JUN 17

SPRINT 3 - MEETING 40

Procesamiento Lenguaje Natural (Parte 2)

Grabación

Presentación

El equipo docente subió contenido extra de la meeting

Modelos específicos de NLP

> *"We're competing with sleep, on the margin. And so, it's a very large pool of time." —Reed Hastings. CEO Netflix.*

En la bitácora anterior, conociste el Procesamiento de Lenguaje Natural o NLP, aprendiste a tokenizar el texto, a filtrar stop words, y a usar lexicons o corpus para pre-entrenar algoritmos. Estos pasos forman parte del preprocesamiento del texto, que es necesario para luego entrenar nuestro algoritmo de NLP.

Como podrás imaginar, NLP tiene sus modelos específicos, especializados en análisis de texto. En esta bitácora conocerás algunos de ellos.

TF-IDF

TF-IDF significa **"Term Frequency-Inverse Document Frequency"**, ¡eso ya nos da información sobre lo que hace!

Term Frequency significa: cuántas veces un término aparece en un documento. La importancia de esto es que se supone que mientras más veces aparece, más importante es ese término en ese documento. Por ejemplo, un texto sobre tiburones va a contener muchas más veces la palabra tiburón que un texto que no es sobre tiburones. Mientras mayor frecuencia tenga la palabra en el documento, mayor "puntaje" tendrá. Esto nos sirve, por ejemplo, para detectar sobre qué tópico trata un texto.

Ahora bien, hay términos (como los artículos "la", "el", o las preposiciones "a", "por") que tienen alta frecuencia en todos los textos. Si les asignamos un puntaje alto, no ganaremos información con respecto a cuál es el tópico del documento. Aquí es donde entra en juego la segunda parte del nombre TF-IDF: **Inverse Document Frequency.** Esto significa que se compara la frecuencia de la palabra en ese documento con la frecuencia en todo el corpus de documentos. A mayor frecuencia de la palabra en todos los documentos, menor puntaje tendrá. Así evitamos puntuar alto a stop words o palabras que no agregan información sobre ese documento específico.

Word2Vec

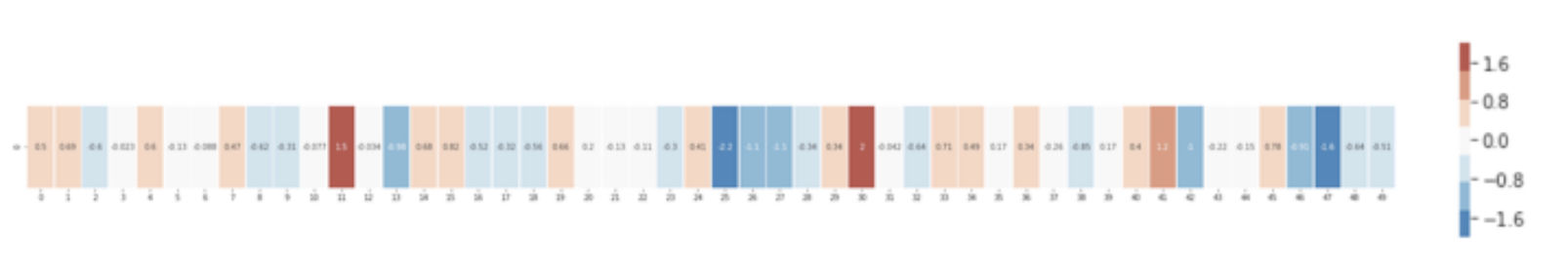
Word2Vec es una técnica que, como su nombre lo indica, transforma las palabras en vectores numéricos y así permite operar matemáticamente sobre ellas. La idea que se encuentra detrás de este método es que las palabras próximas tienen mayor relación en su significado que las palabras lejanas.

