Eindhoven University of Technology

2IMW15  
Web information retrieval and Data Mining

**Guliz Cosan Mutlu - 1034515**

1. General Approach

As rumors will be investigated based on topics of tweets, it is necessary to first find out tweets that are related to same topic. Discovery of each tweet’s topic is essential since further rumor analysis will be performed by analyzing all tweets that has same topic. For this purpose, topic classification, which is the most important part of the overall project, is performed using different approaches. To classify topics, the tasks that are explained in detail later in this document are implemented.

Before starting to process data, a set of operations are done for text normalization.

* Only English words are considered.
* Only certain characters (ASCII letters, digits etc.) are accepted.
* Single words that are smaller than 3 words are removed.
* Stemming is applied by using nltk.stem library (for supervised learning only).

1. TF-IDF
   1. **Motivation**

In order to work with text data for topic classification, a transformation to numerical vectors should be done. This is achieved by using TF-IDF.

* 1. **Description and Goal**

TF-IDF is a method of downscaling. This combined statistic has two different components that are multiplied. These components are namely TF (Term Frequency) and IDF (Inverse Document Frequency). This method is useful to eliminate over-emphasized words. TF is a simple measurement of the frequency of a word in a document. As relative frequency would give more insightful results as many documents will be examined, absolute frequency of occurrence of a specific word is divided by the total number of all words. A vector is generated by applying this method to all words in a document. Yet, it is necessary to multiply TF vector with another measure called IDF to eliminate the misleading TF vector of words that are used too frequently among all the documents such as “an”, “of”, “the”, “but” etc. The idea behind IDF (Inverse Document Frequency) is to raise the importance of less frequent words and to decrease the importance of too frequent words. Similarly, IDF is built as a vector where each component of this vector illustrates the measure of occurrence of each word (within specific input text) among all the documents [1].

Based on the results of this calculation, I infer the most frequent words in each document and cluster documents that has similar topics.

* 1. **Implementation**

Above goal is achieved by using sklearn and nltk libraries. Sklearn library has TfidfVectorizer utility which calculates TF-IDF vectors.

* 1. **Results**

This task enables us to see most frequent words in each topic cluster. TF-IDF calculations are not shown as values but instead, the program helps the classification approach to show top 10 most frequent words in the document.

