UNIVERSIDAD EAFIT

Escuela de
Ciencias Aplicadas
e Ingeniería

Presentación curso

ST7003 Procesamiento Natural del Lenguaje

Profesores

- Edwin Montoya Múnera – emontoya@eafit.edu.co
- Juan David Martínez – jdmartinev@eafit.edu.co

Evaluación

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Tareas (60%), 4 tareas, c/u de 15%, individual
- Examen (10%), individual
- Proyecto (30%), en grupo

Contenido

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Semanas:

1. Intro NLP y Módelos clásicos (Edwin)
2. Modelos de tópicos, Modelo de lenguaje n-gram, clasificación(Edwin)
3. Modelos de lenguaje (Juan David)
4. Redes neuronales y Transformers 1(Juan David)
5. Transfer learning y modelos pre-entrenados (Juan David)
6. Ajuste supervisado y RLHF(Juan David)
7. RAG (Edwin)
8. Agentes y Aplicaciones (Edwin)

Recursos

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Google Colab
- Python
- Diferentes librerías: sklearn, pytorch, API openai, etc
- github

UNIVERSIDAD EAFIT

Escuela de
Ciencias Aplicadas
e Ingeniería

Lecture 01

Contenido

1. Intro NLP
2. Representación del texto
3. Preparación de texto
4. Características y Representación de documentos
5. Modelos de recuperación

UNIVERSIDAD EAFIT

Escuela de
Ciencias Aplicadas
e Ingeniería

NLP Preliminares

Natural Language Processing (NLP) & Text Mining

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Natural Language Processing (o NLP) aplica modelos de Machine Learning o linguistica computacional al texto.
- Enseñar a las máquinas a entender lo que se dice en la palabra hablada y escrita es el foco del Procesamiento del Lenguaje Natural.
- Algunos ejemplos sencillos: siri, traductor google
- También puede usar PNL en una revisión de texto para predecir si la revisión es buena o mala.
- Puede usar NLP en un artículo para predecir algunas categorías de los artículos que está intentando segmentar.
- Puede utilizar la NLP en un libro para predecir el género del libro.
- Puede usar la NLP para construir un traductor automático o un sistema de reconocimiento de voz, y en ese último ejemplo utiliza algoritmos de clasificación para clasificar el lenguaje.

NLP orígenes

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Nace de 2 grandes campos, vigentes aún con amplios desarrollos.
- Information Retrieval
  - Problema: ¿cómo dado un gran volumen de documentos o datos en texto, recuperar un subconjunto de ellos?
- Minería de texto
  - Crear diferentes modelos o métodos para realizar diferentes tareas, como:
    - Clasificación
    - Análisis de sentimientos
    - Análisis de emociones
    - Traducción
    - Extracción de información
    - Q & A
    - Agrupación
    - Resumen de texto
    - Web scrapping
    - Chatbots
    - etc

Retos principales en NLP

- Visión clásica: modelos estadísticos
  - Representación de características, documentos y elementos de lenguaje.
  - Modelos y técnicas para realizar diferentes tareas de lenguaje.
- Representación de características, documentos y elementos de lenguaje
  - Palabras o elementos del lenguaje como característica.
  - Bolsa de palabras – Bag of Words. Conjunto de palabras o tokens diferentes dentro de un conjunto de documentos.
    - V -> conjunto, |V| tamaño del vocabulario.
    - Para conjuntos de documentos muy normales, suelen ser tamaños MUY grandes, lo que representa 2 problemas:
      - Procesamiento computacional alto.
      - Dada la alta dispersión de los datos, requiere gran cantidad de datos y entrenamiento para poder obtener niveles de confiabilidad aceptables.

