Author profiling

Luis Miguel Montes Novella
lmontesnovella@gmail.com

Abstract

En este artículo se describe cómo se ha aproximado la tarea de *author profiling* para la identificación de género y rango de edad. Para llevar a cabo esta tarea se han extraído una serie de características del texto que se cree que pueden ayudar a diferenciar entre géneros o rangos de edad. Las características extraídas son el uso de signos de puntuación, el uso de emoticonos, el uso de ciertas características en el estilo de escritura como el *character flooding*, el uso de la arroba para denotar que la palabra puede contener una a o una o, la inclusión de números, direcciones *web* o correos electrónicos en los textos, etc. Además de estas características se ha obtenido un *BoW (Bag of Words)* con las palabras que ayudan a diferenciar mejor entre géneros para mejorar los resultados en esta tarea. Para la evaluación de los resultados se ha usado el *software Weka*, el modelo se ha entrenado con el algoritmo *Naive Bayes* y para la validación de resultados se ha usado la técnica de validación cruzada.

1 Introducción

El *author profiling* es una tarea que consiste en la identificación de tipos o clases de autores a partir de textos escritos mediante la búsqueda de patrones comunes en el estilo de escritura. Esta clasificación de autores en diferentes grupos tiene numerosas aplicaciones como por ejemplo la identificación de personas sobre las que dirigir las campañas publicitarias o políticas. En concreto en este trabajo se han seguido dos aproximaciones, la detección de género y la detección del rango de edad. Para la primera aproximación sólo tenemos dos posibilidades, detectar si se trata de un hombre o de una mujer. En cuanto a la segunda aproximación tenemos 3 posibles rangos de edad, adolescentes (10s), veinteañeros (20s) y treintañeros (30s).

2 *Dataset*

El *dataset* original de la competición *PAN* sobre *author profiling* contiene textos de 374100 autores extraídos de redes sociales, se compone de una serie de ficheros, uno por autor, con todos los textos que éste ha escrito. Estos textos se encuentran escritos en dos idiomas, hay 283240 textos en inglés y 90860 en castellano. Como el *dataset* es muy grande, para poder afrontar el problema de una forma más sencilla no se ha usado el *dataset* completo sino que se ha empleado un *dataset* reducido, en concreto se ha usado el *dataset* de test en castellano. Este *dataset* contiene un total de 8160 ficheros donde la mitad de los autores son hombres y la otra mitad mujeres, con respecto a las edades hay 288 adolescentes, 4608 veinteañeros y 3264 treintañeros. En la siguiente tabla se puede ver la distribución de las personas por clase, como se puede observar la distribución en cuanto a género es equilibrada pero en lo referente al rango de edad el número de muestras de cada clase es muy diferente.

	10s	**20s**	**30s**	**TOTAL**
Hombre	144	2304	1632	**4080**
Mujer	144	2304	1632	**4080**
TOTAL	**288**	**4608**	**3264**	

Table 1: Distribución de los totales por clase

3 Propuesta del alumno

En este apartado se describen las características extraídas del texto y las variables obtenidas a partir de éstas para entrenar un clasificador mediante el algoritmo *Naive Bayes*. Este clasificador se ha entrenado mediante el *software Weka* y se ha usado

validación cruzada con 10 particiones para obtener una medida del porcentaje de aciertos del clasificador.

3.1 Información extraída del texto

3.1.1 Mejoras en el procesamiento de código *HTML*

En primer lugar se ha mejorado la extracción de *tokens* o palabras desde el código *HTML*. Antes cada vez que aparecía una etiqueta *HTML* de salto de línea (*br*) se ignoraba y las dos palabras que tenía alrededor se unían generándose palabras que no existían dentro del vocabulario de los textos. Esto hacía que las frecuencias de aparición no fueran correctas. Ahora al realizar el preproceso siempre que aparece una etiqueta de salto de línea se sustituye por un espacio para que la *tokenización* sea correcta. Además de esto durante el parseo de *HTML* se extraen nuevas características que se detallarán en las siguientes secciones.

3.1.2 *Character flooding*

El *character flooding* es un estilo de escritura que consiste en el alargamiento de palabras duplicando letras o añadiendo letras que no existen como por ejemplo la o en *hooola* o la h en *tengohh*. Una alta frecuencia de aparición de esta característica en las palabras puede indicar que se trata de una persona joven por lo que puede ayudar a detectar la edad.

Se ha usado un algoritmo para detectar *character flooding* a nivel de palabras. Para ello se ha considerado que una palabra tiene *character flooding* siempre que contiene más de dos vocales seguidas o dos caracteres seguidos que no sean c, l, r o vocal. Se han excluido estos caracteres porque en castellano pueden aparecer juntos. No se ha tenido en cuenta el idioma de la palabra por lo que hay algunas palabras en inglés dentro de los textos que se considera que tienen *character flooding*. Hay que tener en cuenta que algunos *tokens* también pueden ser números por lo que si un *token* es un número no se considera *character flooding*. Otro caso especial de *character flooding* que se ha considerado ha sido el de los signos de interrogación y exclamación, en este caso se considera que hay *character flooding* cuando aparecen dos de estos símbolos juntos.