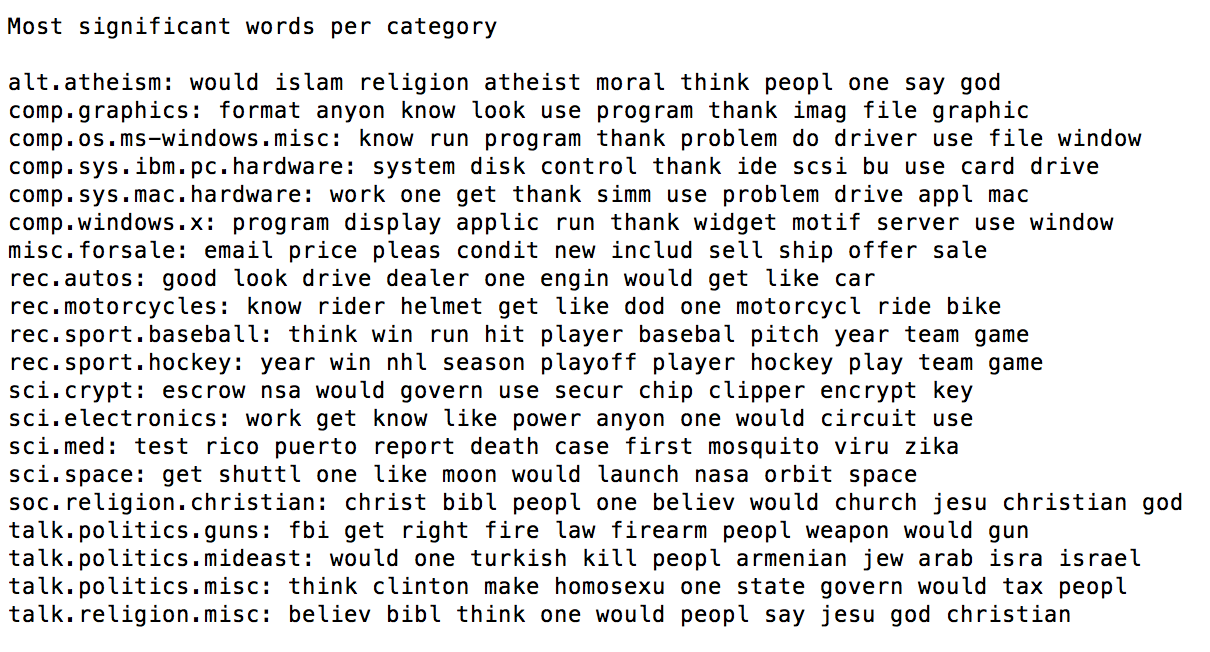


Figure 1: Most Significant Words per Category for 20NewsGroup and ZikaVirus datasets

1. Topic Classifier

As I have two different data, one of them is labelled and the other is not, I performed both supervised and unsupervised learning approaches to implement topic classification task. These two learning methods are explained in detail below:

**3.1 Topic Classification with Supervised Learning Approach**

3.1.1. Motivation

The ultimate motivation of this task is to predict topic of an input tweet per predetermined topics which are the labels of 20 newsgroups data.

3.1.2. Approach

The classification is implemented with 5 different supervised learning methods that are available in sklearn library of Python. These methods are:

* Multinomial Naïve Bayes Classifier[2]
* Bernoulli Naïve Bayes Classifier[2]
* Stochastic Gradient Descent (SGD) Classifier
* Linear Support Vector Classifier (SVC)
* Ridge Classifier

The performances of above classifiers are compared and it is concluded that Multinomial Naïve Bayes Classifier is found to be the most accurate. Basically, this approach calculates the probability distribution of input tweet belonging to each of predetermined topics and then assigns the tweet to the topic that has the highest probability.

3.1.3. Data Exploration

In this part, supervised learning algorithms are used as mentioned before. So, in order to obtain more accurate results and achieve the goal, a well labeled data is needed. At first the data that is provided with name ‘dataset\_2016’ (4200 tweets about Zika Virus in tweets.csv was processed) is considered but it was insufficient individually since it only focuses on one label. Therefore, I have also used ’20 newsgroups’ dataset which is freely available on internet[3]. It is a collection of about 20,000 newsgroup documents with the following categories:

* alt.atheism
* comp.graphics
* comp.os.ms-windows.misc
* comp.sys.ibm.pc.hardware
* comp.sys.mac.hardware
* comp.windows.x
* misc.forsale
* rec.autos
* rec.motorcycles
* rec.sport.baseball
* rec.sport.hockey
* sci.crypt
* sci.electronics
* sci.med
* sci.space
* soc.religion.christian
* talk.politics.guns
* talk.politics.mideast
* talk.politics.misc
* talk.religion.misc

As the names of these categories clearly indicate the context itself, I decided to use same names for the topic labels. The data that is provided is related to Zika Virus, therefore its content is also labelled to ‘sci.med’ which is the label related to health.

3.1.4. Implementation

For the implementation of supervised learning algorithms; nltk, pickle and sklearn libraries are utilized. As two different data sources are used, they are merged as the first step. The merged data is shuffled and trained by using 30% of the whole data as test data and the rest as training data.

As mentioned before, 5 different classifiers are implemented and tested. The performances are shown in Results & Evaluation part.

3.1.5. Results & Evaluation

The performance results of 5 different classifiers is shown below:

|  |  |
| --- | --- |
| **Classifier** | **Accuracy** |
| Multinomial Naïve Bayes Classifier | 0.7063 |
| Bernoulli Naïve Bayes Classifier | 0.5636 |
| Stochastic Gradient Descent (SGD) Classifier | 0.7046 |
| Linear Support Vector Classifier (SVC) | 0.6946 |
| Ridge Classifier | 0.7014 |

As stated before, Multinomial Naïve Bayes Classifier has the best performance. Here, results of Multinomial Naïve Bayes Classifier will be evaluated in detail. In order to show its performance, a confusion matrix can be generated as a visual illustration and also as number matrix by using the program.

