PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC3633 - Sistemas Recomendadores
27 de noviembre de 2016

Informe Final

Jorge Schellman

Resumen

Las plataformas de redes sociales están integradas en ciertas aplicaciones de críticas de ítems. Esto permite la fácil expresión de un sentimiento sobre un ítem, por ejemplo un video. Particularmente en Goodreads (GR) los usuarios pueden tweetear desde la aplicación sobre el libro que consumieron junto a un rating. Estos tweets tienen un contenido predefinido por la aplicación el cual es editable por los usuarios. Este trabajo pretende aprovechar estos tweets para saber el consumo de usuarios sobre libros junto al rating dado para construir un sistema recomendador basado en contenido. Se hace uso tanto de información de los tweets de los usuarios de GR como información rescatable de los libros en la misma aplicación para construir diversos modelos que permitan representar usuarios como ítems en espacios de vectores y construir listas de recomendación en base a los ítems más similares a un usuario según similaridad por distancia coseno y distancia Jaccard. Se encontró que los métodos que mejor se adaptan a este problema son TF-IDF con distancia coseno obteniendo un F-measure de 0.108 y BoW con distancia Jaccard con F-measure=0.32, ambos para una lista de recomendación de 10 ítems. Se vio finalmente que el uso drásticamente distinto de vocabulario entre Twitter y GR puede ser un problema solucionable según la elección de métrica de distancia y modelos.

1. Contexto y Estado del Arte

Plataformas de redes sociales como Twitter, Facebook, Tumblr, etc., permiten a los usuarios expresar sus opiniones de manera rápida y libre, lo que hace que éstos sean una rica fuente de información de la vida diaria de las personas. Estos sistemas de medios sociales incluso fomentan a los usuarios a proveer grandes volúmenes de información, los que anteriormente estarían reacios a compartir, tales como ubicación geográfica, edad, intereses, redes de amigos o las mismas opiniones que ahora fácilmente pueden ser capturadas monitoreando la interacción del usuario con el sistema. Estas nuevas fuentes pueden permitir a los sistemas recomendadores enfocados en recomendaciones sociales a refinar sus técnicas y estrategias. Por otro lado, los recomendadores pueden ayudar a los usuarios de redes sociales en solventar sus necesidades de información al reducir drásticamente la sobrecarga de información para decidir, por ejemplo, qué libro leer o qué película ver.
Dada la rápida velocidad de difusión de información, varias aplicaciones web han sido integradas con plataformas de redes sociales para que sus usuarios expresen fácilmente su sentimiento sobre ciertas cosas. En el caso particular de Twitter, aplicaciones como IMDb, Youtube, Pandora o Goodreads permiten que los usuarios expresen rápidamente sus opiniones sobre los ítems (películas, videos,

canciones o libros, respectivamente) en tweets con contenido predefinido por la aplicación pero que pueden ser editados por los usuarios. Estos tweets con contenido predefinido tienen cierta estructura dependiendo de la aplicación de la que prevenga, pero usualmente consisten en un texto que presenta el rating dado por el usuario sobre el item (en la escala definida por la aplicación) seguido del nombre del ítem y la URL que es la ruta del ítem en la aplicación. Estos datos pueden ser aprovechados para diseñar un sistema recomendador que considere el contenido de estos tweets como datos de input.

