Text Mining en Social Media: Clasificación de Tweets por Género y País

Rafael Alberto Vallejo Antich

ravalan@alumno.upv.es

***Resumen*— Las redes sociales son un recurso que nos permiten disponer de una cantidad ingente de mensajes de texto que podemos identificar por usuario. Estos textos, junto con la información del perfil de los mismos, pueden ser de elevada utilidad a la hora de encontrar distintas relaciones entre las características de los usuarios y el modo en el que se expresan y redactan sus mensajes en las redes sociales. El propósito de este caso de estudio consistió en analizar por medio de text mining y posteriormente clasificar, mediante técnicas de machine learning, a los usuarios de la red social Twitter en base a los mensajes publicados. Así pues, esta memoria incluye la descripción y exploración de los datos, selección de algoritmos y cálculo de errores con la finalidad de clasificar tanto el género como el país al que pertenece un usuario en base a los tweets publicados.**

I. INTRODUCCIÓN

El estudio que se pretende abordar en esta memoria estriba en la predicción del género y del país de un usuario en base a sus tweets publicados. El objetivo principal será el de clasificar el género al que pertenece (másculino o femenino) y el país al que pertenece en base a sus publicaciones. Para ello se analizan tanto el uso de las palabras utilizadas por el mismo, como la sintaxis asociada a estas publicaciones

Para ello se dispone de un dataset de muestras etiquetadas, correspondientes a los tweets de usuarios de distintas nacionalidades. Con el fin de simplificar el estudio se han seleccionado únicamente tweets de 7 países de habla hispana: Argentina, Chile, Colombia, España, Méjico, Perú y Venezuela. Además, para asegurar que las muestras están balanceadas, disponemos únicamente de aquellos usuarios para los que se han podido obtener 100 tweets. Nos encontramos por lo tanto ante un problema de aprendizaje supervisado.

Así pues, el objetivo final será, una vez creado el modelo, el de identificar el género y el país al que pertenece un usuario dado un texto escrito por el mismo.

Se trata por lo tanto de un problema de clasificación, por lo que, tras aplicar las técnicas de text mining sobre los tweets, deberemos crear un modelo que por medio de las técnicas de machine learning sea capaz de predecir las clases a las que pertenece el autor del mismo en base a dos variables objetivo: género y país.

Como veremos en detalle en los siguientes puntos, el algoritmo que mejores prestaciones nos proporcionó en nuestro modelo de clasificación fue el Support Vector Machine, sobre ambas variables con kernel Radial. Si bien es cierto que, a pesar de que en ambos casos el uso de SVM fuera el óptimo, se generaron modelos diferentes para predecir cada una de las variables, ya que presentaban algunas diferencias.

Por último, cabe destacar que mientras que para la clasificación del género conseguimos un accuracy del 75%, para el del país conseguimos una precisión del 88%.

II. EL DATASET

Los datasets con los que contamos para la generación de un modelo se dividen en 2 partes. Por un lado disponemos de un conjunto de train, compuesto por 2.800 muestras etiquetadas y, por otro, de un conjunto de test con 1.400 muestras. Por lo que disponemos de un conjunto de test que supone el 33% sobre las muestras totales. Estas 1.400 muestras serán las que deberemos clasificar, asignándoles un género y un país en base al modelo que crearemos.

También cabe destacar que el dataset asociado a los tweets de cada usuario se tiene en ficheros xml independientes, que contienen los 100 tweets asociados al mismo. Por lo que el primer paso de nuestra labor fue cargar los 100 tweets de los usuarios en una misma tabla, de modo que dispondríamos de una tabla en R para el conjunto de train y otra para el de test. En estas tablas tendríamos tantas filas como muestras tuviéramos y cada tweet se almacenaría en una columna.

Por otro lado, la información asociada al género y país de las 2.800 muestras etiquetadas se tiene en un fichero independiente. Sin embargo, se pueden relacionar sobre las muestras del train por medio de un código único que identifica al usuario.

Uno de los primeros pasos es realizar una vista preliminar de las diferentes variables del dataset de train. De este modo pudimos comprobar que el número de muestras para los distintos géneros y países que disponíamos era la siguiente:

Muestras **Totales del Train: 2.800**

- De Género **Masculino: 1.400**
  - **Argentina**: 200 tweets
  - **Chile**: 200 tweets
  - **Colombia**: 200 tweets
  - **España**: 200 tweets
  - **Méjico**: 200 tweets
  - **Perú**: 200 tweets
  - **Venezuela**: 200 tweets
- De Género **Femenino: 1.400**
  - **Argentina**: 200 tweets
  - **Chile**: 200 tweets
  - **Colombia**: 200 tweets
  - **España**: 200 tweets
  - **Méjico**: 200 tweets
  - **Perú**: 200 tweets
  - **Venezuela**: 200 tweets

Así pues, observamos que las muestras se encuentran perfectamente balanceadas, por lo que no fue necesario descartar ninguna de las muestras de training o recurrir a técnicas de oversampling.