Below, the visual illustration and the number version of confusion matrix is shown respectively:

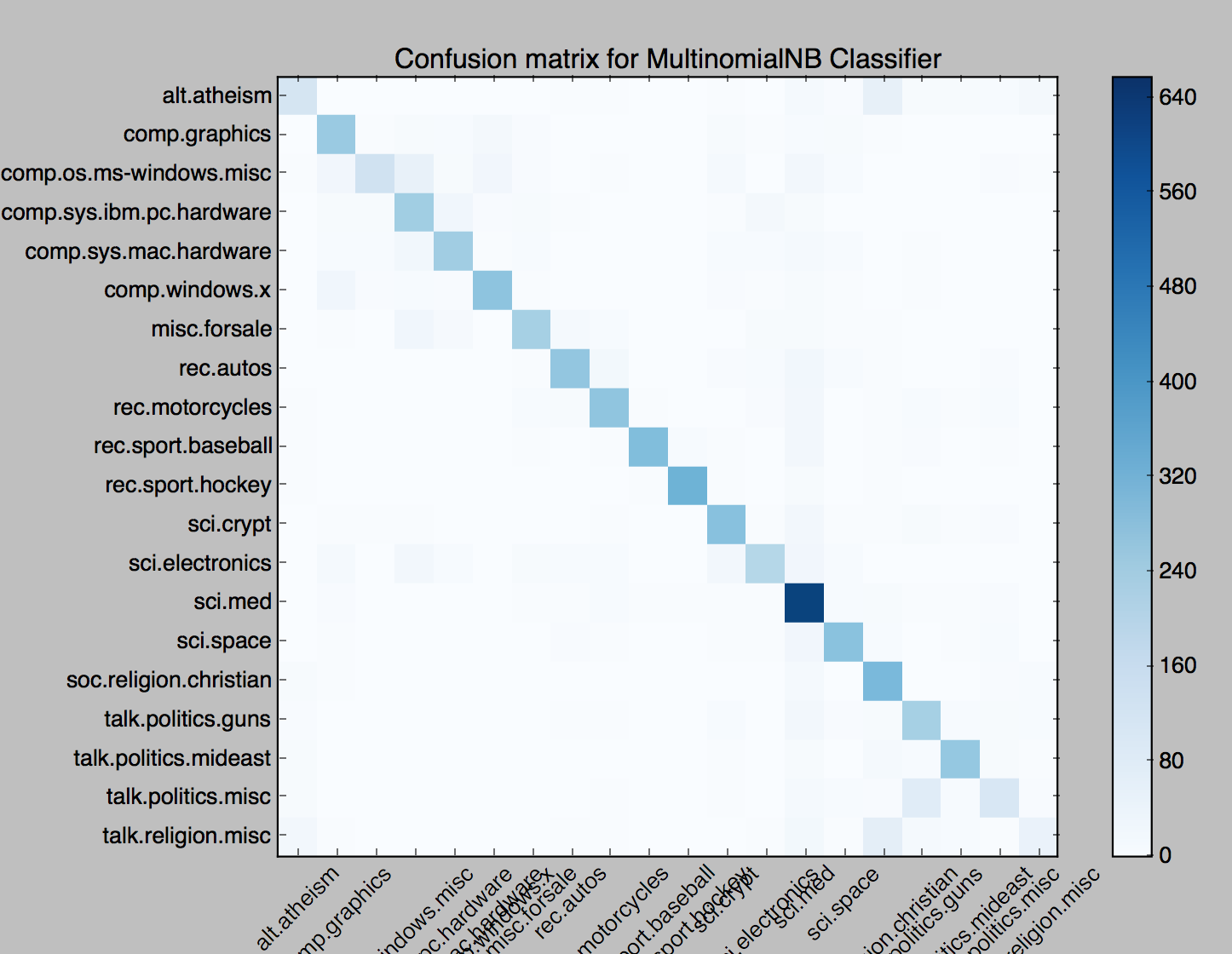


Figure 2: Visual Illustration of Confusion Matrix based on results of Multinomial Naïve Bayes Classifier

