Auto News Categorization - Practical ML-based Mobile Application

KAO Shiu Hong, Spring 2021

Abstract

With the rapid development of modern technology, people often need to deal with much more information nowadays. In this report, we aim to build an app helping people categorizing articles, such as news, faster. We firstly collected hundreds of sample news on the Internet, and then used a machine learning algorithm to do multi-class classification, dividing these samples into three categories - business, engineering, and science. Finally, we built an Android app on Android Studio, uploading articles onto this app, and used the criteria we got in the machine learning to achieve our target. However, we found that the error rates in the training set and the testing set differ a lot. This may be caused by two reasons, resource collection and characteristic ignorance. We hope these problems can be fixed by considering more possible issues which may happen in the algorithm in the future.

1. Introduction

The process can be basically divided into three main parts, specifically, data collection, machine learning, and app development.

In data collection, we used Python module, BeautifulSoup, as our tool, collecting hundreds of recent news articles from online available websites, and saving them in different folders on the local computer based on different categories.

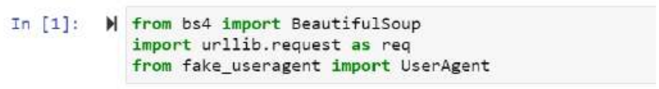
In machine learning, we randomly selected some collected samples as our dictionaries, finding the commonly appeared words as keywords to build our machine learning vector. After that, one-versus-all (OVA) method for multiclass classification and stochastic logistic regression were selected as the main algorithms. We kept updating and modifying the weight of each keyword so as to make the best standard for classification.

Finally, we copied the standard we obtained from machine learning to Android Studio to build our app. The design of the app is basically in two parts, file choosing and decision making. To read the mobile storage, we would request the permission from the user. Afterwards, the app would read the text in the file chosen and further gave the possible values corresponding to each category.

2. Data Collection

We used Jupyter Notebook as the IDE of Python. Three folders for different types of news were created with a .ipynb file inside. We take the type of science news for example in this report as the remaining were the same.

For the first cell of the file, the following content were compiled and run:

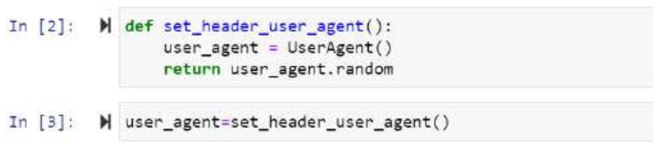


```
In [1]:  from bs4 import BeautifulSoup
         import urllib.request as req
         from fake_useragent import UserAgent
```

figure 2.1

In the web scraping, BeautifulSoup was the main module we used in this project, and it is in the library of bs4. Also we imported urllib to help us send the request to our target websites, and fake_useragent for producing a user agent while sending the request.

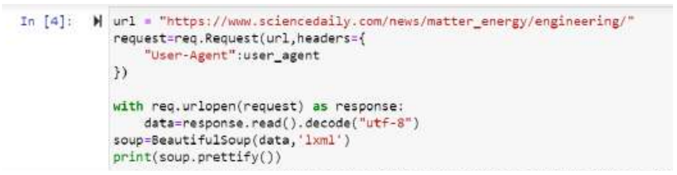
Then we used the following code to generate a fake user agent. One can also use his/her own user agent.



```
In [2]:  def set_header_user_agent():
             user_agent = UserAgent()
             return user_agent.random

In [3]:  user_agent=set_header_user_agent()
```

figure 2.2

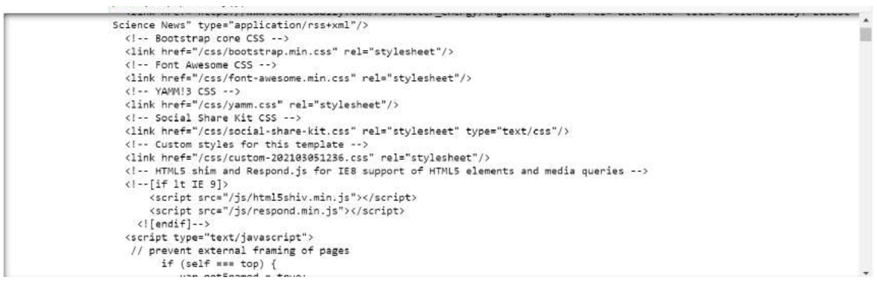
After that, we can have the following code to open the main page of a news website and save it in a BeautifulSoup object (here we take ScienceDaily for example.)



```
In [4]:  url = "https://www.sciencedaily.com/news/matter_energy/engineering/"
         request=req.Request(url,headers={
             "User-Agent":user_agent
         })

         with req.urlopen(request) as response:
             data=response.read().decode("utf-8")
         soup=BeautifulSoup(data,'lxml')
         print(soup.prettify())
```

figure 2.3

This code also helped us to view the html content of the website, like figure 2.4:

