Identificación de tópicos usando LDA en textos de Twitter

López Velasco, José Armando[1 [31315734−3]]

Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas, IIMAS.
Universidad Nacional Autónoma de México, UNAM.

Abstract. La identificación de tópicos en el set de datos mostró que podríamos identificar de manera clara al menos 4 de estos, con un perplexity score de -9.82 y un coherence score de 0.31, eliminando las menciones en el texto y manteniendo los hashtags se obtuvo la mayor cantidad de información, mediante un proceso común de NLP utilizando bi-gramas y lematización.

Keywords: LDA · NLP · Twitter · Lemmatization · N-gram

1 Introducción

Durante la última década el uso de redes sociales como Twitter han tenido un incremento debido a diversos factores, como el avance tecnológico y la adaptación de estas mismas, sin embargo, la cantidad de información que se está generando segundo a segundo es increíblemente grande, lo que hace prácticamente inútil realizar ciertas tareas de manera manual, una de ellas es la identificación de tópicos. tratemos de suponer un escenario: Durante un día obtenemos todos los tweets que se generen en cierta región geográfica, ¿de qué está hablando la población de esa región? Para realizar esta tarea de manera manual la única manera sería leer prácticamente todos los tweets y memorizar de que tema trata, posteriormente tratar de crear grupos, etc. Es claro como esta tarea es muy difícil para un ser humano, por lo que modelos que identifiquen tópicos son altamente necesarios para reducir la complejidad de esta tarea, y de esta manera poder saber día a día el hot-topic de una población de interés.

1.1 Estado del arte y trabajos relacionados

El modelado de tópicos es un área que se encuentra en constante crecimiento, por lo cual diversos autores han contribuido con distintas técnicas que implican el uso de redes neuronales, espacios embebidos, redes recurrentes o simples modelos estadísticos, como fue nuestro caso.

En el 2019 Adji B. de la Universidad de Columbia junto con sus colegas Francisco J.R y David M. publican un artículo llamado Topic Modeling in Embedding spaces [2] el cual busca modelar los tópicos y resolver el problema que se encuentra en diversas técnicas que fallan en la interpretabilidad de corpus muy

grandes y con colas muy largas, es decir, tenemos una cantidad muy grande de palabras con frecuencias de menos de 1%. La clave de el ETM (embedded topic model) consiste en realizar un producto entre la palabra y su categoría con el tópico común que le correspondería el cual se entrena con un modelo variacional amortizado para la inferencia.

Durante el 2015 Yishu Miao y sus colegass exploran un nuevo método para la inferencia de tópicos con técnicas basadas en redes neuronales llamados inferencia variacional neural, en su artículo Neural Variational Inference for Text Processing [3] buscan solucionar el problema de una manera distinta a los modelos comunes que derivan en una aproximación analítica de las distribuciones de las variables latentes que se podían observar en los métodos variacionales, con su método validado en dos distintas aplciaciones de modelaje de texto: modelado generativo de documentos y respuesta a preguntas supervisado. El modelo neural mostró los niveles de perplejidad más bajo en ambas tareas y dos distintos corpus, el cual emplea una capa para la representación estocástica basados con mecanismos de atención para extraer la semántica entre la pregunta y la respuesta.

Por último, LDA (Laten Dirichlet Allocation)[1] el cual es un modelo probabilístico basado en un modelo Bayesiano en el cual cada ítem de la colección de las palabras es modelado como una mezcla bajo la probabilidad de pertenecer a un conjunto de tópicos:

$$P(W, Z, \theta, \varphi; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) P(W_{j,t} \mid \varphi_{Z_{j,t}}), \tag{1}$$

donde:

- K Número de tópicos
- V Número de palabras
- M Número de documentos
- N_d Palabras en el documento d
- N Total número de palabras
- α Peso a prior para los tópicos
- β Peso de la palabra para el tópico
- φ Distribución de palabras en tópicos
- θ Distribución de probabilidades de tópicos en documentos
- Z Identidad del tópico de palabra por documento
- W Identidad de todas las palabras en todos los documentos

2 Metodología

2.1 Pre procesamiento

Lo más importante en un proceso de NLP es tratar de tener el texto lo más limpio posible, debido a que los documentos vienen de una red social tenemos

presente el uso de distintos modos de lenguaje, como lo son el uso de emoticones, insertar enlaces web, además de las características propias de la plataforma como hacer menciones y escribir hashtags. Durante este proceso se realizaron distintas pruebas para obtener el mejor resultado. Por lo que se tomó la decisión de eliminar los enlaces web y las menciones, sin embargo mantenemos los hashtags pero se eliminan los caracteres especiales:

- 1. Guardamos todas las menciones y hashtags que aparezcan, es información que podemos usar más adelante para segmentar los resultados
- 2. Eliminamos las menciones ya que no aportan mucha información y suelen ser altamente frecuentes
- 3. Eliminamos enlaces web
- 4. Eliminamos puntuación

Tokenización Este paso es base para realizar la tarea debido a que debemos reducir toda oración a su menor estructura posible, en este caso a palabras. Este proceso es necesario debido a que con el resultado de la tokenización se construyen distintas estructuras como la matriz término frecuencia. Y de igual manera podemos pasar a identificar el tamaño de nuestra bolsa de palabras.

