# Modelado de perfiles de usuario para la recomendación de contenido en Twitter

María Florencia Rodríguez[1] and Daniela Godoy[2]

[1] PLADEMA, Facultad de Ciencias Exactas, Universidad Nacional del Centro, Tandil, Buenos Aires, Argentina
[2] Isistan, Facultad de Ciencias Exactas, Universidad Nacional del Centro, Tandil, Buenos Aires, Argentina

**Resumen** En este trabajo se investigan diferentes mecanismos para deducir la semántica de los mensajes de Twitter con el fin de modelar perfiles de usuario. Se introducen y analizan métodos de procesamiento de lenguaje natural para plantear diferentes formas de inferir los intereses de los usuarios a partir de sus *tweets*. Luego, esas estrategias son comparadas para analizar el comportamiento al recomendar mensajes de otros usuarios.

**Keywords:** Twitter, perfil de usuario, minería de texto, recomendación de contenido, procesamiento de lenguaje natural

## 1. Introducción y motivación

Con más de 190 millones de usuarios y más de 65 millones de *posts* por día[3], Twitter[4] es el servicio de *micro-blogging* más popular entre las redes sociales. En Twitter los usuarios publican mensajes llamados *tweets*, que están formados por texto plano de corta longitud y se muestran en la página principal del usuario.

En contraste con otras redes sociales como por ejemplo Last.fm[5], donde se deducen los gustos musicales, o Flickr[6] que ofrece información para inferir los intereses de los usuarios en lugares o eventos, los temas discutidos en Twitter no están restringidos a un dominio en particular. Los usuarios de Twitter pueden discutir acerca de cualquier tema que les interese o preocupe.

Dada la gran cantidad de contenido que circula en Twitter uno de los principales desafíos de esta tecnología, es poder brindar al usuario información y contenido que resulte de su interés.

La representación de la semántica de las actividades individuales de Twitter y el modelado de los intereses de los usuarios permite personalización y de esta forma, compensar la sobrecarga de información. Como consecuencia de la

[3] http://www.techcrunch.com/2010/06/08/twitter-190-million-users/
[4] http://www.twitter.com/
[5] http://www.last.fm/
[6] http://www.flickr.com/

diversidad y lo cambiante que son los temas que se discuten en Twitter, los perfiles generados son beneficiosos para otras aplicaciones Web. Sin embargo, inferir automáticamente el significado semántico de mensajes de Twitter no es un problema trivial.

Si bien existen estudios acerca de las estructuras sociales en Twitter, se ha realizado escasa investigación en el área de comprensión de la semántica de las actividades individuales de Twitter y en deducir intereses de los usuarios.

Dada la brevedad de la longitud de los *tweets*, dar sentido a los mensajes individuales y explotarlos para el modelado de usuario representa un desafío. Además se deben aplicar diversos mecanismos de procesamiento de datos ya que es muy común que los usuarios utilicen abreviaciones y no respondan a las reglas léxicas y ortográficas del lenguaje.

## 2. Trabajos Relacionados

Muchas investigaciones sobre sistemas de recomendación en Twitter se enfocan en analizar una fracción de la red social para estudiar la propagación de patrones [13,14,15] o identificar usuarios que tienen mayor influencia[18,7].

Dong et al. [10] explotan Twitter para detectar y establecer un *ranking* de las URLs que han sido creadas recientemente y no han sido indexadas por los motores de búsqueda de la Web. Chen et al.[8] centran su investigación en recomendar URLs que fueron mencionadas en *tweets* y compara diferentes estrategias de selección y *ranking* de URLs.

Abel et al.[3] plantean diferentes técnicas para extraer el significado semántico de los *tweets*, enriquecer el contenido y a partir de ellos, modelan distintos tipos de perfiles de usuario. En su trabajo, observan que el 85 % de los *post* están relacionados con artículos de noticias, entonces se valen del contenido semántico de la noticia relacionada para enmarcar el mensaje dentro de un contexto específico.

Celik, Abel y Houben [6] exploran la semántica de los mensajes de Twitter enriqueciéndolas previamente como proponen Abel et al.[3], realizan un reconocimiento de nombres de entidades y obtienen conceptos relacionados mediante DBpedia[7].

Gao et al.[11] analizan cómo se relacionan los intereses personales de los usuarios con los temas de tendencia y estudian cómo la información de Twitter puede ser aprovechada para generar perfiles que reflejen los intereses de un usuario en noticias actuales.

Las investigaciones en el área han demostrado que la explotación del contenido de los *tweets* es un buen indicador para deducir los intereses de los usuarios. Entonces, en este trabajo se proponen diferentes estrategias de modelado a partir del procesamiento de las actividades de Twitter mediante distintas técnicas de minería de texto.