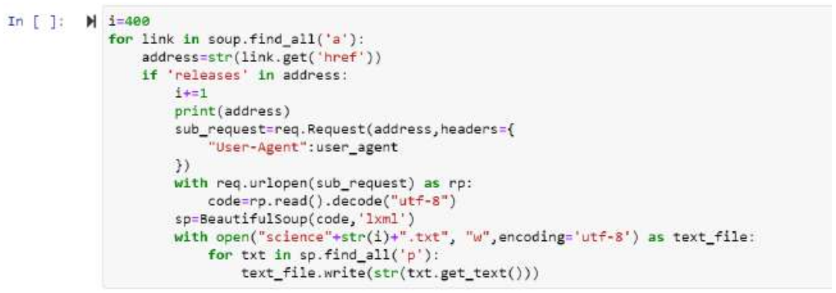


```
Science News" type="application/rss+xml"/>
  <!-- Bootstrap core CSS -->
  <link href="/css/bootstrap.min.css" rel="stylesheet"/>
  <!-- Font Awesome CSS -->
  <link href="/css/font-awesome.min.css" rel="stylesheet"/>
  <!-- YAMM!3 CSS -->
  <link href="/css/yamm.css" rel="stylesheet"/>
  <!-- Social Share Kit CSS -->
  <link href="/css/social-share-kit.css" rel="stylesheet" type="text/css"/>
  <!-- Custom styles for this template -->
  <link href="/css/custom-202103051236.css" rel="stylesheet"/>
  <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
  <!--[if lt IE 9]>
      <script src="/js/html5shiv.min.js"></script>
      <script src="/js/respond.min.js"></script>
    <![endif]-->
  <script type="text/javascript">
   // prevent external framing of pages
        if (self === top) {
```

figure 2.4

If we had not included the user agent within urlopen; that is, we only write req.urlopen(url) instead of req.urlopen(request) in the previous example, a HTTP Error 403: Forbidden will be generated.

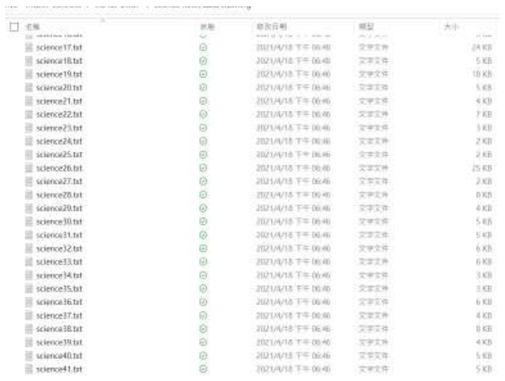
Next, we tried to observe the pattern of the webpage html to find a faster way to get the news on that website. Taking ScienceDaily as an example, we then found that the string ‘release’ frequently appears among different web addresses of news, so we writed the following code:



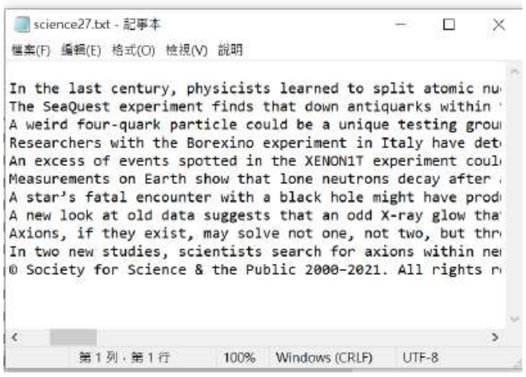
```
In [ ]:  i=400
         for link in soup.find_all('a'):
             address=str(link.get('href'))
             if 'releases' in address:
                 i+=1
                 print(address)
                 sub_request=req.Request(address,headers={
                     "User-Agent":user_agent
                 })
                 with req.urlopen(sub_request) as rp:
                     code=rp.read().decode("utf-8")
                 sp=BeautifulSoup(code,'lxml')
                 with open("science"+str(i)+".txt", "w",encoding='utf-8') as text_file:
                     for txt in sp.find_all('p'):
                         text_file.write(str(txt.get_text()))
```

