INTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Sentiment Analysis with Naïve Bayes

24/04/2018

ANA PAOLA NAVA VIVAS

Contenido

1. Introducción

El análisis de sentimientos o minería de opiniones, es el proceso de determinar si un pedazo de escrito es positivo, negativo o neutral. Para los fines de esta práctica, sólo evaluaremos si un tweet es agresivo o neutral.

Hoy en día es de suma importancia el análisis de sentimientos, existen negocios y organizaciones que buscan conocer la opinión de los consumidores acerca de productos que venden o servicios que ofrecen; personas que toman la decisión de comprar un determinado producto basándose en los "reviews"; o empresas que buscan llegar a su consumidor ideal.

Incluso hay gobiernos que pagan por dicha información sobre los usuario de la web, para saber el estado de ánimo de un país e influir en la gente.

El objetivo de la práctica consistía en clasificar correctamente tweets en agresivos o neutrales. Para eso utilicé el de Naïve Bayes que no es más que un clasificador probabilístico.

2. Desarrollo experimental

La práctica consistía en leer dos archivos de texto, uno con 7700 “tweets” en total, y otro con la información acerca de estos donde se tenía un 1 en caso de que el tweet fuera agresivo o un 0 en caso de que fuera neutral.

Proceso de adaptación:

Los “tweets” venían en orden cronológico, y como el estado de ánimo cambia a través del tiempo era necesario sortear lo tweets para proveer de un mejor entrenamiento al clasificador. Se utilizó un programa que mezclaba los dos archivos por igual, así el 0 o 1 del segundo archivo iba a seguir siendo el del tweet correspondiente, sólo que ahora no iba a estar ordenado cronológicamente.

Preprocesamiento del texto

Para el algoritmo de Naïve Bayes, era necesario recolectar cierta información acerca de los tweets, por ejemplo, la probabilidad de que un tweet fuera agresivo venía siendo el total de tweets agresivos entre el total de tweets.

El procedimiento consistió en leer línea por línea del archivo de entrenamiento y leer al mismo tiempo el archivo con la solución del entrenamiento para contar la frecuencia de tweets negativos, tweets positivos, total de palabras positivas y total de palabras negativas.

Al leer las líneas, se separaron las mismas en palabra por palabra, las cuales se metieron en una lista. El hecho de leer los dos archivos a la vez (el de los tweets y el de sus soluciones), permitió clasificar cada palabra. Cada palabra se convirtió en una llave de un diccionario, que correspondía a una lista de dos cosas, la primera era el total de veces que había aparecido, y la segunda el total de veces que había sido negativa.

Mientras se iban guardando en el diccionario las nuevas palabras, estas se guardaban en minúsculas.

Posteriormente se eliminaron del diccionario las siguientes palabras: el, en, le, al, la, lo, las, les, ellos, ellas, mio, mia, su, sus, se, si, él, ella, de, te, mi, y, por, a, que, me. Esto fue un ajuste ya que dichas palabras son muy comunes en el idioma español, y al ser la mayoría de los tweets positivos, considerar la presencia de esas palabras como relevantes afectaba el resultado final.

No utilicé un ajuste de parámetros en el sentido estricto de la palabra. Simplemente obtuve probabilidades muy bajas tanto para sobre si el tweet era agresivo o no, así que lo que hice fue evaluar cuál de los dos decimales era mayor para decidir si era agresivo.

Variable	Ecuación
Probabilidad de que un tweet sea negativo:	Casos totales negativos / Casos totales
Probabilidad de que un tweet sea positivo:	1 - Probabilidad de que un tweet sea positivo
Probabilidad de que una palabra sea negativa	$\frac{\text{Número de veces que una palabra es negativa + 1}}{\text{Número total de palabras negativas + Total de palabras únicas en el diccionario}}$
Probabilidad de que el caso sea negativo:	(Probabilidad de que un tweet sea negativo) *(probabilidad de que la palabra 1 sea negativa)*(probabilidad de que la palabra 2 sea negativa)*…
Probabilidad de que una palabra sea positiva	$\frac{\text{Número de veces que una palabra es positiva + 1}}{\text{Número total de palabras positiva + Total de palabras únicas en el diccionario}}$
Probabilidad de que el caso sea positivo:	(Probabilidad de que un tweet sea positivo) *(probabilidad de que la palabra 1 sea positiva)*(probabilidad de que la palabra 2 sea positiva)*…

