Natural Language Processing and Machine Learning Project
Twitter topic clustering

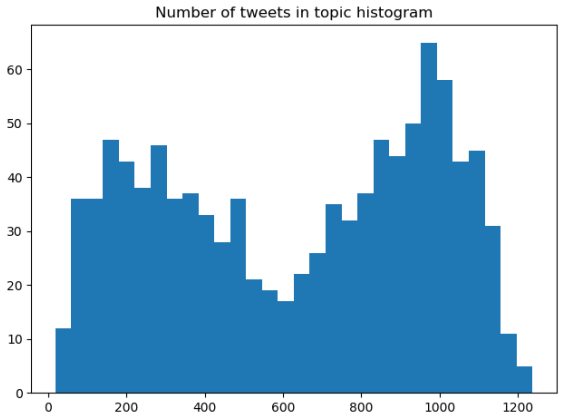
Timur Valiev, BS3-DS-1

Abstract

This document contains the description of implementation of twitter topic clustering project that takes a batch of tweets and attempts to group tweets into topics.

1 Dataset and preprocessing

Dataset Real-Time Classification of Twitter Trends is used for this project. The dataset consists of 1036 topics with ~100-1200 tweets in each topic.



The topic labels assigned to each tweet are ignored during clustering step and used only for evaluation of the models.

1.1 Preprocessing

Tweet preprocessor cleans tweet text from URLs, mentions, hashtags, emojis, and reserved words (such as RT in the beginning of each retweet). In addition, preprocessor extracts *references* from each tweet: mentions, hashtags, links and named entities. Cleanup of tweet text and extraction of mentions, hashtags and links is performed using tweet-preprocessor library. Named entities are extracted using spaCy library with en_core_web_lg model. Types of extracted named entities:

- People, including fictional. (PERSON)
- Nationalities or religious or political groups. (NORP)
- Buildings, airports, highways, bridges, etc. (FAC)
- Companies, agencies, institutions, etc. (ORG)
- Countries, cities, states. (GPE)
- Non-GPE locations, mountain ranges, bodies of water. (LOC)
- Objects, vehicles, foods, etc. (PRODUCT)
- Named hurricanes, battles, wars, sports events, etc. (EVENT)
- Titles of books, songs, etc. (WORK_OF_ART)

2 Feature extraction

Feature extraction pipeline consists of following steps:

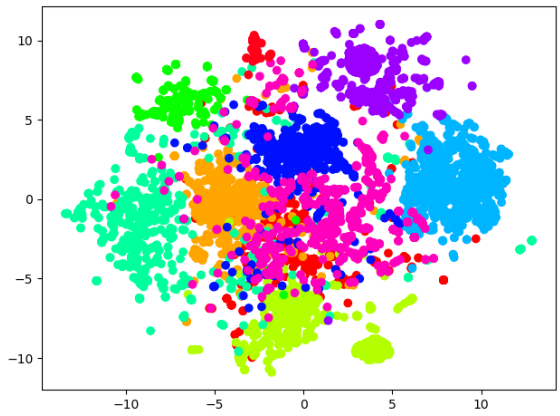
1) Cleaned tweet texts are used to create term frequency vectors for each tweet.
2) References extracted by preprocessor are treated as separate tokens and used to generate reference occurrence vector for each tweet.
3) Term frequency vectors are joined with reference occurrence vectors and passed to TF/IDF vectorizer to generate final vector representation of each tweet.

The goal of this method is to bring named entities, mentions, hashtags and links into vectorized representation of tweets for better representation of tweets, since they have huge impact on topic assignment.

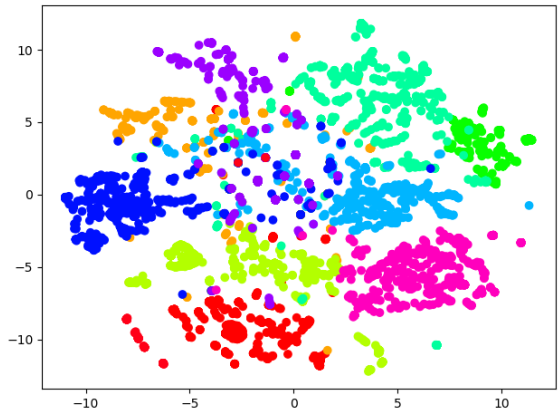
2.1 Visualisation

To visualize impact of this method I used t-SNE on vectorized representation of tweets from 10 topics for 3 vectorization methods (colors represent different topics):

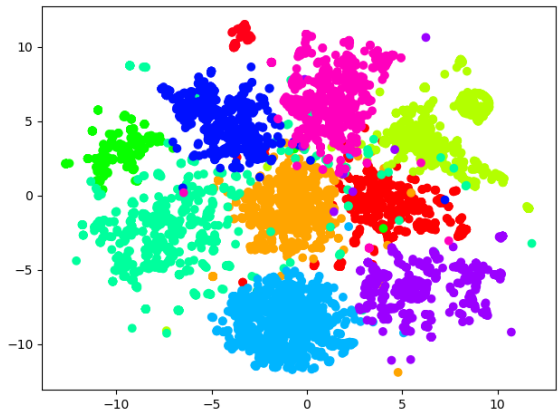
TF/IDF on token frequency vectors:



TF/IDF on reference occurrence vectors:



TF/IDF on combined token frequency and reference occurrence vectors:



As you can see, the last method gives more distinct and dense clusters. Usage of this method gave 0.10-0.27 performance boost for different models (performance measured using Rand index).