Retos principales en NLP

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Visión clásica
  - Modelos y técnicas
    - Supervisados: Regresión, Naive Bayes, Arboles de Decisión, cadenas de markov ocultas, KNN
    - No supervisados: genéricos: k-means, dbscan, etc, específicos de NLP: LDA
  - Modelos de lenguaje basado en n-grams

Natural Language Processing (NLP) & TM

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

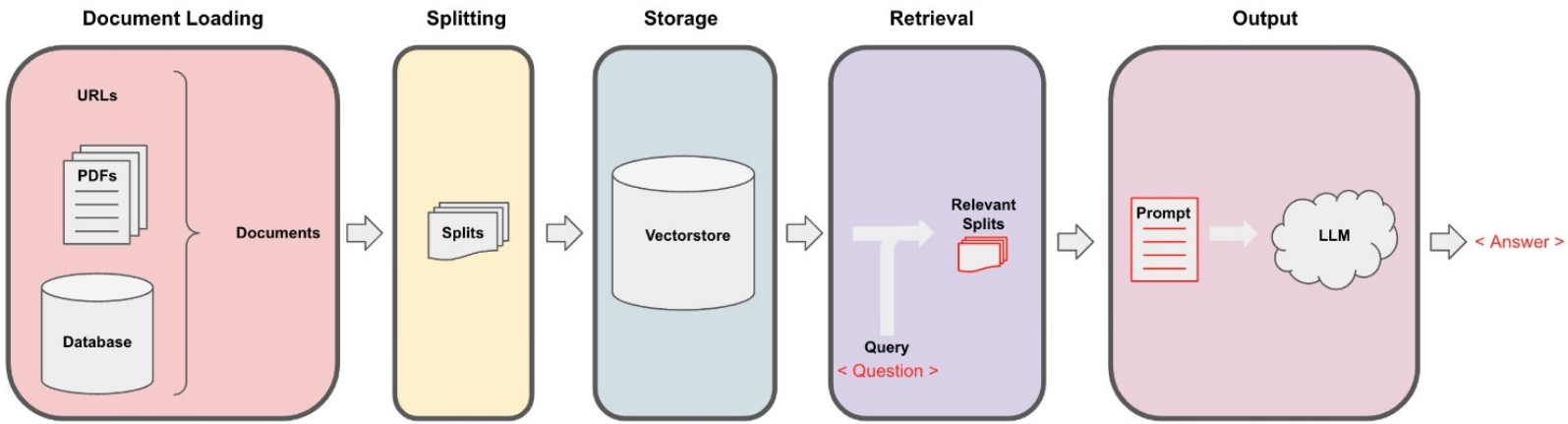
- Visión moderna:
  - Se abandona la representación de palabras como características en pro de otro dominio abstracto: los embeddings.
  - Redes Neuronales
  - Embeddings
  - Modelos de lenguaje Grandes o LLM (Large Language Model)
    - Transformers, modelos de atención
    - BERT (2018), XLNET, GPT-3 (OpenAI,2020), LlaMA (Meta, 2023), GPT-4 (mar/2023)
    - Prompts & Context (memory)

Natural Language Processing (NLP) & TM

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

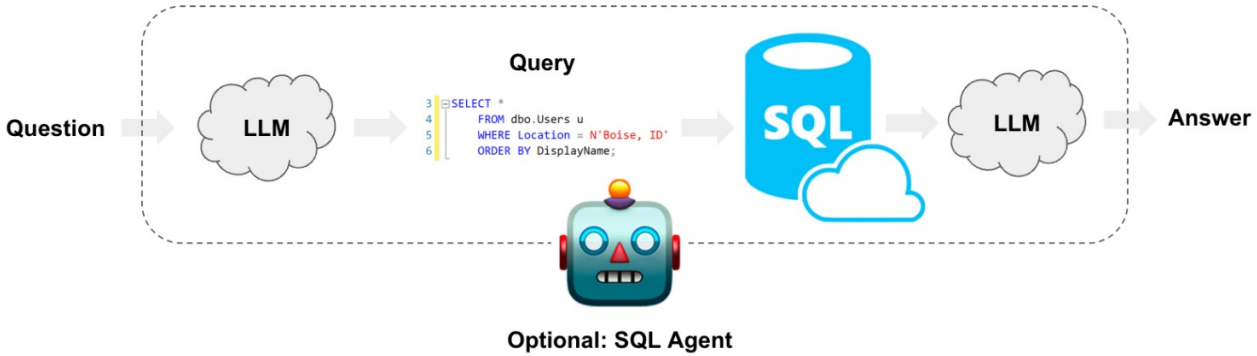
- Visión moderna:
  - Ejemplo: Caso de Question & Answering



