利用机器学习算法对**Trump twitter**的主题分析

靳帅祥 16307130023 2018.01.21

1. 介绍

如今自然语言处理领域迅速发展，研究人员开始尝试对各种文本进行分析。随着智能手机的普及，社交网络呈爆发式发展，从而为研究社交媒体的文本分析产生了兴趣。作为社交网络上最有影响力的人物，莫过于美国总统Donald Trump的twitter了，因其发twitter的频率高且经常在上面宣布一些重大的决定，甚至直接对竞争对手进行攻击，被称为“推特治国”。可以说，他的twitter以作为他本人甚至现在美国的缩影。对其twitter进行研究，分析其中所包含的重要信息。

之后利用LSI、LDA等无监督学习的方法进行了主题分析，得到了16个主题及其关键词；然后利用Word2vec工具建立模型，得到所有tweet的主题相关度分布。同时，还进行了一些其他的分析。

（代码使用方法见后记）

2. 数据获取

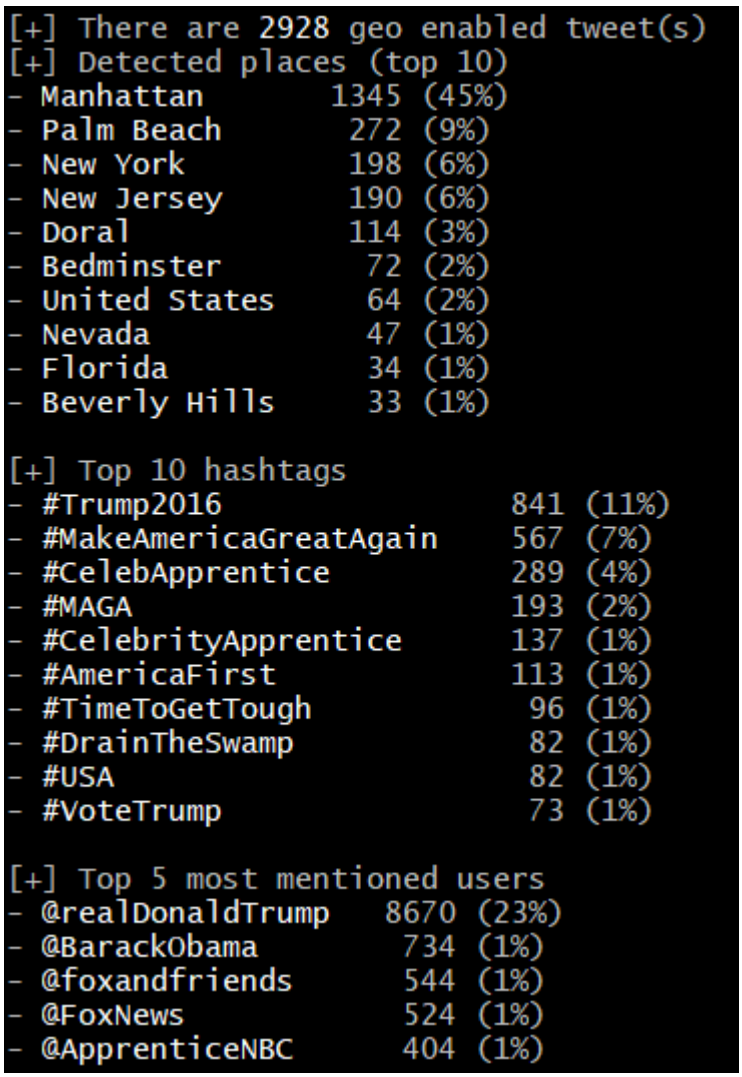
进行自然语言处理的第一步便是获取需要处理的数据，twitter官方提供了相关的API[1]，但官方API有获取twitter数量的限制，并不能获得一个用户的所有数据。但由于本次研究的人物很特殊，网络上有人整理了Donald Trump从注册到现在发送的所有twitter的数据，从这个网站[2]上获取到了需要处理的数据，数据中的格式与官方的API相同，用json储存，共计35402条twitter。

数据中各个标签对应含义如附件中的`./map-of-a-tweet-copy.pdf`所示，包含该条twitter的发布日期、内容、点赞数、转发数、发布地点、发布设备等等，利用这些原始的、繁杂的信息来得到更加精炼的信息。

