

Big Data with ADAMS

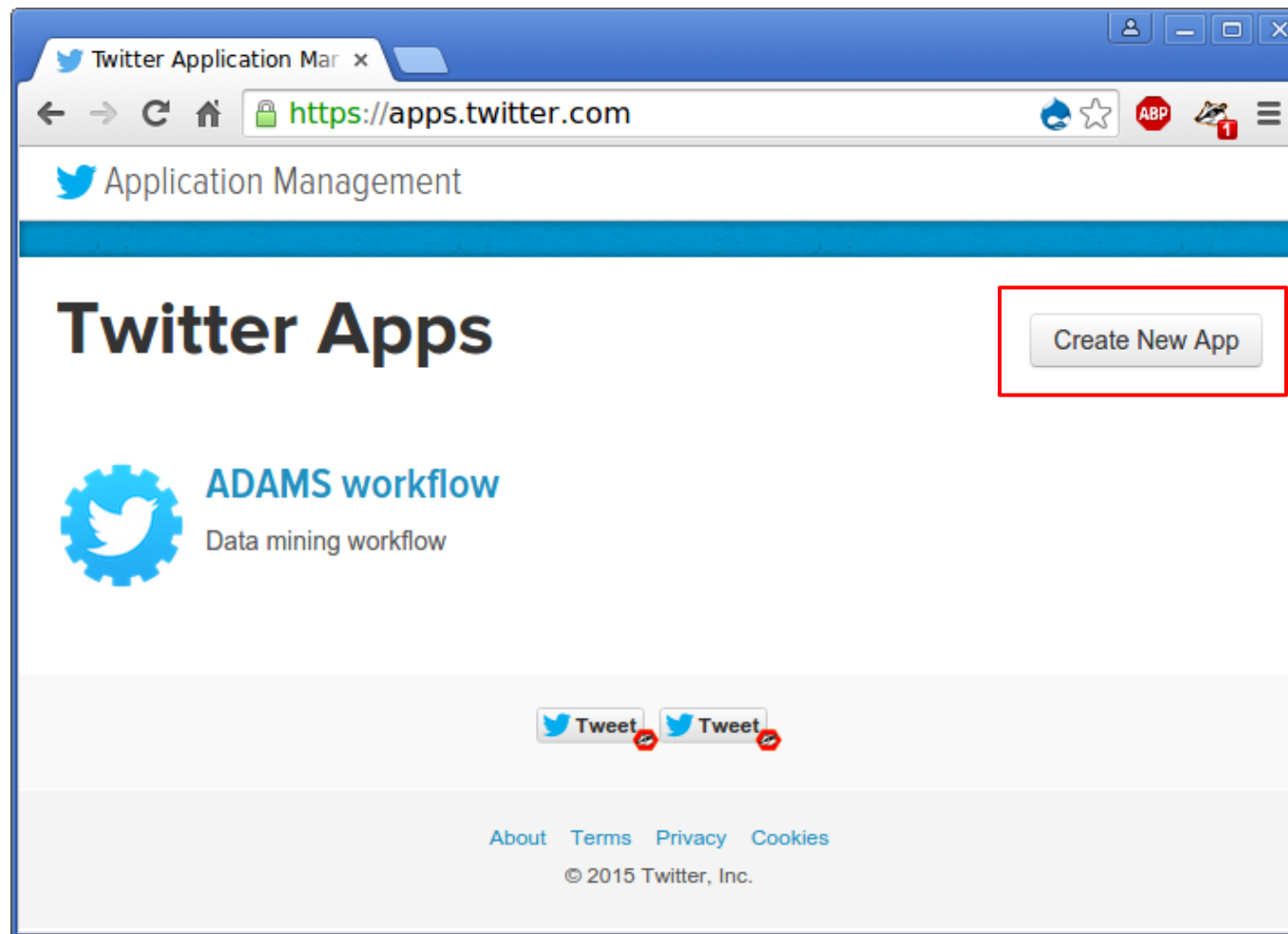
Tweet, tweet, tweet

Collecting tweets

- Twitter allows searches of public tweets
- Twitter offers access to tweets in real time
- ADAMS uses twitter4j.org to access Twitter
- Requires setting up an App
<https://apps.twitter.com>
- Don't worry, it's not that hard...

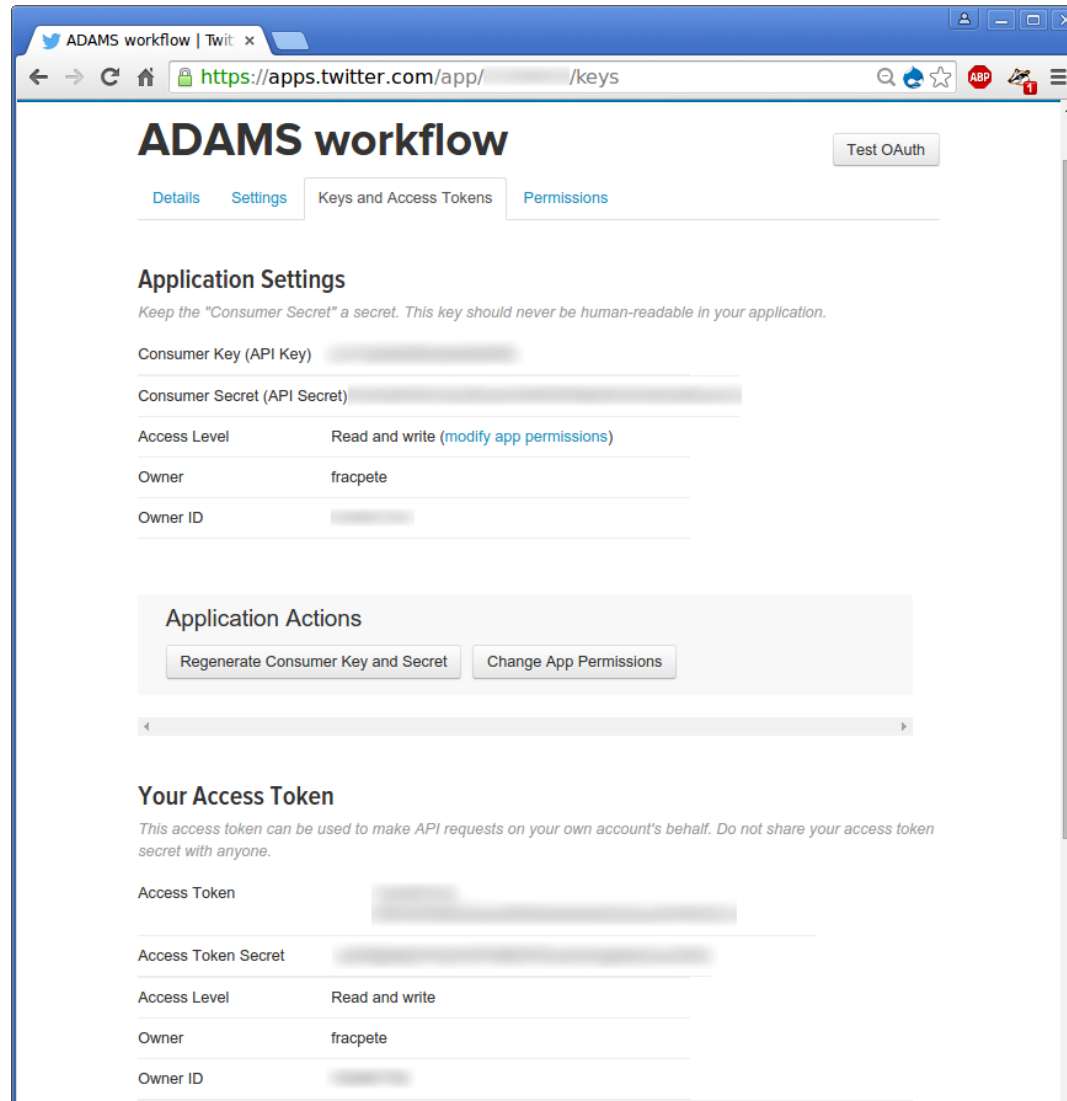
Settings things up

- Create an app



Settings things up (2)