2.2 Stop-words

Las palabras más frecuentes son sin duda los artículos ya que son utilizados para conectar adjetivos, verbos, sujetos, etc. Por lo que si realizamos un conteo de frecuencias, estas palabras crearían una cola de términos que podrían ser más importantes y podrían reducir la probabilidad token-documento para cada tópico, por lo que es una práctica general eliminar estos artículos (el, la, ellos, estos, esos,...etc)

2.3 N-gramas

Una desventaja de la tokenización es que descompone todas las oraciones en palabras y muchas veces estas no debían ser separadas debido a que en conjunto aportar información más general o son palabras que siempre aparecen juntas, es decir, cuan tokenizamos "Estado Unidos" lo descompone en dos tokens "estados" y "unidos" sin embargo en un análisis de frecuencias tendrían prácticamente la misma frecuencia, y en el resultado del modelo LDA ver las palabras separadas no nos aportan mucha interpretabilidad, por lo que cuando hacemos bi-gramas (en nuestro caso) tratamos de aprender esos pequeños casos en los cuales la tokenización no era necesaria hasta la palabra, de esta manera al tokenizar "Estados unidos" tendríamos como resultado "estados_unidos" lo cual nos aporta una mejor interpretación de ese término.

2.4 Lematización

En esta parte tenemos dos opciones, lematizar o estemizar. Sin embargo, cuando probé con técnicas de stemming el performance era menor debido a que no reducía todas las palabras a su raíz, sólo eliminaba el sufijo para tratar de reducirla.

Cuando lematizamos una palabra significa que vamos a reducir esta a su raíz, además de identificar su naturaleza (sujeto, adjetivo, verbo, etc) esto se realiza utilizando un corpus que contenga la etiquetación para poderlos relacionar, en nuestro caso utilizamos el core **"es_core_news_sm"** Una vez realizado esto las palabras que estén conjugadas de distintas maneras podrán ser reducidas al mismo término y por ende aumentar su frecuencia.

2.5 Matriz término-documento

Esta técnica es muy utilizada para entender y representar como están los datos en el texto. La manera de calcularla consiste en descomponer cada oración en sus tokens previamente lematizados (en mi caso), después de esto asignamos un renglón pro documento (tweet) y la matriz nos indicara cuantas veces aparece cada token en la oración, dando como resultado una matriz de tamaño (número de documentos) X (Dimensión de nuestro corpus) La suma a nivel columna nos dará la cantidad de veces que aparece ese término en todo nuestro conjunto y con esta información podemos identificar la distribución de probabilidad para cada token.

```
[[('llegar', 1),
  ('terminar', 1),
  ('solo', 1),
  ('partir', 1),
  ('integrante', 1),
  ('calderon', 1),
  ('sexenio', 1),
  ('colocar', 1),
  ('corta', 1)]]
```

Fig. 1. ejemplo de término frecuencia para el penúltimo documento

2.6 LDA

Una vez que tenemos nuestro texto listo, procedemos a entrenar nuestro modelo, el ajuste fino que se realizó se obtuvo con los parámetros:

- Tópicos: 4
- Chunk: 10
- passes: 5
- per_word_topics=True

Obteniendo la siguiente distribución de palabras clave para los 4 tópicos: Con un Perplexity score de **-9.82** y un Coherence score de **0.31**

```
Most important words in topic: (0, '0.020*"mexico" + 0.016*"hacer" + 0.016*"dar" + 0.013*"solo" + 0.011*"caso" + 0.010*"contagio" + 0.010*"pandemiar" + 0.009
*"coronavirus" + 0.009*"ano" + 0.007*"saber"')

Most important words in topic: (1, '0.025*"hoy" + 0.019*"mejor" + 0.016*"gente" + 0.015*"medida" + 0.014*"seguir" + 0.012*"salir" + 0.011*"virus" + 0.011*"mas"
+ 0.011*"asi" + 0.008*"saludo"')

Most important words in topic: (2, '0.037*"quedateencasar" + 0.024*"persona" + 0.015*"ver" + 0.014*"creer" + 0.010*"cuarentenar" + 0.010*"muerte" + 0.010*"quer
er" + 0.009*"vacunar" + 0.009*"nuevo" + 0.007*"numero"')

Most important words in topic: (3, '0.023*"vacuna" + 0.017*"cuarentena" + 0.015*"salud" + 0.014*"decir" + 0.013*"pandemia" + 0.010*"bien" + 0.010*"covid_mx" +
0.009*"llegar" + 0.008*"ahora" + 0.008*"tener"')
```