¿Cómo lo hace? Consideremos la siguiente frase extraída de El Jardín de los Senderos que se Bifurcan, por Jorge Luis Borges:

"Así combatieron los héroes, tranquilo el admirable corazón, violenta la espada, resignados a matar y a morir"

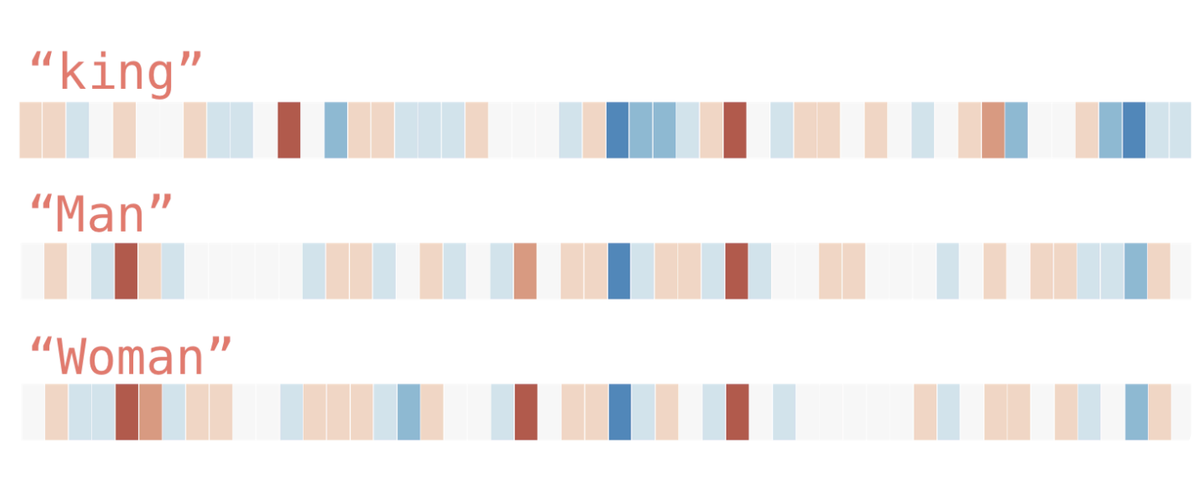
Para cada palabra vamos a considerar las palabras cercanas. Por ejemplo, si tomamos una distancia igual o menor a 2, las palabras cercanas a "admirable" son "tranquilo", "el", "corazón" y "violenta". Podemos tomar un texto entero, medir las distancias que todas las palabras tienen entre sí, y luego entrenar una pequeña red neuronal que prediga la probabilidad de que una palabra se encuentre cercana a otra. A partir de esto se obtendrán los vectores para cada palabra, que nos indicarán su relación con otras palabras.

Cada vector estará compuesto por n cantidad de números entre -2 y 2 (cada uno es una dimensión o *feature*). Podemos visualizarlo como una cadena de números, y colorear cada uno (rojo si está más cerca del 2, blanco si está cerca del 0, azul si está más cerca del -2).



