Universidad de Buenos Aires
Facultad de Ingeniería
Año 2018 - 2º Cuatrimestre

75.69 Simulación

Fecha: 20/11/18

Informe Trabajo Práctico II

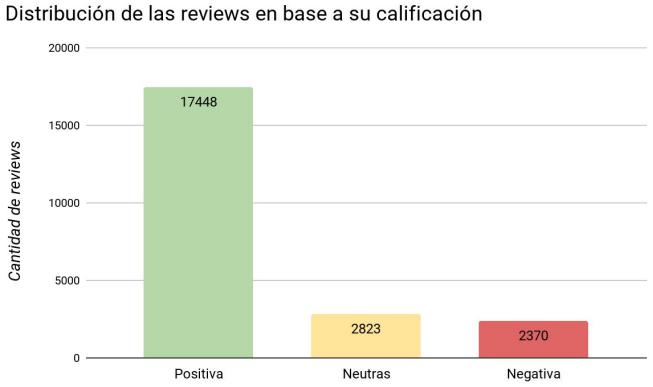
Apellido y nombre	Padrón	Email
Otero, Andrés	96.604	oteroandres95@gmail.com

Índice

Análisis y parseo de datos

Cálculo de sentimientos

A la hora de elegir que es una review positiva ,neutra o negativa se usó como criterio que las reviews positivas son aquellas que superan a los 4 puntos de rating, aquellas que tienen 3 puntos son neutrales y las que tienen 2 o menos son reviews negativas. Este resultó el criterio más conveniente a la hora de encontrar mejores resultados, la distribución de reviews en base a este criterio tiene esta forma:



Filtrado y salida

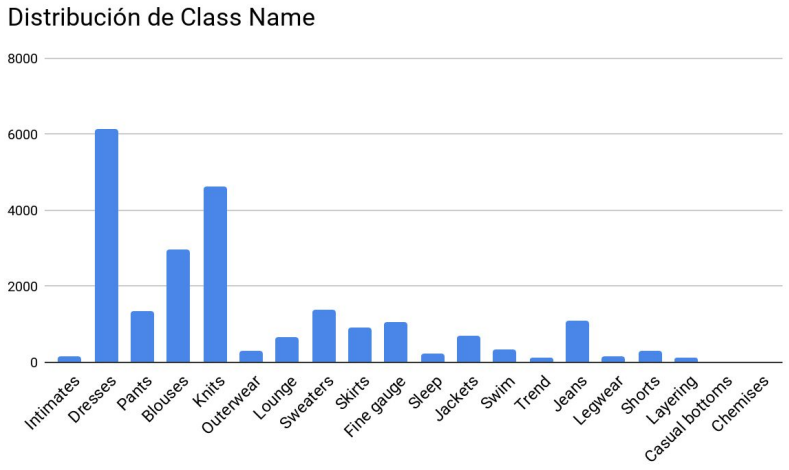
El último paso de este proceso es generar 2 archivos de salida luego de filtrar las columnas interesantes para cada una de los análisis que queremos realizar (sentimiento y class name). En el caso de predecir las clases de las reviews saque un par de los casos particulares ya que eran solo 2 filas con una clase particular cada una (CASUAL BOTTOMOS y CHEMISSES) y generaban mucho ruido al usar los algoritmos de clasificación. Los archivos de salida se verán de la siguiente manera:

Sentimientos

Nombre	Tipo
Sentimiento	Nominal. Tipos:{NEUTRA,POSITIVA, NEGATIVA}
Review Text	Texto

Class Name

Nombre	Tipo
Class Name	Nominal. Tipos:{'Intimates', 'Dresses', 'Pants', 'Blouses', 'Knits', 'Outerwear', 'Lounge', 'Sweaters', 'Skirts', 'Fine gauge', 'Sleep', 'Jackets', 'Swim', 'Trend', 'Jeans', 'Legwear', 'Shorts', 'Layering', 'Casual bottoms', 'Chemises'}
Review Text	Texto