Natural Language Processing (NLP) & TM

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- Visión moderna:
  - Ejemplo: Caso de Question & Answering con datos estructurados



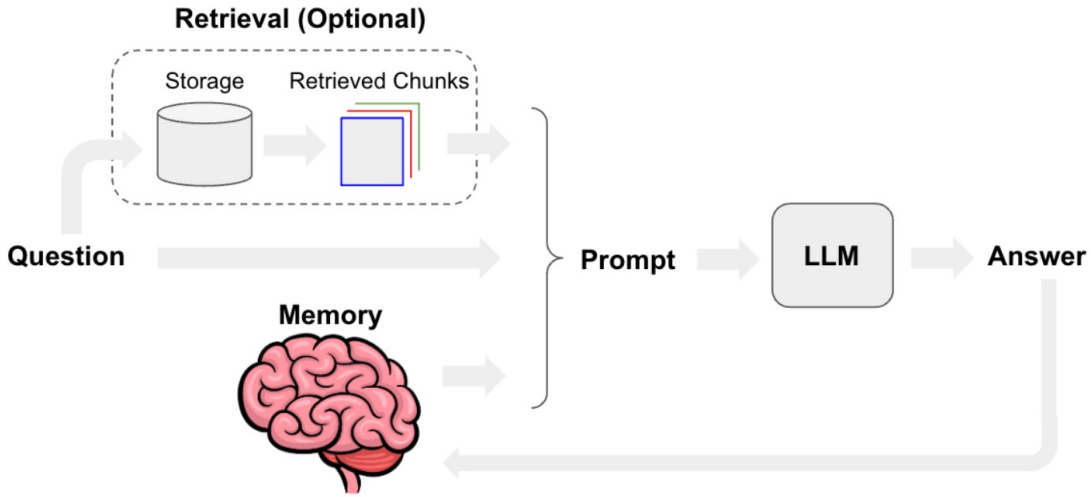
Generar consultas que se ejecutarán en base a preguntas en lenguaje natural.
Crear chatbots que puedan responder preguntas basadas en datos de bases de datos
Creación de paneles personalizados basados en la información que un usuario desea analizar

Natural Language Processing (NLP) & TM

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

- Visión moderna:
  - Ejemplo: chatbot



Contenido

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

1. Intro NLP
2. Representación del texto
3. Preparación de texto
4. Características y Representación de documentos
5. Modelos de recuperación
6. Minería de tópicos / Temas

Representación de texto (clásica)

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- Dependiendo de como represente el texto, podemos realizar una analítica específica
- Múltiples formas:
  - Strings / tokens
  - Estructuras sintácticas
    - Palabras, oraciones, párrafos, documentos.
  - Gráfos de entidad-relación
  - Predicados
- Mucho del procesamiento está basado en ‘strings’ (tokens o n-grams) asociados a palabras y más recientemente ‘sentences’ o ‘chunks’