figure 2.5

One can also make the code after print(address) to be comments, and check whether the target websites were what we wanted. Then the code in figure 2.5 could help us write the text content of these releasing news to our local folder as well as name them with different categories and indices. After all, we got the following result:



名稱	狀態	修改日期	類型	大小
science17.txt	✓	2021/4/18 下午 06:46	文字文件	24 KB
science18.txt	✓	2021/4/18 下午 06:46	文字文件	5 KB
science19.txt	✓	2021/4/18 下午 06:46	文字文件	18 KB
science20.txt	✓	2021/4/18 下午 06:46	文字文件	5 KB
science21.txt	✓	2021/4/18 下午 06:46	文字文件	4 KB
science22.txt	✓	2021/4/18 下午 06:46	文字文件	7 KB
science23.txt	✓	2021/4/18 下午 06:46	文字文件	3 KB
science24.txt	✓	2021/4/18 下午 06:46	文字文件	2 KB
science25.txt	✓	2021/4/18 下午 06:46	文字文件	2 KB
science26.txt	✓	2021/4/18 下午 06:46	文字文件	25 KB
science27.txt	✓	2021/4/18 下午 06:46	文字文件	2 KB
science28.txt	✓	2021/4/18 下午 06:46	文字文件	0 KB
science29.txt	✓	2021/4/18 下午 06:46	文字文件	4 KB
science30.txt	✓	2021/4/18 下午 06:46	文字文件	5 KB
science31.txt	✓	2021/4/18 下午 06:46	文字文件	5 KB
science32.txt	✓	2021/4/18 下午 06:46	文字文件	6 KB
science33.txt	✓	2021/4/18 下午 06:46	文字文件	6 KB
science34.txt	✓	2021/4/18 下午 06:46	文字文件	3 KB
science35.txt	✓	2021/4/18 下午 06:46	文字文件	3 KB
science36.txt	✓	2021/4/18 下午 06:46	文字文件	6 KB
science37.txt	✓	2021/4/18 下午 06:46	文字文件	4 KB
science38.txt	✓	2021/4/18 下午 06:46	文字文件	8 KB
science39.txt	✓	2021/4/18 下午 06:46	文字文件	4 KB
science40.txt	✓	2021/4/18 下午 06:46	文字文件	5 KB
science41.txt	✓	2021/4/18 下午 06:46	文字文件	5 KB

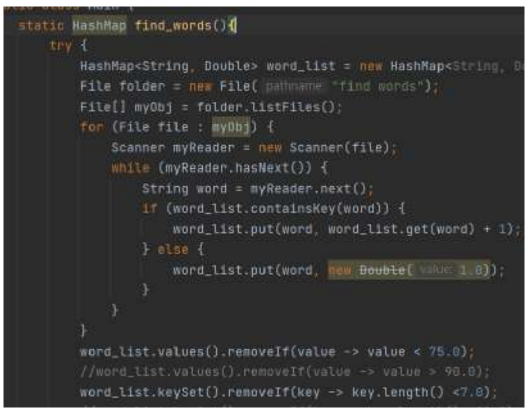


```
In the last century, physicists learned to split atomic nu
The SeaQuest experiment finds that down antiquarks within
A weird four-quark particle could be a unique testing grou
Researchers with the Borexino experiment in Italy have det
An excess of events spotted in the XENON1T experiment coul
Measurements on Earth show that lone neutrons decay after
A star’s fatal encounter with a black hole might have prod
A new look at old data suggests that an odd X-ray glow tha
Axions, if they exist, may solve not one, not two, but thr
In two new studies, scientists search for axions within ne
© Society for Science & the Public 2000-2021. All rights r
```

figure 2.6

3. ML Algorithm

After data crawling, we used machine learning techniques to generate a standard for news categorization, specifically, stochastic regression. To start with this, we selected Java as our programming tools and IntelliJ as the IDE. At the beginning, we randomly selected some samples to find the common-appeared keywords as vector characteristics. To achieve this, a method find_words is written in:

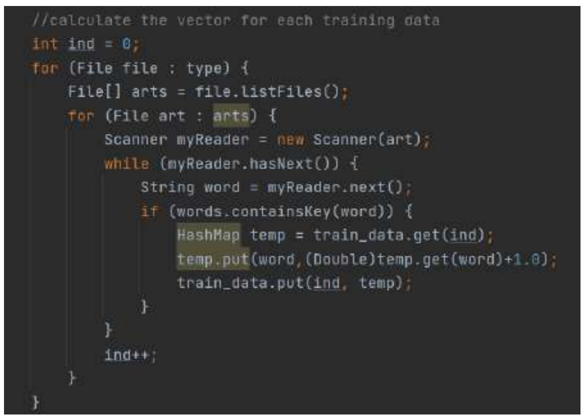


```java
static HashMap find_words(){
    try {
        HashMap<String, Double> word_list = new HashMap<String, D
        File folder = new File( pathname: "find words");
        File[] myObj = folder.listFiles();
        for (File file : myObj) {
            Scanner myReader = new Scanner(file);
            while (myReader.hasNext()) {
                String word = myReader.next();
                if (word_list.containsKey(word)) {
                    word_list.put(word, word_list.get(word) + 1);
                } else {
                    word_list.put(word, new Double( value: 1.0));
                }
            }
        }
        word_list.values().removeIf(value -> value < 75.0);
        //word_list.values().removeIf(value -> value > 90.0);
        word_list.keySet().removeIf(key -> key.length() <7.0);
```

figure 3.1

We selected some frequently appeared words, putting them into a hash map. Afterwards, short words were deleted in order to prevent some meaningless or useless words for us to make the algorithm.

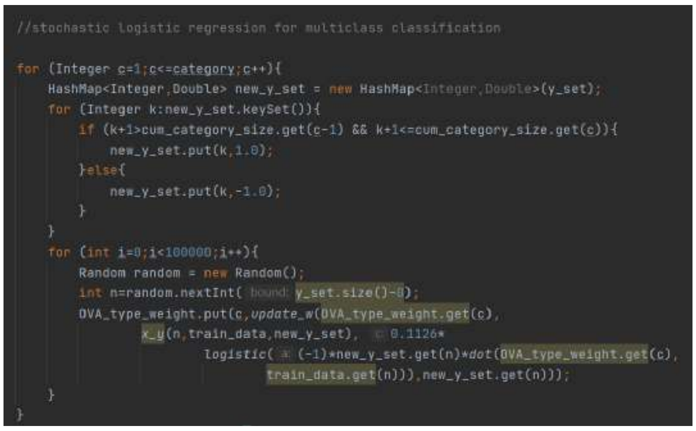
Next, we return the hash map of keywords to the main method, constructing a hashmap storing sample indices associated with it's keyword appearance number (another hashmap, mapping keywords to it's number).



```java
//calculate the vector for each training data
int ind = 0;
for (File file : type) {
    File[] arts = file.listFiles();
    for (File art : arts) {
        Scanner myReader = new Scanner(art);
        while (myReader.hasNext()) {
            String word = myReader.next();
            if (words.containsKey(word)) {
                HashMap temp = train_data.get(ind);
                temp.put(word,(Double)temp.get(word)+1.0);
                train_data.put(ind, temp);
            }
        }
        ind++;
    }
}
```

figure 3.2

Also, we set a hashmap to store different weight vectors for one-versus-all (OVA) multiclass classification, and then train each weight vectors with associated training data with stochastic regression.



```java
//stochastic logistic regression for multiclass classification

for (Integer c=1;c<=category;c++){
    HashMap<Integer,Double> new_y_set = new HashMap<Integer,Double>(y_set);
    for (Integer k:new_y_set.keySet()){
        if (k+1>cum_category_size.get(c-1) && k+1<=cum_category_size.get(c)){
            new_y_set.put(k,1.0);
        }else{
            new_y_set.put(k,-1.0);
        }
    }
    for (int i=0;i<100000;i++){
        Random random = new Random();
        int n=random.nextInt( bound: y_set.size()-0);
        OVA_type_weight.put(c,update_w(OVA_type_weight.get(c),
                x_y(n,train_data,new_y_set),  c: 0.1126*
                        logistic( a: (-1)*new_y_set.get(n)*dot(OVA_type_weight.get(c),
                                train_data.get(n))),new_y_set.get(n)));
    }
}
```

