Mining Online Social Media

CS918
Week 8

Adam Tsakalidis

Module Organiser: Dr Arkaitz Zubiaga

THE UNIVERSITY OF
WARWICK

Overview

- Online Social Media
- Data Aggregation
  - Twitter Streaming API
- Data Understanding
- Text Preprocessing
  - Tokenisation
  - Noise Removal
- Feature Extraction
  - Ngrams
  - Twitter-Specific features
  - Word Embeddings

A.Tsakalidis@warwick.ac.uk

Introduction

- Online Social Media (OSM):

  "a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of user-generated content" [Kaplan and Haenlein, 2010]

- Examples:
  - Facebook Alexa Rank: 3, 2.17B users
  - Twitter Alexa Rank: 13, 0.33B users
  - Instagram Alexa Rank: 17, 0.80B users
  - Sina Weibo Alexa Rank: 21, 0.38B users
  - LinkedIn Alexa Rank: 30, 0.26B users

A.Tsakalidis@warwick.ac.uk

Case Studies

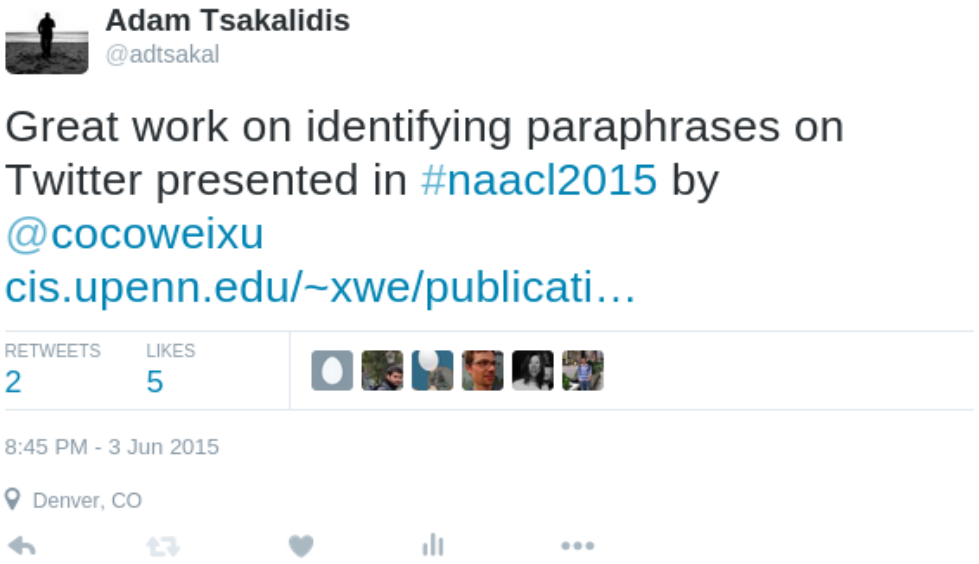
- "Hot" topics:
  - Sentiment Analysis
  - Topic Detection & Tracking
  - SPAM Detection
  - User Influence

- Examples:
  - Flu trends [Lampos et al., 2010; Aramaki et al., 2011]
  - Stock market [Bollen et al., 2013]
  - Elections [Tumasjan et al., 2010; Gay-Avello, 2013]
  - Emerging events [Atefeh et al., 2015]
  - User classification [Preoţiuc-Pietro et al., 2015]