- Set up tokens (consumer and access)



The screenshot shows the 'Keys and Access Tokens' page for an application named 'ADAMS workflow'. The page is divided into two main sections: 'Application Settings' and 'Your Access Token'.

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key) [Redacted]

Consumer Secret (API Secret) [Redacted]

Access Level: Read and write ([modify app permissions](#))

Owner: fracpete

Owner ID: [Redacted]

Application Actions

[Regenerate Consumer Key and Secret](#) [Change App Permissions](#)

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token [Redacted]

Access Token Secret [Redacted]

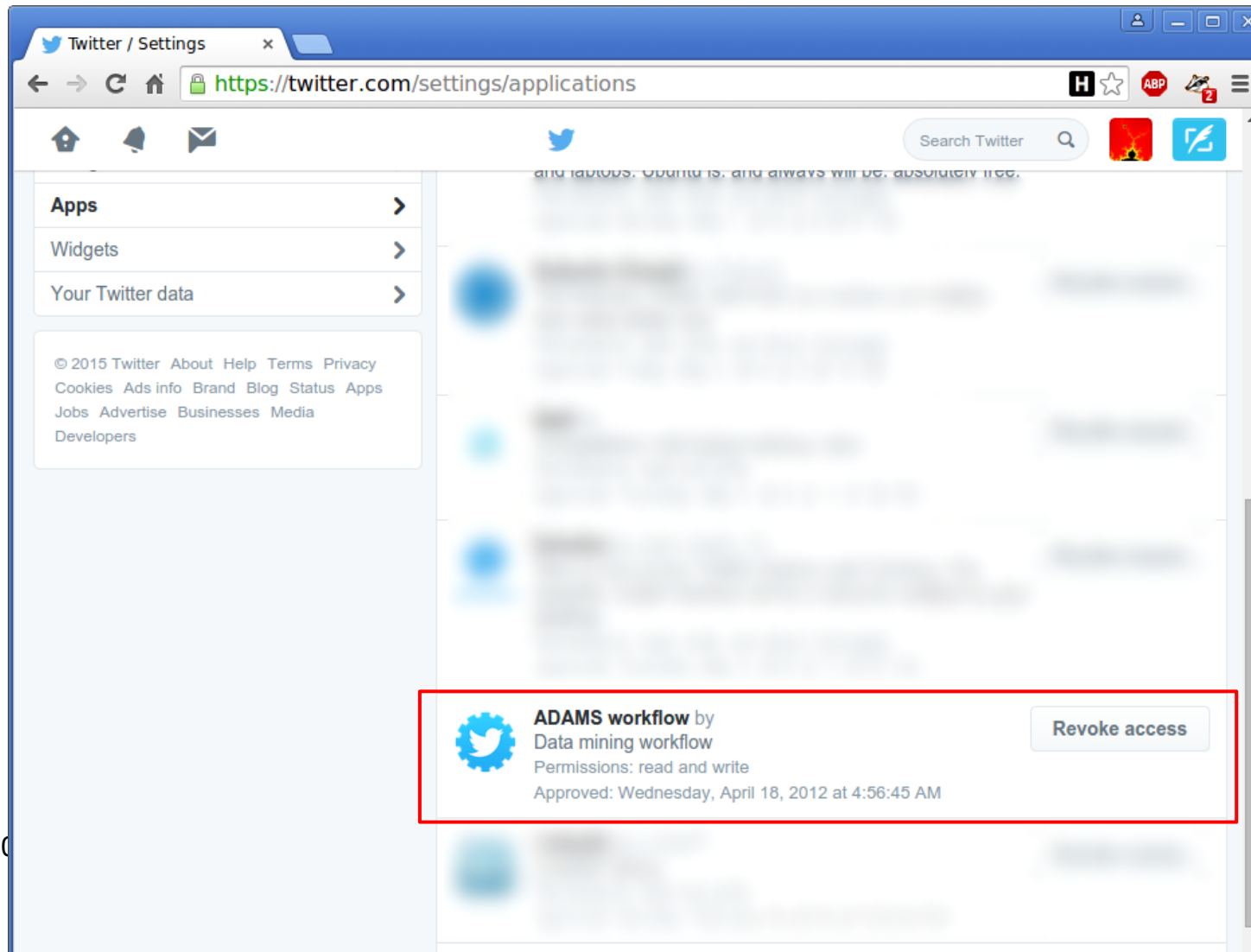
Access Level: Read and write

Owner: fracpete

Owner ID: [Redacted]

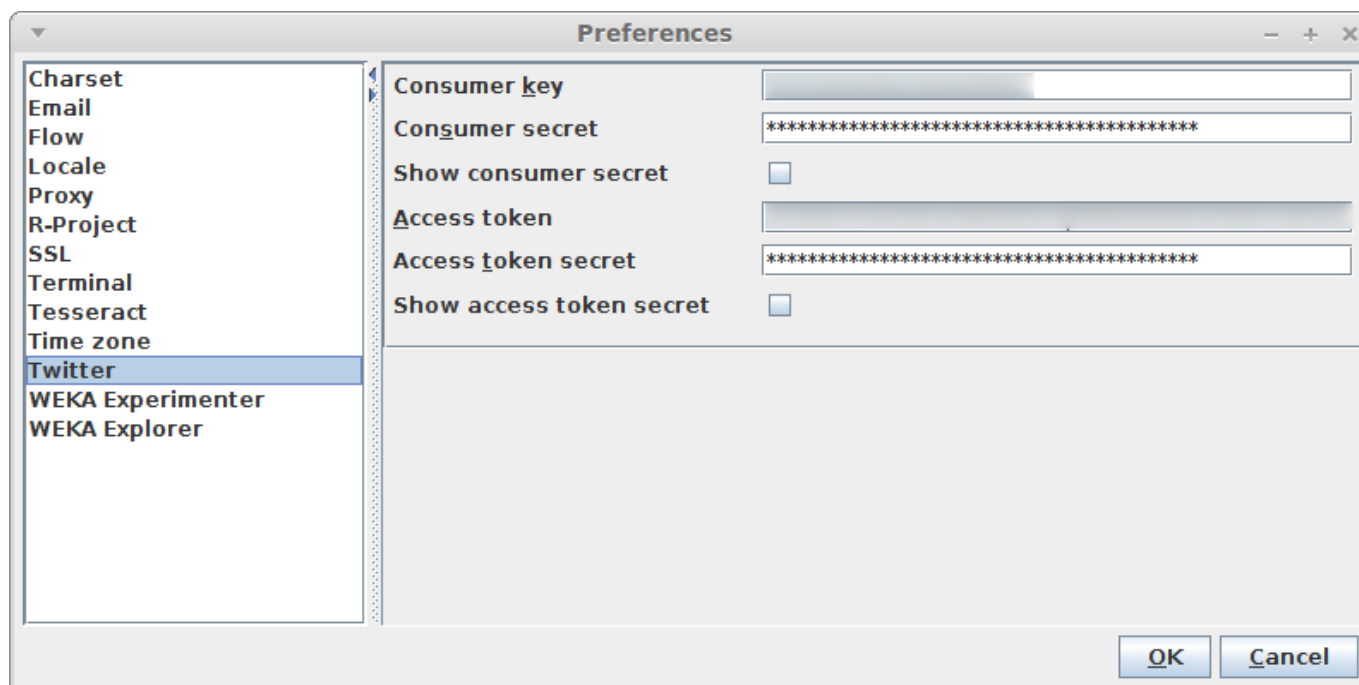
Settings things up (3)