3 Topic detection models

3.1 Baseline model

My baseline solution is following:

- Calculate frequency of each reference in the entire dataset
- Assigns classes to tweets that contain references by setting the most common reference in the tweet as its class.
- Assign classes to tweets without references using KNN with cosine similarity of TF/IDF representation of tweets as metric.

This model assumes that each reference appearing in dataset represents a topic. In case if several references are occurring in a tweet, then the most common reference is the topic of the tweet. KNN is used to assign topics to tweets without references, which are about 5.7% of tweets in the dataset. The models scores 0.39-0.42 similarity to actual topics clustering (measured by Rand index).

3.2 Clustering models

Firstly, I tried to apply k-means clustering on the TF/IDF of tokens and references representation described above in the preprocessing section. Number of clusters is set to be the actual number of topics in the provided batch of tweets. This models scores 0.38-0.43 similarity to actual topics clustering (measured by Rand index) on 20-40 topics, which is almost the same as the baseline solution.

To improve performance of clustering I reduced sparsity of data by applying dimensionality reduction with SVD on the TF/IDF representation of data. In addition, I switched to agglomerative clustering with cosine affinity and complete linkage since it gives much better performance on this data. This models scores 0.76-0.81 similarity to actual topics clustering (measured by Rand index) on 20-40 topics, and this is the best performing model.

The problem of both clustering models is that they require a lot of RAM for calculations and I

could not run the models on more than 40 topics (~30000 tweets) on my machine.

3.3 LSA-based model

As an alternative to the clustering methods, I used Latent Semantic Analysis based model: I apply singular value decomposition to the TF/IDF matrix of tokens and references to get tweet to latent topic matrix, which is used to assign topics to the tweets. This models scores 0.71-0.78 similarity to actual topics clustering (measured by Rand index) on 20-40 topics.

Even though the LSA model has slightly worse performance than the agglomerative clustering model, it has many other benefits:

1) It runs much faster than clustering and requires less RAM: I could run LSA on up to 225 topics (~160000 tweets).
2) It allows to automatically detect number of topics using singular values returned for each topic.
   To do so I take explained variance ratio returned by the SVD calculation model, and pick the number of topics such that sum of explained variances for picked topics is less than 0.18. This value is derived empirically to work best on topic detection with 10 to 100 topics.
   Results of this method:

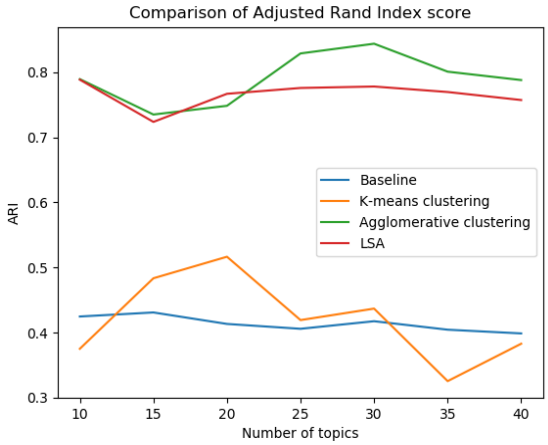
Actual N of topics	10	25	50	75	100
Predicted N of topics	11	33	59	77	97

3) Topic to token matrix returned by SVD allows detecting most representative tokens for each topic by picking tokens with biggest weight in the matrix.
   Results of this method when selecting the most representative references:

Topic name	**Top 3 most representative references**
#mileyonsnl	#MileyOnSNL, @LMAOTwitpics, http://twitpic.com/3rsr3o
Leap Year	Leap Year, HBO, @LMAOTwitpics
Sober Valley Lodge	Charlie Sheen's, Goddesses Speak:, Sober Valley Lodge
Howard Davies	Howard Davies, LSE, Gaddafi
Sky News	Sky News, News Corp, Murdoch
Velez	Velez, Gol de Velez, gol de velez
Brazil♥LadyGaga	@siteladygagabr, @projetoladygaga, @ladygaga
#dealwithit	#dealwithit, #Wisconsin, #OhioState
Ban Jordan Heels	@LMAOTwitpics, http://twitpic.com/3rsr3o, Jordan Heels

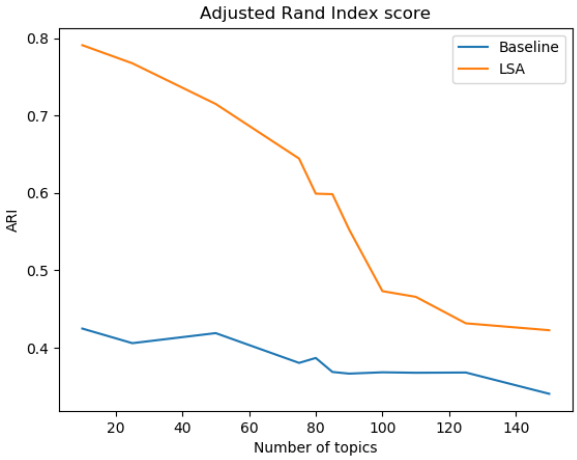
4 Evaluation

To evaluate all the models I used Rand index. Models comparison:



LSA and agglomerative clustering have similar accuracy, but even though agglomerative clustering has a little bit better accuracy, it is limited by its complexity, when LSA is more lightweight and allows detecting number of topics and representative tokens for each topic.

All of the models suffer from reduction of accuracy with higher number of topics in the tweets batch:


Adjusted Rand Index score
Baseline
LSA
ARI
0.8
0.7
0.6
0.5
0.4
20
40
60
80
100
120
140
Number of topics