A.Tsakalidis@warwick.ac.uk

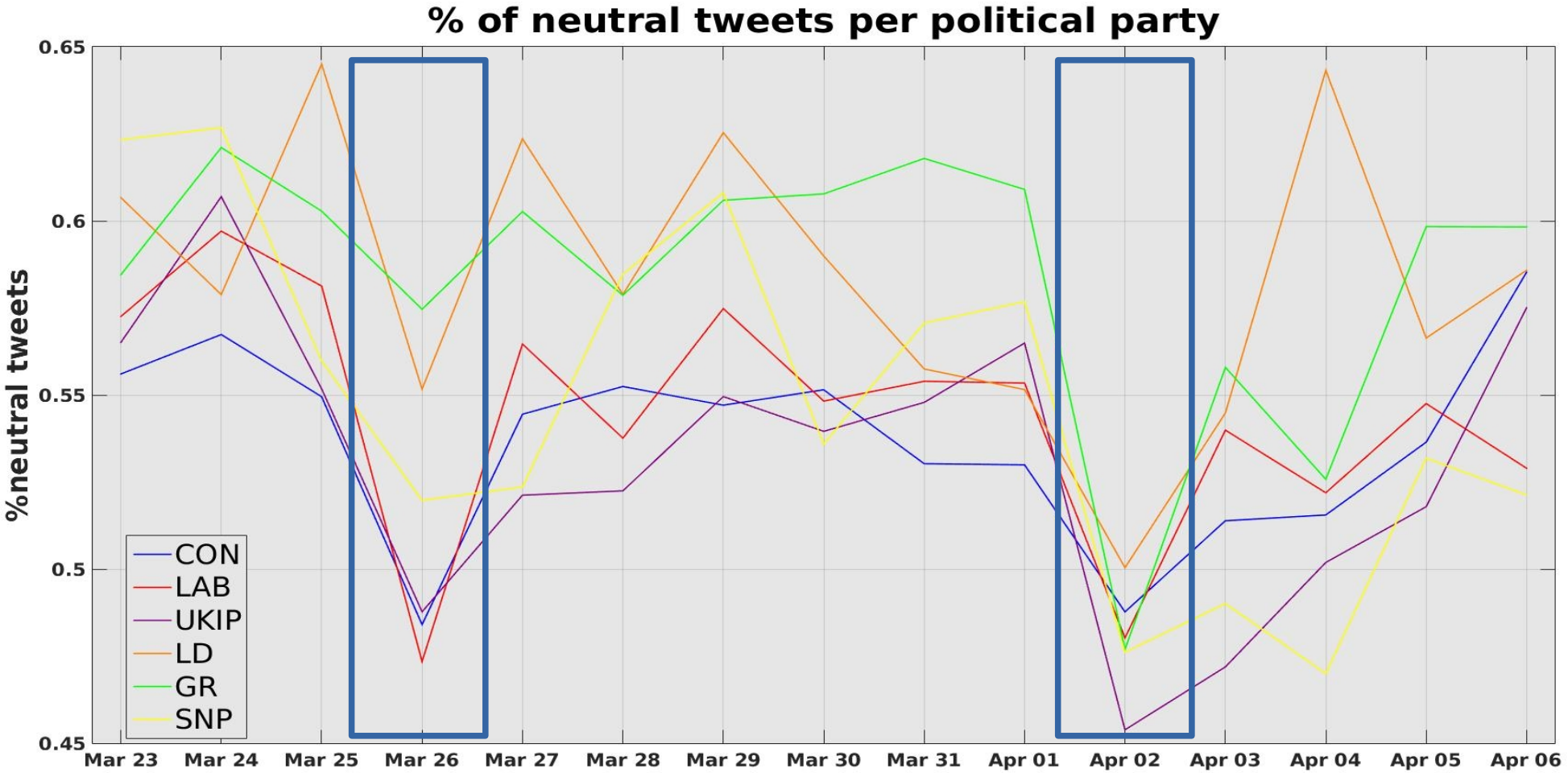
Twitter Basics

- Why Twitter?
  - Past research work...
  - 330M active users
  - Mostly open profiles
  - Streaming/Search API



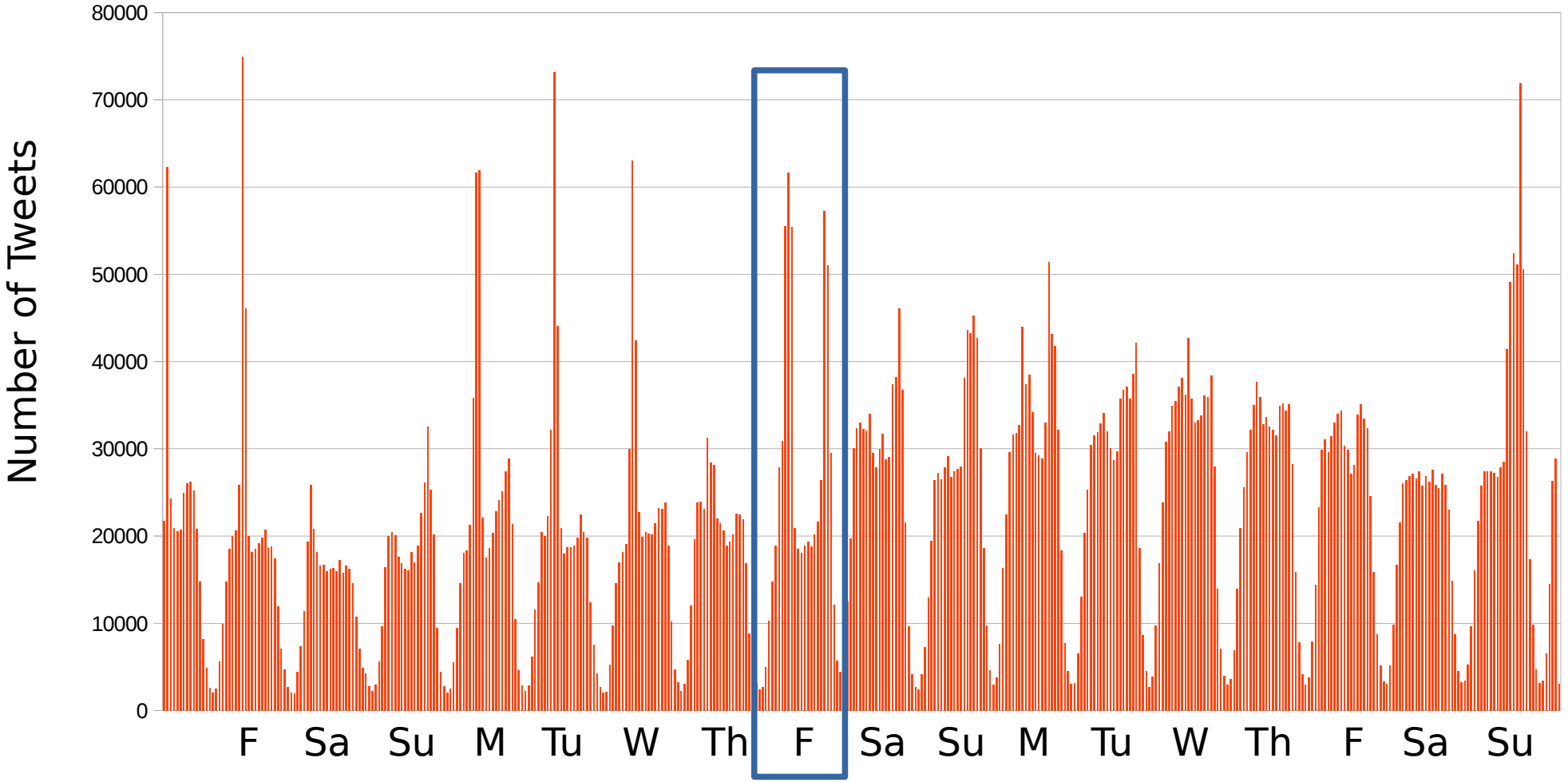
- Basics:
  - Short messages (“tweets”) with a max allowed #chars
  - Following/Followed relationships (not 1-1 communitation)
  - “Key” symbols:
    - # hashtag (pseudo-topic)
    - @ user-mention