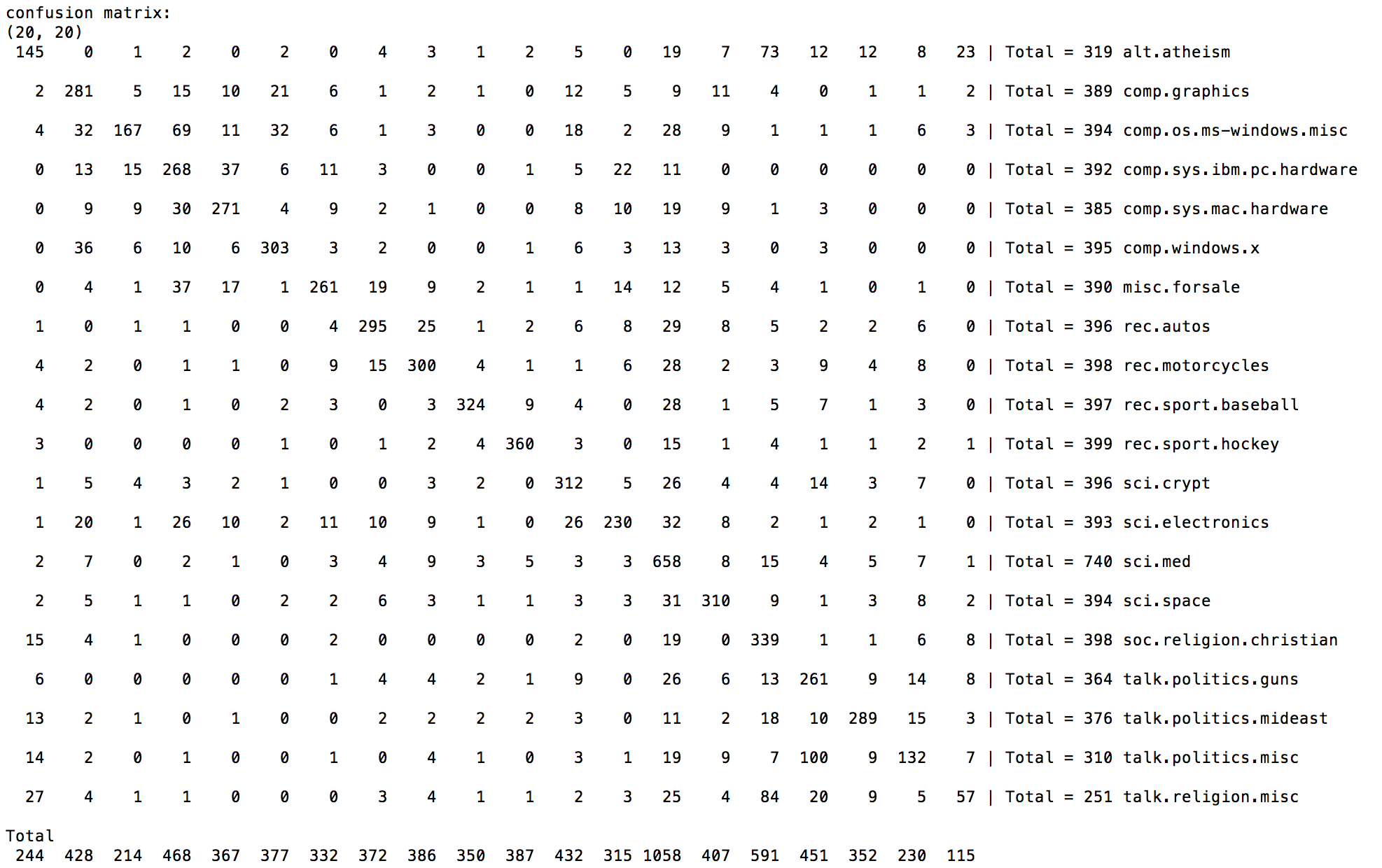


Figure 3: Confusion Matrix based on results of Multinomial Naïve Bayes Classifier

**3.2 Topic Classification with Unsupervised Learning Approach**

The data in hand, which is around 20.42 GB, is crawled from Twitter already on first quartile. As the data is too large for processing, a preprocessing is applied. The data is then used for topic extraction as an option. As the general task was related to topic classification from the beginning, this part is also implemented.

3.2.1. Motivation

The main aim is to extract topic clusters from unlabeled data via unsupervised learning techniques.

3.2.2. Approach

In order to extract topics of the corpus of documents (in our case they are tweets), Non-negative Matrix Factorization and Latent Dirichlet Allocation are applied.

These methods are:

* Latent Dirichlet Allocation (LDA): This algorithm assumes that each document includes a set of different topics and groups the documents accordingly[4].
* Non-negative Matrix Factorization (NMF): This algorithm clusters documents that has same features by data dimension reduction.