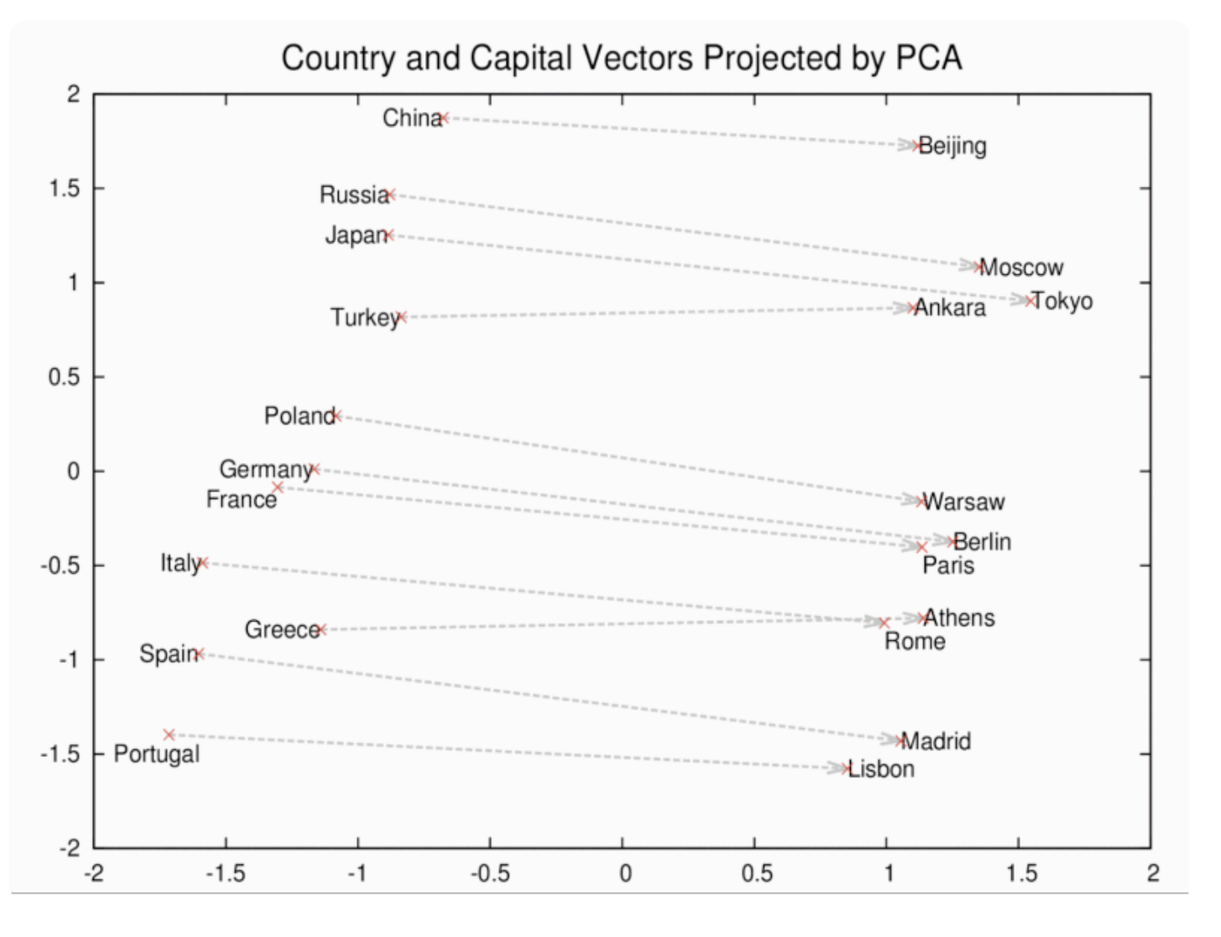
Fuente: GitHub

Para facilitar las cosas, olvidaremos por ahora los números y nos enfocaremos en los colores. Estos son los vectores para las palabras en inglés "king" (rey), "man" (hombre) y "woman" (mujer), luego de entrenar un Word2Vec sobre las entradas en inglés de Wikipedia:



Fuente: GitHub

Podemos ver a simple vista ciertas similitudes entre estos vectores, especialmente entre "man" y "woman". Esto significa que ocupan lugares cercanos en el espacio vectorial. Palabras con relación sintáctica o semántica se convertirán en vectores cercanos con direcciones análogas.



Fuente: GitHub

En esta representación bidimensional (en realidad el espacio tiene muchas más dimensiones, pero no las podemos representar) de vectores podemos ver que existe una relación análoga entre el nombre de cada país y el nombre de su capital. En general, se dice que "China es a Beijing lo que Rusia es a Moscú", lo cual significa que existe una relación similar entre ambos pares.