Insights (1/2)



A.Tsakalidis@warwick.ac.uk

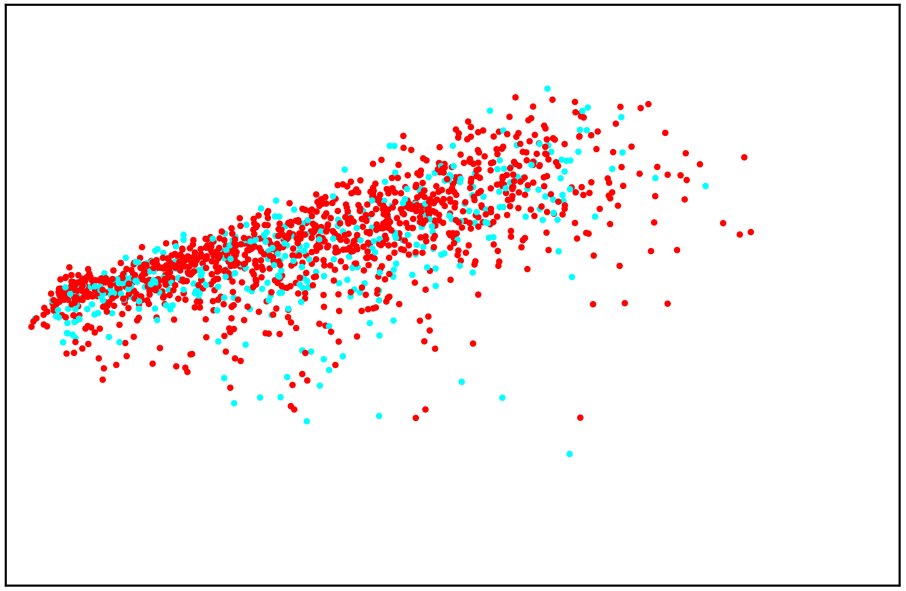
Insights (2/2)



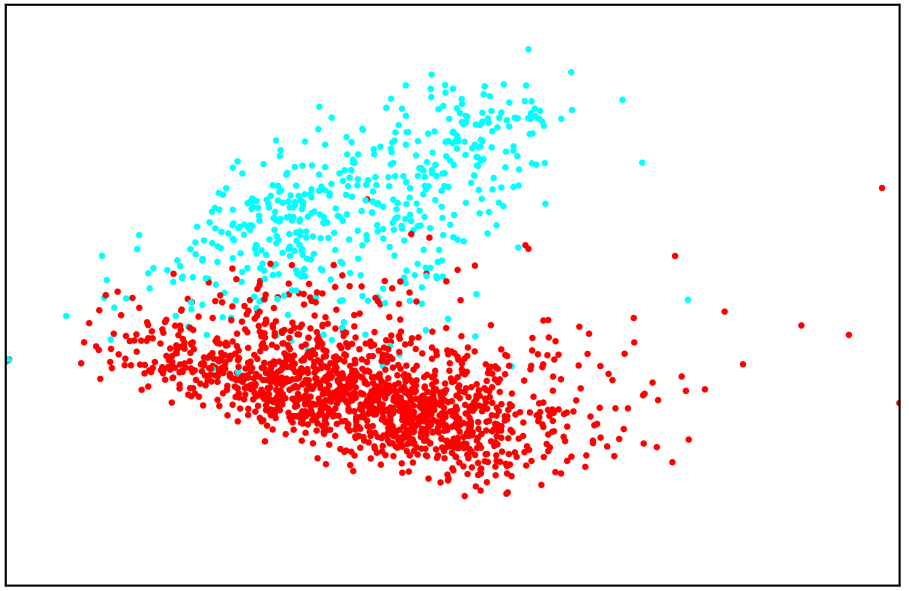
A.Tsakalidis@warwick.ac.uk

Insights (2/2)

a) Manually annotate users (YES/NO)
b) Construct the retweet graph during the two periods
c) Construct latent user graph representations [Tang et al., 2015]



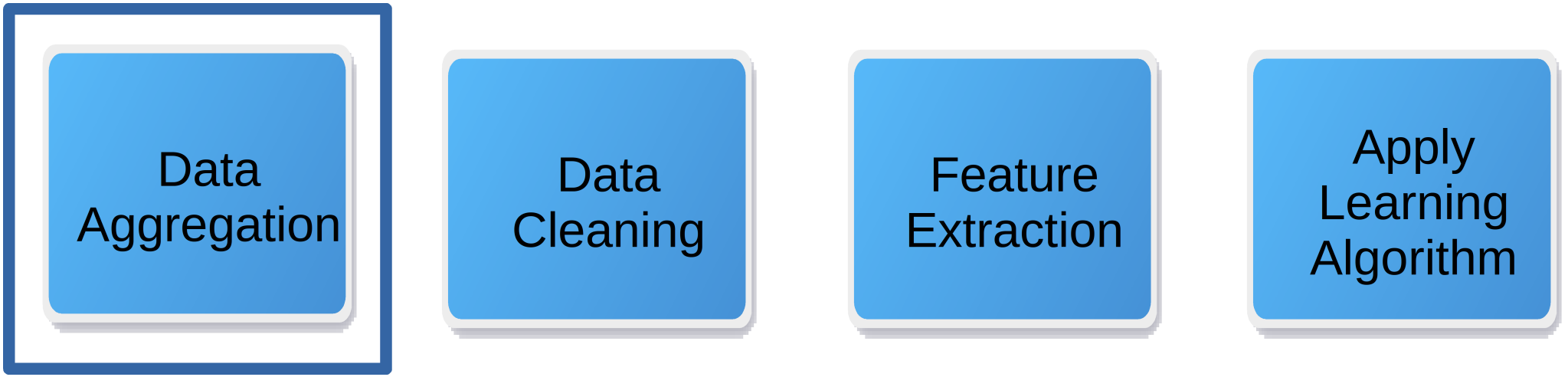
YES/NO voters **before** the referendum announcement