figure 3.3

Consider x[n] be the characteristic vectors of each sample, where n is from 1 to N, and N is the total sample number. Let y[n] be the result of categorization. y[n] = 1 for weight t if x[n] x[n] is in group t; otherwise, y[n] = -1. The pseudo code of the algorithm is provided:

```
OVA Decomposition:
for k∈y_set:
        setting $D\_[k] = \{(x[n], y[n] = 2 * (y[n] - k) + 1)\}$for n=1:N
        use $D\_[k]$ in the stochastic logistic regression to update weight[k]
end
```

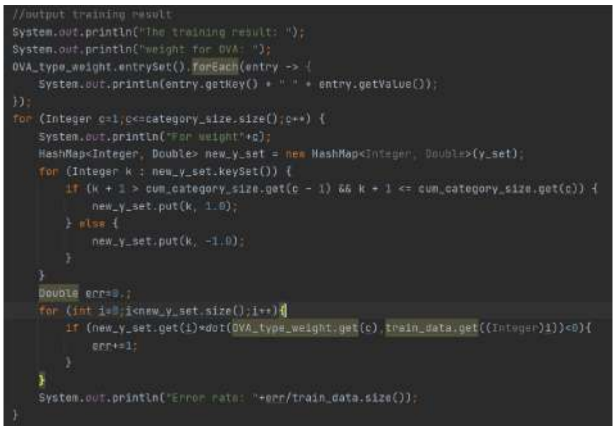
```
Stochastic Logistic Regression:
for $i = 0, 1, ...$:
        $w <- w + \eta * \theta(- y[n] * w^T x[n]) * (y[n]\, x[n])$
        (η is the learning rate)
end
```

```
Logistic function:
        $\theta(a) = 1/(1 + e^{(-a)})$
```

figure 3.4

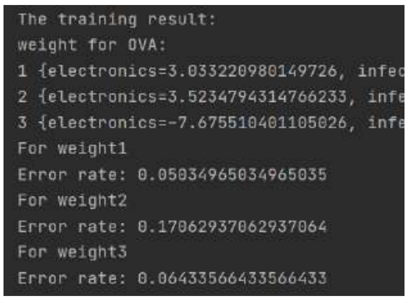
The time of logistic regression loop depends on the size of the training sample set.

Afterwards, we can output the training result:



```java
//output training result
System.out.println("The training result: ");
System.out.println("weight for OVA: ");
OVA_type_weight.entrySet().forEach(entry -> {
    System.out.println(entry.getKey() + " " + entry.getValue());
});
for (Integer c=1;c<=category_size.size();c++) {
    System.out.println("For weight"+c);
    HashMap<Integer, Double> new_y_set = new HashMap<Integer, Double>(y_set);
    for (Integer k : new_y_set.keySet()) {
        if (k + 1 > cum_category_size.get(c - 1) && k + 1 <= cum_category_size.get(c)) {
            new_y_set.put(k, 1.0);
        } else {
            new_y_set.put(k, -1.0);
        }
    }
    Double err=0.;
    for (int i=0;i<new_y_set.size();i++){
        if (new_y_set.get(i)*dot(OVA_type_weight.get(c),train_data.get((Integer)i))<0){
            err+=1;
        }
    }
    System.out.println("Error rate: "+err/train_data.size());
}
```

figure 3.5



```
The training result:
weight for OVA:
1 {electronics=3.033220980149726, infec
2 {electronics=3.5234794314766233, infe
3 {electronics=-7.675510401105026, infe
For weight1
Error rate: 0.05034965034965035
For weight2
Error rate: 0.17062937062937064
For weight3
Error rate: 0.06433566433566433
```

figure 3.6

For a testing sample having characteristic vector x[i], we can then calculate $w[t]^T x[i]$for different weight $t = 1, 2, ...$, and select $arg_{t=1,2,3,...} max\, w[t]^T x[i]$as a best estimated category of x[i].