下载完数据之后，首先对数据进行了简单的统计处理，统计了起止时间、平均每日发送twitter的次数、周活动分布、日活动分布、地理位置的分布等等。

统计的结果如下：

从上图中可以看出，Trump发送Tweet的频率非常之高，达到了平均每日11.2次，被称为"推特治国"并不过分。通过日活动时间统计，发现Trump的睡眠时间并不早，在19:00-21:00时间段最为频繁；对一周的活动，发现Trump作为美国总统也并不是一周七天都是在高强度的工作，在周末也会有适当的休息。



从places的分析中，可以看出在含有位置信息的Tweet中，曼哈顿出现的频率最高。这应该与Trump从纽约市皇后区迁居到曼哈顿区，并在曼哈顿区开展商业活动有关；

从hashtag(twitter中用来标注线索主题的标签)的分析中，出现频率的最高的Trump2016便是与Trump竞选有关的话题，而后面的"Make America Great Again"与"MAGA"同时也是Trump参加竞选的口号，足以看出其对竞选的重视。从后面的其他tag中，也能看出Trump的其他政策。

3. 数据清洗

接下来下载的数据进行清洗，使其能过在接下来的处理中应用。应为要做的是对Trump的twitter的主题分析，所以需要对每条的Tweet的'text'词条进行处理。Twitter文本具有一些特殊的语法，需要做一些预处理工作。

首先假设文本中的每个单词都是用空格隔开的，这里利用nltk中的`nltk.tokenize.WhitespaceTokenizer()`来把Tweet中每个用空格隔开的文本单元分成初步的token，再将这些token的字符全部转为小写。之后对Twitter的特殊规则进行处理。

1. 如果开头为“https”，这个token辨识为网站链接（link）
2. 如果开头为“@”，则为推特的用户名
3. 如果开头为“#”，则代表“hashtag”（twitter中用来标注线索主题的标签）
4. 如果开头为“&”，则基本上是tweet中的“&”符号，在utf-8的文本中以“&”代表（目前暂未发现反例）
5. 通过一个正则表达式，辨识token是否为email地址

排除掉这些特殊token之后，对于剩余token再用Tokenizer再做一次分词，这样可以拆分那些没有被空格隔开的、被标点符号、连字符连接的多词token。最后对每个token再做一次检查，如果是全部由拉丁字母和数字组成的，那基本可以认为是普通类型的token（标记成normal）；否则则说明它包含一些特殊字符，例如原本tweet里的表情符号被强制转换成utf-8文本之后，会变成一些乱码；或者是一些外国语言的人名之类的，这些token统一标记成“special”类型。在后续的NLP分析中，只考虑normal类型的token。这样，每个tweet就转化为了一个token组成的list，每个token有自己对应的类型。

转化后的text类似于下列形式（括号中为辨识到的token类型）：

```
 we(normal) '(special) re(normal) all(normal) thinking(normal) of(normal)
you(normal) stevescalise(tweetUser) teamscalise(Hashtag)
https://t.co/yqf6exhm7x(link)
```

4. 初步统计

数据清洗结束后，可以进行一些基本的分析。

建立一个词袋(Bag of Words)模型，只统计非转发的Twitter中标记为noraml的token，并且过滤掉停止词与一些根据之前的统计结果观察出的需要过滤的词。统计得到的数据中共有551711个normal word，293653个非停止词，17128特殊词，出现次数最多的30个词如下：

```
('great', 4794), ('trump', 4676), ('thank', 2262), ('thanks', 2152), ('president',
2058), ('people', 1722), ('donald', 1714), ('obama', 1543), ('america', 1527),
('would', 1448), ('get', 1399), ('new', 1366), ('like', 1307), ('country', 1281),
('make', 1262), ('time', 1254), ('one', 1223), ('good', 1200), ('big', 1126),
('run', 1083), ('u', 1073), ('again', 1033), ('today', 974), ('never', 945), ('us',
930), ('love', 909), ('best', 900), ('going', 895), ('back', 868), ('think', 864)
```