YES/NO voters **after** the referendum announcement

A.Tsakalidis@warwick.ac.uk

Research Process Overview



A.Tsakalidis@warwick.ac.uk

Getting Started

- Create your Twitter account [https://twitter.com/]
  - Username, email, mobile phone number, password
- Create your first application [https://apps.twitter.com/]
  - Name, description, URL, accept agreement
- Get your keys/access tokens

- Install Tweepy [http://www.tweepy.org/]
- Paste your tokens in myCrawler.py

[http://www2.warwick.ac.uk/fac/sci/dcs/teaching/material/cs918/]

A.Tsakalidis@warwick.ac.uk

MyCrawler.py (1/2)

Run as: **python myCrawler.py -i keyword**

```
def getStreamer():
        consumer_key="CK"
        consumer_secret="CS"
        access_key="123-AK"
        access_secret="AS"
        auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
        auth.set_access_token(access_key, access_secret)
        return tweepy.streaming.Stream(auth, CustomStreamListener())

class CustomStreamListener(tweepy.StreamListener):
        _filename = 'mytweets.json'
        _ffile = open(_filename, 'a')

        def on_status(self, status):
                CustomStreamListener._ffile.write(str(json.dumps(status._json))+'\n')

        def on_error(self, status_code):
                print >> sys.stderr, 'Error:', status_code
                return True

        def on_timeout(self):
                print >> sys.stderr, 'Timeout...'
                return True
```

Paste your tokens

Extend StreamListener()

A.Tsakalidis@warwick.ac.uk

MyCrawler.py (2/2)

Run as: **python myCrawler.py -i keyword**

```
# define a set of keywords to look up for
def getKeywords():
        return ['java', 'python']

# define a list of accounts to follow
def getAccountsToFollow():
        return ['@BBC', '@guardian']

# define some locations to look up for
def getLocations():
        return  [-74.255735, 40.496044, -73.700272, 40.915256,  #NY
                 -118.668176, 33.703692, -118.155289, 34.337306] #LA
```

- Hashtags can also be placed in “getKeywords()”
- Locations in the form: [W_Long, S_Lat, E_Long, N_Lat]
  - Different from [N_Lat, S_Lat, E_Long, W_Long]

  [http://www.mapdevelopers.com/geocode_bounding_box.php]

A.Tsakalidis@warwick.ac.uk

Setting Parameters

- Keywords (up to 400)
- Follow (up to 5,000 accounts)
- Locations (up to 25)

  https://dev.twitter.com/streaming/reference/post/statuses/filter
- Other parameters
  https://dev.twitter.com/streaming/overview/request-parameters

- Search API Limits:

  https://dev.twitter.com/rest/public/rate-limits

A.Tsakalidis@warwick.ac.uk

Exploring Tweet's JSON (1/3)


```
{"contributors": null, "truncated": false, "text": "RT @TheProgressives: #EUWakeUp! #HumanRights in crisis: executions, threats to freedom &amp;amp; #refugee crisis slammed in #AIR16 @amnestyonline h\u2026",
"is_quote_status": false, "in_reply_to_status_id": null, "id": 702638248716328960, "favorite_count": 0, "source": "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitter for iPhone</a>",
"retweeted": false, "coordinates": null, "timestamp_ms": "1456356987538", "entities": {"user_mentions": [{"id": 19287037, "indices": [3, 19], "id_str": "19287037", "screen_name": "TheProgressives",
"name": "S&D Group"}, {"id": 18213483, "indices": [127, 141], "id_str": "18213483", "screen_name": "AmnestyOnline", "name": "AmnestyInternational"}], "symbols": [], "hashtags": [{"indices": [21, 30],
"text": "EUWakeUp"}, {"indices": [32, 44], "text": "HumanRights"}, {"indices": [93, 101], "text": "refugee"}, {"indices": [120, 126], "text": "AIR16"}], "urls": [], "media": [{"source_user_id": 19287037,
"source_status_id_str": "702532694312411136", "expanded_url": "http://twitter.com/TheProgressives/status/702532694312411136/photo/1", "display_url": "pic.twitter.com/pEiACTEYZa", "url": "https://t.co/
pEiACTEYZa", "media_url_https": "https://pbs.twimg.com/media/Cb_lwrgWAAA7dii.png", "source_user_id_str": "19287037", "source_status_id": 702532694312411136, "id_str": "702532691367952384", "sizes":
{"large": {"h": 514, "resize": "fit", "w": 1024}, "small": {"h": 341, "resize": "fit", "w": 680}, "medium": {"h": 514, "resize": "fit", "w": 1024}, "thumb": {"h": 150, "resize": "crop", "w": 150}},
"indices": [143, 144], "type": "photo", "id": 702532691367952384, "media_url": "http://pbs.twimg.com/media/Cb_lwrgWAAA7dii.png"}]}, "in_reply_to_screen_name": null, "id_str": "702638248716328960",
"retweet_count": 0, "in_reply_to_user_id": null, "favorited": false, "retweeted_status": {"contributors": null, "truncated": false, "text": "#EUWakeUp! #HumanRights in crisis: executions, threats to
freedom &amp;amp; #refugee crisis slammed in #AIR16 @amnestyonline https://t.co/pEiACTEYZa", "is_quote_status": false, "in_reply_to_status_id": null, "id": 702532694312411136, "favorite_count": 3, "source":
"<a href=\"http://twitter.com\" rel=\"nofollow\">Twitter Web Client</a>", "retweeted": false, "coordinates": null, "entities": {"user_mentions": [{"id": 18213483, "indices": [106, 120], "id_str":
"18213483", "screen_name": "AmnestyOnline", "name": "AmnestyInternational"}], "symbols": [], "hashtags": [{"indices": [0, 9], "text": "EUWakeUp"}, {"indices": [11, 23], "text": "HumanRights"}, {"indices":
[72, 80], "text": "refugee"}, {"indices": [99, 105], "text": "AIR16"}], "urls": [], "media": [{"expanded_url": "http://twitter.com/TheProgressives/status/702532694312411136/photo/1", "display_url":
"pic.twitter.com/pEiACTEYZa", "url": "https://t.co/pEiACTEYZa", "media_url_https": "https://pbs.twimg.com/media/Cb_lwrgWAAA7dii.png", "id_str": "702532691367952384", "sizes": {"large": {"h": 514,
"resize": "fit", "w": 1024}, "small": {"h": 341, "resize": "fit", "w": 680}, "medium": {"h": 514, "resize": "fit", "w": 1024}, "thumb": {"h": 150, "resize": "crop", "w": 150}}, "indices": [121, 144],
"type": "photo", "id": 702532691367952384, "media_url": "http://pbs.twimg.com/media/Cb_lwrgWAAA7dii.png"}]}, "in_reply_to_screen_name": null, "id_str": "702532694312411136", "retweet_count": 8,
"in_reply_to_user_id": null, "favorited": false, "user": {"follow_request_sent": null, "profile_use_background_image": true, "default_profile_image": false, "id": 19287037, "verified": true,
"profile_image_url_https": "https://pbs.twimg.com/profile_images/515452443195228162/0sl1ukhb_normal.jpeg", "profile_sidebar_fill_color": "D8CCFF", "profile_text_color": "333333", "followers_count": 46747,
"profile_sidebar_border_color": "FFFFFF", "id_str": "19287037", "profile_background_color": "DDE69A", "listed_count": 1120, "profile_background_image_url_https": "https://pbs.twimg.com/
profile_background_images/451745336474607616/XoN2l3ds.jpeg", "utc_offset": 3600, "statuses_count": 15109, "description": "The Socialists and Democrats Group in the European Parliament works for social
justice & equality for all EU citizens.", "friends_count": 7936, "location": "Europe", "profile_link_color": "B3001E", "profile_image_url": "http://pbs.twimg.com/
profile_images/515452443195228162/0sl1ukhb_normal.jpeg", "following": null, "geo_enabled": true, "profile_banner_url": "https://pbs.twimg.com/profile_banners/19287037/1450166015",
"profile_background_image_url": "http://pbs.twimg.com/profile_background_images/451745336474607616/XoN2l3ds.jpeg", "name": "S&D Group", "lang": "en", "profile_background_tile": false, "favourites_count":
405, "screen_name": "TheProgressives", "notifications": null, "url": "http://www.socialistsanddemocrats.eu", "created_at": "Wed Jan 21 13:26:48 +0000 2009", "contributors_enabled": false, "time_zone":
"Brussels", "protected": false, "default_profile": false, "is_translator": false}, "geo": null, "in_reply_to_user_id_str": null, "possibly_sensitive": false, "lang": "en", "created_at": "Wed Feb 24
16:37:01 +0000 2016", "filter_level": "low", "in_reply_to_status_id_str": null, "place": null, "extended_entities": {"media": [{"expanded_url": "http://twitter.com/TheProgressives/
status/702532694312411136/photo/1", "display_url": "pic.twitter.com/pEiACTEYZa", "url": "https://t.co/pEiACTEYZa", "media_url_https": "https://pbs.twimg.com/media/Cb_lwrgWAAA7dii.png", "id_str":
"702532691367952384", "sizes": {"large": {"h": 514, "resize": "fit", "w": 1024}, "small": {"h": 341, "resize": "fit", "w": 680}, "medium": {"h": 514, "resize": "fit", "w": 1024}, "thumb": {"h": 150,
"resize": "crop", "w": 150}}, "indices": [121, 144], "type": "photo", "id": 702532691367952384, "media_url": "http://pbs.twimg.com/media/Cb_lwrgWAAA7dii.png"}]}}, "user": {"follow_request_sent": null,
"profile_use_background_image": false, "default_profile_image": false, "id": 55207471, "verified": false, "profile_image_url_https": "https://pbs.twimg.com/profile_images/700405597208977412/
GpmrVB8q_normal.jpg", "profile_sidebar_fill_color": "000000", "profile_text_color": "000000", "followers_count": 4601, "profile_sidebar_border_color": "000000", "id_str": "55207471",
"profile_background_color": "000000", "listed_count": 125, "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme14/bg.gif", "utc_offset": 3600, "statuses_count": 4225,
"description": "MEP for #Malta and #Gozo, @PL_Malta, S&D @TheProgressives, Member of @EP_Environment, @EP_Industry, @EP_Justice, @EP_Petitions", "friends_count": 1235, "location": "Malta",
"profile_link_color": "89C9FA", "profile_image_url": "http://pbs.twimg.com/profile_images/700405597208977412/GpmrVB8q_normal.jpg", "following": null, "geo_enabled": false, "profile_banner_url": "https://
pbs.twimg.com/profile_banners/55207471/1452118037", "profile_background_image_url": "http://abs.twimg.com/images/themes/theme14/bg.gif", "name": "Miriam Dalli", "lang": "en", "profile_background_tile":
false, "favourites_count": 1577, "screen_name": "Miriamdalli", "notifications": null, "url": "http://www.miriamdalli.com", "created_at": "Thu Jul 09 11:19:06 +0000 2009", "contributors_enabled": false,
"time_zone": "Rome", "protected": false, "default_profile": false, "is_translator": false}, "geo": null, "in_reply_to_user_id_str": null, "possibly_sensitive": false, "lang": "en", "created_at": "Wed Feb
24 23:36:27 +0000 2016", "filter_level": "low", "in_reply_to_status_id_str": null, "place": null, "extended_entities": {"media": [{"source_user_id": 19287037, "source_status_id_str": "702532694312411136",
"expanded_url": "http://twitter.com/TheProgressives/status/702532694312411136/photo/1", "display_url": "pic.twitter.com/pEiACTEYZa", "url": "https://t.co/pEiACTEYZa", "media_url_https": "https://
pbs.twimg.com/media/Cb_lwrgWAAA7dii.png", "source_user_id_str": "19287037", "source_status_id": 702532694312411136, "id_str": "702532691367952384", "sizes": {"large": {"h": 514, "resize": "fit", "w":
1024}, "small": {"h": 341, "resize": "fit", "w": 680}, "medium": {"h": 514, "resize": "fit", "w": 1024}, "thumb": {"h": 150, "resize": "crop", "w": 150}}, "indices": [143, 144], "type": "photo", "id":
702532691367952384, "media_url": "http://pbs.twimg.com/media/Cb_lwrgWAAA7dii.png"}]}}
```