Existe trabajo previo que investigue algoritmos de recomendación que de alguna manera tome en cuenta el uso de redes sociales [1][2][3]. Particularmente hay trabajos que se refieren a Twitter, como Hamed et al. (2015) en el que usan datos de tweets de diferentes aplicaciones web para mejorar el desempeño de evaluación de *engagement* del usuario en la aplicación. Ellos propusieron un método adaptativo basado en aprendizaje *multi-task* para detectar involucramiento positivo en tweets con contenido predefinido proveniente de varios dominios (los mismos 4 mencionados anteriormente). Encontraron que este método puede transferir conocimiento entre los dominios para mejorar la evaluación de *engagement* del usuario.
Rodriguez et al. (2016) proponen uno de los últimos métodos, usando lógica difusa, para recomendar usuarios a quienes seguir (hacerles *follow*). Adicionalmente hay algoritmos para recomendar hashtags como el de Dhingra et al. (2016), en el que hacen una extensión a nivel de tweet del método de generación de *embeddings* de palabras word2vec, llamado *tweet2vec*. Una de las ventajas importantes es que este modelo logra capturar el habla informal y coloquial de los tweets al extender enormemente el set tradicional de caracteres unicode y al tomar en cuenta el uso de jergas y corriente aparición de errores de ortografía y abreviaciones. En sus experimentos, tweet2vec logra mejores resultados que el baseline de embedding a nivel de palabra word2vec, el que se suele extender a nivel de documento al sumar o promediar los vectores de palabras que lo constituyen.
Otros trabajos analizan recomendaciones de tweets de otros usuarios, retweets, noticias y nuevos *followers* que difundan los mensajes del usuario activo[1][3]. No hay trabajo previo alguno que, aprovechando la generación de tweets con contenido predefinido, haga un modelo a partir del consumo y rating puesto por usuario de un ítem para recomendar a usuarios de Twitter ítems del dominio en el que ellos estén familiarizados (es decir, que hayan consumido varios ítems de un cierto dominio). Este proyecto pretende crear y evaluar tales métodos mediante un sistema recomendador basado en contenido, construido a base de distintas técnicas de procesamiento de lenguaje natural (NLP).

2. Dataset

El dataset inicial es un corpus de 4098 documentos JSON recolectado por Hamed et al. (2015). Cada documento corresponde a toda la información rescatable de tweets de un mismo usuario a partir de las APIs públicas de Twitter. Estos tweets tienen formato predefinido de 4 aplicaciones web. La información que incluyen es: texto e id del tweet, URLs contenidos en el tweet y su span (en unidades de índice), cantidad de retweets, si fue *favorited*, informacion del usuario que emitió tal tweet como su *screen name*, su ID, descripción del perfil dado por el mismo usuario, locación geográfica, nombre real (muestrado en su perfil), fecha de creación de la cuenta, número de followers, etc. De todos los documentos, 73 están vacíos, por lo que efectivamente se tienen tweets de 4025 usuarios.
Estadísticas del dataset original se resumen en la figura 1. Por simplificación, y dado que este trabajo no es de análisis en ámbitos cross-domain, se decidió por usar sólo la información proveniente del sitio de críticas de libros Goodreads.

[1] Útil para gente de negocios, políticos, etc.

	IMDb	YouTube	Goodreads	Pandora
Items type	movie	video clip	book	music
# of tweets	100,206	239,751	65,445	98,212
# of users	6,852	6,480	3,813	3,312
# of items	13,502	154,041	31,558	32,321
Avg eng.	0.1097	0.4737	0.1632	0.0778
% of tweets with eng>0	4.139	14.193	6.931	6.285

Figura 1: Características del dataset.

Este dataset se ha ido actualizando. Originalmente se tiene lo que aparece en la figura 1. Con lo que se cuenta ahora -y que seguiremos refiriéndonos a éste como el 'dataset original'- lo vemos en el cuadro 1. Esta información es en su gran mayoría de tweets con contenido predefinido. A pesar de que los usuarios pueden editar el mensaje, esto no es suficiente para capturar el habla coloquial de Twitter ni para hacer un modelo de usuario apropiado, pues se requiere de mayor información. Dado esto, se complementó el dataset con una recolección de mayor cantidad de tweets de los mismos usuarios, por lo que ahora se tienen más de 2 millones de tweets. La recolección se hizo con interfaces para las APIs públicas de Twitter en R y con bindings de Selenium para Python[2] para la automatización de tareas de scrapping.

