Emerging topic detection on twitter based on temporal and social terms evaluation. In Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD '10, pages 4:1{4:10, New York, NY, USA, 2010. ACM.

Charikar, M.; Chen, K.; Farach-Colton, M. Finding frequent items in data streams. Automata, Languages and Programming, pages 784{784, 2002.

Cormode, G. and Hadjieleftheriou, M. Finding frequent items in data streams. Proceedings of the VLDB Endowment, 1(2):1530{1541, 2008.

Cormode, G and Muthukrishnan, S. What's new: Finding significant differences in network data streams. In INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, volume 3, pages 1534{1545. IEEE, 2004.

Cormode, G. and Muthukrishnan, S. An improved data stream summary: the count-min sketch and its applications. Journal of Algorithms, 55(1):58{75, 2005.

Evert, Stefan 2004. The Statistics of Word Cooccurrences: Word Pairs and Collocations. PhD dissertation, University of Stuttgart. (www.collocations.de)

Evert, Stefan 2013. UCS toolkit. Retrieved from http://www.collocations.de/software.html

Fang, M.; Shivakumar, N.; Garcia-Molina, H.; Motwani, R; Ullman. J.D. Computing iceberg queries efficiently. In Internaational Conference on Very Large Databases (VLDB'98), New York, August 1998. Stanford InfoLab, 1999.

Fellbaum, Christiane. 1998. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

Gibbons, P.B. and Matias, Y. New sampling-based summary statistics for improving approximate query answers. In ACM SIGMOD Record, volume 27, pages 331{342. ACM, 1998.

Greenwald, M. and Khanna, S. Space-efficient online computation of quantile summaries. In ACM SIGMOD Record, volume 30, pages 58{66. ACM, 2001.

Hung, R.; Lee, L.; Ting, H. Finding frequent items over sliding windows with constant update time. Information Processing Letters, 110(7):257{260, 2010.

Karp, R.; Shenker, S.; Papadimitriou, C. A simple algorithm for finding frequent elements in streams and bags. ACM Transactions on Database Systems (TODS), 28(1):51{55, 2003.

Lee, L. and Ting, H. A simpler and more efficient deterministic scheme for finding frequent items over sliding windows. In Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 290{297. ACM, 2006.

Lin, Jimmy; Dyer, Chris; Hirst, Graeme. 2010. Data-Intensive Text Processing with MapReduce (Synthesis Lectures on Human Language Technologies) Morgan and Claypool Publishers

Miller, George A.   1995. WordNet: A Lexical Database for English.  Communications of the ACM Vol. 38, No. 11: 39-41.

Manku, G. and Motwani, R. Approximate frequency counts over data streams. In Proceedings of the 28th international conference on Very Large Data Bases, pages 346{357. VLDB Endowment, 2002.

Mathioudakis, M. and Koudas, N. Twittermonitor: trend detection over the twitter stream. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, SIGMOD '10, pages 1155{1158, New York, NY, USA, 2010. ACM.

Metwally, A.; Agrawal, D.; Abbadi, A. An integrated efficient solution for computing frequent and top-k elements in data streams. ACM Transactions on Database Systems (TODS), 31(3):1095{1133, 2006.

Misra, J. and Gries, D. Finding repeated elements. Science of Computer Programming, 2(2):143{152, 1982.

Motwani, R.; Widom, J.; Arasu, A.; Babcock, B.; Babu, S.; Datar, M.; Manku, G.; Olston, C.; Rosenstein, J.; Varma, R. Query processing, resource management, and approximation in a data stream management system. CIDR, 2003.

Naaman, M.; Becker, H.; Gravano, L. Hip and trendy: Characterizing emerging trends on twitter. J. Am. Soc. Inf. Sci. Technol., 62(5):902{918, May 2011.

Sakaki, T.; Okazaki, M.; Matsuo, Y. Earthquake shakes twitter users: real-time event detection by social sensors. In Proceedings of the 19th international conference on World wide web, WWW '10, pages 851{860, New York, NY, USA, 2010. ACM.

Shrivastava, N.; Buragohain, C.; Agrawal, D.; Suri, S. Medians and beyond: new aggregation techniques for sensor networks. In Proceedings of the 2nd international conference on Embedded networked sensor systems, pages 239{249. ACM, 2004.