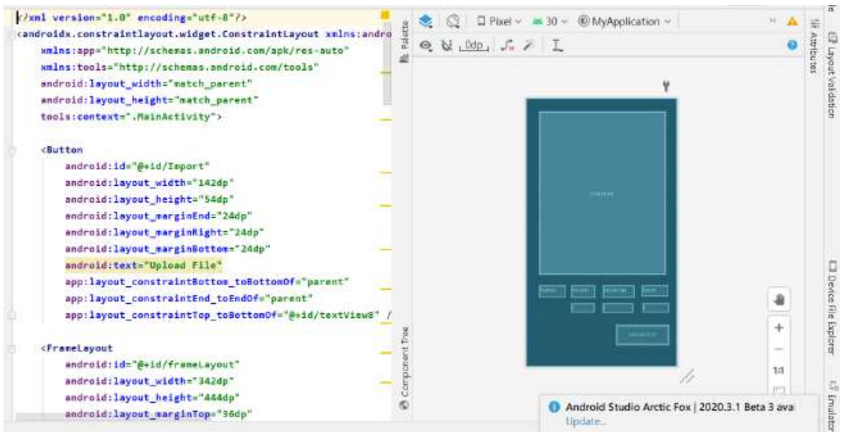
4. Android App Development

Android Application was built in Java and Android Studio. We first open a new project and add the following line in AndroidManifest.xml to help us request the user permission, so we can upload our testing news article to the app.

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

figure 4.1

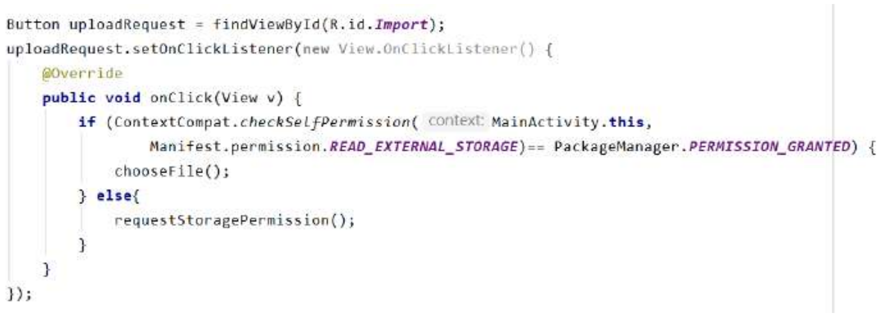
Then we open the activity_main.xml to draw the user interface as well as declare the associated methods to each view.



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:andro
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/Import"
        android:layout_width="142dp"
        android:layout_height="54dp"
        android:layout_marginEnd="24dp"
        android:layout_marginRight="24dp"
        android:layout_marginBottom="24dp"
        android:text="Upload File"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView8" /

    <FrameLayout
        android:id="@+id/frameLayout"
        android:layout_width="342dp"
        android:layout_height="444dp"
        android:layout_marginTop="36dp"
```

figure 4.2

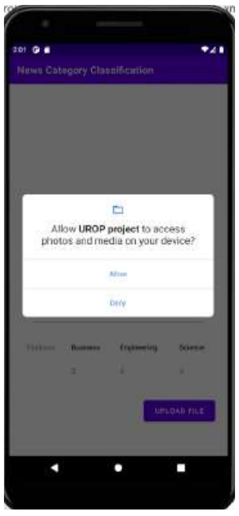
We put a content view at the top of the activity. This will show up the text when we upload a file to the app. Then we put three labels below to demonstrate the testing result $(w[t]^T x)$for $t = 1, 2, 3$. Finally, a button was placed at the bottom right corner of the activity. We can upload the file by clicking on this button. The associated method to this button is shown below:

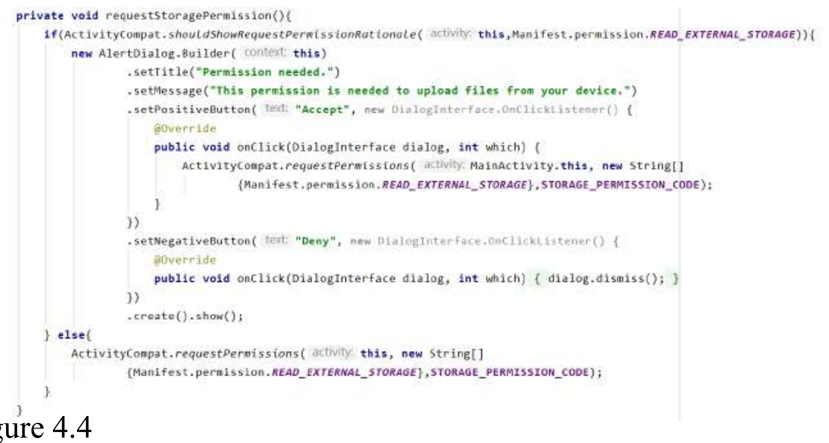