Análisis de sentimientos

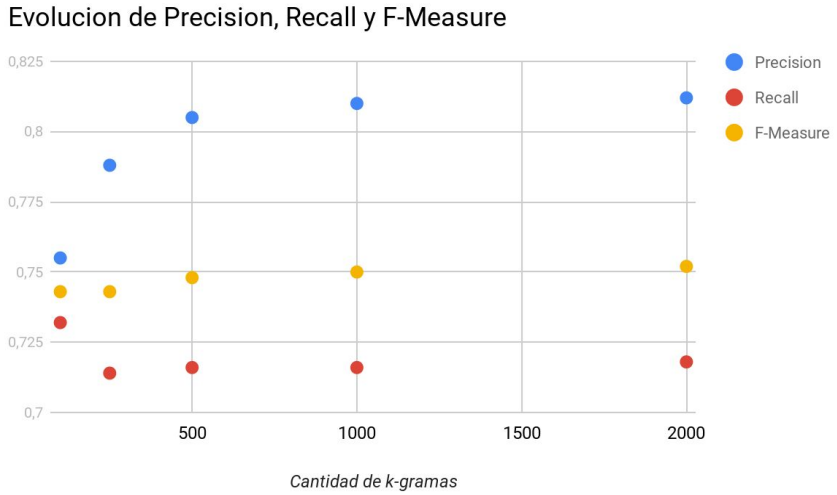
Procedimiento

Para empezar con el software ***Weka*** se abre el csv que se generó con el parser, el primer paso es que el campo "*review text*", que se interpreta como un campo nominal, se convierta en un string así que con el filtro de preprocesamiento de atributos no supervisado ***NominalToString*** y guardar el archivo como *reviewsSentimientos.arff.*

Luego de este cambio, lo próximo es aplicar otro filtro al campo de texto, ***stringToWordVector*** usando como tokenizer un N-Gram-Tokenizer de 1 a 3 palabras. Luego de usar este filtro decido cuántas k-gramas se van a utilizar, finalmente se usa un clasificador de ***NaiveBayes***, usando como porcentaje de aprendizaje 66%.

Resultados

Probando varias cantidad de k-gramas vemos que los resultados finales son estos:



Esto trae resultados interesantes, a medida que se aumenta los k-gramas , aumenta la precisión pero al mismo tiempo baja el recall. Esto significa que cada vez hay menos falsos positivos pero al mismo tiempo se aumentan los falsos negativos. Podemos ver que mientras tanto el F-Measure se mantiene más o menos constante alrededor de 0,75.

ClassName

Procedimiento

El procedimiento es muy parecido al del análisis de sentimientos, para empezar con el software ***Weka*** se abre el csv que se generó con el parser, el primer paso es que el campo "*review text*", que se interpreta como un campo nominal, se convierta en un string así que con el filtro de preprocesamiento de atributos no supervisado ***NominalToString*** y guardar el archivo como *reviewsClassName.arff.*

Luego de este cambio, lo próximo es aplicar otro filtro al campo de texto, ***stringToWordVector*** usando como tokenizer un N-Gram-Tokenizer de 1 a 3 palabras. Se usaron un set de 100 k-gramas en este caso, ya que la mayor cantidad de clases hacía que el proceso de clasificación con muchos k-gramas fuera excesivamente lento. Luego de usar este filtro decido cuántas k-gramas se van a utilizar, finalmente se usa un clasificador de ***SMO***, usando como porcentaje de aprendizaje 66%. Probé varios tipos de clasificadores y el SMO resultó el más eficiente. Los resultados obtenidos con este método son:

Precisión	Recall	F-Measure
0,585	0,597	0,578

Repositorio

Link al repositorio: https://github.com/AndresOtero/AnalisisDeSentimientosFallas
Como correr el parser: Ejecutar el script Parser.py en la carpeta con una version de Python mayor a 3.0.0 .