Robust Social Event Detection in Twitter

S. E. Bekçe & F. Amira

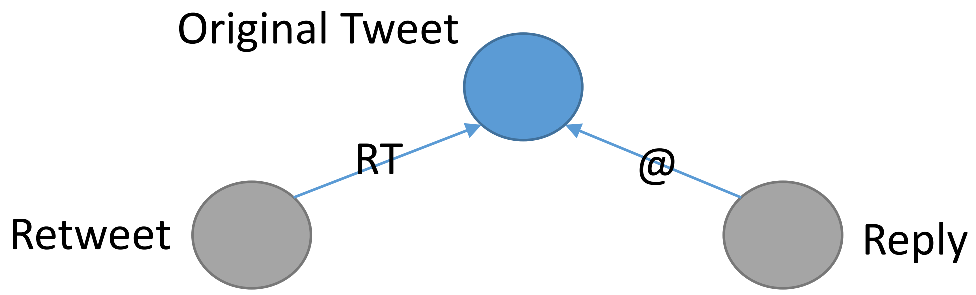
CS533 Spring 2016

Motivation & Goals

- Protests in Turkey are common but they don’t often broadcasted by news agencies because of political pressure
- It is often not possible to understand the intensity and effects of a protest via an external view of point
- Twitter is highly used amongst people to report nearby activity
- We can use those tweets to detect such events
- Our input data contains 100M Turkish-only tweets from 2011 to 2014

Preprocessing: Pre-filtering

1. Discard ‘retweet’s
   Retweets contain no original information
2. Discard very short tweets
   Short tweets usually offer no information at all
3. Discard replies to other tweets
   Replies often introduce information redundancy and offer no original information



Preprocessing: Standardization

1. Replace links with ‘URL’
   Keep tweets with links because they may be photos or other important things, such as location or emergency link, etc
2. Convert tweet content to lowercase
3. Trim tweet content (remove excess whitespace)
4. Tokenize tweets via Zemberek [1]
   Important for finding stems for our classifier: ‘protestolar’ -> ‘protesto’