Abel et al.[4,2,3,1] sostienen que el reconocimiento de las entidades mencionadas en los *tweets* es un buen mecanismo de deducción de intereses. El trabajo

[7] http://wiki.dbpedia.org/

realizado por Celik, Abel y Houben [6] ha demostrado que una alternativa útil es extraer las entidades de los *tweets*, junto con los conceptos relacionados según DBpedia. En base a esas dos investigaciones, en este trabajo se evalúa el comportamiento de estas y otras estrategias de modelado de perfiles de usuario para la recomendación de *tweets* de otros usuarios que muestran intereses similares.

## 3. Cómo explotar Twitter para construir perfiles de usuario?

La propuesta de solución al problema de la recomendación de contenido se basa en la exploración del texto de los mensajes de Twitter para deducir intereses de usuario, y en base a ellos modelar el perfil. Ese perfil es utilizado como motor de clasificación para recomendar *tweets* que tengan contenido que coincida con los intereses del usuario.

Los perfiles de usuario son modelados mediante vectores, en los que cada elemento es un concepto que representa el interés de un usuario en determinado tema. Cada elemento del perfil es ponderado de acuerdo a una función que mide el nivel de importancia que tiene dicho elemento para el usuario.

### 3.1. Perfil de usuario

Un perfil de usuario se define como una lista de palabras claves relevantes para el usuario [17]. Dentro del proceso de personalización, los perfiles funcionan como una estructura de almacenamiento de información sobre las preferencias del usuario. En este caso, aportan información acerca del conjunto de conceptos sobre los que suele escribir.

En el contexto de Twitter, Abel et al.[4] define perfiles de usuario como una estructura de información sobre los intereses del usuario obtenidos a través de los *tweets*.

**Definition 1.** *Formalmente, el perfil de usuario $P(u)$ correspondiente al usuario $u$, del conjunto de usuarios $U$, es un vector de pares $(c, w(u,c))$ donde $c$ es un concepto, del conjunto de conceptos $C$, extraído de los tweets $T_u \in T$ del usuario $u \in U$, y $w(u,c)$ es el resultado de aplicar una función de ponderación $w$ (Eq. 1) que indica que tan significativo es el concepto $c$ para el usuario $u \in U$.*

$$P(u) = \{c, w(u,c) : u \in U; c \in C\}$$

En la Eq. 1 se puede observar la expresión correspondiente a $w(u,c).Aqui$, $f_(u,c)$ es la frecuencia de aparición del concepto $c \in C$ en el conjunto de *tweets* $T_U \in T$ del usuario $u \in U$.

De esta forma, el peso $w(u,c)$ de un concepto $c \in C$ queda determinado por el número de veces en los que el usuario $u \in U$ se refiere al concepto $c$ en el conjunto de *tweets* $T_u \in T$. Por ejemplo, un valor de ponderación $w(u, technology) = 5$ indica que el usuario $u \in U$ ha escrito 5 veces la palabra "*technology*" en su conjunto de *tweets* $T_u \in T$.

$$w(u,c) = \frac{f_{u,c}}{max f_{u,c}} \quad (1)$$

### 3.2. Estrategias de procesamiento

En la Definición 1 se ha introducido la noción de *concepto*. Un concepto es una unidad cognitiva de significado, es un elemento que aporta información acerca de los intereses de los usuarios. Durante el desarrollo de este trabajo hemos planteado tres estrategias de extracción de conceptos (ver Tabla 1).

| Dimensión de diseño | Alternativas de diseño |
|---|---|
| Tipo de perfil | (i) basado en bolsa de palabras<br>(ii) basado en entidad ó<br>(iii) basado en concepto |
| Fuente de información | (i) únicamente tweets |
| Restricciones temporales | (i) sin restricciones |

**Cuadro 1.** Espacio de diseño de las estrategias de modelado de perfiles de usuario.

**Estrategia basada en bolsa de palabras.** Una primera estrategia -ingenua- consiste en procesar cada *tweet* como una tira de *tokens*, y luego calcular el peso de cada *token* según la Eq. 1 para armar el perfil de usuario basado en bolsa de palabras $P_B(u)$. Este mecanismo considera todas las palabras que forman parte del texto como conceptos, de modo que cada *tweet* es transformado en una tira de *tokens*.

**Definition 2.** *Formalmente, un tweet $t_u \in T$ es representado mediante un vector $t_u = \{(c_1, f_1), (c_2, f_2), ...(c_n, f_n)\}$ donde cada elemento $c_i$ corresponde a un término extraído de $t_u$ y $f_i$ es la frecuencia con la que $c_i$ aparece en $t_u$.*

*El perfil $P_B(u)$ para un usuario $u \in U$ es el conjunto de todos los tokens que aparecen en el conjunto de tweets $T_u \in T$ de $u$, junto con los valores de ponderación $w_i$ para cada $x_i$ calculado en base a la función $w(u,x)$.*

*El valor $w(u,x)$ es el peso asociado al token $x$ para el usuario $u \in U$ correspondiente al conjunto $U$ de usuarios.*

De esta forma, cada elemento del vector $P_B(u)$, es un par $(x, w(u,x))$ donde $x$ corresponde a un *token* y $w(u,x)$ es la función de ponderación definida en Eq. 1.