- App should show up in your profile settings





Settings things up (4)

- Finally, fill in Twitter preferences in ADAMS



User queries

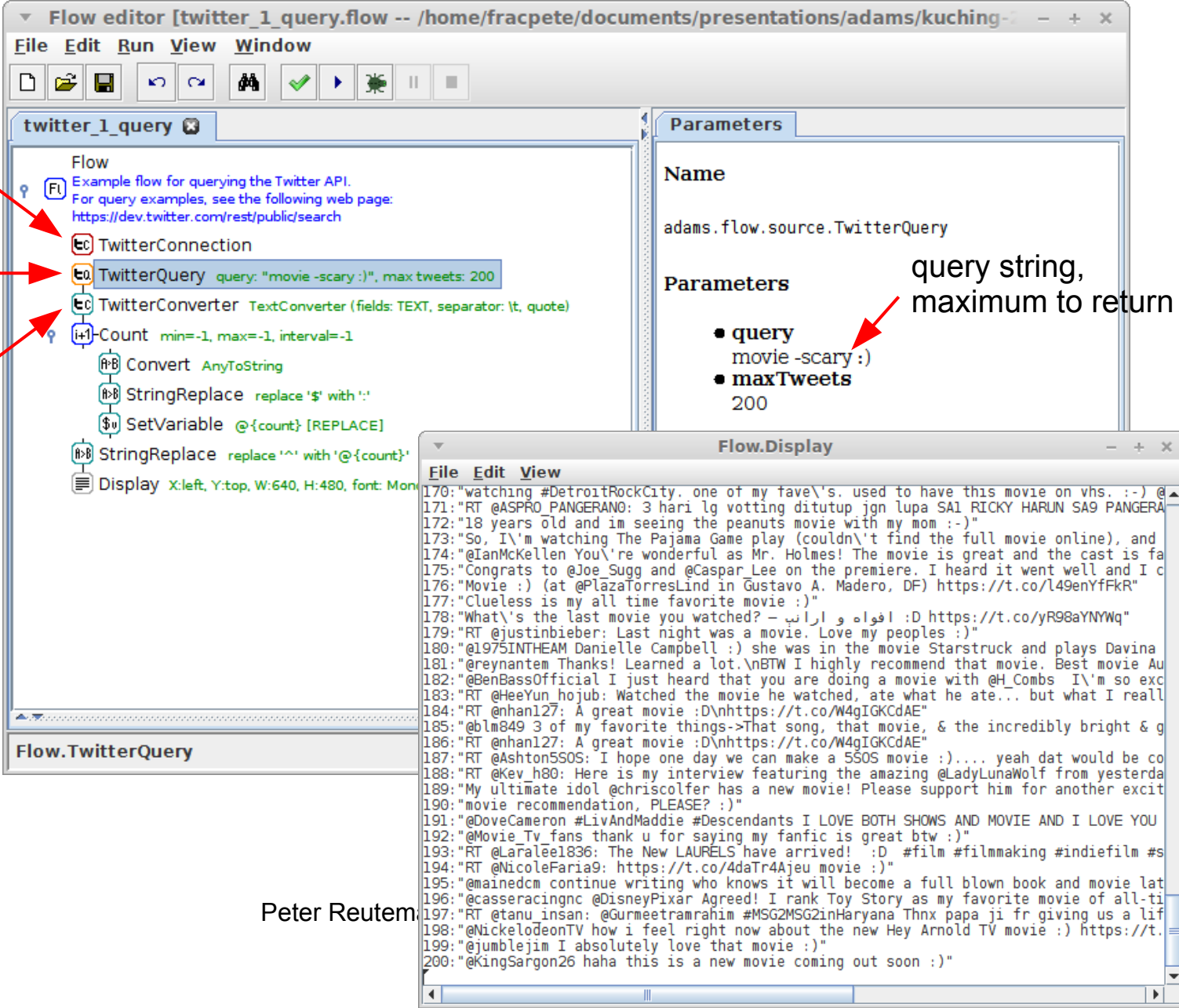
- You can query twitter using something like:
movie -scary :)
[tweets containing “movie” and “:)” but not “scary”]
- Actors to use
 -  TwitterQuery
 -  TwitterConverter
- Use case?
 - when looking for specific keywords

User queries

connection settings;
can override global
preference settings

send query

convert tweet
into textual format



The screenshot shows a flow editor window titled "Flow editor [twitter_1_query.flow -- /home/fracpete/documents/presentations/adams/kuching-]". The flow is named "twitter_1_query" and contains the following steps:

- Flow: Example flow for querying the Twitter API. For query examples, see the following web page: <https://dev.twitter.com/rest/public/search>
- TwitterConnection
- TwitterQuery query: "movie -scary :)", max tweets: 200
- TwitterConverter TextConverter (fields: TEXT, separator: {t, quote})
- Count min=-1, max=-1, interval=-1
- Convert AnyToString
- StringReplace replace '\$' with ':'
- SetVariable @-{count} [REPLACE]
- StringReplace replace '^' with '@-{count}'
- Display X:left, Y:top, W:640, H:480, font: Monospace


The Parameters panel on the right shows the following settings:

- Name: adams.flow.source.TwitterQuery
- Parameters:
 - query: movie -scary :)
 - maxTweets: 200