符合其嚣张狂妄的形象，利用`WordCloud`做出词云如下`cloudimg.png`：

5. 主题分析

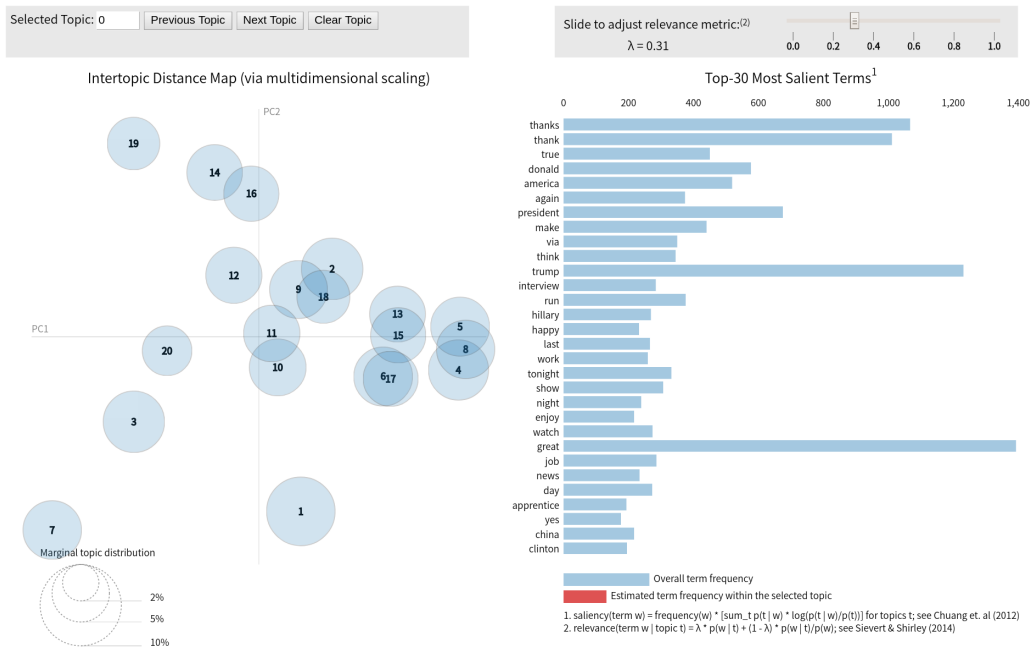
主题分析是NLP的一个重要组成部分，Trump的每条tweet可能分属不同的主题，比如竞选、抨击竞争对手、环境问题、相关政策等等。由于获取到的是没有进行标注的数据，人工标注也不现实，所以需要采用无监督学习的方式，在这里利用无监督学习方法(如PCA、LSI、LDA等)进行分析。

LSI的全名是Latent Semantic Indexing，是一种基于SVD(Singular Value Decomposition)的方法，用矩阵分解的方式将文本中的单词投射到向量空间中，从而抽取主题关键词。使用gensim来实现LSI的分析，利用gensim中corpora模块建立BOW模型，再用models模块把它转化为tf-idf模型（统计了单词的相对出现频率而不是出现次数），最后学习出一个LSI模型，从而得到20个topic的keyword，以及每个keyword在对应的余弦空间的投射。通过similarities模块，可以计算任意一个tweet文本与所有topic之间的余弦相似度。

LSI虽然可以提取topic，但是它只是一个纯粹的数学手段，并没有考虑单词间出现概率的影响；而LDA（Latent Dirichlet Allocation，不是Linear Discriminant Analysis）是基于三层贝叶斯概率模型的主题提取方法，包含了很多上下文之间的语义信息。

利用scikit-learn里面的feature_extraction模块将新的文本集转化为tf-idf模型，然后用decomposition模块里的LDA进行分析。利用工具包pyLDAvis可以打开一个网页，里面是LDA结果的可视化展示，截图如下

`LDA_result.png`



网页的左侧用不同圈代表topic，圈的大小代表该topic的比重；横坐标和纵坐标分别为前两大的主成分（PC）。网页的右侧显示了所有词语的频率,与初步统计章节中的结果是基本吻合的（如果用鼠标点击某一个圈的话，还会显示该topic下同样词语的频率占比），调节右上角滑块可以改变单词的在本topic占比和在全文中占比的权重。