3.2.3 Implementation

These two algorithms are implemented by using sklearn library of Python. The output is a list of topics, each represented as a list of terms.

3.2.3. Results

The unlabeled data is grouped into 20 topics and most important 10 words of the topics are shown as output. As the data does not have an already clustered version, it is unfortunately not possible to test the performance of the results.

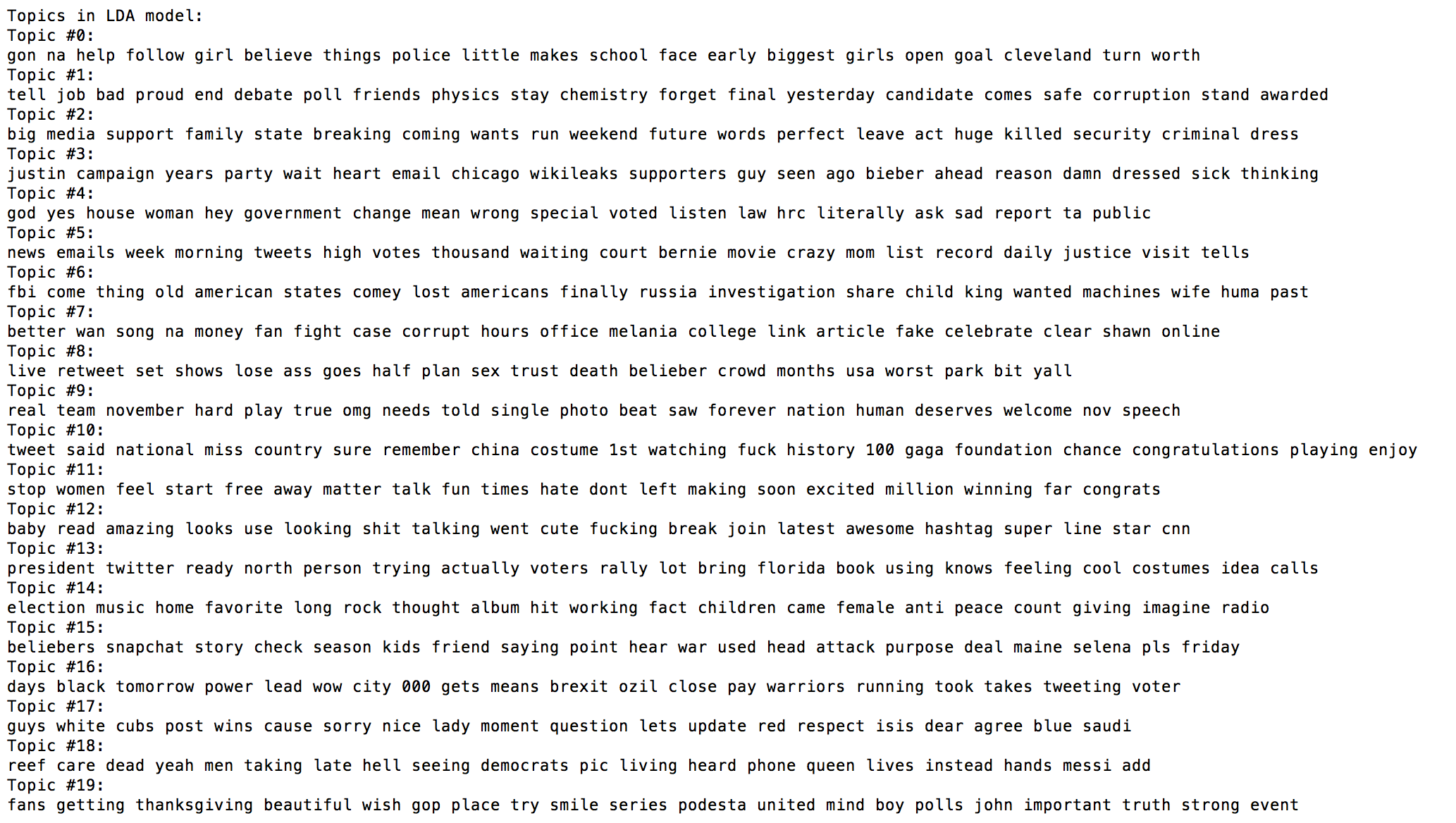


Figure 4: Extracted Topics in LDA Mode