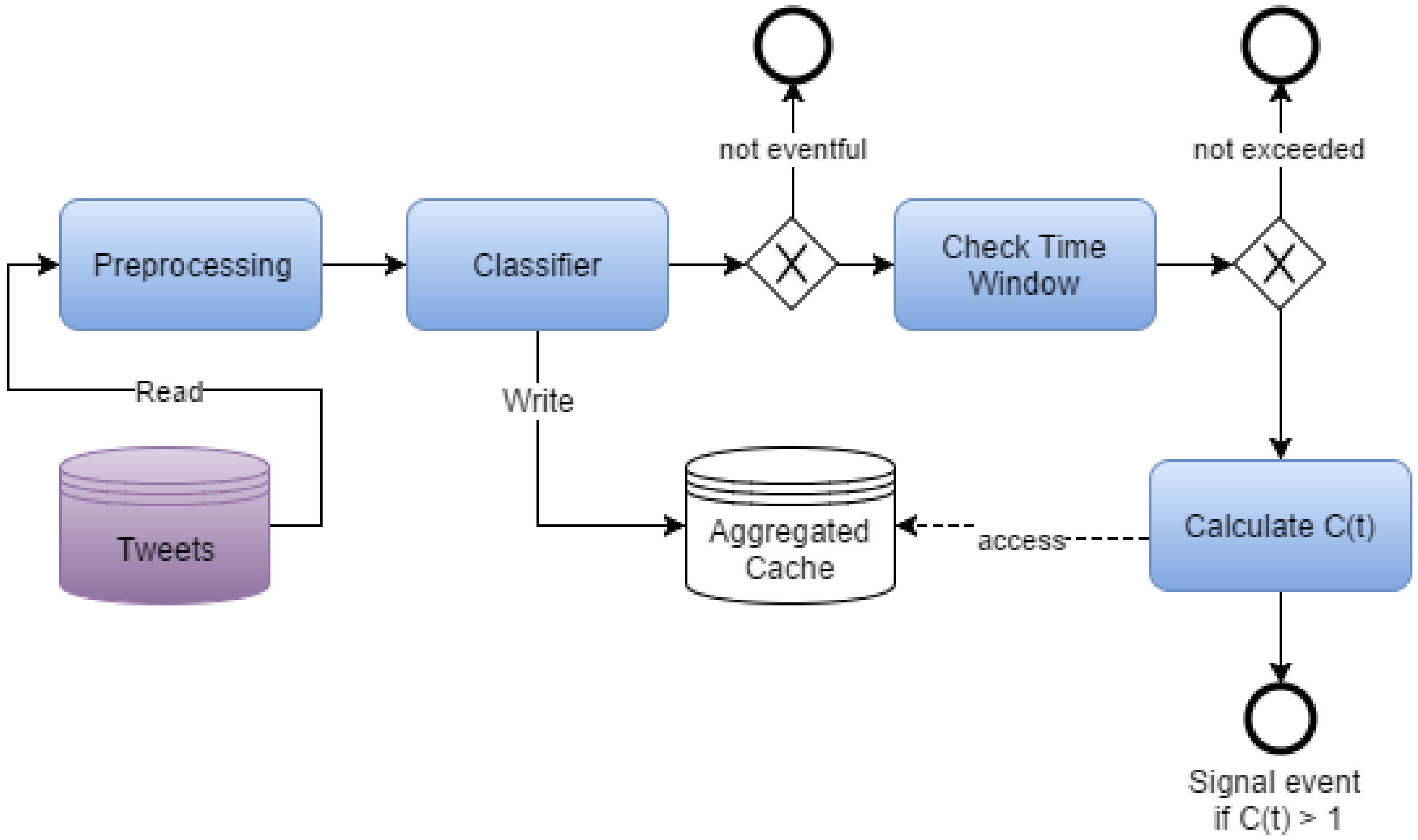
Classifier

- Design a simple classifier to classify each tweet as ‘eventful’ or ‘not eventful’
- Based on possible keywords: (protesto, eylem, toma, saldırı, barikat, direniş, olay)
- Also search for present tense ‘-yor’ in the tweet
  “Ankara Kızılay'da madenci heykeli önünde Soma'daki iş cinayeti protesto ediliyor.”
- Prefer tweets with embedded photos
  - First class (eventful) contains the Tweets which mention the keywords
  - Second class (not eventful) contains irrelevant Tweets

Detection – Characteristic Func.

- Generate time series (histogram) data by aggregating tweets by 15 second intervals and counting them
- Apply Characteristic Function
  - *C(t) = STA/(mLTA+b)* [2]
  - Short Term Average (STA): 1 minute -> possible event
  - Long Term Average (LTA): 1 hour -> background noise
  - *m* and *b* are parameters
  - Declare event when *C(t)* > 1
  - C(t) requires higher signal levels (STA) to trigger at higher noise levels (LTA)