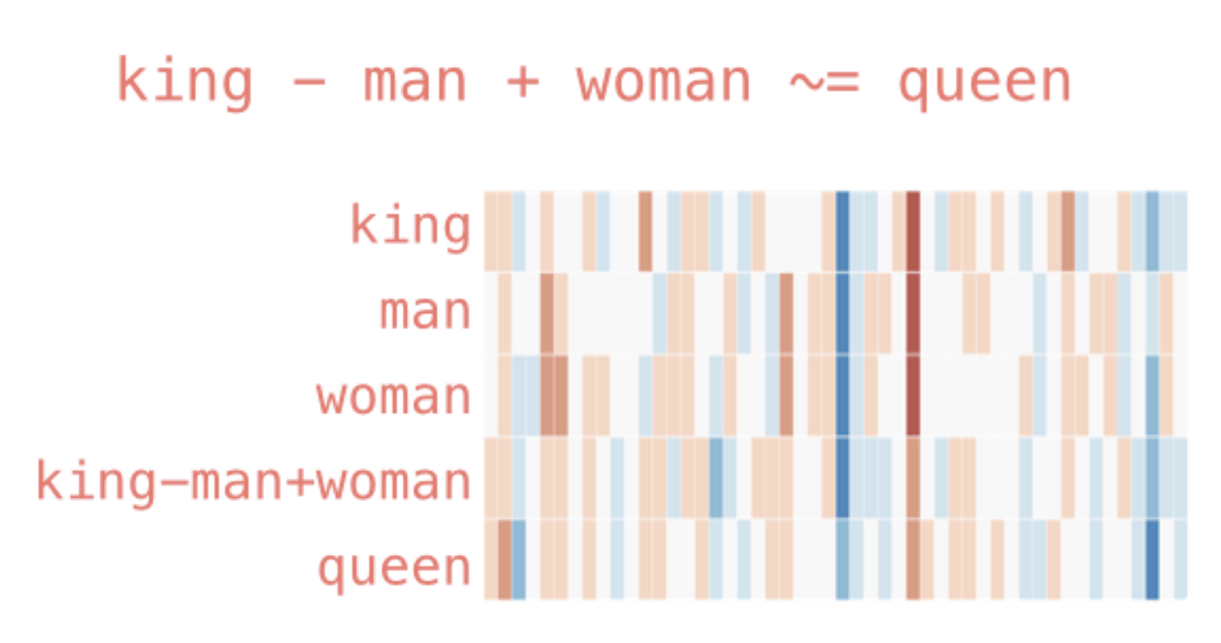
Como dijimos, podemos hacer operaciones matemáticas con estos vectores. Volviendo al ejemplo anterior, si, tomamos el vector de la palabra "king", le restamos el vector de la palabra "man" y le sumamos el vector de la palabra "woman", estas serán las predicciones resultantes:

```
model.most_similar(positive=["king","woman"], negative=["man"])
[('queen', 0.8523603677749634),
 ('throne', 0.7664333581924438),
 ('prince', 0.7592144012451172),
 ('daughter', 0.7473883032798767),
 ('elizabeth', 0.7460219860076904),
 ('princess', 0.7424570322036743),
 ('kingdom', 0.7337411642074585),
 ('monarch', 0.721449077129364),
 ('eldest', 0.7184862494468689),
 ('widow', 0.7099430561065674)]
```