The Flow.Display window shows the output of the flow, which is a list of tweets. The first few tweets are:

```
170: "watching #DetroitRockCity. one of my fave\'s. used to have this movie on vhs. :-) @
171: "RT @ASPRO PANGERANO: 3 hari lg votting ditutup jgn lupa SA1 RICKY HARUN SA9 PANGERA
172: "18 years old and im seeing the peanuts movie with my mom :-)"
173: "So, I\'m watching The Pajama Game play (couldn\'t find the full movie online), and
174: "@IanMcKellen You\'re wonderful as Mr. Holmes! The movie is great and the cast is fa
175: "Congrats to @Joe_Sugg and @Caspar Lee on the premiere. I heard it went well and I c
176: "Movie :) (at @PlazaTorresLind in Gustavo A. Madero, DF) https://t.co/l49enYfFkR"
177: "Clueless is my all time favorite movie :)"
178: "What\'s the last movie you watched? - افواه و اراب :D https://t.co/yR98aYNYWq"
179: "RT @justinbieber: Last night was a movie. Love my peoples :)"
180: "@1975INTHEAM Danielle Campbell :) she was in the movie Starstruck and plays Davina
181: "@reynantem Thanks! Learned a lot.\nBTW I highly recommend that movie. Best movie Au
182: "@BenBassOfficial I just heard that you are doing a movie with @HCombs I\'m so exc
183: "RT @HeeYun hojub: Watched the movie he watched, ate what he ate... but what I reall
184: "RT @nhan127: A great movie :D\nhttps://t.co/W4gIGKcDAE"
185: "@blm849 3 of my favorite things->That song, that movie, & the incredibly bright & g
186: "RT @nhan127: A great movie :D\nhttps://t.co/W4gIGKcDAE"
187: "RT @Ashton5SOS: I hope one day we can make a 5SOS movie :)... yeah dat would be co
188: "RT @Kev_h80: Here is my interview featuring the amazing @LadyLunaWolf from yesterda
189: "My ultimate idol @chriscolfer has a new movie! Please support him for another excit
190: "movie recommendation, PLEASE? :)"
191: "@DoveCameron #LivAndMaddie #Descendants I LOVE BOTH SHOWS AND MOVIE AND I LOVE YOU
192: "@Movie_Tv fans thank u for saying my fanfic is great btw :)"
193: "RT @LaFaleel1836: The New LAURELS have arrived! :D #film #filmmaking #indiefilm #s
194: "RT @NicoleFaria9: https://t.co/4daTr4Ajeu movie :)"
195: "@mainedcm continue writing who knows it will become a full blown book and movie lat
196: "@casseracingnc @DisneyPixar Agreed! I rank Toy Story as my favorite movie of all-ti
197: "RT @tanu_insan: @Gurmeetramrahim #MSG2MSG2inHaryana Thnx papa ji fr giving us a lif
198: "@NickelodeonTV how i feel right now about the new Hey Arnold TV movie :) https://t.
199: "@jumblejim I absolutely love that movie :)"
200: "@KingSargon26 haha this is a new movie coming out soon :)"
```


Listening

- Rather than posting queries, listen to tweets in real time
- Public access to 1% sample of tweets
 - “garden hose” vs “fire hose”
- Sample bias?
 - <http://arxiv.org/abs/1212.1684>
- Use case?
 - create tweet archives for repeatable experiments
 - capture “the moment”
- Actor
 -  TwitterListener

convert tweets into
spreadsheet format
and save to CSV
“archive tweets”



Continuous archiving

- “Fleeting” nature of tweets makes it hard to repeat experiments
- Archival and replay of tweets solves this
- Next flow shows
 - continuous archival
 - compressed CSV spreadsheets
 - one archive per day

Continuous archival

create filename based
on current day

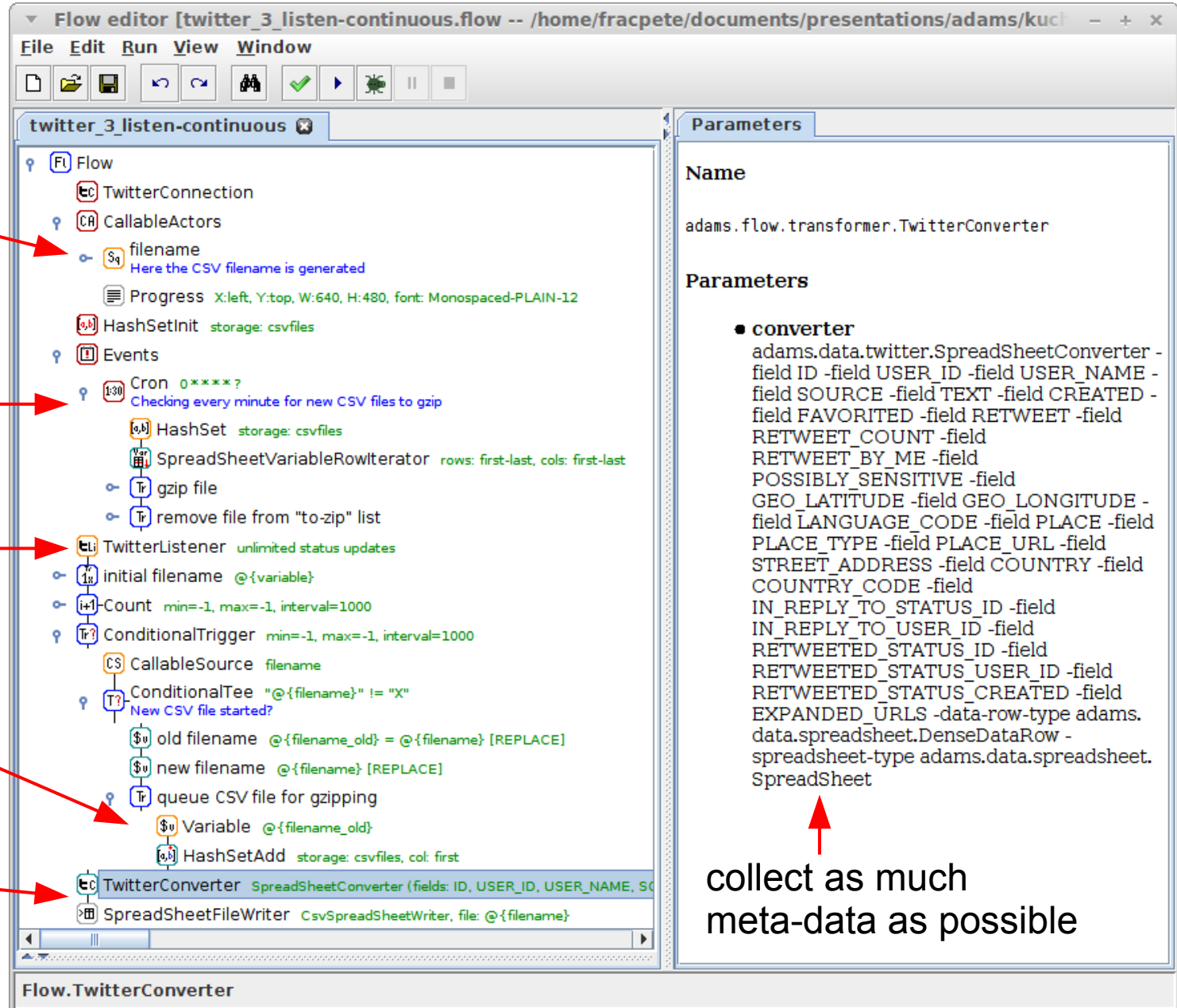
check "queue"
whether new
spreadsheet to
compress

listen till flow
gets stopped

"queue" new
spreadsheet
to be gzip'ed

create spreadsheet
row and append file

26 & 27 Nov 2015



Flow editor [twitter_3_listen-continuous.flow -- /home/fracpete/documents/presentations/adams/kuc] - + x