Cuadro 1: Estadísticas del dataset

# de usuarios	4024
# de libros	46811
#total de tweets (dataset original)	81434
#total de tweets (ahora)	2334851

Las distribuciones de tweets por usuario se pueden observar en la figura 2 y 3. En la figura 2 vemos claramente una distribución como Pareto, en el que la gran mayoría de los tweets totales se le atribuyen a una cantidad reducida de usuarios. De hecho, sólo 808 de 4024 usuarios (el 20 %) han emitido 55836 de los tweets, correspondiente a un aproximadamente 70 % del total, lo que se aproxima a la regla del 80/20. En la figura 3, con los datos que se tienen ahora, podemos distinguir tres secciones: la primera es de unos 600 usuarios quienes tweetean pero en cantidades bajas. Luego, aproximadamente hasta la marca del usuario 3800, son de usuarios que utilizan activamente twitter. La tercerca parte son de los usuarios que se llevan la mayor cantidad de tweets.

En la figura 4 vemos la distribución de ratings que se rescatan de los tweets con contenido predefinido. Tiene la típica forma de una distribución de ratings que los usuarios proveen en una aplicación web. Son en total 80904 ratings, los que vienen mayoritariamente del dataset original. Notar que hay una columna de rating 0. Estos son tweets que fueron capturados incorrectamente según los propósitos de la expresión regular de la recolección original.

[2]http://selenium-python.readthedocs.io/

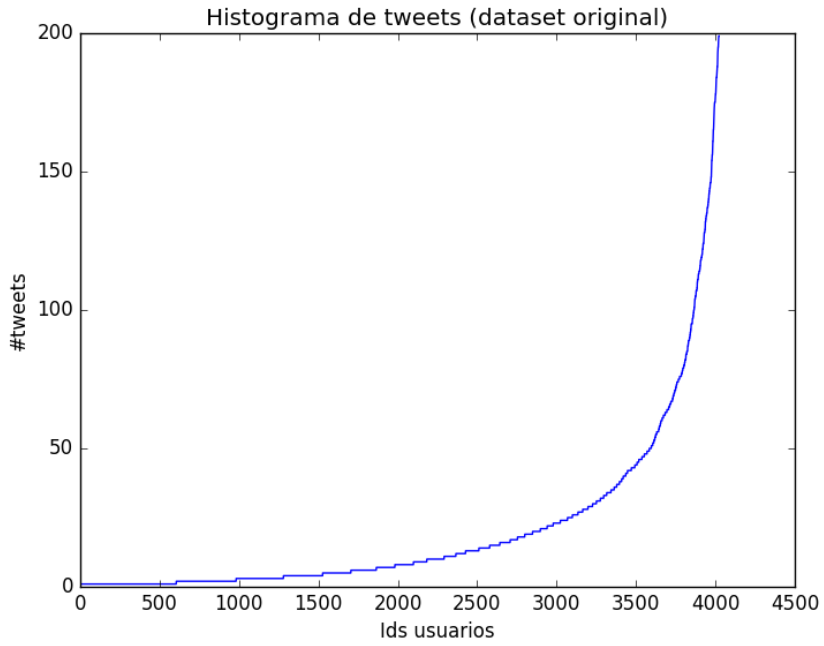

Figura 2: Cantidad de tweets emitidos por usuario en el dataset original.