Fuente: GitHub

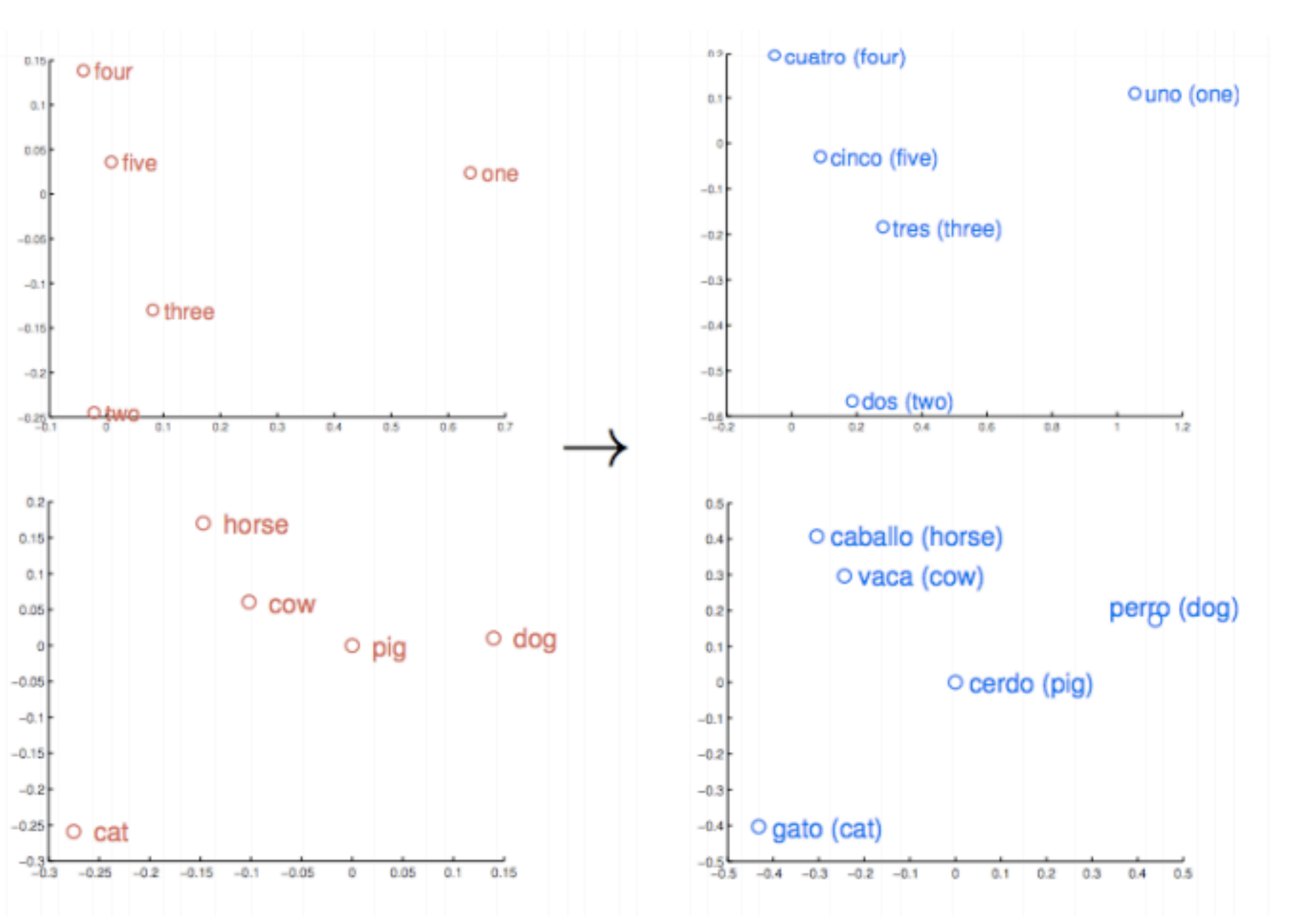
Esto sucede siempre y cuando le "prohibamos" al algoritmo que la predicción sea una de las palabras usadas en el cálculo, en cuyo caso será otra vez "king".

Aquí vemos exactamente el vector que resulta de hacer la operación numérica king - man + woman, y su cercanía al vector queen (no siempre Word2Vec da resultados tan prolijos, aunque por ahora no veremos las posibles excepciones).



Fuente: GitHub

Podemos usar esta misma lógica para encontrar similitudes entre palabras de dos idiomas diferentes, buscando aquellas que ocupan lugares cercanos, para hacer traducciones.