El proceso de extracción de conceptos se realizó utilizando el algoritmo de *tokenización* de la librería Lucene[8]. Lucene es un API para la recuperación de información. Es un proyecto *open source*, implementado en Java, es miembro del Apache Software Foundation y se distribuye bajo Apache Software Licence.

[8] http://lucene.apache.org/core/

El algoritmo *tokenizador* de Lucene acepta una cadena de caracteres como entrada (un *tweet*), luego procesa la cadena para dividirla en palabras individuales, y emite una cadena de componentes léxicos como salida.

**Estrategia basada en bolsa de palabras sin *stop words*.** Un refinamiento aplicado a la estrategia anterior, consiste en remover las palabras vacías (*stop words*) del vector $P_B(u)$, manteniendo la definición de la función de ponderación $w(u, c)$ y alterando $P_B(u)$, de manera que las palabras que corresponden a *stop words* sean eliminadas.

Las palabras vacías son términos que por su frecuencia y/o semántica no poseen valor discriminatorio alguno, es decir no permiten distinguir un texto de otro en una colección. Habitualmente se trata de artículos, pronombres, preposiciones, verbos muy frecuentes, adverbios, etc.

**Definition 3.** *El perfil basado en bolsa de palabras sin palabras vacías $P_S(u)$ para un usuario $u \in U$ es el conjunto de todos los tokens que aparecen en el conjunto de tweets $T_u \in T$ de $u$, excepto los que forman parte del conjunto de stop words, junto con los valores de ponderación $w_i$ para cada $x_i$ calculado en base a la función $w(u, x)$.*

*El valor $w(u, x)$ es el peso asociado al token $x$ para el usuario $u \in U$ corresponden al conjunto de usuarios.*

Cada elemento del vector $P_S(u)$, es un par $(x, w(u, x))$ donde $x$ corresponde a un *token* que no es *stop word* y $w(u, x)$ es la función de ponderación definida en Eq. 1.

La eliminación de *stop words* se realiza chequeando el contenido del texto contra un listado de palabras. Lucene provee un módulo procesador de texto, llamado *StandardAnalyzer*, que realiza una tokenización que elimina *stop words*. Durante el proceso de conversión a *token*, el *StandardAnalyzer* extrae el texto que se debe analizar mientras se le aplica lógica de transformación como ser el uso de *stop words*.

Los enfoques planteados en esta sección realizan un análisis léxico, y si bien obtienen el conjunto de conceptos a los que el usuario se refiere con mayor frecuencia, no construyen perfiles con valor semántico. Entonces, planteamos dos estrategias que realizan análisis semántico. Una de ellas está basada en reconocimiento de nombres de entidades y la otra en extracción de conceptos de DBpedia.

**Estrategia basada en reconocimiento de nombres de entidades.** El reconocimiento de nombres de entidades (o NER) etiqueta la secuencia de palabras de un *tweet* como nombre de cosas, por ejemplo nombres de personas, compañías, o lugares, y genera perfiles basados en entidad $P_E(u)$. Utilizando esta técnica, un perfil basado en entidades es modelado en función de los intereses del usuario en un determinado conjunto de entidades, como las personas, organizaciones o eventos. Los *tweets* son definidos como un vector donde cada elemento contiene las entidades mencionadas y la frecuencia con que aparecen esas entidades.

**Definition 4.** *El perfil basado en entidad $P_E(u)$ para un usuario $u \in U$ es el conjunto de nombres de entidades $e_i \in E$ que aparecen en el conjunto de tweets $T_u \in T$ de u, junto con los valores de ponderación $w_i$ para cada $e_i$ calculado en base a la función $w(u, e)$.*

*El valor $w(u, e)$ es el peso asociado con la entidad e para el usuario u. E y U corresponden al conjunto de entidades y de usuarios respectivamente.*

De esta forma, cada elemento del vector $P_E(u)$, es un par $(e, w(u, e))$ donde $e$ corresponde a una entidad y $w(u, e)$ es la función de ponderación definida en Eq. 1.

Las entidades son extraídas directamente a partir del contenido del *tweet* utilizando TwitIE[5]. TwitIE es un *pipeline* GATE (General Architecture for Text Engineering) [9] diseñado para la extracción de entidades sobre *tweets*.

