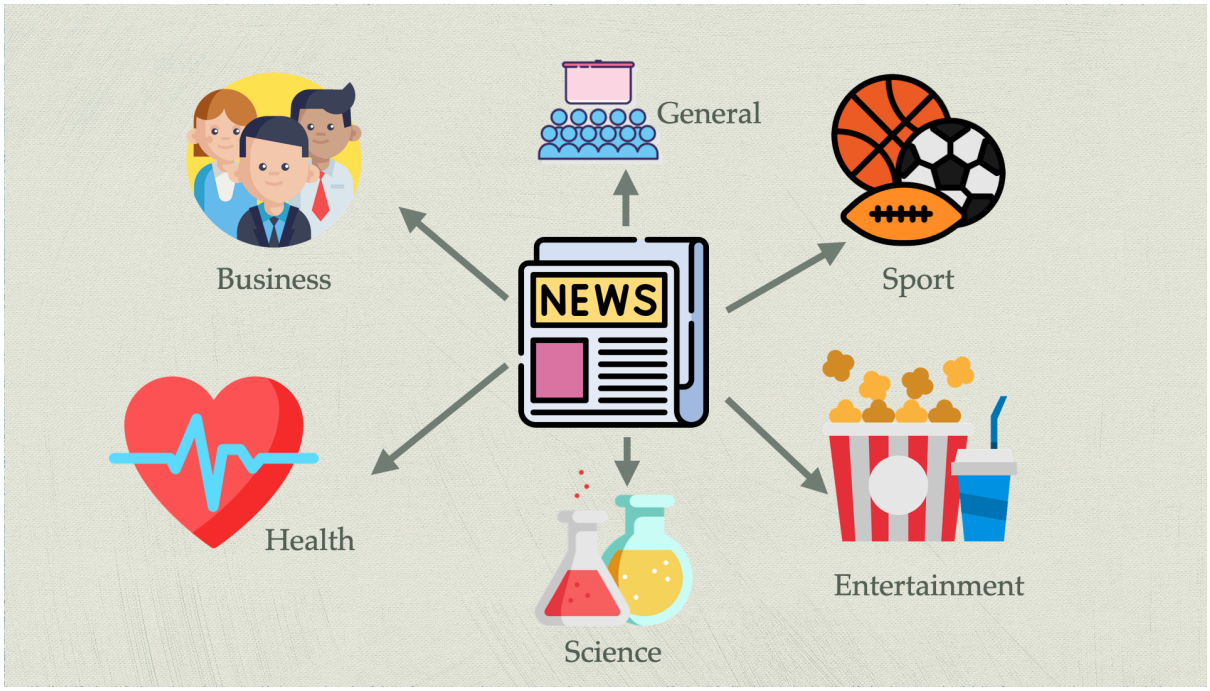
# News Classification

Yu-Chieh Wang 04/01/2021


It is a system which loads daily news articles in 6 categories automatically.



# Data mining

Through the news API, it is able to load real-world data by JSON format.

```
source : {'id': None, 'name': 'CNBC'}
author : Saheli Roy Choudhury
title : Papua New Guinea's coronavirus cases spike, health system 'at risk of collapsing' - CNBC
description : The situation in PNG is said to be dire and international organizations have warned of an imminent collapse of the
url : https://www.cnbc.com/2021/04/01/papua-new-guinea-coronavirus-cases-spike-health-system-on-the-brink.html
urlToImage : https://image.cnbcfm.com/api/v1/image/106861969-1617180959399-gettyimages-1231885263-AFP_96K4B7.jpeg?v=1617180707
publishedAt : 2021-04-01T01:15:00Z
content : Australian officials carry boxes containing some 8,000 initial doses of the AstraZeneca vaccine following their arrival
```

The system aggregates the data obtained daily into a json file to facilitate subsequent calls.

```
Loading.. 2021-03-31 13:44:34
Get news articles from newsapi.NewsApiClient in 6 categories
Articles: {'science': 13, 'general': 34, 'health': 24, 'business': 46, 'entertainment': 46, 'sports': 49}
Finish loading -> machine learning is available now.
```

# Data Preprocessing

Due to the limited amount of daily news, after accumulating a large amount of data, the data must be processed before implement machine learning on it.

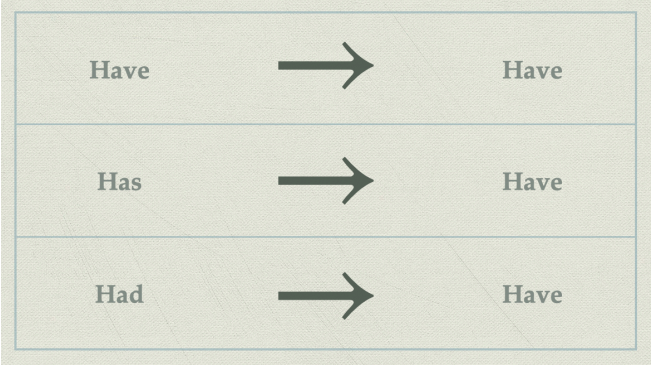
In the next step, in order to use the keywords appearing in the article to judge the classification of the article, the content of the article must be handled carefully.

Here, it uses NLTK library to help lemma words.

Separate each word from the sentence of the article, check out the type of each word, restore the word to a simple form, and then replace the word in the sentence.