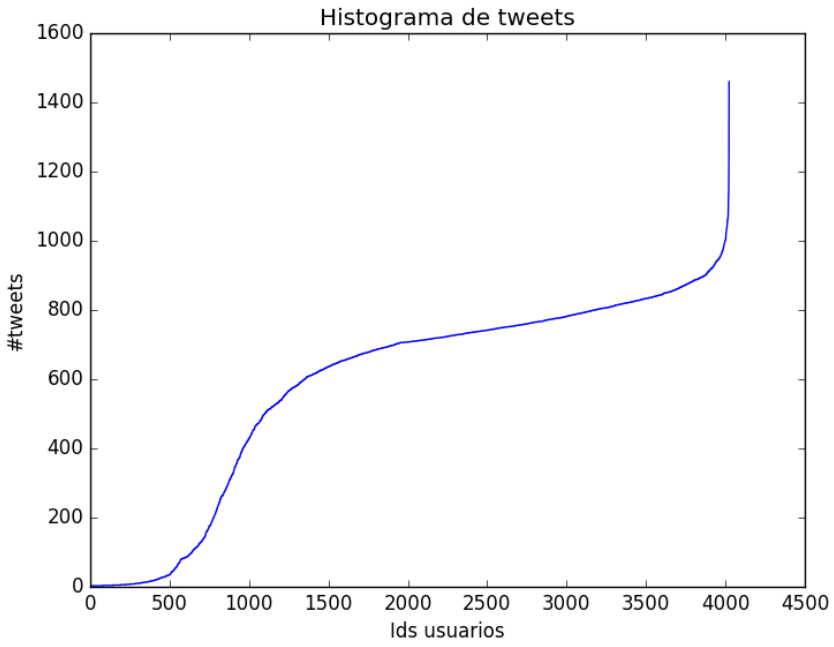

Figura 3: Cantidad de tweets emitidos por usuario.

3. Metodología

A parte de los tweets de los usuarios se hizo un scrapping de la información de los libros consumidos y guardadas en un archivo de texto para cada libro. Esta información corresponde al título

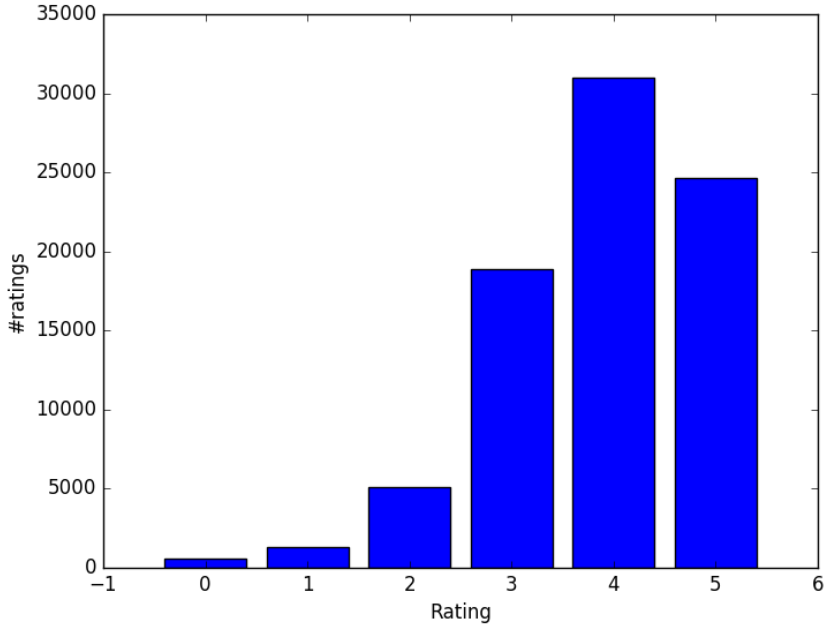

Figura 4: Distribución de ratings dados en Goodreads.

del libro, lista de géneros (que son dados por los mismos usuarios de GR), descripción/reseña del libro, nombre y biografía del autor, fecha de estreno, citas textuales (pasajes del libro) y opiniones y críticas dejadas por los usuarios en la ruta del libro en la aplicación. En total se recolectó información de 18048 libros diferentes, los que en su mayoría están en idioma inglés (varios de ellos en español, árabe, alemán y otros idiomas). Con estos dos corpus se procedió a formar un diccionario con *stop words* y palabras muy poco frecuentes removidas. Paralelamente se asignó identificadores únicos tanto a usuarios como a ítems. En el caso de los ítems son números al azar. En el caso de los usuarios se les asignaron dependiendo del ID de la cuenta, el que se puede encontrar en los archivos JSON del dataset original. Con herramientas de la librería gensim hecha para Python por Radim Řehůřek[3] serializamos los corpus y en base a ellos se crearon modelos TF-IDF, LSI y word2vec, utilizando la suma de estos dos corpus para los entrenamientos. Para TF-IDF convertimos los corpus previamente en representación de bolsa de palabras (BoW). El corpus completo lo transformamos al espacio TF-IDF y eso se lo pasamos al LSI para armar un modelo de 200 tópicos. Para la representación word2vec usamos igualmente el corpus completo para que el modelo pueda capturar las relaciones semánticas de las palabras tanto en lenguaje formal (textos de GR) como coloquial (tweets de usuarios). El vocabulario resultante cuenta con una extensión total de 217268 palabras. Con los modelos listos calculamos las distancias entre la query inicial (el usuario representado por sus tweets) y los documentos de informacion de libros. Para calcular las coincidencias entre los top-N libros más similares y aquellos realmente consumidos se usó una métrica a base de precision y recall. Como baseline tenemos la comparación de documentos en BoW con métricas de distancia por similaridad de coseno y distancia Jaccard.