File Edit Run View Window

twitter_3_listen-continuous

Flow

- TwitterConnection
- CallableActors
- filename
Here the CSV filename is generated
- Progress X:left, Y:top, W:640, H:480, font: Monospaced-PLAIN-12
- HashSetInit storage: csvfiles
- Events
- Cron 0* * * * ?
Checking every minute for new CSV files to gzip
- HashSet storage: csvfiles
- SpreadSheetVariableRowIterator rows: first-last, cols: first-last
- gzip file
- remove file from "to-zip" list
- TwitterListener unlimited status updates
- initial filename @{variable}
- Count min=-1, max=-1, interval=1000
- ConditionalTrigger min=-1, max=-1, interval=1000
- CallableSource filename
- ConditionalTee "@{filename}" != "X"
New CSV file started?
- old filename @{filename_old} = @{filename} [REPLACE]
- new filename @{filename} [REPLACE]
- queue CSV file for gzipping
- Variable @{filename_old}
- HashSetAdd storage: csvfiles, col: first
- TwitterConverter SpreadsheetConverter (fields: ID, USER_ID, USER_NAME, SOURCE, TEXT, CREATED, FAVORITED, RETWEET, RETWEET_COUNT, RETWEET_BY_ME, POSSIBLY_SENSITIVE, GEO_LATITUDE, GEO_LONGITUDE, LANGUAGE_CODE, PLACE, PLACE_TYPE, PLACE_URL, STREET_ADDRESS, COUNTRY, COUNTRY_CODE, IN_REPLY_TO_STATUS_ID, IN_REPLY_TO_USER_ID, RETWEETED_STATUS_ID, RETWEETED_STATUS_USER_ID, RETWEETED_STATUS_CREATED, EXPANDED_URLS -data-row-type adams.data.spreadsheet.DenseDataRow -spreadsheet-type adams.data.spreadsheet.SpreadSheet)
- SpreadSheetFileWriter CsvSpreadSheetWriter, file: @{filename}

Parameters

Name

adams.flow.transformer.TwitterConverter




Parameters

- converter**
adams.data.twitter.SpreadSheetConverter -
field ID -field USER_ID -field USER_NAME -
field SOURCE -field TEXT -field CREATED -
field FAVORITED -field RETWEET -field
RETWEET_COUNT -field
RETWEET_BY_ME -field
POSSIBLY_SENSITIVE -field
GEO_LATITUDE -field GEO_LONGITUDE -
field LANGUAGE_CODE -field PLACE -field
PLACE_TYPE -field PLACE_URL -field
STREET_ADDRESS -field COUNTRY -field
COUNTRY_CODE -field
IN_REPLY_TO_STATUS_ID -field
IN_REPLY_TO_USER_ID -field
RETWEETED_STATUS_ID -field
RETWEETED_STATUS_USER_ID -field
RETWEETED_STATUS_CREATED -field
EXPANDED_URLS -data-row-type adams.
data.spreadsheet.DenseDataRow -
spreadsheet-type adams.data.spreadsheet.
SpreadSheet

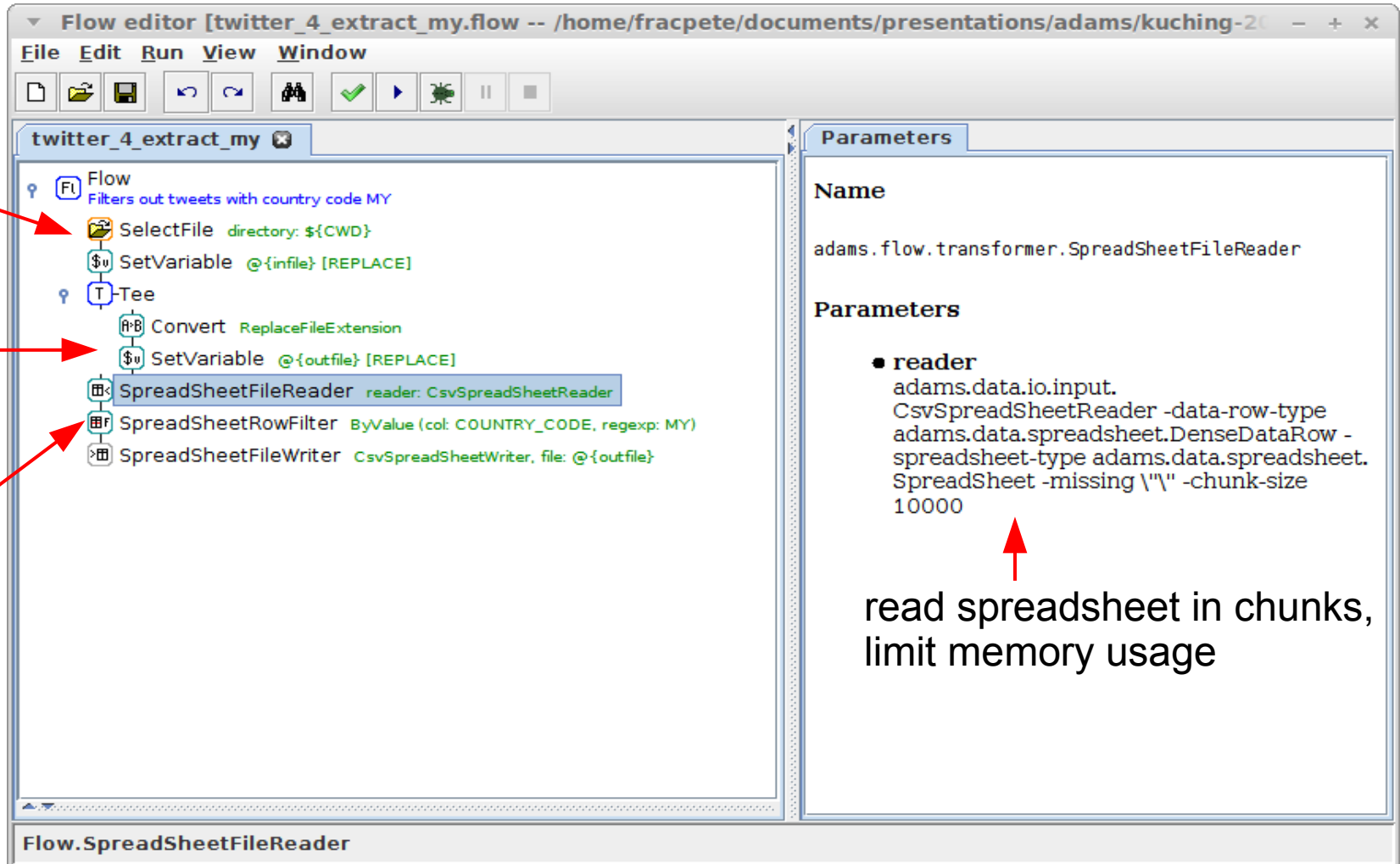
Flow.TwitterConverter

collect as much
meta-data as possible

Filtering archives

- At time of writing ~3 million tweets per day
- Experiments can operate on subsets to speed things up considerably
 - pre-filter spreadsheet archives
- Actors to use
 -  SpreadSheetFileReader
 -  SpreadSheetRowFilter
 -  SpreadSheetFileWriter

Filtering by country



Flow editor [twitter_4_extract_my.flow -- /home/fracpete/documents/presentations/adams/kuching-20 -- + x]