For example,

```python
def get_wordnet_pos(tag):
    if tag.startswith('J'):
        return wordnet.ADJ
    elif tag.startswith('V'):
        return wordnet.VERB
    elif tag.startswith('N'):
        return wordnet.NOUN
    elif tag.startswith('R'):
        return wordnet.ADV
    else:
        return None
```



Before v.s. After

| |
|---|
| By that time, health had been damaged. |
| By that time, health have be damage. |

## Data Modeling

In this project, we consider supervised learning and deep learning together.

| Supervised Learning | Deep Learning |
|---|---|
| Naive Bayes | CNN |
| Linear Regression | RNN |
| SVM | LSTM |

According to the algorithms of deep learning, which is used to calculate continuous data, only the supervised learning model will use the processed data, while the deep learning model will use the data that has not been lemma processed.

More over, the Naive Bayes model is able to consider unigrams and bigrams at the same time.

In the following demo, the experiment will focus on Naive Bayes in training and testing steps.

# Data Training

Since the amount of data in each catalog is very different, a random selection function is used to obtain training and test data respectively.

```
# make training data and testing data
from sklearn.model_selection import train_test_split
trainX, testX, trainY, testY = train_test_split(df[:]['content'], df[:]['label'],
                                                test_size=0.25, random_state=1000)
return (trainX, trainY), (testX, testY)
```

# Data Testing

Here, it shows the results of data collection and training in 7 days.

## Day 1

Collecting Data

```
Loading.. 2021-03-29 13:44:21
Get news articles from newsapi.NewsApiClient in 6 categories
Articles: {'science': 8, 'general': 37, 'health': 20, 'business': 50, 'entertainment': 45, 'sports': 50}
Finish loading -> machine learning is available now.
```

Training & Testing

```
Articles: {'science': 8, 'general': 37, 'health': 20, 'business': 50, 'entertainment': 45, 'sports': 50}
Get lemma data successfully! training size: 157 testing size: 53
Final accuracy: 0.2830188679245283
```

## Day 2

Collecting Data

```
Loading.. 2021-03-30 13:44:27
Get news articles from newsapi.NewsApiClient in 6 categories
Articles: {'science': 8, 'general': 33, 'health': 31, 'business': 50, 'entertainment': 34, 'sports': 50}
Finish loading -> machine learning is available now.
```

Training & Testing

```
Articles: {'science': 16, 'general': 70, 'health': 51, 'business': 100, 'entertainment': 79, 'sports': 100}
Get lemma data successfully! training size: 312 testing size: 104
Final accuracy: 0.38461538461538464
```

## Day 3

Collecting Data

```
Loading.. 2021-03-31 13:44:34
Get news articles from newsapi.NewsApiClient in 6 categories
Articles: {'science': 13, 'general': 34, 'health': 24, 'business': 46, 'entertainment': 46, 'sports': 49}
Finish loading -> machine learning is available now.
```

Training & Testing

```
Get news articles from newsapi.NewsApiClient in 6 categories
Articles: {'science': 29, 'general': 104, 'health': 75, 'business': 146, 'entertainment': 125, 'sports': 149}
Get lemma data successfully! training size: 471 testing size: 157
Final accuracy: 0.4840764331210191
```

## Day 4

## Day 5

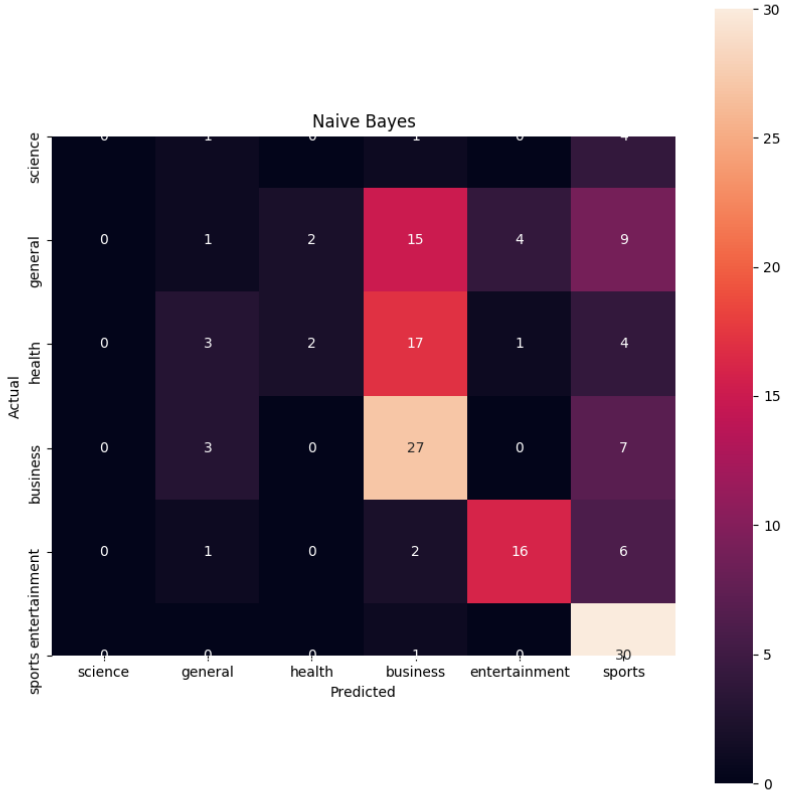
## Day 6

## Day 7

# Evaluation

From the previous results of Naive Bayes, it is easy to find out that the percentage of accuracy is increasing by 10% every day due to the increasing training data.

In addition to checking out the accuracy results, a confused matrix is able to help understanding the classification, either.



According to the matrix, obviously, too little scientific data will cause classification difficulties. Moreover, many health and general articles are misclassified as business categories, which may also be due to the lack of data. Besides that, the system works very well in categorizing entertainment and sports articles. In particular, the classification success rate of entertainment articles is as high as 76%.

The current results, with a training volume of 628 articles, the classification effect of the system is good. If more data can be collected in the future, the success rate of classification must be greatly improved.