A.Tsakalidis@warwick.ac.uk

Exploring Tweet's JSON (2/3)

tweetExplorer.py

```
In [90]: explore("mytweets.json")
{
        "contributors": null,
        "truncated": false,
        "text": "RT @TheProgressives: #EUWakeUp! #HumanRights in crisis: executions, threats to freedom &amp;amp; #refugee crisis slammed in #AIR16 @amnestyonline h\u2026",
        "is_quote_status": false,
        "in_reply_to_status_id": null,
        "id": 702638248716328960,
        "favorite_count": 0,
        "source": "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitter for iPhone</a>",
        "retweeted": false,
        "coordinates": null,
        "timestamp_ms": "1456356987538",
        "entities": {
                "user_mentions": [
                        {
                                "indices": [
                                        3,
                                        19
                                ],
                                "screen_name": "TheProgressives",
                                "id": 19287037,
                                "name": "S&D Group",
                                "id_str": "19287037"
                        },
                        {
                                "indices": [
                                        127,
                                        141
                                ],
                                "screen_name": "AmnestyOnline",
                                "id": 18213483,
                                "name": "AmnestyInternational",
                                "id_str": "18213483"
                        }
                ],
                "symbols": [],
                "hashtags": [
                        {
                                "indices": [
                                        21,
                                        30
                                ],
                                "text": "EUWakeUp"
                        },
```


A.Tsakalidis@warwick.ac.uk

Exploring Tweet's JSON (3/3)

tweetExplorer.py

```
# print out the values of a user-specific field
def printUserStuff(filename):
        with open(filename, 'r') as f:
                for line in f:
                        user = json.loads(line)["user"]
                        name = user["name"]
                        username = user["screen_name"]
                        description = user["description"]
                        location = user["location"]
                        timezone = user["time_zone"]
                        numFollowing = user["friends_count"]
                        numFollowers = user["followers_count"]
                        print username + '\t' + str(numFollowers) + '\t' + str(numFollowing)
        f.close()
```

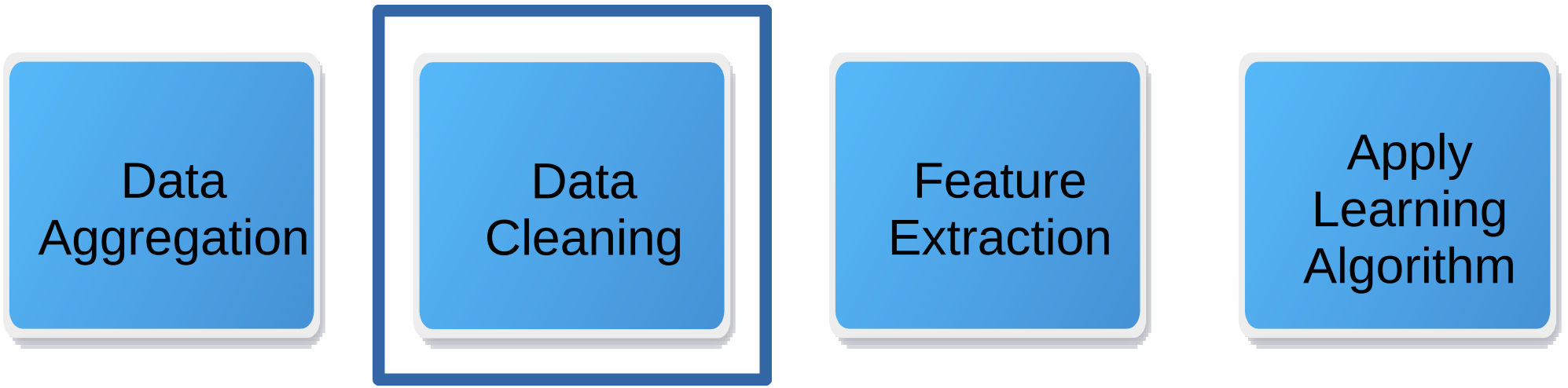
```
In [100]: printUserStuff("mytweets.json")
Miriamdalli     4601    1235
katscags62      1103    1222
cyqwtusohzo1    181     0
metelyk 61      176
onlydjhr        7225    644
sanneweber      1376    2005
Justpeachijo    1291    728
Cim_Katja       555     638
fretdem_hyman   63      228
z_eisberg       2178    2665
YBbianahR       39      168
Fight4UK        10793   10498
MyAutoBlogging  8451    48
Master_Synaps   1789    2092
blakesbassett   111     349
6262Dersim6262  541     1050
classicmedia_ng 25560   22368
ClarionByPEC    704     995
ublocks 14627   13112
NappyNoel004    177     147
nsaidian        5524    5407
news4Glasgow    2432    2880
```

```
# print out the values of a specific field
def printStuff(filename, field):
        with open(filename, 'r') as f:
                for line in f:
                        tweet = json.loads(line)
                        print tweet[field]
        f.close()
```

```
In [101]: printStuff("mytweets.json", "text")
RT @TheProgressives: #EUWakeUp! #HumanRights in crisis: executions, threats to freedom &amp;
RT @PP15146407: GO TRUMP
Texas is excited too.
Explain America's spiral of death.
Illegal Alien Invasion.
Rubio REFUGEE ACT OF 2016 https:…
https://t.co/DhgG0XvvwU Find the newest amazon tips here https://t.co/42X9k74HXD
```