GATE es una suite de herramientas Java originalmente desarrollada por la Universidad de Sheffield utilizada para las tareas de procesamiento de lenguaje natural, incluyendo extracción de información en varios idiomas. El *framework* de código abierto GATE se encuentra pre-empaquetado con el *pipeline* ANNIE.

ANNIE (A Nearly-New Information Extraction System) consiste en una serie de módulos para la extracción de la información que implementan los siguientes procesos: *tokenizador*, divisor de sentencias, etiquetador POS (POS *tagger*), listas *gazetteer*, un reconocedor de nombres de entidades y un detector de co-referencias.

**Estrategia basada en conceptos de DBpedia.** A menudo, los usuarios suelen publicar *tweets* acerca de determinados conceptos que no corresponden a entidades; entonces, planteamos una estrategia que consiste en mapear el contenido de los *tweets* a recursos de DBpedia (que es la representación estructurada de Wikipedia[9]).Así, los conceptos que forman parte del perfil, pueden ser contextualizados con el contenido que provee Wikipedia.

**Definition 5.** *El perfil basado en conceptos de DBpedia $P_D(u)$ para un usuario $u \in U$ es el conjunto de conceptos $d_i \in D$ que aparecen en el conjunto de tweets $T_u \in T$ de u, junto con los valores de ponderación $w_i$ para cada $d_i$ calculado en base a la Eq. 1 $w(u, d)$.*

*El valor $w(u, d)$ es el peso asociado con la entidad e para el usuario u. E y D corresponden al conjunto de conceptos de DBpedia y de usuarios respectivamente.*

Cada elemento del vector $P_B(u)$, es un par $(d, w(u, d))$ donde $d$ corresponde a un concepto de DBpedia y $w(u, d)$ es la función de ponderación definida en Eq. 1.

Para construir perfiles utilizando esta estrategia, se toma como entrada el conjunto de *tweets* y se intenta buscar el recurso correspondiente en DBpedia.

[9] http://www.wikipedia.org/

### 3.3. Perfil combinado

El tipo de conceptos que forman el conjunto $C$ determina el tipo de perfil, pudiendo ser una tira de *tokens* con o sin *stop-words* -$P_B(u)$ o $P_S(u)$ respectivamente-, un vector de entidades $P_E(u)$ o un conjunto de conceptos $P_D(u)$. Además, hemos planteado perfiles que combinan las estrategias explicadas hasta el momento. Formalmente un perfil de usuario mixto, que combina las cuatro estrategias se define como sigue:

**Definition 6.** *Un perfil de usuario $P_M(u)$, para un usuario $u \in U$ que combina los perfiles $P_B(u)$, $P_S(u)$, $P_E(u)$ y $P_D(u)$ generados a través de las estrategias basadas en bolsa de palabras (con y sin palabras vacías), en entidades y en recursos de DBpedia, respectivamente, se define mediante la función.*

$$P_M(u) = \{n1 * P_B(u) + n2 * P_S(u) + n3 * P_E(u) + \\ + n4 * P_D(u) : u \in U, n1 + n2 + n3 + n4 = 1\}$$

*Donde $n1$, $n2$, $n3$ y $n4$ indican la ponderación (es decir, el peso) de cada perfil (estrategia).*

Según la Definición 6, valores $n1 = n2 = n3 = n4 = 0{,}25$ indican que todos los perfiles tendrán el mismo peso; valores tales como $n1 = n2 = 0$ y $n3 = 0,5$ y $n4 = 0,5$ indican que los perfiles basados en bolsa de palabras no tendrán peso en la ponderación, y la influencia será dada por los perfiles basados en entidades y conceptos de DBpedia de forma equitativa.

## 4. Modelo de recomendación

La creación de perfiles de usuario tiene como objetivo armar perfiles en base a los intereses extraídos a partir del contenido de los *tweets*. Dado un usuario, esos perfiles son usados como modelos, para clasificar si el *tweet* contiene información que le interesa (o no) a ese usuario.

El conjunto de elementos que usamos para el desarrollo de este trabajo está compuesto por un *dataset* formado por 2316204 *tweets* de 1619 usuarios[10]. Para los experimentos consideramos los *post* de 500 usuarios que cuentan con una cantidad mínima de 100 *tweets*.

La evaluación de las diferentes estrategias fue realizada mediante *Holdout*. Esta técnica de validación consiste en dividir el *dataset* en dos partes: una utilizada para realizar la construcción del perfil (llamada *conjunto de entrenamiento*), y otra para realizar evaluaciones (*conjunto de evaluación*).

A su vez, los *tweets* del conjunto de evaluación fueron etiquetados según dos clases: una formada por *tweets* de la clase "*Interesa*" (*ejemplos positivos*), y otra clase formada por *tweets* marcados como "*No interesa*" (*ejemplos negativos*).