Representación de texto

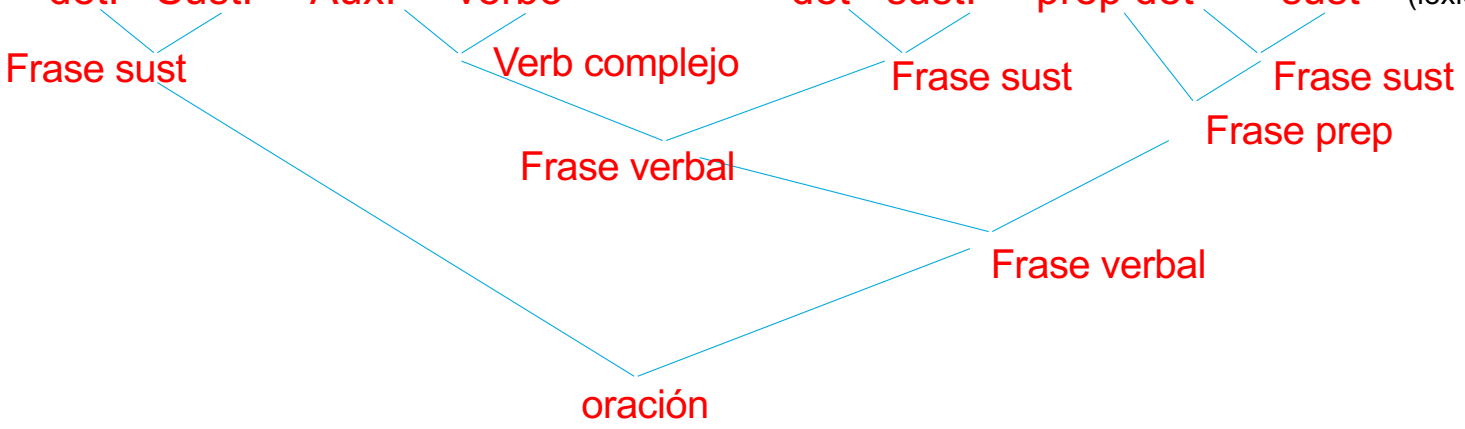
UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

"Un perro esta persiguiendo un joven en el parque" -> + string de caracteres

"un", "perro", "esta", "persiguiendo","un","joven","en","el","parquet" -> + secuencia de palabras

•Un perro esta persiguiendo un joven en el parque -> + POS tagging

det. Sust. Aux. verbo det sust. prep det sust (lexico)



+ Estructuras sintácticas

•Análisis semántico

'un perro' -> animal 'un joven' -> persona 'el parque' -> un lugar + entidades y relaciones

perseguir **donde**

Representación de texto

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

•Análisis semántico

‘un perro’ -> animal ‘un joven’ -> persona ’el parque’ -> un lugar + entidades y relaciones

perseguir **donde**

Perro(d1). Joven(b1), Parque(p1). Perseguir(d1,b1,p1) + lógica de predicados

Reto: Todos estos bases sintácticas y semánticas, requieren gran esfuerzo Humano, con baja precisión.

-> más cerca de REPRESENTACIÓN DE CONOCIMIENTO

Representación de conocimiento y tipos de análisis

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Text rep	Generalización	Análisis habilitado	Ejemplos
String	XXXXXXXXXXXXXXX	Procesamiento a nivel de string	Compresión
Palabras	XXXXXXXXXXXXXX	Análisis de relaciones de palabras, análisis de tópicos, análisis de sentimientos	Tesauros, análisis de tópicos, sentimientos, emociones, en general ML
+ estructuras sintácticas	XXXXXXXX	Análisis de grafos sintácticos	Análisis de estilo, extracción de características basada en estructura
+ entidades & relaciones	XXXXX	Análisis de grafos de conocimiento	Descubrimiento de conocimiento, opiniones acerca de entidades especificas
+ lógica de predicados	XX	Análisis de conocimiento disperso. Inferencia lógica	Conocimiento asistido para xxx (biólogos, químicos, etc)

Representación de texto (moderna)

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

- En embeddings, próximas clases.