[3]https://radimrehurek.com/gensim/index.html

4. Resultados

Usamos las definiciones de precision y recall dadas por Gunawardana et al. (2009) [9] que adaptamos para este contexto.

Cuadro 2: Clasificación de los resultados de la recomendación de ítems a un usuario

	Recomendados	No Recomendados
Consumidos	Verdaderos-Positivos (tp)	Falsos-Negativos (fn)
No Consumidos	Falsos-Positivos (fp)	Verdadero-Negativo (tn)

Vemos en el cuadro 2 las definiciones que nos permiten delinear de manera clásica las medidas de precision y recall:

$$Precision = \frac{\#tp}{\#tp + \#fp}$$

$$Recall = \frac{\#tp}{\#tp + \#fn}$$

Es decir, en precision el divisor es el largo de la lista de recomendaciones N; en recall, el largo de ítems consumidos por un cierto usuario.

Los resultados con distancia coseno los podemos ver en el cuadro 3.

Cuadro 3: Precision, recall y F-measure para distintas representaciones de documentos.

	top-N	P	R	F
BoW	top-10	0.0	0.0	0.0
	top-50	0.0	0.0	0.0
TF-IDF	top-10	**0.220**	**0.720**	0.108
	top-50	0.188	0.175	**0.181**
LSI	top-10	0.0	0.0	0.0
	top-50	0.002	0.050	0.004
W2V	top-10	0.020	0.010	0.013
	top-50	0.001	0.100	0.018

Los resultados para BoW con distancia Jaccard se presentan en el cuadro 4.

Cuadro 4: Distancia Jaccard con BoW

	top-N	P	R	F
BoW	top-10	0.220	0.017	0.032
	top-50	0.160	0.638	0.091

En general vemos cómo obviamente al aumentar de tamaño la lista de recomendaciones baja la precisión, apreciable en TF-IDF, word2vec y BoW con Jaccard.
Vemos que los mejores resultados según similaridad por distancia coseno son obtenidos por el modelo TF-IDF tradicional. No malos resultados logra el método BoW con distancia Jaccard y word2vec. Esto se cree que porque al usar frecuencia de términos, nos abstraemos del largo de documento, lo que alivia el problema de usuarios con pocos tweets. También se piensa que es porque esto ayuda a normalizar las diferencias en uso de vocabulario tanto en Twitter como en Goodreads, lo que sirve para captar mejor las similitudes entre usuarios y libros.

Se tuvo que al obtener los scores midiendo similaridades con distancia coseno en representación BoW tenemos 0 F-measure, tanto para N=10 como N=50. Esto puede deberse a que se le tuvo que hacer un clipping al dataset debido al gran número de documentos en nuestro dataset y a restricciones de tiempo. Comparar menos usuarios con menos items aumenta la inconveniencia de que los vectores de usuarios con los vectores de libros sean tan disímiles (debido a usos muy distintos del lenguaje en un mismo idioma). La distancia coseno, definido vagamente, es el número de términos comunes (muy pocos) dividido el número total de posibles términos (bastantes), razon por la que los resultados son prácticamente 0. Esto a diferencia de Jaccard, definido vagamente como el número de términos comunes dividido por el número de términos existentes entre ambos documentos: los comunes más los no compartidos, el cual es un buen índice para medir similaridad entre documentos que tratan de lo mismo pero son relativamente disímiles.
Por último se piensa que los resultados de LSI son bajos debido a que no se logra diferenciar tan categóricamente los tópicos del un usuario representado por sus tweets. Esto porque los mensajes de los tweets deben ser no mayor a 140 caracteres, lo que se cree que no es suficiente para extraer tópicos, y a la inversa, no es suficiente para que un usuario desarrolle en torno a un tema que sea fácilmente clasificable en tópicos.