Una vez explorado cómo se estructuran los datos de nuestro dataset, analizamos la bolsa de palabras más usadas sobre el conjunto de tweets del training. Para ello representamos en un histograma la frecuencia de las 10 palabras más usadas:

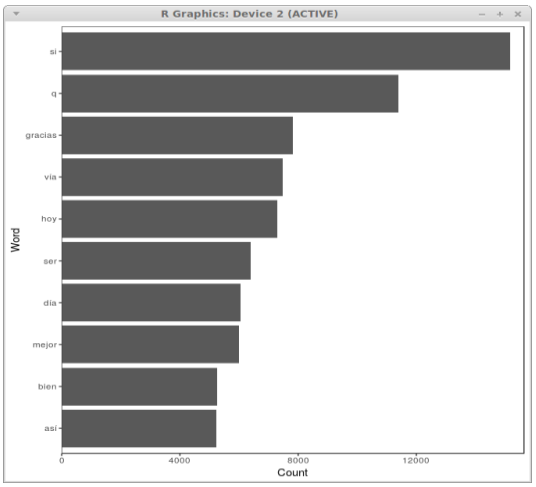

Fig. 1: Bolsa de 10 palabras

Asimismo, vimos que la distribución de una bolsa de 1.000 palabras seguía también una distribución logarítmica:

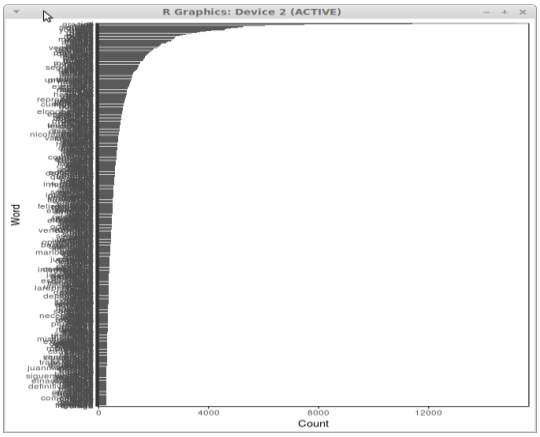

Fig. 2: Bolsa de 1.000 palabras

III. PROPUESTA

Uno de los apartados que más tiempo consumen en las tareas de ciencia de datos consiste en el preproceso de los mismos y, en este caso de estudio, no fue una excepción.

En primer lugar disponíamos de los tweets de los usuarios en ficheros xml independientes, por lo que el primer paso consistió en cargar estos datos en RStudio, uniéndolos y componiendo una única tabla donde cada columna representase un tweet. Esto lo hicimos mediante la librería *XML*.

Una vez cargados los datos decidimos asignarlos a 2 dataframes distintos. Al igual que se ha comentado anteriormente, se creó una tabla para el conjunto de train que contenía los tweets y variables a predecir (género y país) y, otra, para los 1.400 tweets de test.

En cuanto a los valores cargados, observamos algún caso para el que sólo disponíamos de 99 tweets. Sin embargo, puesto que sólo se trataba de un usuario, se optó por mantenerlos ya que consideramos que no supondría un fuerte impacto de cara a la implementación del modelo.

Los 2 principales puntos a tener en cuenta de cara a la implementación de cada uno de los modelos fueron los siguientes:

- **Signos de puntuación:** En cuanto al procesamiento de textos había 2 alternativas: mantener o eliminar los signos de puntuación de los tweets con los que entrenaríamos el modelo. Para ello optamos por mantenerlos para la clasificación del género y eliminarlos para la clasificación del país. Esto es debido a que intuimos que la sintaxis de un mensaje es más útil para predecir el género que el país de procedencia del usuario.
- **Preposiciones y palabras sin significado:** En este caso se optó por la misma medida que el caso anterior. Es decir, mientras que se optó por mantener estas palabras para la predicción del sexo, se decidieron eliminar para la clasificación del país. Ya que a la hora de determinar el país de procedencia es más útil comprobar si el tweet contiene palabras que son claves a la hora de identificar la nacionalidad del usuario.

Una vez realizadas las tareas de preproceso, nuestra propuesta consistió en implementar el mayor número posible de modelos mediante los algoritmos de clasificación de machine learning. Nuestra experiencia previa en anteriores tareas de data science nos hizo pensar que para la creación de un modelo con una precisión razonable en un tiempo reducido, la mejor opción era la de reducir al máximo posible las tareas de preprocesado (muy costosas en tiempo) para centrarnos más en el uso de los algoritmos de las distintas librerías de ML disponibles en RStudio.