File Edit Run View Window

twitter_4_extract_my

Flow
Filters out tweets with country code MY

SelectFile directory: \${CWD}

SetVariable @-{infile} [REPLACE]

Tee

Convert ReplaceFileExtension

SetVariable @-{outfile} [REPLACE]

SpreadsheetFileReader reader: CsvSpreadSheetReader

SpreadsheetRowFilter ByValue (col: COUNTRY_CODE, regexp: MY)

SpreadsheetFileWriter CsvSpreadSheetWriter, file: @-{outfile}

Parameters

Name
adams.flow.transformer.SpreadSheetFileReader

Parameters

- reader
adams.data.io.input.
CsvSpreadSheetReader -data-row-type
adams.data.spreadsheet.DenseDataRow -
spreadsheet-type adams.data.spreadsheet.
SpreadSheet -missing \"\" -chunk-size
10000

Flow.SpreadSheetFileReader

select archive files to filter

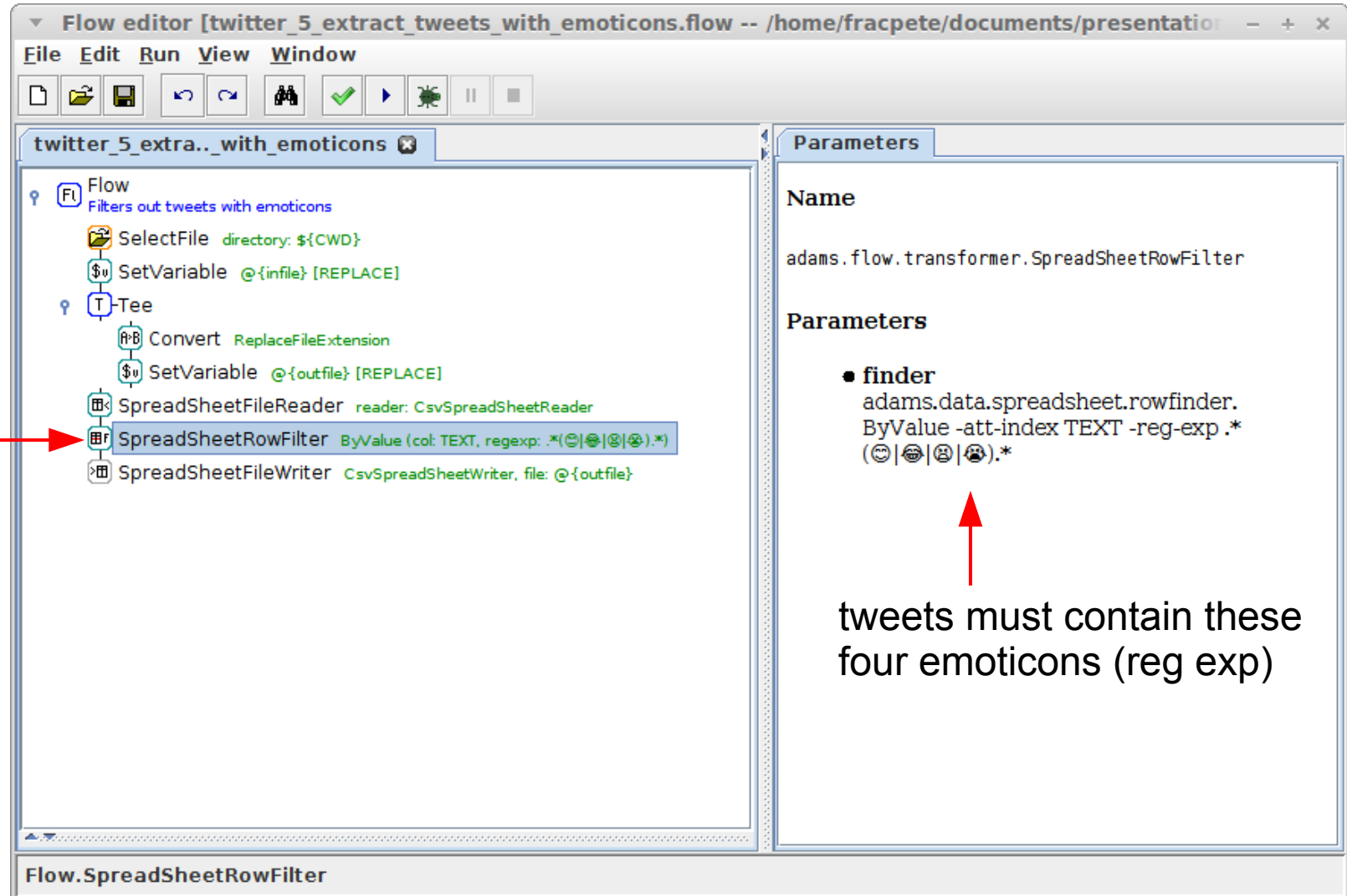
create output filename based on input file

leave only rows with value "MY" in column "COUNTRY"

read spreadsheet in chunks, limit memory usage

Filtering by content

filter column
"TEXT" using
a regular
expression



Flow editor [twitter_5_extract_tweets_with_emoticons.flow -- /home/fracpete/documents/presentation] - + x

File Edit Run View Window

twitter_5_extra.._with_emoticons

Flow
Filters out tweets with emoticons

- SelectFile directory: \${CWD}
- SetVariable @-{infile} [REPLACE]
- Tee
 - Convert ReplaceFileExtension
 - SetVariable @-{outfile} [REPLACE]
 - SpreadsheetFileReader reader: CsvSpreadSheetReader
 - SpreadsheetRowFilter ByValue (col: TEXT, regexp: .* (☺|☹|😄|😡).*)**
 - SpreadsheetFileWriter CsvSpreadSheetWriter, file: @-{outfile}

Parameters

Name
adams.flow.transformer.SpreadSheetRowFilter


Parameters

- finder
adams.data.spreadsheet.rowfinder.
ByValue -att-index TEXT -reg-exp.*
(☺|☹|😄|😡).*

Flow.SpreadSheetRowFilter

tweets must contain these
four emoticons (reg exp)

Replaying archives

- Replaying archives is excellent for repeatable experiments
- Use cases
 - further filtering
 - feature extraction
 - plotting
- Actor
 -  TweetReplay

