Tarea 3 Análisis Inteligente de Datos

Juan Ávalo, Bastien Got

July 13, 2016

Contents

1 Reconocimiento de Texto

1. Tanto el set de entrenamiento como el de pruebas tienen **3554** datos. Los datos se separan en textos positivos y negativos de acuerdo a la tabla:

Sets	Positivos	Negativos
Entrenamiento	1770	1784
Prueba	1803	1751

2. Se creó una función que extrae palabras usando *stemming* y quitando *stopwords* si así es pedido. Sobre los resultados obtenidos se puede observar con los ejemplos usados que *stemming* lo que hace es hacer que una palabra tome una pseudoraiz. Además, el proceso de quitar *stopwords* efectivamente quita palabras como ”I” o ”to”.

   Como ejemplo, las frases: ”I love to eat cake” y ”I love eating cake” ambas se reducen a ”love”, ”eat” y ”cake”.

   Pero un ejemplo más interesante es el de aplicar *stemming* sobre palabras como ”absolutely” y ”dislike”, las cuales se traducen en ”absolut” y ”dislik”. Ninguna de las pseudoraices es una palabra del inglés verdadera, pero van a ser útiles para saber que palabras están relacionadas o no.

3. También se creó una función análoga a la del punto anterior, pero lematizando. Para poder lograr ésto no se pudo usar la función `WordNetLemmatizer` directamente como estaba en los ejemplos.

   Lematización involucra hacer un análisis sintáctico de las palabras, por lo que la función usada pide marcar cada palabra con la posición dentro de la oración que tiene (verbo, sustantivo, adjetivo, adverbio), la cual se obtuvo mediante la función `pos_tag`.

   Los efectos en las palabras de ejemplo son aparentes. En casos como el de ”I love to eat cake” el lematizador las reduce de la misma forma que usando *stemming*. La diferencia se nota al usar las palabras ”dislike” y ”absolutely”, las cuales se mantienen iguales. O con palabras como ”are” e ”is” las cuales se reducen a ”to be”.

4. Se generaron cuatro representaciones vectoriales para los dos conjuntos de datos. La razón de ésto es porque se necesita extraer palabras con *stemming* y con *lemmatize*, con *stopwords* y sin ellas.

La representación del texto consiste en resumir cada comentario a un vector binario con todo el vocabulario obtenido de todos los mensajes. Si el mensaje tiene una palabra dentro del vocabulario, el valor de la variable correspondiente a esa palabra es **1**. Sino es **0**.

Luego las etiquetas son **0** si el mensaje es negativo, y **1** si es positivo.

Considerando como se tratan los datos, se puede rankear las palabras que globálmente se encontraron por frecuencia. Un ejemplo de ello es la siguiente tabla:

Frecuencia	Palabra
115	way
125	get
127	well
128	much
129	work
143	even
143	time
145	comedy
163	character
169	good
176	story
246	one
254	like
264	make
481	movie
573	film

5. El evaluador de desempeño considera las siguientes medidas:
   - La precisión del modelo sobre los datos de entrenamiento.
   - La precisión del modelo sobre los datos de prueba.
   - La *precisión*, esto es, el porcentaje de datos bien clasificados dentro de todos los datos clasificados.
   - El *recall* el cual es el porcentaje de los datos seleccionados bien clasificados dentro de los datos de su clase.
   - El *f1-score*, el cual es la media armónica entre la precisión y el recall.
   - El support, que cuenta cuantos datos de cada clase hay.
6. Se creó la función para ajustar un modelo *Naive Bayesian* sobre los datos.