3.1.3 Signos de puntuación

Se ha tenido en cuenta el uso de signos de puntuación como puntos, comas y acentos ya que un mayor uso de estos símbolos puede indicar que la persona tiene mayor edad.

3.1.4 Emoticonos

En los mensajes de *chat* y redes sociales es muy frecuente el uso de emoticonos, la frecuencia de uso podría estar relacionada con la edad ya que su uso se extendió con la aparición de los teléfonos móviles y esto es algo bastante reciente. También podrían estar muy relacionados con el género ya que se usan frecuentemente para expresar sentimientos y en este aspecto suele haber muchas diferencias entre hombres y mujeres. A la hora de procesar los textos se ha considerado que todas las etiquetas *HTML* de imagen (*img*) que aparecen en el texto se corresponden con emoticonos.

3.1.5 Etiquetas *HTML* de énfasis

En el lenguaje *HTML* existen una serie de etiquetas que sirven para resaltar el significado del texto que contienen, estas etiquetas hacen que el texto aparezca con un estilo diferente como negrita cursiva o subrayado. La frecuencia del uso de estilos al escribir podría ayudar a diferenciar el género y edad de las personas. Por tanto se ha tenido en cuenta si en el código *HTML* del mensaje aparecen etiquetas de énfasis como *em*, *strong*, *b*, *u* e *i*.

3.1.6 Detección de risas

La frecuencia del uso de risas en los mensajes de *chat* podría ayudar a detectar el género o edad de una persona. Para ello se ha realizado el parseo del texto buscando si los *tokens* que aparecen se corresponden con risas. Se ha considerado una risa toda cadena de texto que contiene en su interior una subcadena como *jaj*, *jej*, *jij*, *joj* o *juj*. Esta detección se ha hecho mediante el uso de expresiones regulares sobre cada uno de los *tokens*.

3.1.7 Número de palabras diferentes

Se ha contado el número de palabras diferentes que usa una persona ya que se puede pensar que cuanto mayor sea su edad más rico será su vocabulario y por tanto usará un mayor número de palabras diferentes.

3.1.8 Otros

Además de todo lo anterior también se ha procesado el texto con expresiones regulares en busca de direcciones *web*, correos electrónicos y números para ver si la aparición de éstos en un texto puede ayudar a diferenciar entre rango de edad o género.

También se ha buscado si en las palabras aparece el símbolo @ para denotar que esa letra puede ser tanto una a como una o ya que es algo que está de moda desde hace pocos años y podría ayudar a diferenciar el rango de edad. Para evitar confundir estas palabras con los correos electrónicos sólo se cuentan los *tokens* que no son correos electrónicos.

3.1.9 *Bag of Words*

Un *BoW (Bag of Words)* es un conjunto de palabras que se usa para representar un texto, normalmente se escogen palabras muy comunes o que ayudan a discriminar entre las diferentes clases que hay que diferenciar. En el *baseline* se usa un *BoW* con las frecuencias de aparición de las 1000 palabras más comunes dentro de cada autor.

En este caso se ha modificado la forma de obtener el *BoW* para intentar mejorar los resultados en la tarea de género, para ello en el preproceso del texto se ha obtenido un ratio que indica la capacidad de discriminación de una palabra. Se ha contado el número de veces que aparece una palabra ($frec_w$) y el número de veces que usa esa palabra algún hombre ($frec_{wh}$). Con estos datos se ha obtenido un ratio $ratio_w$ para cada palabra como indica la siguiente fórmula:

$$ratio_w = \frac{frec_{wh}}{frec_w}$$

Con este ratio podemos saber si una palabra es más usada por hombres o por mujeres, los valores alrededor de 0.5 indican que esa palabra no ayuda a diferenciar entre género, los valores mayores de 0.5 indican que esa palabra suele ser más usada por los hombres y valores menores de 0.5 indican que esa palabra suele ser más usada por las mujeres. Una vez calculado este ratio se han ordenado las palabras por su valor. Para evitar que palabras con pocas apariciones tengan ratios muy altos se han eliminado todas las palabras que aparecen menos de 200 veces a lo largo de los textos. A la hora de escoger las palabras para incluir en el *BoW* se han cogido las 200 con mayor valor y las 200 con menor valor para asegurarnos de que sus ratios tienen un valor que se diferencia bastante de 0.5 y que el número de palabras que ayudan a identidificar a los hombres es igual que el número de palabras usado para las mujeres.