[10] http://wis.ewi.tudelft.nl/umap2011/#dataset/

Para cada uno de los usuarios, el subconjunto de clase "*Interesa*" fue obtenido de los *tweets* del mismo usuario. Los *tweets* que forman parte de los *ejemplos negativos*, fueron seleccionados a partir del conjunto de *tweets* que no forman parte del conjunto de *tweets* o *retweets* del usuario a analizar.

### 4.1. Algoritmo recomendador de *tweets*

Con el fin de evaluar la efectividad de las estrategias de modelado hemos aplicado un algoritmo basado en clasificar contenido de acuerdo a la similitud entre el *tweet* a recomendar y el perfil de usuario. De esta forma, el problema de la recomendación se convierte en una búsqueda y clasificación, donde el perfil de usuario es utilizado como motor de clasificación.

**Definition 7.** *Dado un perfil de usuario $P(u)$ representado mediante un vector y un conjunto de tweets $T = \{P(t_1), \dots P(t_n)\}$ representados mediante perfiles (con la misma representación vectorial), el algoritmo de recomendación clasifica los tweets según la similitud de coseno:*

$$sim_{coseno}(P(u), P(t_i)) = \frac{P(u).P(t_i)}{||P(u)||.||P(t_i)||} \tag{2}$$

**Función de similitud.** La recomendación consiste en calcular la similitud entre el perfil de cada usuario $P(u_i)$ y los perfiles $P(t_j)$ de cada uno de los *tweets* $t_j \in T$ que forman parte del conjunto de prueba.

Medimos la similitud entre dos perfiles de usuario mediante el *coseno* (ver Eq. 2). La función coseno es una medida de similaridad de patrones de datos, que permite comparar usuarios o documentos, ya que el coseno mide el ángulo entre dos vectores en un espacio N-dimensional [16,12]. Cuando los perfiles son idénticos, la función da como resultado 1, mientras que por el contrario, para perfiles totalmente diferentes, el coseno da cero.

De esta forma, si el resultado de la función entre el perfil del usuario $P(u_i), u_i \in U$ y un *tweet* del conjunto de prueba $t_j \in T$ es mayor a determinado umbral, el *tweet* $t_j$ es clasificado como "*Interesa*". Sino, como "*No interesa*".

## 5. Resultados experimentales

En esta sección presentamos los resultados de los experimentos obtenidos luego de aplicar el algoritmo de recomendación (ver Definición 7) para las diferentes estrategias de construcción de perles de usuario, especificadas en la Sección 3.

### 5.1. Evaluación por estrategia

Si comparamos el comportamiento de las estrategias con respecto a la tasa de error (Fig. 1) podemos ver que la que menor error posee es la estrategia basada en bolsa de palabras con eliminación de palabras vacías. Luego sigue la estrategia

basa en bolsa de palabras y finalmente, las estrategias que mayor tasa de error presentan, son las basadas en NER y en extracción de conceptos de DBpedia.

De estos resultados se puede deducir que las palabras vacías no aportan información acerca del contenido semántico del *tweet*, es decir, no son buenos indicadores de los intereses de usuario y es necesario eliminarlas si se desea realizar una recomendación con un bajo margen de error.

La estrategia que presenta mayor exactitud (ver Fig. 1) es la basada en bolsa de palabras sin *stop words*, y por una menor diferencia continúa la estrategia basada en bolsa de palabras. Luego siguen las estrategias basadas en NER y extracción de recursos de DBpedia.

Conforme a los resultados anteriores, si analizamos el valor F (ver Fig. 1) podemos observar que la construcción de perfiles con la estrategia basada en bolsa de palabras sin palabras vacías es la que presenta mayor valor. Luego sigue la estrategia basada en bolsa de palabras y finalmente las que realizan un análisis semántico.

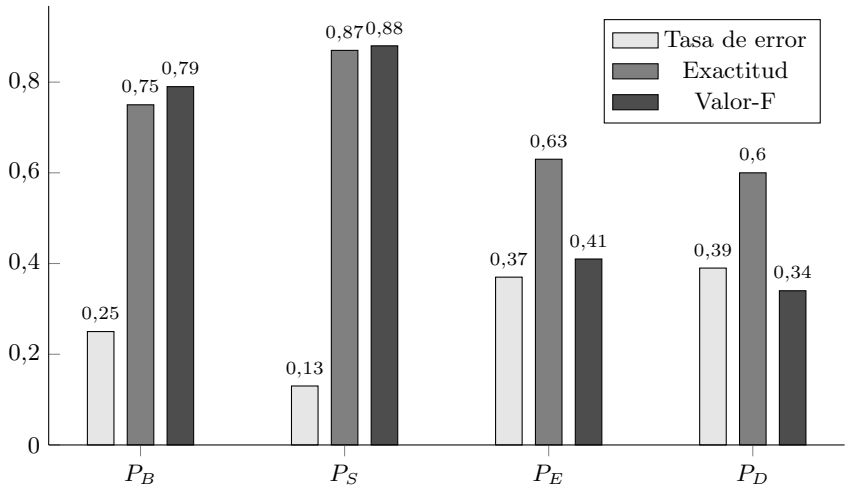


**Figura 1.** Tasa de error, exactitud y valor-F para cada estrategia.

Como se puede observar en la Tabla 2, la estrategia que genera modelos de menor tamaño es la estrategia basada en NER. Luego sigue la estrategia basada en extracción de conceptos de DBpedia, y finalmente las basadas en bolsas de palabras.