通过人工筛选出16个Topic如下:

```
vote          ['run', 'president', 'please', 'agree', 'needs', 'mr', 'running', 'vote', 'someone', 'mess', 'save']
fakeNews      ['fake', 'news', 'media', 'poll', 'luck', 'story', 'cruz', 'ted', 'dishonest', 'rubio', 'report']
golf          ['golf', 'course', 'hotel', 'tower', 'honor', 'scotland', 'chicago', 'welcome', 'building', 'club', 'sign']
othercountry  ['china', 'leader', 'iran', 'isis', 'oil', 'find', 'iraq', 'respect', 'trade', 'syria', 'mexico']
whitehouse    ['white', 'house', 'obamacare', 'north', 'healthcare', 'republicans', 'loser', 'korea', 'chance', 'global', 'warming']
HRC           ['hillary', 'clinton', 'crooked', 'politics', 'bernie', 'fbi', 'thoughts', 'truth', 'may', 'touch', 'beat']
police        ['tax', 'economy', 'security', 'home', 'illegal', 'border', 'wall', 'immigration', 'weak', 'crime', 'reform']
life          ['work', 'hard', 'trying', 'trump', 'decision', 'thought', 'wonderful', 'proud', 'beautiful', 'talking', 'meeting']
America       ['again', 'make', 'america', 'interesting', 'video', 'gets', 'follow', 'safe', 'god', 'problems', 'future']
speech        ['think', 'congratulations', 'champion', 'makes', 'become', 'general', 'fan', 'days', 'continue', 'winning', 'happen']
congratulations['forward', 'looking', 'soon', 'speech', 'crowd', 'loved', 'politicians', 'learn', 'leaving', 'past', 'governor']
jobs          ['high', 'since', 'jobs', 'successful', 'wind', 'fight', 'government', 'market', 'stock', 'hate', 'unemployment']
unemployment  ['last', 'night', 'join', 'live', 'job', 'team', 'debate', 'set', 'absolutely', 'romney', 'failed']
join          ['success', 'getting', 'million', 'something', 'bush', 'ready', 'congrats', 'presidential', 'winner', 'jeb', 'donaldtrump']
veterans      ['star', 'party', 'day', 'disaster', 'far', 'ago', 'call', 'close', 'press', 'veterans', 'taking']
interview     ['true', 'interview', 'happy', 'yes', 'awesome', 'discussing', 'birthday', 'everything', 'women', 'open', 'men']
```

左侧为自己总结的Topic名字，Topic的名字并不能完全概括主题的内容。

从主体分析的结果中可以发现一些有趣的事情，例如：HRC(希拉里)话题中keywords的分布表明Trump很喜欢用crooked形容希拉里，在这个主题中还有Bernie，另一个总统候选人。

6. Word2vec分析

由于一共有551711左右的词，数据量虽然不太大，但是应该足够用来训练一个简单的word2vec词向量模型。训练出来的word2vec模型还可以用于更深层次的分析。比如利用CNN、RNN进行进一步分析。使用gensim.models中的Word2Vec模块，用下载到的twitter数据作为输入，设定每个单词向量维度为300，训练出一个小型模型；

对训练出的模型进行测试：

```
w2v_Trump.wv.most_similar(positive=['hillary'])
[('crooked', 0.6706148386001587),
 ('clinton', 0.6076074838638306),
 ('excoriates', 0.4195103049278259),
 ('wikileaks', 0.41540080308914185),
 ('rodham', 0.4114377796649933),
 ('overtaxes', 0.41111671924591064),
 ('softball', 0.4106309413909912),
 ('urged', 0.40947848558425903),
 ('exonerating', 0.4090154767036438),
 ('catches', 0.40578269958496094)]
```