New Years “Happiness”

plots happy vs sad

select archives to replay

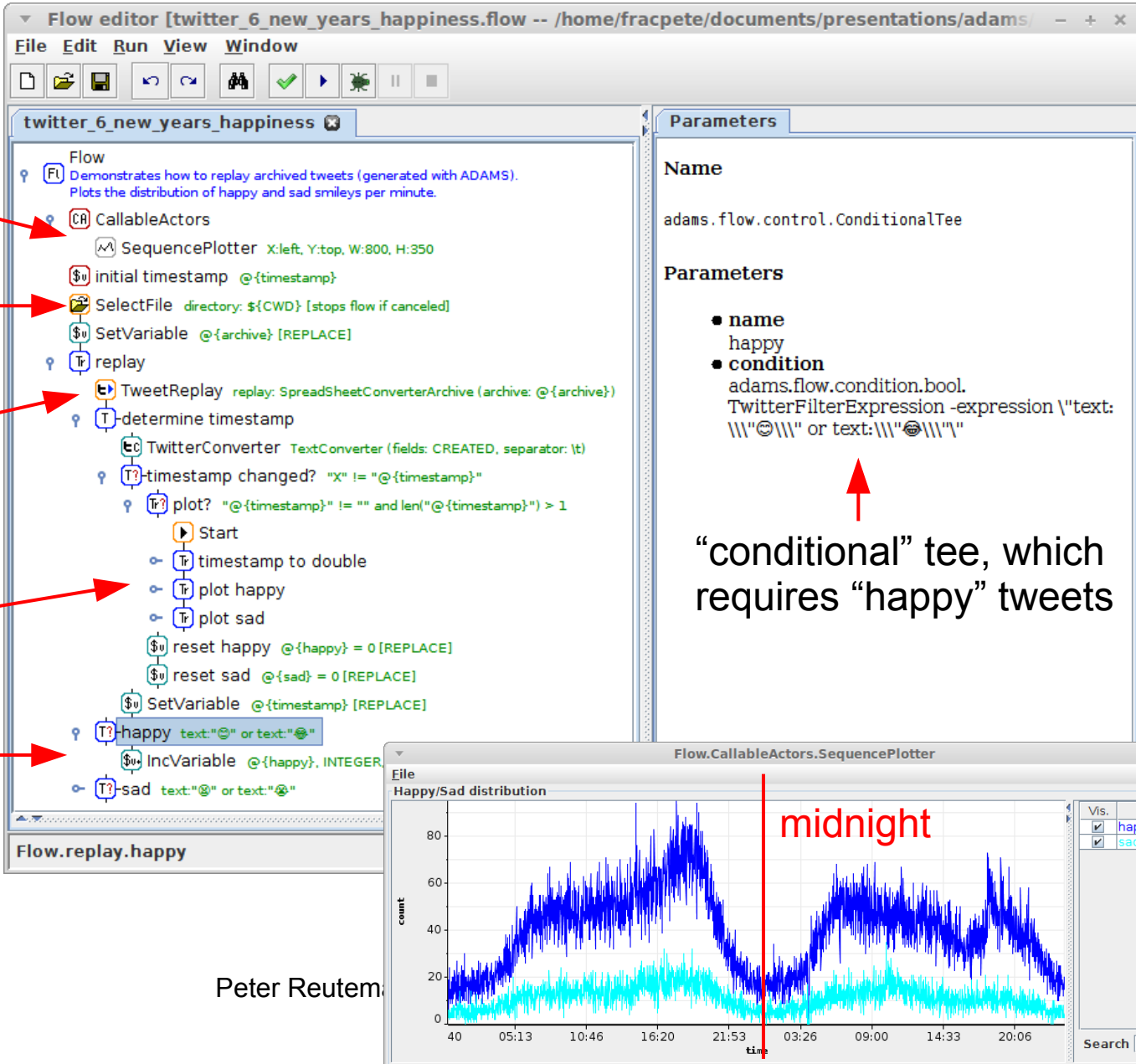
replay current archive, creating “fake” tweets from archived data

create plot containers each minute

count “happy” and “sad” tweets

“conditional” tee, which requires “happy” tweets

midnight



The screenshot shows a Flow editor window titled "Flow editor [twitter_6_new_years_happiness.flow -- /home/fracpete/documents/presentations/adams]". The flow is named "twitter_6_new_years_happiness" and contains the following steps:


- Flow: Demonstrates how to replay archived tweets (generated with ADAMS). Plots the distribution of happy and sad smileys per minute.
- CallableActors: SequencePlotter (X:left, Y:top, W:800, H:350)
- initial timestamp: @timestamp
- SelectFile: directory: \${CWD} (stops flow if canceled)
- SetVariable: @archive [REPLACE]
- replay: TweetReplay (replay: SpreadsheetConverterArchive (archive: @archive))
- determine timestamp: TwitterConverter (TextConverter (fields: CREATED, separator: \t))
- timestamp changed? "X" != "@timestamp"
- plot? "@timestamp" != "" and len("@timestamp") > 1
- Start
- timestamp to double
- plot happy
- plot sad
- reset happy @happy = 0 [REPLACE]
- reset sad @sad = 0 [REPLACE]
- SetVariable @timestamp [REPLACE]
- happy text:"😊" or text:"😄"
- IncVariable @happy, INTEGER
- sad text:"😞" or text:"😡"

The Parameters panel on the right shows the following details for the "adams.flow.control.ConditionalTee" actor:

- Name: adams.flow.control.ConditionalTee
- Parameters:
 - name: happy
 - condition: adams.flow.condition.bool. TwitterFilterExpression -expression "\"text: \\\"😊\\\" or text: \\\"😄\\\""

The bottom right window shows a plot titled "Flow.CallableActors.SequencePlotter" with the subtitle "Happy/Sad distribution". The plot shows the count of happy (blue) and sad (cyan) tweets over time. A vertical red line marks "midnight" at 21:53. The x-axis is labeled "time" and the y-axis is labeled "count".

Create dataset

- Build dataset from tweets as basis for predictive model
- Use happy vs sad as the “state” of the tweet (= class attribute)
- Add class attribute column using
 -  SpreadSheetInsertColumn

Create dataset

Flow editor [twitter_7_happiness_csv.flow -- /home/fracpete/documents/presentations/adams/kuching

File Edit Run View Window

twitter_7_happiness_csv

Flow

Fl Demonstrates how to replay archived tweets (generated with ADAMS).
Creates CSV file from archives, labeling happy and sad tweets.

Start

select output file

SelectFile directory: \${CWD} [stops flow if canceled]

SetVariable @outfile [REPLACE]

process archives

SelectFile directory: \${CWD} [stops flow if canceled]

SetVariable @archive [REPLACE]

replay

TweetReplay replay: SpreadsheetConverterArchive (archive: @archive)

Continue !text:"@|" or text:"@|" or text:"@|" or text:"@|" allow only happy/sad tweets

happy or sad?

IfThenElse text:"@|" or text:"@|"

then @state = happy [REPLACE]

else @state = sad [REPLACE]

TwitterConverter SpreadsheetConverter (fields: ID, CREATED, TEXT, GEO_LATITUDE, GEO_LONGITUDE, LANGUAGE_CODE, COUNTRY_CODE, State)

SpreadSheetInsertColumn header: 'State', after: last, insert: @state

SpreadSheetFileWriter CsvSpreadSheetWriter file: @outfile

Parameters

Name

adams.flow.control.Continue

Parameters

- annotations
allow only happy/sad tweets
- condition
adams.flow.condition.bool.Not -condition \"
adams.flow.condition.bool.
TwitterFilterExpression -expression \"\"
text: \"\" or text: \"\" or text: \"\" or text: \"\"
text: \"\" or text: \"\" or text: \"\" or text: \"\"
\"

boolean tweet
filter expression

only let happy/sad
tweets pass through

determine "state" of
tweet set in variable

insert column
with "state"

26 & 27 Nov 2015

Pet

er [blah.csv -- /home/fracpete/temp]

Associate Select attributes Visualize Forecast

Preprocess Classify Cluster

Filter

Choose AllFilter Apply

Current relation

Relation: Attributes: 8
Instanc... Sum of weights: 139047

Attributes

All ... I... ..