A.Tsakalidis@warwick.ac.uk

Research Process Overview



A.Tsakalidis@warwick.ac.uk

Tweet Text Preprocessing

- Much more difficult than “any” other type of text!
- Tokenisation
  - NOT straight-forward
- Removal of non-alphanumeric tokens
  - hashtags, user-mentions, urls might be important!
- Lowercasing
  - all-upper-case tokens might be important
- Spelling correction
  - Beware: “lol”, "coooooollll"

A.Tsakalidis@warwick.ac.uk

Tokenisation (1/3)

- Q: How would you tokenise a string?
- A: Tokenise on white space

> @user this is AWESOME!!1Let's get the#party started!

- Ideally:
  - [@user, this, is, AWESOME, !!1, Let's, get, the, #party, started, !]
  - Keep “@user” and “#party”, separate the rest
- White Space tokenisation:

  [@user, this, is, AWESOME!!1Let's, get, the#party, started!]

A.Tsakalidis@warwick.ac.uk

Tokenisation (2/3)

@user this is AWESOME!!1Let's get the#party started
[@user, this, is, AWESOME, !!1, Let's, get, the, #party, started, !]

- Consider non-alphanumeric chars:

  [@, user, this, is, AWESOME, !!, 1Let, 's, get, the, #, party, started, !]

- Give up, just use nltk:

  [@, user, this, is, AWESOME, !, !, 1Let, 's, get, the, #, party, started, !]

- Twitter-specific tokeniser [Owoputi et al., 2013]:

  [@user, this, is, AWESOME!!1Let's, get, the, #party, started, !]

[https://github.com/myleott/ark-twokenize-py]

A.Tsakalidis@warwick.ac.uk

Tokenisation (3/3)

Hands-on: **tokenisers.py**

```
In [50]: tweet
Out[50]: "@user this is AWESOME!!1Let's get the#party started!http://someurl.com"
```

```
In [51]: ws = tokenise_ws(tweet)

In [52]: ws
Out[52]:
['@user',
 'this',
 'is',
 "AWESOME!!1Let's",
 'get',
 'the#party',
 'started!http://someurl.com']
```

```
In [58]: twok = tokenise_twok(tweet)

In [59]: twok
Out[59]:
['@user',
 'this',
 'is',
 "AWESOME!!1Let's",
 'get',
 'the',
 '#party',
 'started',
 '!',
 'http://someurl.com']
```

```
In [56]: nltk_tok = tokenise_nltk(tweet)

In [57]: nltk_tok
Out[57]:
['@',
 'user',
 'this',
 'is',
 'AWESOME',
 '!',
 '!',
 '1Let',
 "'s",
 'get',
 'the',
 '#',
 'party',
 'started',
 '!',
 'http',
 ':',
 '//someurl.com']
```

Testing three different tokenising methods. Think about the tokens after removing non-alphanumeric characters.

A.Tsakalidis@warwick.ac.uk

Noise Removal

- Remove non-alphanumeric
  - Issue: :) :(
- Deal with negation?
- Replace user mentions, extract hashtags
  - Allowed username symbols: alphanumeric and “_”
  - Regex: ?
- Replace URLs
  - Regex: ?
- Convert to lower-case

A.Tsakalidis@warwick.ac.uk

Other Preprocessing

- All-Uppercase tokens?
  - Might be useful to keep them!
- Stop-word removal/stemming?
  - Task-dependent
- Expand abbreviations?
- Part-of-Speech Tagging?
  - Stanford, GATE...
- Named Entity Recognition?
  - Difficult!