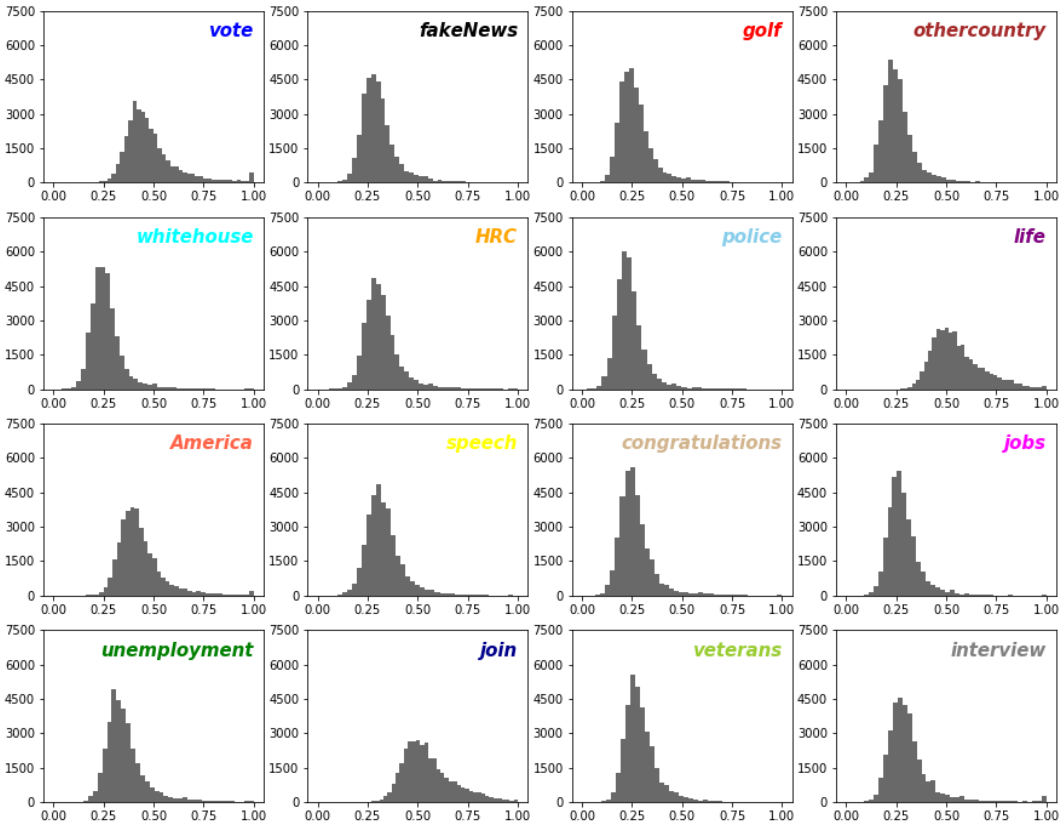
可以看到很多熟悉的词汇，比如'crooked'，在之前的主题分析中曾经出现过。'wikileaks'爆料了关于希拉里的众多新闻，clinton、rodham都是希拉里名字的一部分；以及softball事件；相关度较高。

```
w2v_Trump.wv.most_similar(positive=['fake'])
[('news', 0.6794359683990479),
 ('suppression', 0.4579120874404907),
 ('fabricate', 0.44496068358421326),
 ('rigging', 0.4443070590496063),
 ('distorted', 0.43214553594589233),
 ('media', 0.4269370436668396),
 ('partisan', 0.4256756007671356),
 ('mainstream', 0.42544591426849365),
 ('oversampling', 0.42067909240722656),
 ('orgs', 0.41753920912742615)]
```