Así pues, optamos por hacer uso de las siguientes librerías:

```
library(caret)

library(e1071)
```

Y los algoritmos de Clasificación que íbamos a contemplar para nuestros modelos serían los siguientes:

- SVM (Linear)
- SVM (Radial)
- Naive Bayes
- k-NN
- Random Forest

Además, el planteamiento consistió no solamente en probar el algoritmo que mejor funcionaba en función del tamaño de la bolsa de palabras. Si no que trataríamos posteriormente de ajustar la precisión por medio de los parámetros de éstos si procedía.

IV. RESULTADOS EXPERIMENTALES

A. Clasificación de Género

Recordemos que para la clasificación por género optamos por mantener signos de puntuación y preposiciones. Además, en este caso generamos los modelos mediante una bolsa de 1.000 palabras, ya que no fuimos capaces de crear estos mismos modelos con bolsas de tamaño superior debido a los tiempos de ejecución. Los resultados obtenidos en este caso fueron los siguientes:

Algoritmo	Kappa	Accuracy
Baseline	0.32	0.66
SVM Linear	0.34	0.67
SVM Radial	**0.50**	**0.75**
Naive Bayes	0.34	0.67
kNN	0.42	0.71
Random Forest	0.40	0.69

Tabla 1: Resultados de Clasificación por Género

Como podemos observar en esta tabla, el mejor resultado lo obtuvimos con un SVM Radial, que nos generó un accuracy del 0.75.

B. Clasificación de País

Para tener una primera aproximación de los modelos que mejor ajustaban a la predicción del país, se optó por probar primero diferentes modelos sobre una bolsa de palabras de tamaño reducido (100 palabras). De este modo, mediante unos tiempos de ejecución bastante rápidos podíamos tener una idea acerca de cuáles funcionaban mejor.

Para una bolsa de 100 palabras los resultados obtenidos fueron los siguientes:

Algoritmo	Kappa	Accuracy
SVM Linear	**0.4433**	**0.5229**
SVM Radial	**0.5193**	**0.5193**
Naive Bayes	0.325	0.4214
kNN	0.1375	0.355

Tabla 2: Clasificación por País con Bolsa de 100 Palabras

A la vista de estos resultados intuímos que el mejor modelo se construirá mediante un SVM, pero con estas diferencias tan reducidas no podemos garantizar que el kernel Lineal funcione mejor que el Radial.

Así pues, repitiendo la prueba anterior pero con una bolsa de 200 palabras obtenemos:

Algoritmo	Kappa	Accuracy
SVM Linear	0.5717	0.6329
SVM Radial	**0.6217**	**0.6757**
Naive Bayes	0.4775	0.5521
kNN	0.335	0.43

Tabla 3: Clasificación por País con Bolsa de 200 Palabras

Observamos, cómo a medida que aumentamos el tamaño de la bolsa de palabras el modelo creado mediante un SVM Radial va mejorando su precisión con respecto al resto de modelos.

Generando a continuación un modelo basado en SVM Radial con una bolsa de 500 palabras obtenemos los siguientes resultados:

Algoritmo	Kappa	Accuracy
SVM Radial	0.7508	**0.7864**

Tabla 4: Clasificación por País con SVM Radial y Bolsa de 500 Palabras

Así pues, una vez disponemos del accuracy de nuestro mejor modelo para una bolsa de 500 palabras (78,64% de precisión), pasamos a probar diferentes valores de los parámetros *C* y *Sigma* con el propósito de incrementar aún más la precisión del modelo. Para ello probamos con los siguientes valores:

- **Sigma:** 0.005, 0.010, 0.015
- **C:** 1.25, 1.50, 2.00

Esto lo realizamos por medio del parámetro *tuneGrid* de la función *train*:

```
grid <- expand.grid(sigma = c(.005, .01,
.015), C = c(1.25,1.5,2))
model_SVM_variety <- train(theclass~.,
data= training_variety,
trControl = train_control,
method = "svmRadial",
preProc = c("center","scale"),
tuneGrid = grid)
```

Tras probar con todas las combinaciones el mejor resultado lo obtenemos con el Sigma más bajo (0.005) y un valor de C=1.50:

Algoritmo	Kappa	Accuracy
SVM Radial	0.7542	**0.7107**

Tabla 5: SVM Radial con Sigma=0.005 y C=1.5

No obstante, en el preproceso RStudio hace una estimación de la precisión con un conjunto más reducido de muestras, siendo el siguiente el resultado de todas las combinaciones:

sigma	C	Accuracy	Kappa
0.005	1.25	0.6710879	0.6162678
0.005	**1.50**	**0.6718025**	**0.6171011**
0.005	2.00	0.6707311	0.6158517
0.010	1.25	0.4974796	0.4137461
0.010	1.50	0.4971227	0.4133289
0.010	2.00	0.4967658	0.4129120
0.015	1.25	0.3189177	0.2054301
0.015	1.50	0.3171332	0.2033495
0.015	2.00	0.3171332	0.2033499

Donde comprobamos que la opción de un sigma=0.005 y C=1.50 era la mejor opción. Sin embargo, la precisión obtenida con estos parámetros (71,07%) vemos que no supera los resultados del modelo generado con los valores por defecto (Tabla 4). Sin embargo, vemos que la modificación de éstos tiene un impacto relevante sobre el accuracy del modelo generado.