A.Tsakalidis@warwick.ac.uk

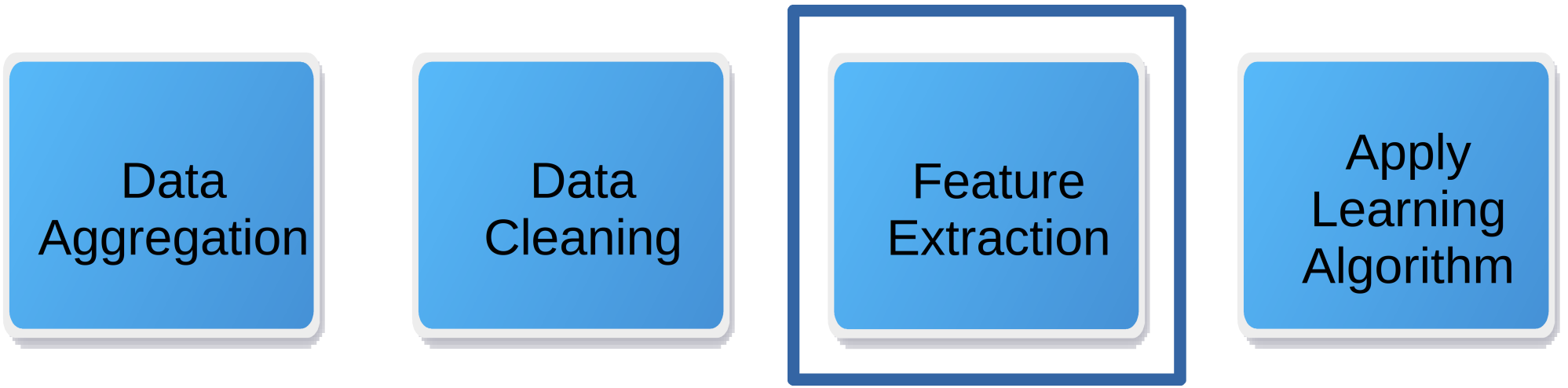
All together

```
from nltk.tokenize import word_tokenize
import twokenize, re, json

#tokenise using nltk.tokenize word_tokenize
def nltk_tokenise(tweet):
        tokens = word_tokenize(tweet)
        tokenised = ''
        for token in tokens:
                tokenised += str(token.encode('utf-8')+' ')
        return tokenised.strip()


#replace URLs with "URLLINK" (beware: the REGEX here is not necessarily correct)
def replaceURLs(tweet):
        return re.sub(r"http\S+", "URLLINK", tweet)


#replace user mentions with "USERMENTION"
def replaceUserMentions(tweet):
        return result = re.sub("(@[A-Za-z0-9_]+)", "USERMENTION", tweet)


#replace all non-alphanumeric (beware: the REGEX used here is not necessarily correct)
def replaceRest(tweet):
        result = re.sub("[^a-z]", " ", tweet)
        return re.sub(' +',' ', result)


def testit():
        with open("mytweets.json", 'r') as f:
                for line in f:
                        text = json.loads(line)['text']
                        newtext = nltk_tokenise(text).lower()
                        newtext = replaceURLs(newtext)
                        newtext = replaceUserMentions(newtext)
                        newtext = replaceRest(newtext)
                        print(text, '\n', newtext, '\n')
        f.close()
```

textPreprocessor.py

A.Tsakalidis@warwick.ac.uk

Research Process Overview



Feature Extraction

- Ngrams
  - {binary, tf, tfidf}, {POS}
- Twitter-specific
  - Hashtags, uppercase words, tweet length...
- Word Embeddings
  - Use functions over vector dimensions
- Others
  - Topics, Clusters, Lexicons...

A.Tsakalidis@warwick.ac.uk

Ngram Features (1/2)

- Doc#1: “great piece of art have a look”
  - Unigrams: {great, piece, of, art, have, a, look}
  - Bigrams: {great piece, piece of, of art, art have, have a, a look}
- Doc#2: “omg this notdoes look great”
  - Unigrams: {omg, this, notdoes, look, great}
  - Bigrams: {omg this, this notdoes, notdoes look, look great}
- Construct vocabulary (e.g., binary unigrams):

Doc	a	art	great	have	look	notdoes	of	omg	piece	this
Doc#1	1	1	1	1	1		1		1	
Doc#2			1		1	1		1		1

A.Tsakalidis@warwick.ac.uk

Ngram Features (2/2)

- Implement your ngram extractor...
- ...or use sklearn's one (read the documentation!)
- Typically, combinations of n={1, 2, (3)} are used

featureExtractor.py

```
from sklearn.feature_extraction.text import CountVectorizer

# returns matrix
# one row per document
# one column per unigram
# Source: http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
def getUnigrams():
        vectorizer = CountVectorizer()
        corpus = ['this is a document.',
        'another document.',
        'yet another document!']
        X = vectorizer.fit_transform(corpus)
        print vectorizer.vocabulary_
        print X.todense()
```

```
In [22]: getUnigrams()
{u'this': 3, u'is': 2, u'document': 1, u'yet': 4, u'another': 0}
[[0 1 1 1 0]
 [1 1 0 0 0]
 [1 1 0 0 1]]
```


A.Tsakalidis@warwick.ac.uk

Twitter-Specific Features

- Ngrams are quite basic
  - Fail to capture pseudo-semantics
- Other linguistics might be very important:
  - All-uppercase words, hashtags, consecutive letters, smileys, punctuation, user mentions...
- New sets of features:
  - e.g., #all-uppercase or #elongated words, #smileys/frowns...
  - tweet length, number of words
  - keeping the hashtags is important for many tasks

A.Tsakalidis@warwick.ac.uk

Word Embedding Features

- WE: Vector representations of words
  - But how do we use them?
  - Apply functions over each dimension
- Example:

"are you feeling good today"

Token	dim1	dim2
are	0.8	0.2
you	0.7	0.3
feeling	0.2	0.2
good	0.5	0.9
today	0.1	0.1

ftr_avg1 = avg(dim1) = .46
ftr_avg2 = avg(dim2) = .34

Typically: avg, min, max

NumFeatures = d*f

- d: #dimensions of WE
- f: #functions applied

A.Tsakalidis@warwick.ac.uk

Other Features

- Topic modelling
  - LDA, LSA, LSI
- Word Clusters
- Semantics
  - Enrichment, replacement... [Saif et al., 2012]
- **Sentiment Lexicons**
  - Map every word to a lexicon
- Other lexicons
  - Can we create WE based on existing knowledge bases? [Faruqui et al., 2015]

A.Tsakalidis@warwick.ac.uk

Summary

- Introduction to OSM Mining
- Covered three stages of OSM Mining:
  - Data Aggregation
  - Text Preprocessing
  - Feature Extraction
- Build on the code provided (can save some time!)

- Next Seminar:
  - Feature Extraction for Sentiment Analysis
  - Sentiment Classifiers

A.Tsakalidis@warwick.ac.uk