Fuente: GitHub

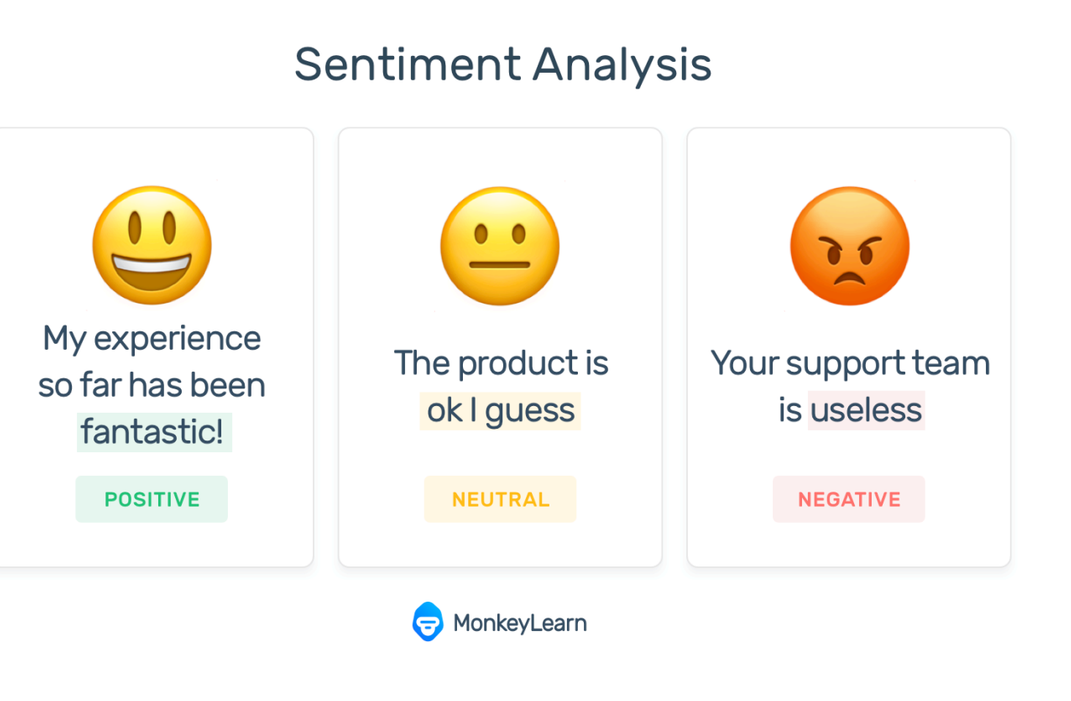
Desafío mental: Imagina que quieres que un algoritmo aprenda palabras nuevas de un idioma a través de la lectura de sus definiciones en el diccionario. El algoritmo estará pre-entrenado sobre algunas palabras: usando esas palabras que sabe, comprenderá las definiciones de palabras nuevas y las incorporará a su vocabulario.¿Cuántas palabras tiene que saber como mínimo para poder entender todas las definiciones del diccionario? Si sabes algo de matrices, esto equivaldría a decir: ¿cuál es la mínima cantidad de vectores necesaria, es decir cuál es el rango de la matriz "vocabulario"?

Sentiment Analysis

Una aplicación muy difundida de NLP, sobre todo en el campo de marketing o business intelligence, es la de **sentiment analysis** o **análisis de opiniones**.

Imagina que quieres saber qué opinan los consumidores de un modelo de auto. Puedes buscar en internet las páginas e-commerce con publicaciones de ese modelo, grupos en redes sociales y foros de aficionados a los autos y **scrapear** los comentarios de los/as usuarios/as sobre el auto.

Luego, aplicando un sencillo *sentiment analysis*, podrías clasificar los comentarios en positivos o negativos. Incluso podrías identificar sensaciones como alegría, enojo o desilusión. Con un poco más de complejidad, podrías tratar de distinguir qué aspectos del auto están asociados a opiniones positivas o negativas. Por ejemplo, si la palabra "motor" está asociada a oraciones con opinión positiva y la palabra "consumo" de combustible está asociada a oraciones que denotan una opinión negativa.