```
Button uploadRequest = findViewById(R.id.Import);
uploadRequest.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (ContextCompat.checkSelfPermission( context: MainActivity.this,
                Manifest.permission.READ_EXTERNAL_STORAGE)== PackageManager.PERMISSION_GRANTED) {
            chooseFile();
        } else{
            requestStoragePermission();
        }
    }
});
```

figure 4.3

This method calls the method chooseFile() if the permission has already been granted, otherwise a method named requestStoragePermission() will be called to ask for the permission. Let’s look at requestStoragePermission() first.


News Category Classification
Allow UROP project to access photos and media on your device?
Allow
Deny
Business
Engineering
Science
UPLOAD FILE

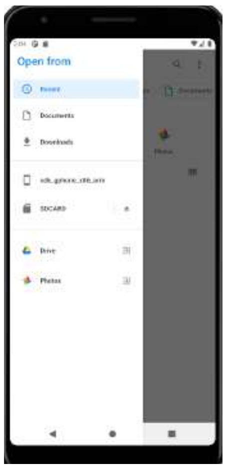


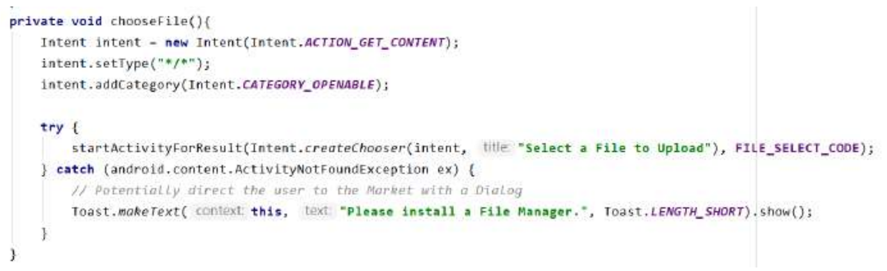
```
private void requestStoragePermission(){
    if(ActivityCompat.shouldShowRequestPermissionRationale( activity: this,Manifest.permission.READ_EXTERNAL_STORAGE)){
        new AlertDialog.Builder( context: this)
                .setTitle("Permission needed.")
                .setMessage("This permission is needed to upload files from your device.")
                .setPositiveButton( text: "Accept", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        ActivityCompat.requestPermissions( activity: MainActivity.this, new String[]
                                {Manifest.permission.READ_EXTERNAL_STORAGE},STORAGE_PERMISSION_CODE);
                    }
                })
                .setNegativeButton( text: "Deny", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) { dialog.dismiss(); }
                })
                .create().show();
    } else{
        ActivityCompat.requestPermissions( activity: this, new String[]
                {Manifest.permission.READ_EXTERNAL_STORAGE},STORAGE_PERMISSION_CODE);
    }
}
```

figure 4.4

This method shows a rationale for the user to approve the permission. If the user denies the permission, another rationale pops up and tells the user why the permission is needed.

Next, let’s look at the chooseFile() method.

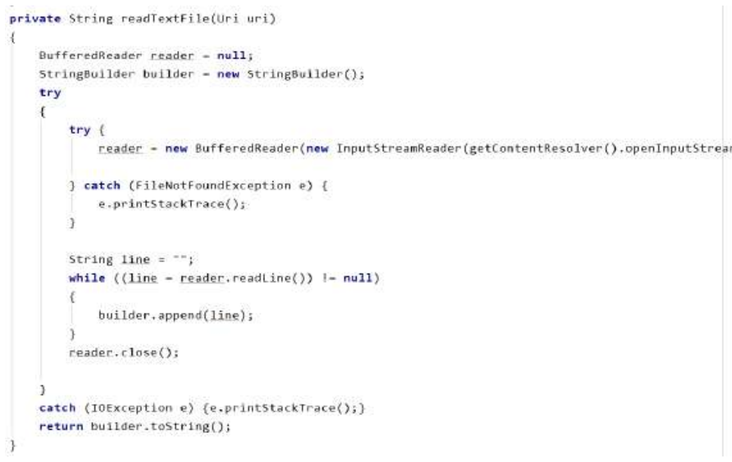




