Procesamiento natural del lenguaje

- Objetivo del NLP es construir la capacidad de procesar texto:
  - Reconocer
  - Analizar
  - Interpretar
  - Generar
- ¿Dónde se utiliza NLP?
  - Twitter, Facebook, Google, etc.
  - Chatbots
  - Autocompletado de texto

Representación vectorial del texto

- Construir “embeddings” o vectores que representen un texto
- Una vez se tienen embeddings, es posible analizar y comparar distintos textos/entidades. Ejs:
  - Twitter: ¿son dos personas políticamente afines?
  - Stack Overflow/Yahoo respuestas/Reddit: ¿esta pregunta ya se ha hecho antes? ¿algún post ha contestado esta pregunta?
  - Rappi: ¿este restaurante es parecido a este otro?
  - Chatbots: ¿el cliente está satisfecho? ¿la petición a qué categoría pertenece?
  - Política: ¿qué temas suelen tratar los congresistas del CD? ¿qué temas menciona el presidente en tiempos de crisis?

¿Por qué es complejo?

- Ambiguedad:
  - Vi a Juan caminando
  - Saldré de vacaciones solo unos días
  - Juan vio a Gabriel con sus gafas
- Modismos:
  - Tirar la toalla
  - La oveja negra de la familia
- Lenguaje no estándar:
  - pq
  - tqm

Tokenización

- Idea: separar texto entero en unidades más pequeñas para hacer el análisis
- “Voy a jugar fútbol mañana”
  - En palabras: “voy”, “a”, “jugar”, “fútbol”, “mañana”
  - En letras: “v”, “o”, “y”, …
  - En n-gramas:
    - 1-gramas: “voy”, “a”, “jugar”, “fútbol”, “mañana”
    - bigramas: “voy a”, “a jugar”, “jugar fútbol”, “fútbol mañana”
    - trigramas: “voy a jugar”, “a jugar fútbol”, “jugar fútbol mañana”
- Dificultad: “Los Ángeles”
  - ¿“Los Ángeles” o “Los”, “Ángeles”?

Stop Words, acentos y puntuación

- Stopwords: limpiar palabras que son muy comunes y, por lo tanto, no agregan información en el texto
  - Español: para, con, de, la, los, en, también, como, cuando
  - Inglés: for, a, the, on, in, at
- Suelen borrarse para mejorar el desempeño de modelos
- Acentos:
  - papá vs papa, sólo vs solo
  - en ciertos contextos, el texto no siempre tiene acentos
- Puntuación y símbolos:
  - expresidente vs ex-presidente

Lematizar y stemmizar

librería: nltk
stemmer: SnowballStemmer

- Stemizar:
  - quedarse solamente con la raíz:
    - corremos, corrí, corren, corríamos => corr
    - niñas, niños, niña => niñ

librería: nltk
stemmer: WordNetLemmatizer

- Lematizar:
  - transformar palabra a forma base:
    - wanting, wanted, want => want
    - rejection, rejecting => reject

    } Verbo

    - wanting, wanted, want => wanted
    - rejection, rejecting => rejected

    } Sustantivo

  - Depende del contexto: traje puede venir de traer o de traje
  - Complejo: se debe especificar si palabra es verbo, sustantivo, etc.

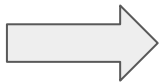
Similaridad de restaurantes

- Usar descripción de platos de restaurantes para construir embeddings:

Producto	Tipo	Descripción
Pizza	Italiana	Pizza de pepperoni con queso mozzarella
Raviolis	Italiana	Raviolis de espárragos con queso ricotta
Hamburguesa	Americana	Doble carne con queso mozzarella
Pizza	Italiana	Cuatro quesos

1. Stemizar palabras: “quesos” vs “queso”
2. Remover stop words: “con”, “de”, …
3. Limpiar acentos: “espárragos”
4. etc.

Queso: 4
Italiana: 3
Pizza: 3
...



Idea:
Encontrar restaurantes con una proporción similar de: “pizza”, “italiana” y “queso”

Tokens y corpus

- El corpus del texto, o “vocabulario”, es la lista de todas las posibles palabras que aparecen en el menú:
  - pizza, pollo, peperoni, pasta, atún, queso, mozzarella
- Pasos:
  - Limpieza:
    - remover stop words
    - limpiar acentos, signos de puntuación, números (depende del contexto)
    - stemizar o lematizar
  - Construir corpus
  - Tokenizar

Count vectorizer

- Idea:
  - Construir corpus o diccionario. Ej: tomate, espinaca, pizza, queso, lechuga, carne, pollo, hamburguesa
  - Construir vectores con 8 dimensiones - cada dimensión corresponde a una palabra:

	Tomate	Espinaca	Pizza	Queso	Lechuga	Carne	Pollo	Hamburguesa
restaurante 1:	0	0	3	4	0	0	0	1
restaurante 2:	5	4	0	1	2	2	0	0
restaurante 3:	3	1	1	0	0	7	0	5

Term Frequency - Inverse Document Frequency

- TFIDF (term frequency - inverse document frequency):

  TFIDF = TF x IDF

  TF = N_{# de veces la palabra aparece en el texto}

  IDF = log(1+# textos/ 1+# textos con la palabra})

⇨ TFIDF da más peso a palabras más escasas en el mercado

◇ **Ej:** fríjoles vs. pad thai