La estrategia basada en bolsas de palabras con eliminación de palabras vacías construye perfiles con una cantidad de palabras muy cercana a la cantidad de la estrategia basada en bolsas de palabras sin refinamientos. De la misma forma, la estrategia basada en extracción de recursos y la estrategia basada en NER construyen perfiles de un tamaño similar. Para algunos perfiles, la estra-

tegia basada en extracción de recursos de DBpedia genera modelos con menos cantidad de conceptos que la estrategia basada en entidades, sin embargo, este comportamiento aparece en la menor cantidad de casos.

**Reducción de tamaño del modelo para las estrategias basadas en bolsa de palabras.** Luego de realizar el análisis del comportamiento de cada estrategia, se puede concluir que la que mejores resultados arroja en la recomendación es la estrategia que utiliza todos los términos que aparecen en el *tweet* eliminando las palabras vacías.

El principal problema que presenta es el gran tamaño del modelo, entonces se ha analizado el comportamiento del algoritmo al reducir la cantidad de conceptos por perfil, y se ha llegado a la conclusión de que usando el 10 % de los conceptos se reduce notablemente la longitud del modelo y además, se agrega mayor cantidad de aciertos en la clasificación, debido a que elimina una gran cantidad de palabras a las que el usuario no se refiere comúnmente.

| Perfil | Cantidad promedio de conceptos |
|---|---|
| $P_B(u)$ | 3985 |
| $P_S(u)$ | 3358 |
| $P_E(u)$ | 379 |
| $P_D(u)$ | 544 |

**Cuadro 2.** Tamaño promedio de los perfiles para cada estrategia.

**Conclusión.** Si bien la evaluación indica que las estrategias que presentan menor tasa de error y mayor exactitud son las basadas en bolsa de palabras, son estrategias que construyen perfiles carentes de contenido semántico y los conceptos que ocurren con mayor peso se refieren a términos usados frecuentemente por el usuario. Al mismo tiempo, las estrategias basadas en NER y extracción de recursos de DBpedia filtran esos términos y en consecuencia, tienen una tendencia a tener mayor número de errores durante la etapa de evaluación del modelo.

Además, se observó que este comportamiento es consecuencia del hecho de que las estrategias basadas en NER y en extracción de conceptos filtran contenido no interesante para el usuario con 100 % de precisión, pero no sucede lo mismo al clasificar contenido de interés para el usuario.

Las estrategias basadas en NER y extracción de conceptos de DBpedia son las que tienen un mayor nivel de entendimiento. Las otras estrategias no aportan información semántica detallada y en consecuencia agregan mayor complejidad a la interpretación. Sin embargo, las estrategias basadas en bolsa de palabras presentan mayor exactitud y menor tasa de error.

### 5.2. Evaluación para perfiles combinados

Para analizar cómo varía la recomendación de un perfil generado a partir de las combinaciones de las estrategias planteadas, hemos construido perfiles según la Definición 6, asignando distintos pesos a cada tipo de perfil como se muestra en la Tabla 3.

| Perfil $P_M(u)$ | $n_1$ | $n_2$ | $n_3$ | $n_4$ |
|---|---|---|---|---|
| $P_{M_0}(u)$ | 0,25 | 0,25 | 0,25 | 0,25 |
| $P_{M_1}(u)$ | 0 | 0,33 | 0,33 | 0,33 |
| $P_{M_2}(u)$ | 0 | 0,5 | 0,3 | 0,2 |
| $P_{M_3}(u)$ | 0 | 0,5 | 0,5 | 0 |
| $P_{M_4}(u)$ | 0 | 0,5 | 0,5 | 0 |
| $P_{M_5}(u)$ | 0 | 0,7 | 0,3 | 0 |

**Cuadro 3.** Pesos asignados a cada estrategia para armar perfiles combinados.

Según se puede observar en la Tabla 4, las ponderaciones que mayor tasa de error arrojan son las de los perfiles $P_{M_1}(u)$, $P_{M_3}(u)$ y $P_{M_4}(u)$. En estos tres casos a los perfiles se les ha asignado el mismo valor de ponderación; entonces, la primer conclusión que se puede obtener es que al asignarle el mismo peso a todos los perfiles se obtiene una mayor tasa de error. Luego, las combinaciones que mayor tasa de error presentan son las $P_{M_0}(u)$ y $P_{M_2}(u)$. En estos casos los perfiles basados en bolsa de palabras tienen un mayor peso. Finalmente, el valor más bajo de tasa de error se obtiene mediante la ponderación de $P_{M_5}(u)$, que es la combinación que utiliza el 10 % de los términos más frecuentes para la estrategia basada en bolsa de palabras (sin *stop words*).

