

# Twitter Topic Analysis using Apache Mahout

Eitan Romanoff  
Thad Billing  
Egezon Berisha

## What's this all about?

As it becomes easier to share thoughts and sentiment online through social networking services, the amount of publicly available data grows with it. This information, in the form of natural text, can be acquired, processed, and analyzed to identify features such as trending topics across all users of these services. Companies can leverage this data to market intelligently, but as the amount of data increases, so does the need for scalable systems. In this project, we attempt to create a system to identify changes in the topical landscape over time.

## Acquire.py

JSON

```
{
  "coordinates": null,
  "truncated": false,
  "created_at": "Wed May 01 21:15:32 +0000 2013",
  "favorited": false,
  "entities": {
    "urls": [ ],
    "hashtags": [ "testing" ],
    "user_mentions": [ ]
  },
  "text": "Hello, world! Natural text is hard...",
  "annotations": null,
  "contributors": [ ],
  "id": 15812958191125,
  "id_str": "15812958191125",
  "retweet_count": 0,
  "geo": null,
  "retweeted": false,
  "in_reply_to_user_id": null,
  "in_reply_to_user_id_str": "null",
  "in_reply_to_screen_name": "null",
  "user": {
    "id": 123456,
    "id_str": "123456",
    "source": "Web",
    "place": null,
    "in_reply_to_status_id": null,
    "in_reply_to_status_id_str": "null"
  }
}
```

Local

Data was acquired through usage of the Twitter Stream API, where we get a 1% sample of the overall twitter stream. Tweets come in as JSON, delimited by carriage returns, and we store these as a "dump" file on the local disk.

## Fragment.py

id\_str


created\_at

15812958191125

Fri, 01 May 2013 21:15:32

text

entities['hashtags']



15812958191125

Fri, 01 May 2013 21:15:32

Hello, world! Natural text is hard.. #testing

text

entities['hashtags']

Each tweet is split into three fragment portions: JSON, Content, and Hashtags. The JSON portion is simply for the whole, pure storage of the tweet's metadata for potential future use. The Content fragment is the text of the tweet without any metadata. This portion of the tweet gets cleaned in a preprocessing stage prior to being stored as a file fragment, where the text is normalized, and stop words are removed. The Hashtag fragment contains only the hashtags of the tweet. These are stored on the local disk in a directory hierarchy, but ideally, this would be placed on the HDFS. Unfortunately, HDFS is not designed for many small data files, which remains an unsolved issue.

Local

Each tweet is split into three fragment portions: JSON, Content, and Hashtags. The JSON portion is simply for the whole, pure storage of the tweet's metadata for potential future use. The Content fragment is the text of the tweet without any metadata. This portion of the tweet gets cleaned in a preprocessing stage prior to being stored as a file fragment, where the text is normalized, and stop words are removed. The Hashtag fragment contains only the hashtags of the tweet. These are stored on the local disk in a directory hierarchy, but ideally, this would be placed on the HDFS. Unfortunately, HDFS is not designed for many small data files, which remains an unsolved issue.

Each file was binned based on the timestamp of the tweet. Creating the file hierarchy on the disk in this way is necessary for running the Seq2sparse utility in Mahout. Furthermore, this allows us to create vector "snapshots" between time periods. In the future, we may want to make the binning even more granular for small time steps.

Seqdirectory was utilized to create a sequence file containing the collected tweets for a time period into a large file to be used in processing.

```
[root@ip-10-245-3-81 ~]# /usr/lib/mahout/trunk/bin/mahout seqdir
actory -i /root/twitter_files/fragments/content_fragments/2013/5
/4/23/ -o /root/twitter_files/seqfiles/23/ -c UTF-8
```

Seq2sparse was executed to create TF vectors for utilization in LDA. Each term needed at least a frequency of 50 to be considered a topic for LDA. A dictionary of topics was generated for reference in retrieval of LDA Topics.

```
[root@ip-10-245-3-81 ~]# /usr/lib/mahout/trunk/bin/mahout seq2sp
arse -i /root/twitter_files/vectors23/tf-vectors/part-r-00000 -o /roo
t/twitter_files/vectors/23/dictionary.file-0 -dt sequencefile
```

rowid converts text indexes to numerical values for input to LDA.

```
[root@ip-10-245-3-81 ~]# /usr/lib/mahout/trunk/bin/mahout rowid
-i /root/twitter_files/vectors23/tf-vectors/part-r-00000 -o /roo
t/twitter_files/sparse_vectors_cvb/23/
```

LDA is executed against the data set for 30 topics over 20 iterations.


```
[root@ip-10-245-3-81 ~]# /usr/lib/mahout/trunk/bin/mahout cvb -i
/root/twitter_files/sparse_vectors_cvb/23/matrix -dict /root/tw
itter_files/vectors/23/dictionary.file-0 -o /root/twitter_files/
output/23k30/ -k 30 -x 20 -ntt 400
```

A vectordump output is generated for the resulting LDA topics.

```
[root@ip-10-245-3-81 ~]# /usr/lib/mahout/trunk/bin/mahout v
ectordump -i /root/twitter_files/output/23k30/part-m-00000 -d /r
oot/twitter_files/vectors/23/dictionary.file-0 -dt sequencefile
-vs 10 -sort true
```

## Current

Our hardware architecture leverages Amazon Web Services to host our worker machines. We run our worker machine on top of a newly created AMI running centos 5.4 with both a Hadoop and Mahout installation. Furthermore, we use S3 as a data farm service to store our large quantity of tweets. Currently, our configuration only runs on a single EC2 instance, but should be scalable to multiple EC2 instances, using the machines as a cluster. Future work includes migrating all operations that run on the local disk to run on HDFS. Furthermore, Acquire and Fragment operations should be adapted to use the MapReduce workflow.



Our project used a service called s3fs (S3 FileSystem) which allowed us to mount an S3 bucket and work with it as if it was on the local filesystem. But beware! The s3fs service did not scale well with many small files, and the overhead for each transaction rendered the service unusable at any non-trivial scale!

## Ideal

## Results

In reviewing the results over the two timeslices we can see overlap in the hashtags and topics generated over an hour apart. It also served to highlight the difficulties in foreign language detection and trivial content contained in tweets.

Hashtag	Frequency
rt	125
teamfollowback	95
gameinsight	90
lifewouldbeatbetterif	85
android	80
fbj	75
openfollow	70
androidgames	65
unboxtherapygives	60
#	55
nitasasa	50

Topic	Frequency
people	125
cant	95
hate	90
just	85
love	80
when	75
photo	70
what	65
picture	60
away	55
around	50
cause	45
morning	40
shes	35
verdad	30