A continuación, y a la vista de que el parámetro sigma es el que más influye a la hora de determinar la precisión del modelo, lo que hacemos es explorar nuevos valores de sigma manteniendo un C igual a 1.50. Así pues, probamos a continuación las siguientes combinaciones de valores:

- **C:** 1.50
- **Sigma:** 0.00001, 0.0001, 0.001, 0.005

Al igual que antes, por medio del parámetro *tuneGrid* pasamos al train los valores de C y Sigma a contemplar.

```
grid <- expand.grid(sigma = c(.00001,
.0001, .001, .005), C = c(1.5))
```

En el preproceso que hace RStudio obtenemos ya los siguientes valores:

sigma	Accuracy	Kappa
1e-05	0.4272117	0.3320385
1e-04	0.7185640	0.6716578
1e-03	**0.7528551**	**0.7116650**
5e-03	0.6785749	0.6250128

Por lo que vemos que el mejor modelo se generará con un valor de Sigma = 0.001. Creado el modelo con estos valores sobre el dataset total de train obtenemos el siguiente resultado:

Algoritmo	Kappa	Accuracy
SVM Radial	0.7542	**0.7893**

Tabla 6: SVM Radial con Sigma=0.001 y C=1.5

Finalmente, una vez conocido el modelo que mejor funciona en nuestro caso (SVM Radial) y, más concretamente, los valores de los parámetros que generan una mayor aproximación, pasamos a crear el modelo sobre una bolsa de 5.000 palabras. El incremento de la bolsa de palabras hemos visto que permite clasificar el país con una mayor precisión, por lo que aplicando además los valores de C y Sigma correctos nos ha permitido generar un modelo con una precisión notable:

Algoritmo	Kappa	Accuracy
SVM Radial	0.8617	**0.8814**

Tabla 7: Clasificación por País con SVM Radial y Bolsa de 5.000 Palabras

En conclusión, hemos conseguido crear un modelo mediante Support Vector Machine con kernel Radial que nos permite predecir el país al que pertenece el autor de un tweet con un porcentaje de acierto del 88,14%.

V. CONCLUSIONES Y TRABAJO FUTURO

En caso de continuar con este caso de estudio sería posible trabajar otras ideas para mejorar la precisión del modelo. La primera de ellas sería profundizar más en el valor de los parámetros con los que construir los modelos, haciendo uso de **optimización bayesiana**.

En nuestra propuesta se ha profundizado principalmente en el uso y comparación de los distintos modelos de clasificación de ML. Sin embargo, tan importante como la aplicación de modelos es el pre-procesado del dataset, donde hay muchas mejoras que se pueden realizar.

Algunas de las tareas asociadas al pre-proceso con las que se podría mejorar nuestro modelo serían las siguientes:

- **Longitud del tweet**: Sería interesante incluir la longitud de cada tweet al dataset, ya que es posible que haya cierta correlación entre la longitud de los tweets y género y(o) país al que pertenece el autor.
- **Número de preposiciones**: Otra alternativa consistiría en concatenar al dataset el número de preposiciones utilizadas en el tweet. Podría haber una correlación entre este dato y el sexo o país del autor del mismo.
- **Palabra más usada**: Un dato relevante sería también considerar la palabra más usada en un tweet. Es posible que este dato nos proporcione más información para clasificar correctamente.
- **Reducción de dimensionalidad**: El uso de Principal Components Analysis (PCAs) puede permitirnos generar un modelo con una precisión mayor.

Obviamente, cualquier idea que pueda proporcionar información acerca del género o país al que pertenece el autor de un tweet, podría ser interesante contemplarla e incluirla en este caso de estudio.

En conclusión, hemos construido un modelo relativamente preciso para predecir el sexo y país al que pertenece el autor de un mensaje en Twitter. Procesando los tweets de 2.800 usuarios, diferenciados por género y país, y aplicando el modelo de SVM hemos sido capaces de obtener una precisión del 75% a la hora de determinar su género (es decir, si se trata de un hombre o de una mujer) y una precisión del 88,14% a la hora de determinar su nacionalidad. Pasos como ahondar más en los valores de los parámetros, trabajar más en el pre-procesado de datos para incluir nuevas variables, o reducir la dimensionalidad mediante el uso de PCAs, pueden potencialmente mejorar los resultados en el futuro.