Contenido

1. Intro NLP
2. Representación del texto
3. Preparación de texto
4. Características y Representación de documentos
5. Modelos de recuperación

Preparación de texto

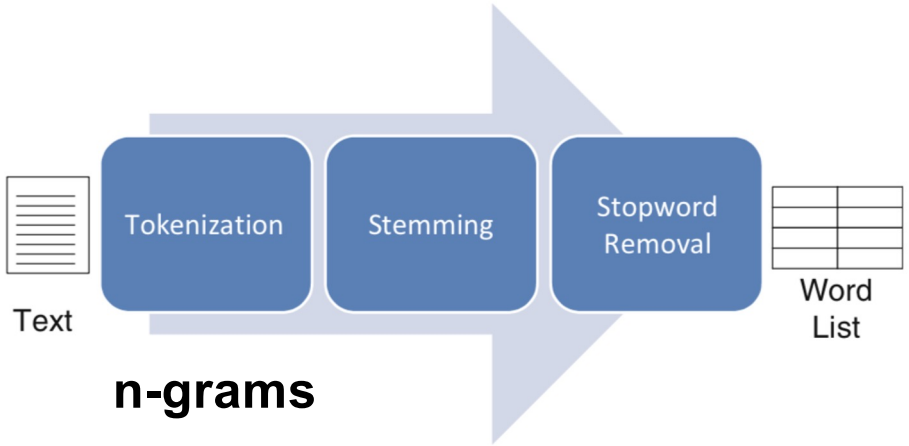
UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Diferentes técnicas
  - Tokenización
  - Remoción de stop-words (palabras de parada)
  - Stemming, lemmatization
  - POS tagging
  - BoW – bolsa de palabras – generación de vocabulario del corpus.
  - VSM – representación vectorial
  - TF-IDF – representación de características
  - Incrustaciones (embeddings) representación de características

Preparación de texto

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- pasos básicos

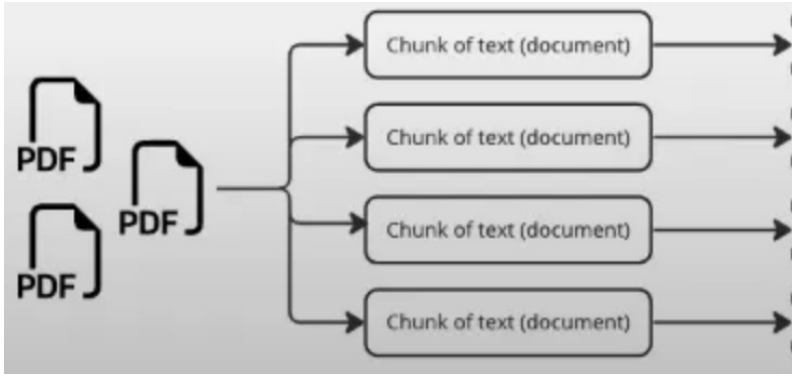


¿Como es con LLMs?

Preparación de texto

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- pasos básicos con LLMs



embeddings

Preparación de texto
TOKENIZACIÓN

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- La tokenización se define como el proceso de convertir un texto o textos en una lista de palabras (tokens) bajo un esquema de n-gram. (1-gram (default), bigram, trigram, n-gram)

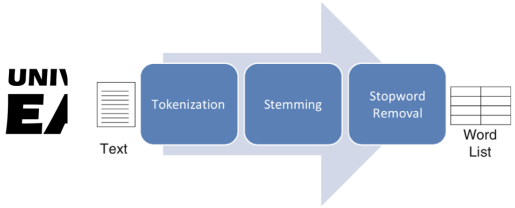
- Se tokeniza, para facilitar la representación ‘procesable’ numéricamente.

Preparación de texto
TOKENIZACIÓN

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

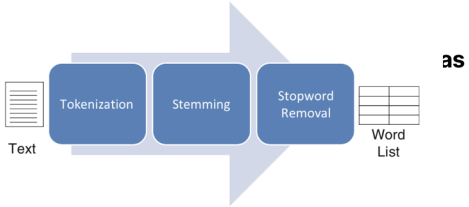
- Es imposible aplicar operaciones numéricas SOBRE textos directamente y no es fácil codificar TODO un texto en su propio valor numérico.
- Un texto es una cadena demasiado larga que difiere de una cadena corta, por lo que es muy difícil darle a un texto un valor único categórico.
- Por lo tanto, hay necesidad de segmentar un texto en tokens para luego realizar la representación.
- Opción de tokenización: n-grams:
  - 1-gram – 1 palabra,
  - 2-gram – 2 palabras seguidas,
  - 3-gram – 3 palabras seguidas, etc)
  - O chunks para embeddings (doc2vec, LLM)