Fig. 2. Palabras clave para los 4 tópicos

3 Resultados

Los tópicos se ven representados en un espacio 2d de la siguiente manera: Las

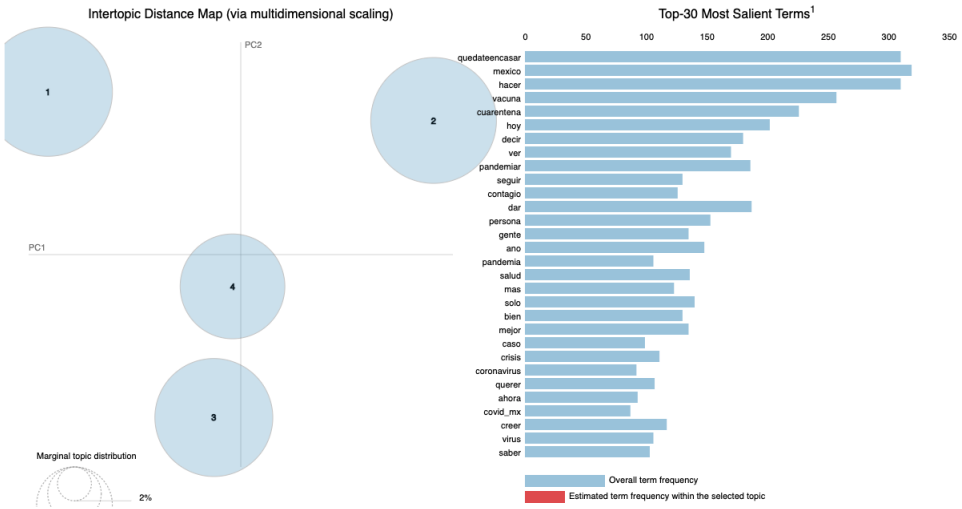

Fig. 3. Tópicos LDA

etiquetas identificadas asignadas pro mi fueron:

- Gobierno Federal
- Gobierno Estatal
- Vida cotidiana afectada por COVID
- Critica a servicios y empleos

Cabe recalcar que todos ellos giraban entorno al COVID-19, por lo que se presupone, todos los tweets fueron extraídos bajo cierto criterio relacionado a la pandemia. Un análisis por keywords y tópicos lo podrá encontrar al final del repositorio en github [1]

[1] https://github.com/ArmandoLp/LDA-Twitter

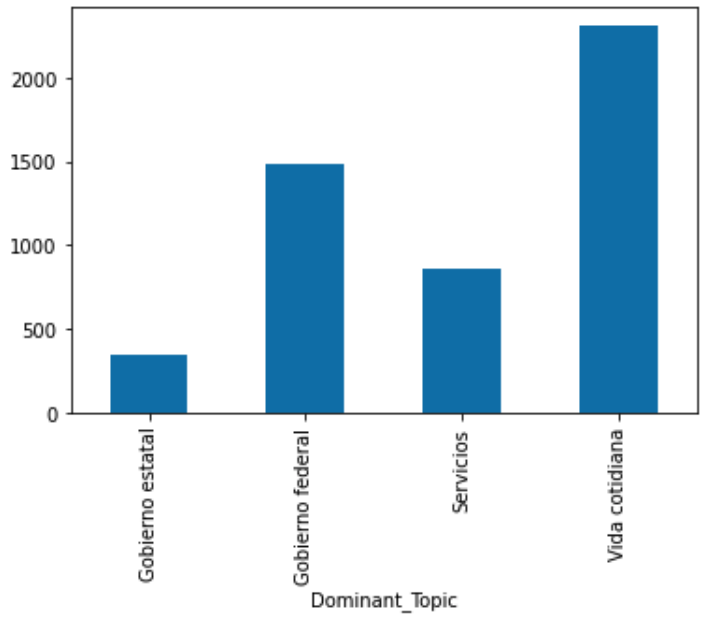

Fig. 4. Cantidad de textos por tópico

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**(null), 993–1022 (Mar 2003)
2. Dieng, A.B., Ruiz, F.J.R., Blei, D.M.: Topic modeling in embedding spaces. CoRR **abs/1907.04907** (2019), http://arxiv.org/abs/1907.04907
3. Miao, Y., Yu, L., Blunsom, P.: Neural variational inference for text processing. CoRR **abs/1511.06038** (2015), http://arxiv.org/abs/1511.06038