Por otro lado, las combinaciones que arrojan una exactitud más baja son la $P_{M_1}(u)$, $P_{M_3}(u)$ y $P_{M_4}(u)$; luego siguen $P_{M_0}(u)$ y $P_{M_2}(u)$ y finalmente, la combinación que tiene mayor exactitud a la hora de utilizarla como modelo recomendador es la $P_{M_5}(u)$.

Como se puede ver en la Tabla 4, la ponderación que tiene un valor F más cercano al óptimo es la combinación $P_{M_5}(u)$; luego siguen las $P_{M_0}(u)$ y $P_{M_2}(u)$ y finalmente, las que peor valor F presentan son las de las $P_{M_1}(u)$, $P_{M_3}(u)$ y $P_{M_4}(u)$.

### 5.3. Comparación entre perfil basado en entidad, basado en bolsa de palabras y combinación $P_{M_5}$

Al comparar las métricas arrojadas por el perfil combinado según $P_{M_5}(u)$ con los perfiles basado en bolsa de palabras y basado en entidad, se puede observar que el perfil combinado tiene menor tasa de error que el perfil basado en entidad (ver Fig. 2). A su vez, este valor es muy cercano al del perfil basado en bolsa de palabra. Lo mismo sucede con la exactitud y el valor F. Sin embargo, la principal ventaja que incorpora el perfil combinado con respecto al perfil basado en bolsa

| $P_M(u)$ | $n_0$ | $n_1$ | $n_2$ | $n_3$ | **Tasa de error** | **Exactitud** | **Valor F** |
|---|---|---|---|---|---|---|---|
| $P_{M_0}(u)$ | 0,25 | 0,25 | 0,25 | 0,25 | 0.23 | 0.77 | 0.7 |
| $P_{M_1}(u)$ | 0 | 0,33 | 0,33 | 0,33 | 0,3 | 0,7 | 0,57 |
| $P_{M_2}(u)$ | 0 | 0,5 | 0,3 | 0,2 | 0,22 | 0,77 | 0,71 |
| $P_{M_3}(u)$ | 0 | 0,5 | 0,5 | 0 | 0,3 | 0,7 | 0,57 |
| $P_{M_4}(u)$ | 0 | 0,5 | 0,5 | 0 | 0,3 | 0,7 | 0,57 |
| $P_{M_5}(u)$ | 0 | 0,7 | 0,3 | 0 | 0,15 | 0,85 | 0,83 |

**Cuadro 4.** Tasa de error, exactitud y valor F de los perfiles combinados.

de palabra es un aumento en la precisión al momento de filtrar el contenido que no es interesante para el usuario. Por otro lado, posee la desventaja de generar un modelo de mayor tamaño al de los perfiles que lo componen.