Preparación de texto



- tokenización, que es el proceso de segmentar un texto mediante espacios en blanco o signos de puntuación en tokens.
- Remoción de caracteres especiales / normalización de tokens.
- stemming es el proceso de convertir cada token en su propia forma de raíz utilizando diferentes reglas (sintácticas, gramaticales, etc)
- stop-words o palabras de parada, es el proceso de eliminar las palabras gramaticales como artículos, conjunciones y preposiciones.
- lematización, llevar a una forma base diferentes palabras. forma canónica o base. ej: be por is, are, were, etc.

Preparación de texto

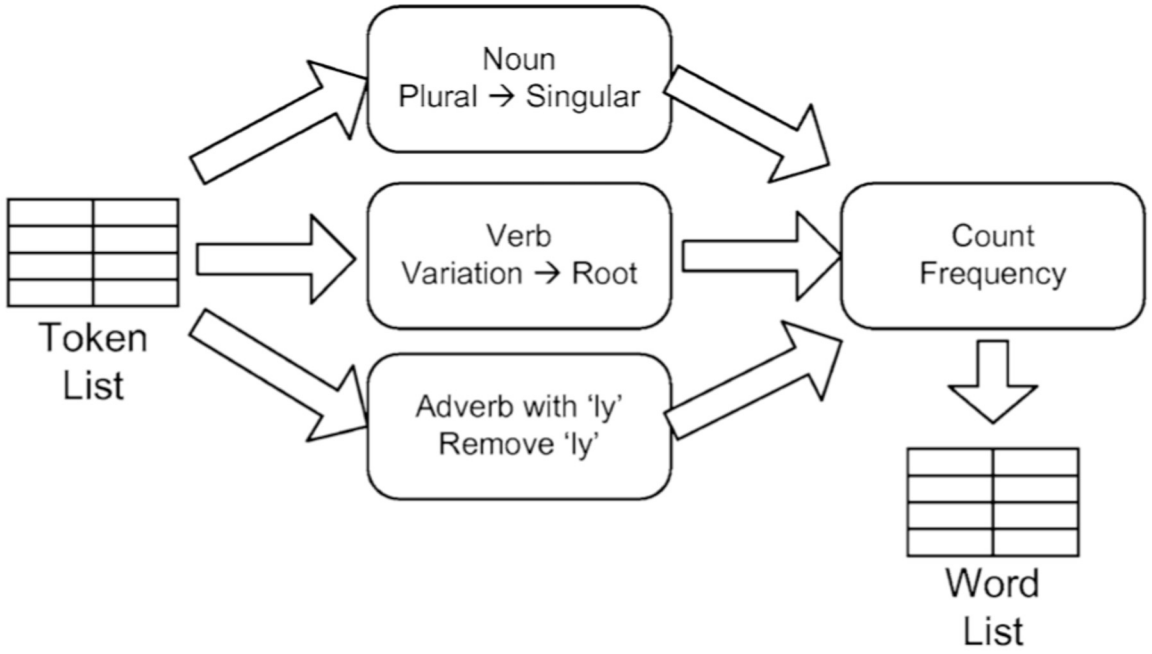


- otras etapas:
  - POS (part of speech) tagging: proceso de clasificar palabras en sustantivos, verbos, adjetivos, etc., gramaticalmente a la indexación de texto como un módulo adicional.
  - NER: Named Entity Reconigtion.

Text categorization refers to the process of assign a category or some categories among predefined ones to each document, automatically. Text categorization is a pattern classification task for text mining and necessary for efficient management of textual information systems.