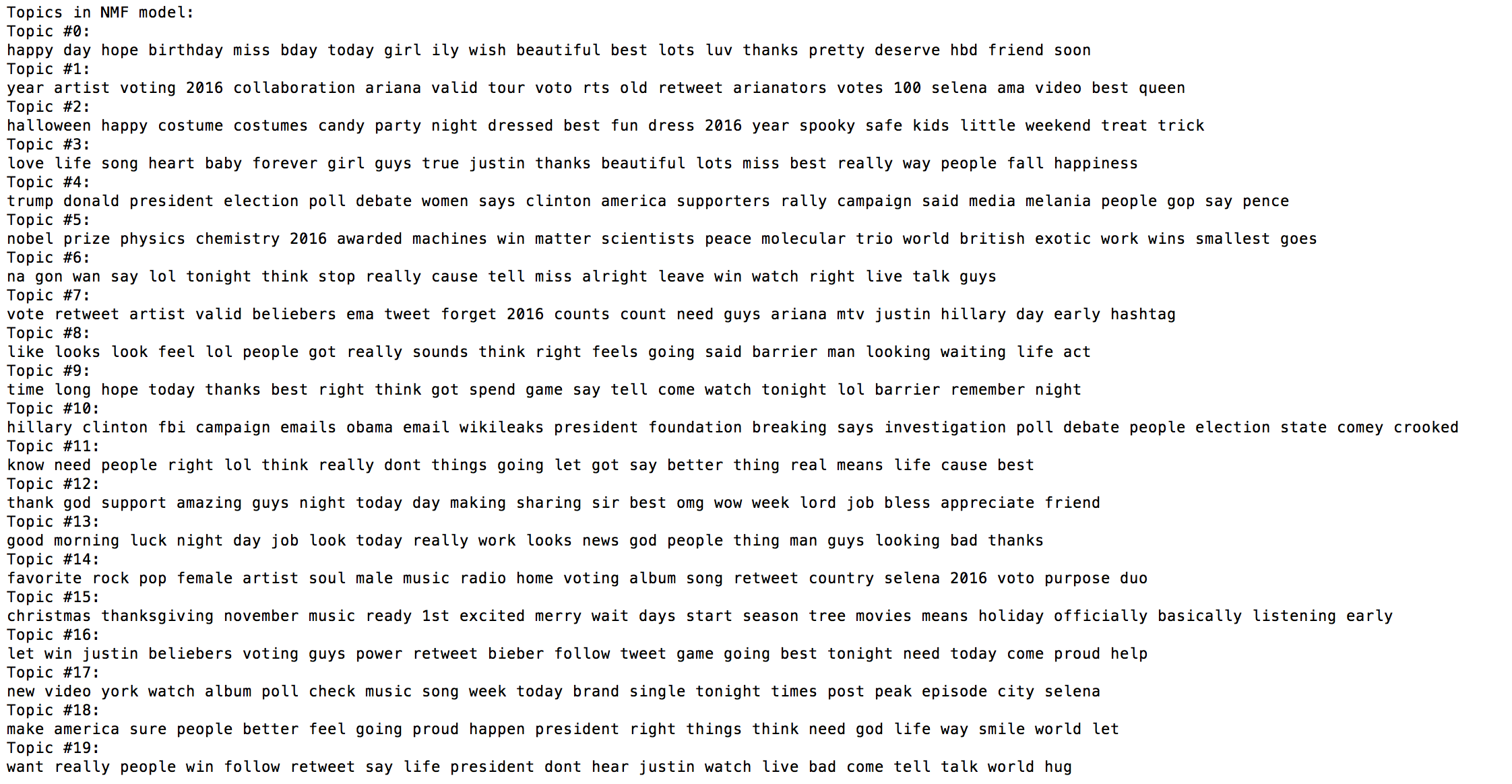


Figure 5: Extracted Topics in NMF Model

# Bibliography

1. Ramos, J. (n.d.). *Using TF-IDF to Determine Word Relevance in Document Queries.* Piscataway, NJ: Department of Computer Science, Rutgers University.

2. *scikit-learn*. (2010). Retrieved 01 11, 2017, from http://scikit-learn.org/stable/datasets/twenty\_newsgroups.html

3. *Naive Bayes text classification.* (2008). Cambridge University Press.

4. David M. Blei, A. Y. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*.