Sesgos en NLP

Las relaciones sociales son complejas. Sobre cualquier tema de interés podemos señalar, como mínimo, puntos de vista contradictorios, cuando no verdaderos conflictos sociales. ¡El lenguaje no es ajeno a estas disputas! Sabes, por ejemplo, en que existen palabras prejuiciosas para designar a grupos de personas. Esto tiene que advertirte algo importante: el lenguaje está cargado de **subjetividad** y de sesgos. Nunca podemos considerarlo como una entidad plenamente objetiva, aislada de un contexto social. Al analizar nuestros datasets y armar nuestros modelos, debemos tenerlo en cuenta.

Hay muchos casos donde los modelos fracasan en manejar bien determinados usos del lenguaje. Por ejemplo, un estudio por Sap et al. (2019) halló que algoritmos que detectan **"discurso de odio"** tendían a señalar erróneamente los tweets de personas afroamericanas como discurso de odio más que el resto. Un caso era el uso de la palabra "n*gga", que viene de las épocas de la esclavitud y fue reapropiada por la comunidad afroamericana. En boca de una persona afroamericana, es un término cordial, pero dicha por una persona caucásica tiene una connotación racista.

¿Cuáles serían posibles soluciones a este problema? Una de las soluciones que se propusieron fue analizar si la persona que había emitido el tweet era afroamericana o no. ¿Podría esta solución llevar a otros problemas?

Los **sesgos de género** embebidos en el lenguaje también tienden a ser "aprendidos" por algoritmos de procesamiento de lenguaje natural. Por ejemplo, dado que actualmente los hombres son mayoría en el campo de la programación el algoritmo que autocompleta las búsquedas en Google Search tiende a autocompletar las búsquedas con sugerencias que amplifican los prejuicios existentes.

Un estudio del Pew Research Center mostró que las búsquedas de imágenes de profesiones comunes sobre-representaban a los hombres (es decir, mostraban menos mujeres de las que realmente hay en esos trabajos). Por ejemplo, la búsqueda de CEOs mostraban solamente 10% de mujeres a pesar de que las mujeres representan un 28%. ¿Tendrá que ver esto con sesgos en las imágenes que se cargan a las páginas web y que el algoritmo de Google levanta sin filtrar por diversidad? ¿O con lo que clickean los usuarios que hacen las búsquedas y el algoritmo aprende subiendo la visibilidad de algún grupo social en desmedro de otros? ¿O alguna otra opción, o más de una a la vez? ¿Cómo podría ser una posible solución?

Es importante incluir chequeos contra los sesgos en el diseño de los algoritmos, ya que muchas veces son invisibles para el ojo no entrenado y pasan desapercibidos hasta que es demasiado tarde y ya contribuyó a introducir más desigualdad en el mundo.

Por último, no podemos dejar de mencionar los grandes avances en Procesamiento del Lenguaje Natural que representan arquitecturas de Deep Learning como GPT-2 y GPT-3. No entraremos en detalles sobre el funcionamiento de estos modelos - los detalles técnicos exceden los contenidos de la carrera - pero sí podemos plantear algunas preguntas asociadas a su uso. ¿Está bien que exista - al menos sin regulación - herramientas capaces de generar noticias y comentarios falsos?¿Qué ocurrirá cuando podamos escribir un texto sobre algún tema dando una simple instrucción a una computadora? ¿Qué impactos tendrá sobre la manera de comunicarnos?¿Y sobre el mercado laboral? Incluso, hay demostraciones donde dando una simple instrucción, estos modelos pueden generar bloques de código - por ahora sencillo - y éste es ejecutado correctamente. ¿Creés que puede ser usado con fines poco éticos?

Empieza a trabajar en el notebook

Para el encuentro, trae ya hecha tal parte, y completen durante la clase esto...

DS_Bitácora_39_y_40_NLP

¡Prepárate para el próximo encuentro!

Profundiza
Te invitamos a conocer más sobre el tema de esta bitácora.

Contenido extra de la meeting

Grabación
Código de acceso para acceder: Acamica321*

Lista de recursos

- DS_Bitácora_39_y_40_NI
- Profundiza

Contenido extra

- Grabación