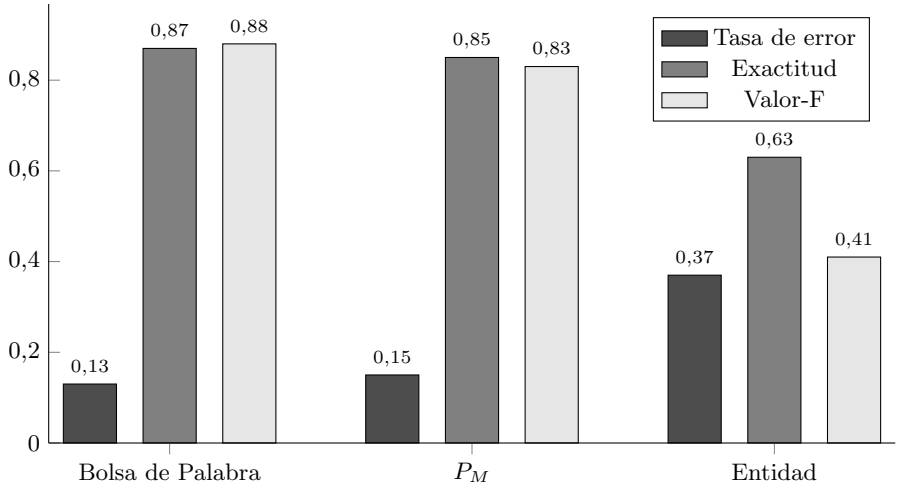


**Figura 2.** Tasa de error, exactitud y valor-F para las estrategias basada en bolsa de palabras, entidades y $P_M$.

## 6. Conclusión y Trabajo Futuro

En este trabajo se ha realizado un análisis de distintas técnicas de recuperación de información para recomendación de contenido en Twitter. Se ha centrado el análisis en la explotación de la información contenida en los *tweets* para inferir los intereses de los usuarios. A partir de los intereses se construyen perfiles de usuarios que son utilizados como motores de recomendación de *tweets*. La recomendación se ha realizado comparando el perfil del usuario con el ítem a recomendar. Luego el comportamiento fue validado usando un conjunto de datos

estándar del área y una metodología de evaluación conocida como *Holdout* para la obtención de métricas de eficiencia.

La información extraída de Twitter puede ser utilizada por diversas aplicaciones que se valen de los perfiles de usuario para recomendar contenido. El presente trabajo ha sido orientado a la recomendación de mensajes, sin embargo puede ser extendido para recomendar otro tipo de contenido, como por ejemplo noticias, otros usuarios de la red, páginas web, etc.

Además, se pueden plantear otras alternativas para extraer información. Por ejemplo los *Hashtags* suelen ser buenos indicadores del contenido de los mensajes. Por otro lado, los *tweets* pueden ser contextualizados explotando los *links* que incluyen o vinculándolos a conferencias o eventos.

Debido a los intereses y preferencias cambiantes de los usuarios, una nueva alternativa es analizar las variaciones que presentan los usuarios a través del tiempo y analizar cómo los patrones temporales influyen en los intereses de los usuarios. Otro factor a considerar es la influencia que presentan los temas de tendencia sobre los intereses de los usuarios.

Finalmente, para validar la calidad de las recomendaciones se puede incluir la opinión de los usuarios reales.

## Referencias


1. Abel, F., Gao, Q., Houben, G.-J., and Tao, K. Supporting website: code, datasets and additional findings (2011).
2. Abel, F., Gao, Q., Houben, G.-J., and Tao, K. Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In *Proceedings of the 3rd International Web Science Conference* (2011), ACM, p. 2.
3. Abel, F., Gao, Q., Houben, G.-J., and Tao, K. Analyzing user modeling on twitter for personalized news recommendations. In *User Modeling, Adaption and Personalization*. Springer, 2011, pp. 1–12.
4. Abel, F., Gao, Q., Houben, G.-J., and Tao, K. Semantic enrichment of twitter posts for user profile construction on the social web. In *The Semanic Web: Research and Applications*. Springer, 2011, pp. 375–389.
5. Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., and Aswani, N. Twitie: An open-source information extraction pipeline for microblog text. In *RANLP* (2013), pp. 83–90.
6. Celik, I., Abel, F., and Houben, G.-J. Learning semantic relationships between entities in twitter. In *Web Engineering*. Springer, 2011, pp. 167–181.
7. Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, P. K. Measuring user influence in twitter: The million follower fallacy. *ICWSM 10*, 10-17 (2010), 30.
8. Chen, J., Nairn, R., Nelson, L., Bernstein, M., and Chi, E. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), ACM, pp. 1185–1194.
9. Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th annual meeting on association for computational linguistics* (2002), Association for Computational Linguistics, pp. 168–175.

10. Dong, A., Zhang, R., Kolari, P., Bai, J., Diaz, F., Chang, Y., Zheng, Z., and Zha, H. Time is of the essence: improving recency ranking using twitter data. In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 331–340.
11. Gao, Q., Abel, F., Houben, G.-J., and Tao, K. Interweaving trend and user modeling for personalized news recommendation. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01* (2011), IEEE Computer Society, pp. 100–103.
12. Krishnapuram, R., Joshi, A., Nasraoui, O., and Yi, L. Low-complexity fuzzy relational clustering algorithms for web mining. *Fuzzy Systems, IEEE Transactions on 9*, 4 (2001), 595–607.
13. Kwak, H., Lee, C., Park, H., and Moon, S. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 591–600.
14. Lerman, K., and Ghosh, R. Information contagion: An empirical study of the spread of news on digg and twitter social networks. *ICWSM 10* (2010), 90–97.
15. Sakaki, T., Okazaki, M., and Matsuo, Y. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 851–860.
16. Salton, G. Developments in automatic text retrieval. *Science 253*, 5023 (1991), 974.
17. Tang, J., Yao, L., Zhang, D., and Zhang, J. A combination approach to web user profiling. *ACM Transactions on Knowledge Discovery from Data (TKDD) 5*, 1 (2010), 2.
18. Weng, J., Lim, E.-P., Jiang, J., and He, Q. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining* (2010), ACM, pp. 261–270.