Tokens
text
categorization
refers
to
the
process
of
assign
a
category
.....

Stemming y Lemmatization



Varied Form	Root Form
better	good
best	good
simpler	simple
simplest	simple
assigning	assign
assigned	assign
assignment	assign
complexity	complex
analysis	analyze
categorization	categorize
categorizing	categorize
categorizes	categorize

Contenido

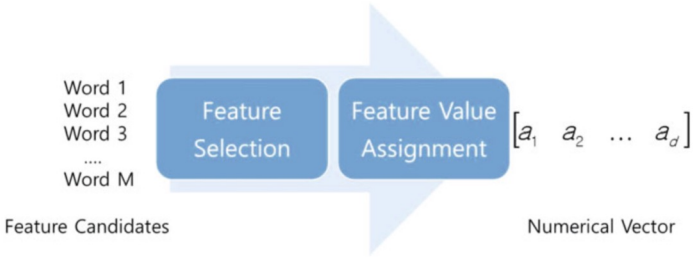
UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

1. Intro NLP
2. Representación del texto
3. Preparación de texto
4. Características y Representación de documentos
5. Modelos de recuperación

Text Encoding - representación

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- Proceso de codificación de tokens en vectores numéricos con sus valores (features – características)
- Después de la tokenización, entra a un proceso codificación (REPRESENTACIÓN) de texto
- A los tokens del texto se les definen características y se les asignan valores numéricos al codificar cada Token.
- El vector numérico que se genera a partir del proceso representa un texto



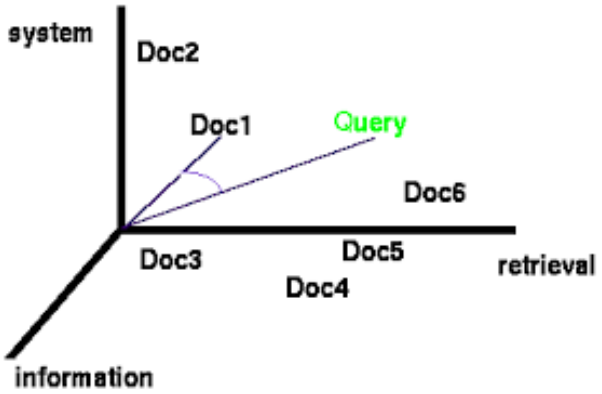
Selección de características

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- Una característica, es una representación numérica del texto (tokens)
- Hay varios modelos: bit-one, Term Frequency (TF), Document Frequency (DF), Inverse DF (IDF), Document Length (DL), etc. o combinación
- Visión actual de representación como Embbedings (word2vec, doc2vec, LLM)

ASIGNANDO VALORES A LAS CARACTERÍSTICAS:

- Esquemas de asignación (TF, DF, IDF, DL, etc)
- Para que? cálculo de similaridad para el ranking o la recuperación (una vez se cruza Las Características con la Representación) (indexación, búsqueda, ranking, etc)
  - Funciones de similitud: producto punto, coseno, distancia euclidiana, jaccard
  - Indexación: Indice invertido / vector storage
  - Ranquin: funciones de similitud
  - Modelos de ML



Selección de características

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Es el proceso de extraer palabras/términos importantes de un documento para ser considerado como entrada a un clasificador.
- Métodos comunes:
  - TF-IDF (term frequency – inverse document frequency)
  - Reducción de dimensionalidad: PCA (principal component analysis) y métodos para el texto: HashingTF, CountVectorizer, etc)
  - Embeddins: word2vec, doc2vec, LLMs, etc

Extracción de características desde el Texto

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- BoW
- Text vectorization
  - Matriz enorme, altamente dispersa
- Problemas:
  - Se pierde el orden de las palabras
  - La métrica no es normalizada

"good song"
"not a good song"
"did not like"

good	song	not	a	did	like
1	1	0	0	0	0
1	1	1	1	0	0
0	0	1	0	1	1

Problemas con el BoW

- Solución: utilizar representación n-gram, tokens (1-gram), bigram, trigram.