3. Ejemplos de uso

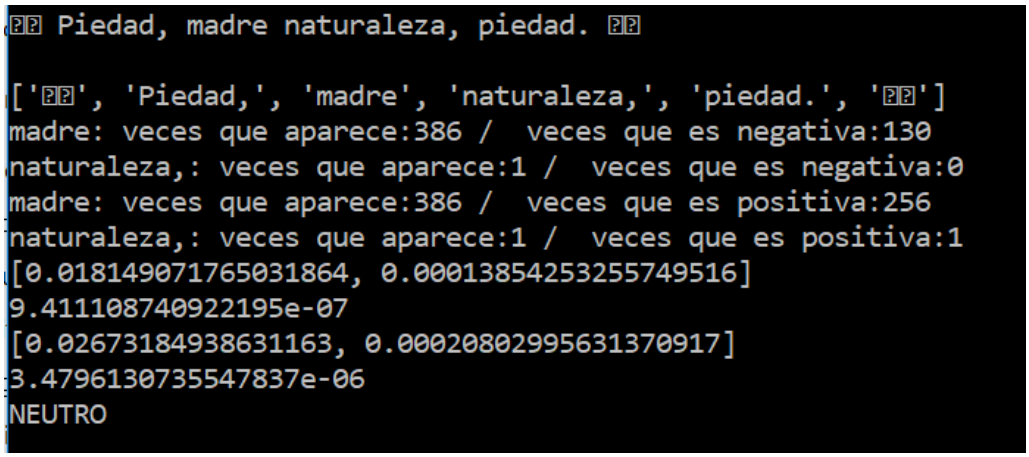
Así trabaja el clasificador. Evalúa la cadena “Piedad, madre naturaleza, piedad”, la separa por palabras, las convierte a minúsculas, evalúa si ya están en el diccionario o no, averigua cuántas veces aparecen y cuántas veces ha sido negativa o positiva.

```
    linea=train.readline().split()
    print(linea)
    resp=sol.readline()
    print(str(i+1)+":   "+str(resp))
    if(int(resp)==1):
        WtotalesNeg=WtotalesNeg+len(linea)#palabras negativas totales en el entrenamiento
        CasostotalesNeg=CasostotalesNeg+int(resp)#casos totales negativos en el entrenamiento
    else:
        WtotalesPos=WtotalesPos+len(linea)#palabras positivas totales en el entrenamiento
        CasostotalesPos=CasostotalesPos+1#casos positivos totales en el entrenamiento
    for j in linea:
        if(j.lower() in dic):
            tmp=dic[j.lower()]
            tmp[0]=tmp[0]+1
            tmp[1]=tmp[1]+int(resp)
            dic[j.lower()]=tmp
        else:
            dic[j.lower()]=[1,int(resp)]
```

Calcula la probabilidad para cada palabra de ser agresiva o no, las multiplica por la probabilidad de que un tweet sea agresivo o neutro dependiendo del caso y finalmente evalúa cuál probabilidad es mayor. En este caso era más probable que el tweet fuera neutro.

```
def calcularPROB(cad,dic,WtotalesNeg,WtotalesPos,PtotalNeg,PtotalPos):#función que calcula si es más probable que un tweet sea qué...
    cade=cad.split()
    print(cade)
    probNEG=[]
    probPOS=[]
    probabilidadCasoSeaNeg=PtotalNeg
    probabilidadCasoSeaPos=PtotalPos
    for word in cade:
        if(word.lower() in dic):
            tmp2=dic[word.lower()]
            Wtotal=tmp2[0]#total de veces que aparece la palabra
            Wneg=tmp2[1]#cuantass veces la palabra es negativa
            print(word.lower()+": veces que aparece:"+str(Wtotal)+" /  veces que es negativa:"+str(Wneg))
            probNEG.append((Wneg+1)/(WtotalesNeg+len(dic)))
            probabilidadCasoSeaNeg=probabilidadCasoSeaNeg*((Wneg+1)/(WtotalesNeg+len(dic)))
    for word in cade:
        if(word.lower() in dic):
            tmp2=dic[word.lower()]
            Wtotal=tmp2[0]
            Wpos=Wtotal-tmp2[1]
```

```
            print(word.lower()+": veces que aparece:"+str(Wtotal)+" /  veces que es positiva:"+str(Wpos))
            probPOS.append((Wpos+1)/(WtotalesPos+len(dic)))
            probabilidadCasoSeaPos=probabilidadCasoSeaPos*((Wpos+1)/(WtotalesPos+len(dic)))
    print(probNEG)
    print(probabilidadCasoSeaNeg)
    print(probPOS)
    print(probabilidadCasoSeaPos)
    if(probabilidadCasoSeaNeg>probabilidadCasoSeaPos):
        print('AGRESIVO')
        return 1
    else:
        print('NEUTRO')
        return 0
```



4. Tabla de resultados

Entrenando con una muestra de 770 tweets (10%), se obtuvo que en el otro 90% de los tweets, coincidieron 4961 de 6930, eso es un 71.5873% de éxito.

Entrenando con una muestra de 1540 tweets (20%), se obtuvo que en el otro 80% de los tweets, coincidieron 4548 de 6160, eso es un 73.8311% de éxito.

Entrenando con una muestra de 3850 tweets (50%), se obtuvo que en el otro 50% de los tweets, coincidieron 2956 de 3850, eso es un 76.7792% de éxito.

Entrenando con una muestra de 6930 tweets (90%), se obtuvo que en el otro 10% de los tweets, coincidieron 599 de 770, eso es un 77.7922% de éxito.

Muestra de entrenamiento	Muestra a comparar	Coincidencias	Coincidencias en %
Los 770 primeros tweets	6930	4961	71.5873%
1540	6160	4548	73.8311%
3850	3850	2956	76.7792%
6930	770	599	**77.7922%**

Otra manera de analizar qué tan confiable es el programa, sería dividiendo los archivos de entrenamiento en cinco partes, tomar un quinto de los archivos (diferente cada vez) para comparar con los resultados del programa, y promediar el porcentaje de éxitos.

Sea ‘ * ’ 1540 tweets a entrenar y ‘ o ’ 1540 tweets a comparar:

****o

***o*

o

*o***

o****

Se comparó desde	Hasta	Coincidencias	Coincidencias en %
0	1540	1204	78.1818%
1540	3080	1187	77.0779%
3080	4620	1229	79.8051%
4620	6160	1197	77.7272%
6160	7700	1217	79.0259%
		Promedio: 1206.2	Promedio: 78.36358%