No. Name

1 ID

2 CREATED

3 TEXT

4 GEO_LATITUDE

5 GEO_LONGITUDE

6 LANGUAGE_CODE

7 COUNTRY_CODE

8 State

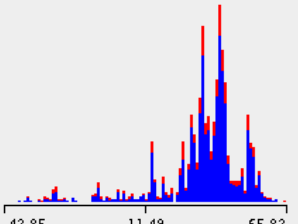
Remove

Selected attribute

Name: GEO_LATITUDE... Type: ..
Missi... 1... Distinct: Unique: ..

Statistic	Value
Minimum	-42.845
Maximum	65.816
Mean	33.67
StdDev	16.135

Class: State (N... Visualize All



Create features

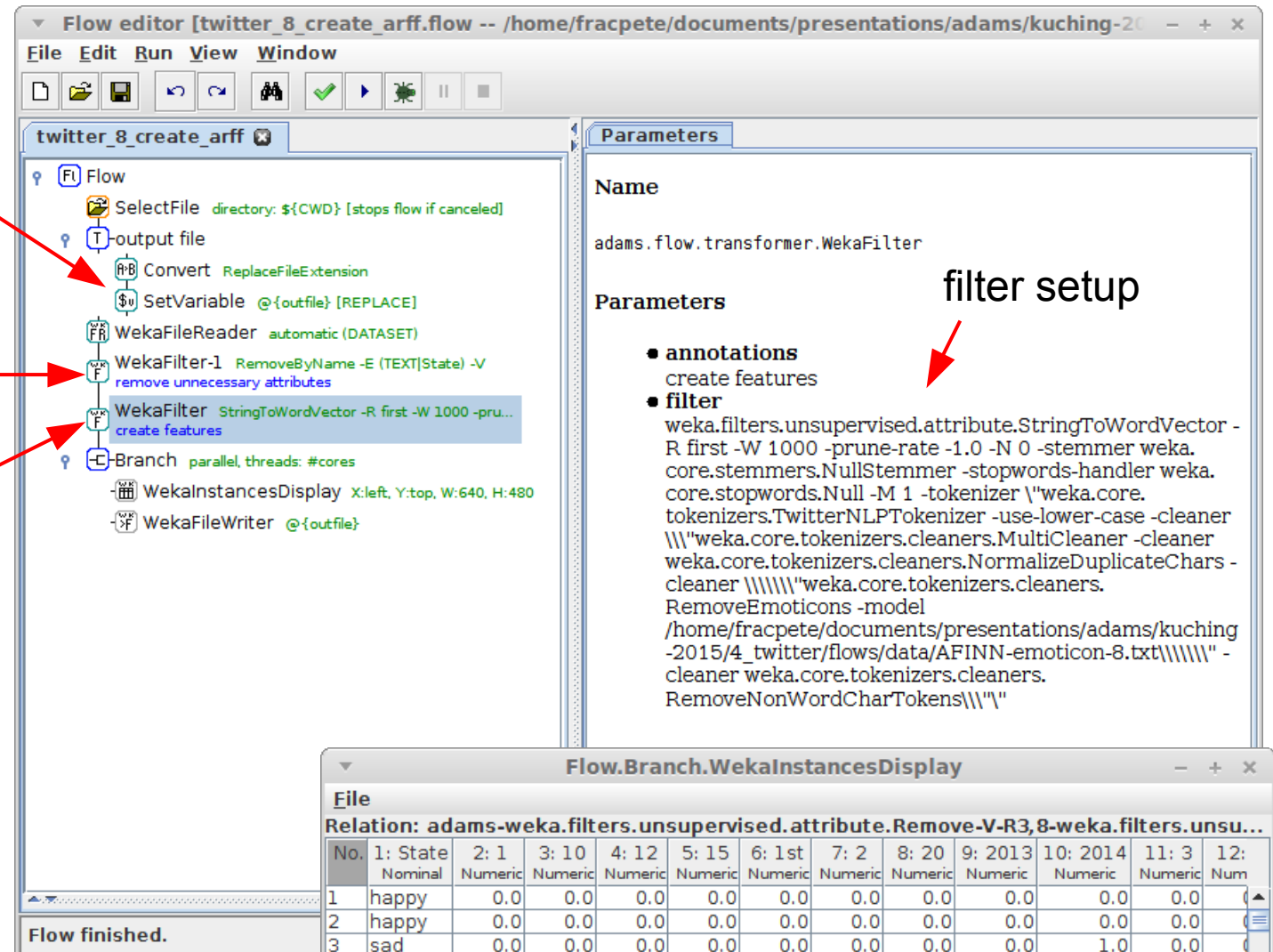
- Using WEKA's StringToWordVector to create features from tweet text
- Apply token cleaners to remove unwanted content
 - users
 - URLs
 - hashtag
 - emoticons

Create features

create output
filename

remove some
attribute

generate features
from tweet text



Flow editor [twitter_8_create_arff.flow -- /home/fracpete/documents/presentations/adams/kuching-20 -- + x]

File Edit Run View Window

twitter_8_create_arff

Flow

- SelectFile directory: \${CWD} [stops flow if canceled]
- output file
- Convert ReplaceFileExtension
- SetVariable @-{outfile} [REPLACE]
- WekaFileReader automatic (DATASET)
- WekaFilter-1 RemoveByName -E (TEXT)State -V remove unnecessary attributes
- WekaFilter StringToWordVector -R first -W 1000 -pru... create features
- Branch parallel, threads: #cores
- WekaInstancesDisplay X:left, Y:top, W:640, H:480
- WekaFileWriter @-{outfile}

Flow finished.

Parameters

Name

adams.flow.transformer.WekaFilter

Parameters

- annotations
create features
- filter
weka.filters.unsupervised.attribute.StringToWordVector -R first -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "\"weka.core.tokenizers.cleaners.MultiCleaner -cleaner weka.core.tokenizers.cleaners.NormalizeDuplicateChars -cleaner \"\"weka.core.tokenizers.cleaners.RemoveEmoticons -model /home/fracpete/documents/presentations/adams/kuching-2015/4_twitter/flows/data/AFINN-emoticon-8.txt\"\" -cleaner weka.core.tokenizers.cleaners.RemoveNonWordCharTokens\"\""

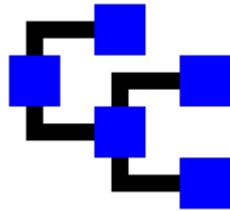
Flow.Branch.WekaInstancesDisplay

File

Relation: adams-weka.filters.unsupervised.attribute.Remove-V-R3,8-weka.filters.unsu...

No.	1: State Nominal	2: 1 Numeric	3: 10 Numeric	4: 12 Numeric	5: 15 Numeric	6: 1st Numeric	7: 2 Numeric	8: 20 Numeric	9: 2013 Numeric	10: 2014 Numeric	11: 3 Numeric	12: Num
1	happy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	happy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	sad	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	
4	happy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	sad	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	sad	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
7	happy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
8	happy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9	happy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
10	happy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
11	happy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
12	happy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Questions?



<https://adams.cms.waikato.ac.nz/>

@TheAdamsFlow