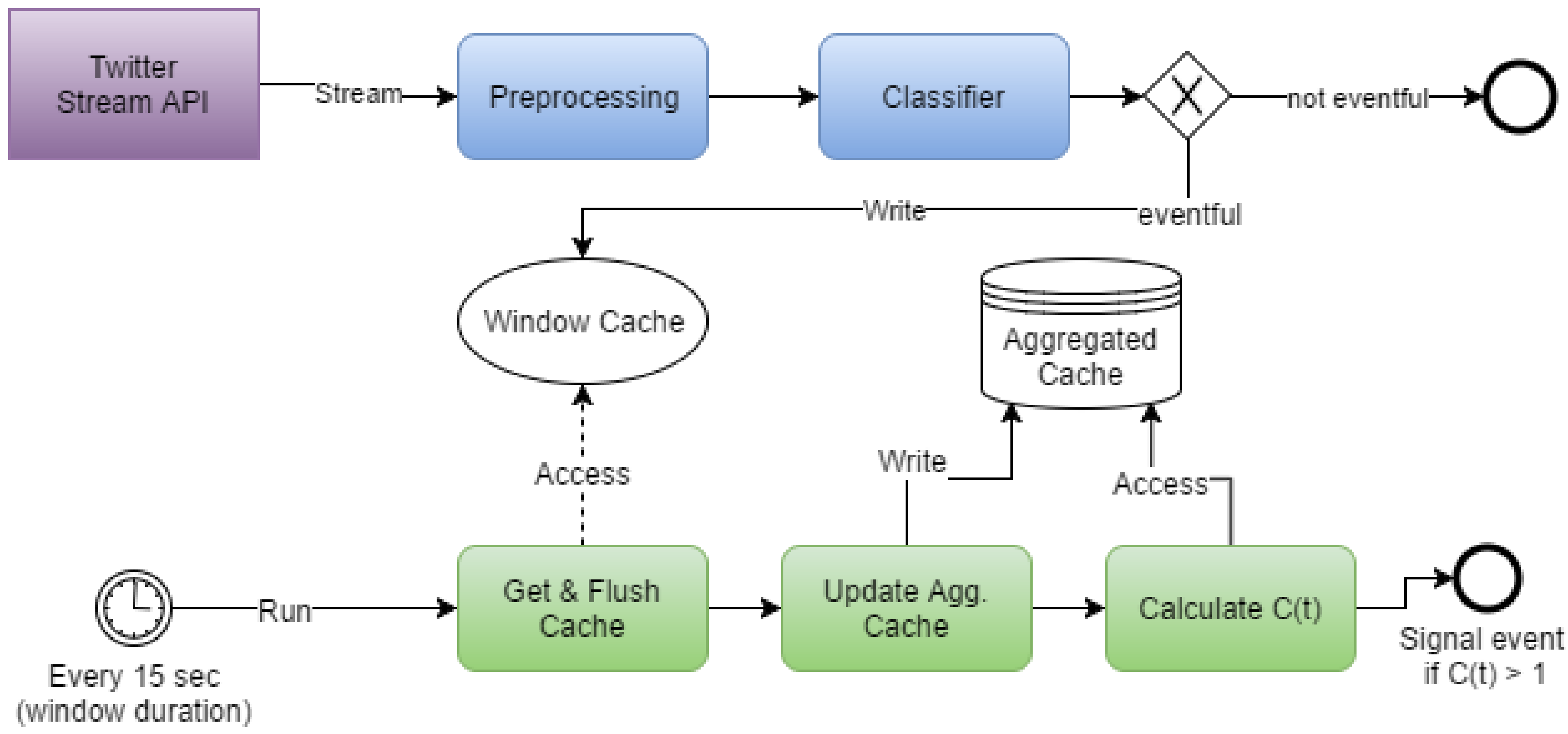
The Algorithm



Online Algorithm

- Instead of working on existing tweets to detect existing events, work on ‘unseen data’ to detect ‘new events’ as they occur on the fly
- Connect to Twitter Streaming API and work on each new tweet sequentially as they arrive
- STA / LTA semantics works nicely with streaming data: Only store the last LTA window (1 hour) amount of histogram (count per window)
- It is important that standardization (tokenization, etc) operations are optimized so that the system can sustain high amount of live tweets

Online Algorithm



Location Estimation

- Use tweet’s geo-tag if possible
- Most tweets have no location information
- In this case, infer a location from
  1. User’s profile information
  2. Moving average of last 10-20 geo-tag tweets from same user
- Events will be mapped to area(s) by using tweets that are mapped to this event

Evaluation

- Precision is calculated by checking the correctness of the detected events
    - P = TP / TP + FP
    - Can be done by looking at news archives
- Calculating Recall is a challenge: Need a structured and reliable way to get list of events
    - Can be done by manually crawling some (reliable) news sites or by human annotators
- Tune parameters (*m* and *b*) by using feedback from the evaluation

References

[1] Zemberek NLP
https://github.com/ahmetaa/zemberek-nlp

[2] P. Earle, D. Bowden and M. Guy, "Twitter earthquake detection: Earthquake monitoring in a social world", Annals of Geophysics, vol. 54, no. 6, pp. 708-715, 2011