```
private void chooseFile(){
    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
    intent.setType("*/*");
    intent.addCategory(Intent.CATEGORY_OPENABLE);

    try {
        startActivityForResult(Intent.createChooser(intent, title: "Select a File to Upload"), FILE_SELECT_CODE);
    } catch (android.content.ActivityNotFoundException ex) {
        // Potentially direct the user to the Market with a Dialog
        Toast.makeText( context: this, text: "Please install a File Manager.", Toast.LENGTH_SHORT).show();
    }
}
```

figure 4.5

After uploading the file, we can then read the text in the file, and change the value of text view. The following method introduces how we can read the uploading file.

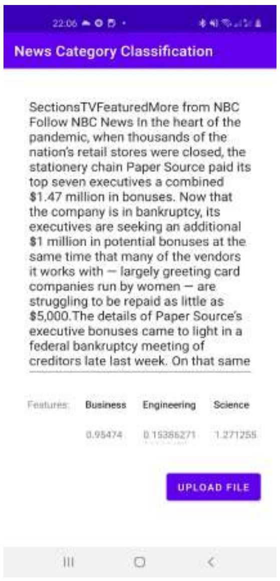


```
private String readTextFile(Uri uri)
{
    BufferedReader reader = null;
    StringBuilder builder = new StringBuilder();
    try
    {
        try {
            reader = new BufferedReader(new InputStreamReader(getContentResolver().openInputStream(uri)));

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

        String line = "";
        while ((line = reader.readLine()) != null)
        {
            builder.append(line);
        }
        reader.close();

    }
    catch (IOException e) {e.printStackTrace();}
    return builder.toString();
}
```

figure 4.6

Now, we can read the article we upload, and surely we can get its keyword vectors. We move the weight vectors of our training result from Intellij to Android Studio in the machine learning algorithm by copying the values. Afterwards, we multiply it with the current keyword vector one by one to determine the possibility in each category. Finally, show it on the main activity.

As the result of the example in figure 4.7, since the value for science category is the largest, we can therefore categorize this news into scientific type.



22:06

News Category Classification

SectionsTVFeaturedMore from NBC Follow NBC News In the heart of the pandemic, when thousands of the nation's retail stores were closed, the stationery chain Paper Source paid its top seven executives a combined $1.47 million in bonuses. Now that the company is in bankruptcy, its executives are seeking an additional $1 million in potential bonuses at the same time that many of the vendors it works with — largely greeting card companies run by women — are struggling to be repaid as little as $5,000.The details of Paper Source's executive bonuses came to light in a federal bankruptcy meeting of creditors late last week. On that same

Features:	Business	Engineering	Science
	0.95474	0.15386271	1.271255

figure 4.7

5. Conclusion

This app helps the user categorize the articles in a faster manner. However, there are also some problems we need to solve in the future as well.

First, the way we copied the machine learning result is just copying and pasting the keywords and their weight one-by-one. This will raise a problem if we have a larger sample set and a vector with a larger size.

Second, sometimes the testing result is not as we expected. This is mainly because we are dealing with news articles on the Internet. In most of the time, the news will not cover only one aspect but may consider the technological reasons and business results. Also, different websites may have a different standard to categorize an article, which means one may be put in scientific type on website A, but business type on another website.

Finally, during the keyword recognition, we ignored the letter of the alphabet and the punctuation. This may also increase the error rate of our module. In the future, some concepts in natural language processing are needed to build a better module in this application for sure.

Reference

1. Android Developers documentation - App fundamentals

   https://developer.android.com/guide/components/fundamentals

2. Android Developers documentation - Request app permission

   https://developer.android.com/training/permissions/requesting

3. Hsuen-tien Lin, National Taiwan University, Machine Learning Foundations

   https://www.csie.ntu.edu.tw/~htlin/course/mlfound18fall/