fake news是trump一直在强调的话题，fabricate 捏造 distorted 扭曲 media 等等

7. 主题结果分析

首先利用word2vec的结果，对全部用这16个Topic对所有的tweet的相似度数据做一个柱状图，如下图所示：
`topics_hist.png`

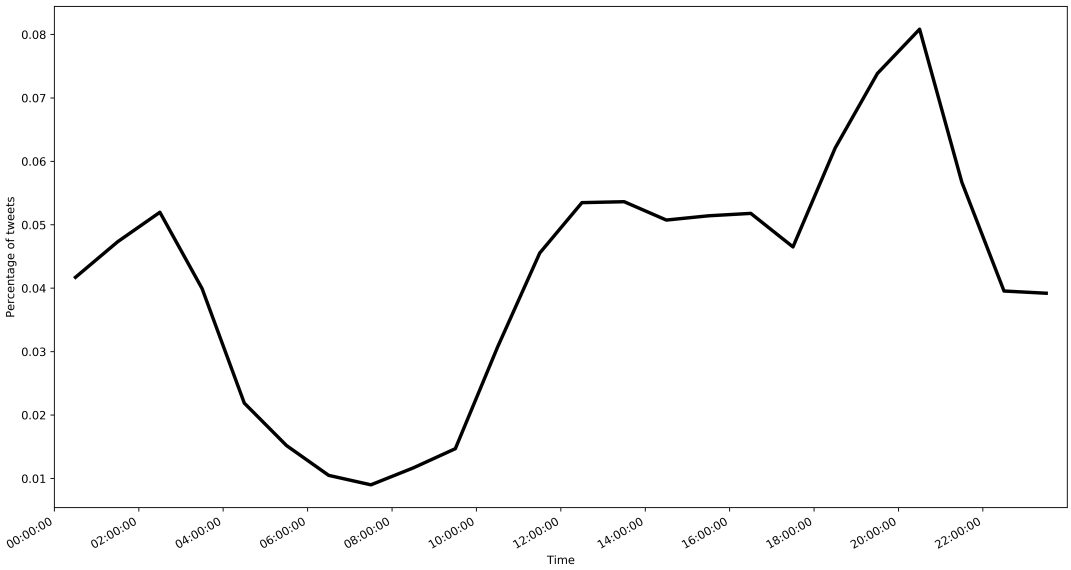


不同的topic的相似度分布的区别很是比较大的，例如othercountry的相似度普遍低一些。原因并不一定是和这个topic有关的tweet数量少，而是它的keyword对应了不同的事件，比如keyword中的China Iran Mexico Syria这四个国家，同时提到这四个国家的概率就很低。另一个原因可能是我在选择keyword的时候，有一些词权重相同，就随机选择了一个。

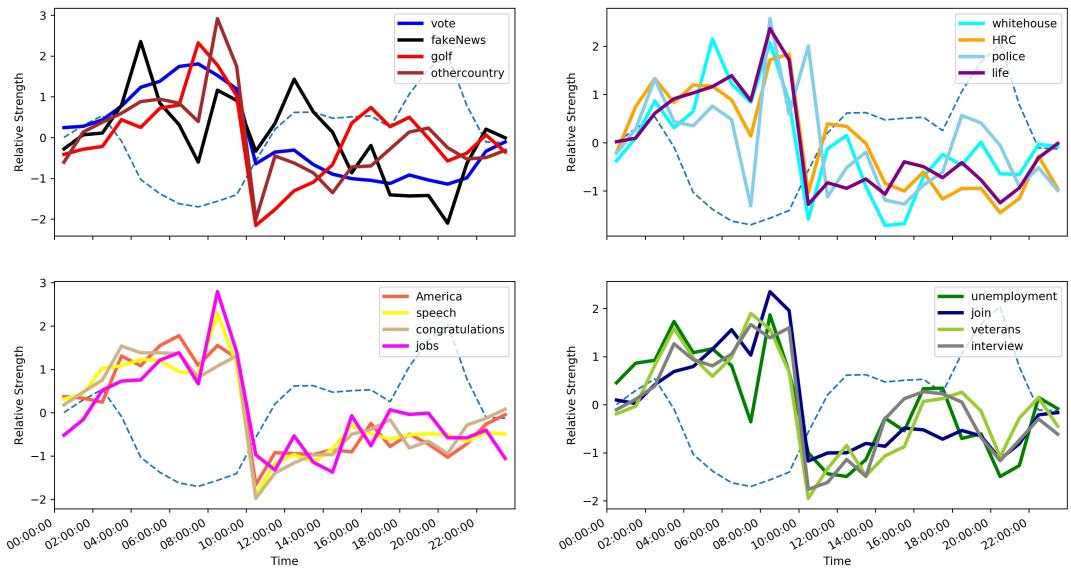
很多topic多存在相似度等于100%的一个集中分布，说明一部分tweet是可以完美符合该主题的。比较典型的就是vote，拉票的话也没其他的话可以说……

8. 其他分析

利用得到的数据，还对所有tweet发送时间做了统计，x轴为发送时间，y轴为所占比例，bili如图
`tweets_time.png`：



对每个主题，基于时间对其相对强弱进行了分析`Topic_Relative_Strength.jpg`：



9. 后记

本次用到的库如下(不完全统计)：

```
nltk
numpy
pandas
scripy
wordcloud
imread
gensim
csvpy
ascii-graph
tqdm
pickle
pyLDAvis
matplotlib
dateutil
```

由于训练数据需要花费一定的时间，所以，训练好的LSI、LDA、word2vec模型在./test/Pickles文件夹下。在代码中将直接引用这些数据。

如需运行程序，可先`pip3 install -r requirements.txt`,再用`jupyter`打开`analyize.ipynb`

---

1. https://developer.twitter.com/en/docs/api-reference-index↩

2. http://www.trumptwitterarchive.com/↩