3.2 Variables usadas

A la hora de generar los ficheros de datos para *Weka* se han usado las siguientes variables:

- **ratio de comas**: número de comas en el texto entre el total de palabras del texto.
- **ratio de puntos**: número de puntos en el texto entre el total de palabras del texto.
- **ratio de acentos**: número de acentos en el texto entre el total de palabras diferentes del texto.
- **ratio de *character flooding***: número de palabras con *character flooding* entre el total de palabras del texto.
- **ratio de palabras diferentes**: número de palabras diferentes entre el total de palabras del texto.
- **ratio de emoticonos**: número de emoticonos entre el total de palabras del texto.
- **ratio de énfasis**: número de veces que aparecen etiquetas de énfasis en *HTML* entre el total de palabras del texto.
- **ratio de risas**: número de risas entre el total de palabras del texto.
- **ratio de números**: cantidad de veces que aparecen números en un texto entre el total de palabras del texto.
- ***character flooding* de símbolos**: *flag* que indica si se usa *character flooding* de símbolos de exclamación e interrogación en el texto, sólo puede tener el valor 0 (si se no usa) y 1 (si se usa).
- **uso de direcciones *web***: *flag* que indica si aparecen direcciones *web* en el texto, su valor sólo puede ser 0 o 1.
- **uso de direcciones de correo electrónico**: *flag* que indica si aparecen direcciones de correo electrónico, su valor sólo puede ser 0 o 1.
- **uso de la arroba en palabras**: *flag* que indica si aparecen palabras que contienen una arroba y no son correos electrónicos.
- ***BoW***: *Bag of Words* con 400 palabras obtenido con el proceso del apartado 3.1.9. Para cada palabra se indica si aparece en el texto mediante un 1 y si no aparece mediante un 0.

4 Resultados experimentales

Para evaluar el modelo de clasificación se ha usado el *software Weka* con los algoritmos *Naive Bayes* y *DMNBText* usando validación cruzada con 10 particiones.

En la siguiente tabla se pueden observar los resultados obtenidos para el algoritmo *Naive Bayes*, en la primera fila se muestra el *baseline* para comparar. Al usar únicamente las 13 variables descritas en el apartado 3.2 pero sin usar el *BoW* obtenemos una gran mejora con respecto al *baseline* para la tarea de edad y los resultados empeoran ligeramente para la tarea de género. Si además de estas variables añadimos el *BoW* obtenido con las palabras que ayudan a distinguir mejor entre géneros conseguimos superar el *baseline* para la tarea de género y mejoramos un poco el porcentaje de acierto para la tarea de edad.

	Género	**Edad**
Baseline	54.88%	27.13%
Variables sin *BoW*	52.55%	44.36%
Variables con *BoW*	58.87%	46.64%

Table 2: Resultados obtenidos con el algoritmo *Naive Bayes*

En cuanto a los resultados obtenidos con el algoritmo *DMNBText* se puede ver que el porcentaje de acierto para la tarea de edad mejora en gran medida y para la tarea de género los resultados mejoran ligeramente. Además cabe destacar que en ambas tareas se supera el *baseline* sin necesidad de usar el *BoW*.

	Género	**Edad**
Variables sin *BoW*	58.01%	57.49%
Variables con *BoW*	61.65%	63.33%

Table 3: Resultados obtenidos con el algoritmo *DMNBText*

5 Conclusiones y trabajo futuro

Tras observar los resultados obtenidos se puede concluir que no siempre es necesario el uso del *BoW* para realizar una tarea de *author profiling* ya que los resultados pueden ser muy parecidos o mejores que los obtenidos usando esta técnica. Sin embargo una elección más elaborada de las palabras que forman parte del *BoW* permite mejorar los resultados tanto para género como para edad. En esta parte también es muy importante elegir adecuadamente el algoritmo usado para el entrenamiento del modelo ya que también influye considerablemente en los resultados obtenidos.

En lo referente al tiempo de ejecución se ha hecho uso de la clase *StringBuilder* de *Java* a la hora de generar los ficheros para *Weka*, con esto se ha conseguido reducir el tiempo de ejecución casi 13 veces si se compara con el tiempo usado para generar el mismo fichero realizando la concatenación de *Strings* en *Java*. En cuanto al uso de expresiones regulares estas pueden penalizar la ejecución del código pero este proceso se podría distribuir sobre los nodos de un clúster ya que el proceso de los textos de cada autor es independiente.

En cuanto a posibles mejoras se puede intentar mejorar la generación del *BoW* y extraer otras características mediante el uso de diccionarios, por ejemplo se podría usar un diccionario con palabras en inglés para detectar el uso de este idioma en los textos en castellano ya que es algo bastante frecuente. Además el uso del diccionario puede ayudar a no detectar como como *character flooding* algunas palabras en inglés ya que en este idioma es más frecuente la aparición de dos consonantes seguidas. Otra mejora sería la eliminación del *BoW* de algunos *tokens* como direcciones *web* correos electrónicos o números ya que incluir esto en el *BoW* puede causar un sobreajuste del algoritmo. Por último se pueden usar otros tipos de *BoWs* como por ejemplo de n-gramas de caracteres o palabras para ver si los resultados mejoran.

References

[PAN] www.uni-weimar.de/medien/webis/events/pan-13/pan13-web/author-profiling.html

[DATASET] www.uni-weimar.de/medien/webis/corpora/corpus-pan-labs-09-today/pan-13/pan13-data/pan13-author-profiling-test-corpus2-2013-04-29.zip

[CODIGO] www.github.com/lmontes/text_mining