5. Conclusiones

Se encontró que es posible hacer recomendaciones siguiendo los métodos sugeridos, a saber, armar modelos de usuarios a partir de la información de sus tweets y recomendarle ítems del dominio en el que él suele consumir al dejarlo mostrado por sus tweets.
Vimos que las diferencias en registro lingüístico pueden ser un problema que debe ser solucionado con una correcta elección de modelos y métricas de distancia. En este caso, se vio que lo que daba mejores resultados (para el dataset reducido con el que se hicieron los experimentos) era TF-IDF con métrica coseno y BoW con Jaccard.
Una obvia extensión de esto es aprovechar los ratings puestos en los mensajes predefinidos para hacer un sistema híbrido. También se pueden usar otras métricas. En el caso de BoW se ha observado que el método de ranqueo Okapi BM25 obtiene buenos resultados. En el caso de TF-IDF y LSI, y de forma más clara la extensión probabilística de LSI, LDA, se podría intentar con métricas que miden distancia entre distribuciones de probabilidad como Hellinger o Kullback-Leibler. Kusner et al. (2015) propusieron un interesante nuevo método para calcular distancias entre documentos cuyas palabras están embebidas, llamada *word mover's distance*, calculada como la distancia mínima que las palabras embebidas de un documento tienen que viajar para llegar a las palabras embebidas del otro documento[10].

Referencias

[1] M. Agrawal, M. Karimzadehgan, C. Zhai. An online news recommender system for social networks. In *Proceedings of ACM SIGIR workshop on Search in Social Media.* (2009).

[2] Z. Sun, L. Han, W. Huang, X. Wang, X. Zeng, M. Wang, H. Yan. Recommender systems based on social net-works. In *Journal of Systems and Software.* (2015).

[3] W. Geyer, J. Freyne, B. Mobasher, S. S. Anand, C. Dugan. 2nd workshop on recommender systems and the social web. In *Proceedings of the fourth ACM conference on Recommender systems* (RecSys '10). ACM, New York, NY, USA, 379-380. (2010). DOI=http://dx.doi.org/10.1145/1864708.1864798

[4] H. Zamani, P. Moradi, A. Shakery. Adaptive User Engagement Evaluation via Multi-task Learning. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '15). ACM, New York, NY, USA, 1011-1014. (2015). DOI=http://dx.doi.org/10.1145/2766462.2767785

[5] S. M. Kywe, E.-P. Lim, F. Zhu. A survey of recommender systems in Twitter. In *Social Informatics*. Berlin, Heidelberg,Germany: Springer, 420–433. (2012).

[6] F.M. Rodríguez, L. M. Torres, S. E. Garza. Followee recommendation in Twitter using fuzzy link prediction. In *Expert Systems*. 33: 349–361. (2016). doi: 10.1111/exsy.12153.

[7] G. Morales, A. Gionis, C. Lucchese. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining* (WSDM '12). ACM, New York, NY, USA, 153-162. (2012). DOI=http://dx.doi.org/10.1145/2124295.2124315

[8] B. Dhingra, Z. Zhou, D. Fitzpatrick, M. Muehl, W. Cohen. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of ACL*. (2016).

[9] A. Gunawardana, G. Shani. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. In *J. Mach. Learn.* Res. 10, 2935-2962. (2009).

[10] M. J. Kusner, Y. Sun, N. I. Kolkin, K. Q. Weinberger. From Word Embeddings To Document Distances. in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 957-966. (2015).