"good song"
"not a good song"
"did not like"

good song	song	did not	a	...
1	1	0	0	...
1	1	0	1	...
0	0	1	0	...

- Nuevos problemas:
  - Muchas más características
    - Solución:
      - Remover n-grams: stop-words, y n-grams de baja frecuencia. Quedarse con los de frecuencia media
      - No todo es ideal, los n-grams de baja frecuencia son más discriminantes

TF-IDF

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- El peso de la palabra t en el documento d es:
  - **Peso bajo** – t poca frecuencia en el documento.
  - **Peso 0** – t no existe en el documento.
  - **Peso alto** – t tiene alta frecuencia en d, y baja frecuencia en otros documentos en el conjunto de entrenamiento.

- $w(t,d) = tf(t,d) * idf(t,d) = f(t,d) * \log(\frac{N}{1+cf(t)})$

- d es el documento.
- t es una palabra en d.
- f(t,d) es la frecuencia de t en d.
- N número total de documentos en el training set
- cf(t) es la frecuencia acumulada de t en todo el training set.

TF-IDF

- TF -> f(t,d)
  - Pesos del TF
    - Binario, 0,1
    - Contador: f(t,f)
    - Term frecuency: f(t,d) / ∑ f(t',d) … t' ∈ d
    - Normalización logaritmica: 1+log(f(t,d))

- IDF: Inverse Document Frequency
  - Priorizar la distribución de un termino t en el dataset
  - a menor distribución, mayor importancia.
  - A mayor distribución, menos importancia (p.e. stop words)
  - Porque invertido?

Ejemplo TF-IDF en python con scikit-learn

UNIVERSIDAD EAFIT

Escuela de **Ciencias Aplicadas e Ingeniería**

```python
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
texts = [
    "good song",
    "not a good song",
    "did not like",
    "i like it",
    "good one"
]
# using default tokenizer in TfidfVectorizer
tfidf = TfidfVectorizer(min_df=2, max_df=0.5, ngram_range=(1, 2))
features = tfidf.fit_transform(texts)
pd.DataFrame(
    features.todense(),
    columns=tfidf.get_feature_names()
)
```

	good song	like	not	song
0	0.707107	0.000000	0.000000	0.707107
1	0.577350	0.000000	0.577350	0.577350
2	0.000000	0.707107	0.707107	0.000000
3	0.000000	1.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000

UNIVERSIDAD EAFIT

Escuela de
Ciencias Aplicadas
e Ingeniería

Porque Reducción de Dimensionalidad?

Alta Dispersión de los vectores y matrices
Alta dimensionalidad, baja representación

Reducción de dimensionalidad: PCA para texto

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- PCA es una técnica muy utilizada en reducción de variables en procesamiento estadístico.
- En esencia es reducir los datos en algo menor que capture la esencia de los datos.
- Dataset X, que contiende m documentos, y cada documento Di representado como un VSM
  - Di = [t0,t1,t2, …., tn] 1 * n
    - ti será alguna métrica como TF por ejemplo.
- Input: matriz X = m * n
- Proceso: calcular la descomposición propia de la matriz de covariancias Xt.
  - Descomposición propia: hallar los valores propios y vectores propios
- Output: features o palabras (words) que tienen un valor mayor o menor en los vectores propios.

CountVectorizer y HashingTF

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

- CountVectorizer
  - Transforma el texto en una matriz de conteo de palabras, donde cada columna representa una palabra única del vocabulario y el valor es la cantidad de veces que aparece esa palabra en un documento.
  - **Funcionamiento**:
    - Construye un vocabulario a partir del corpus.
    - Asigna un índice a cada palabra del vocabulario.
    - Genera una matriz dispersa donde cada entrada representa el conteo de palabras.
  - Filtrar características por altas frecuencias.
  - TopTF, ejemplo: # caractecticas = 1000, toma las 1000 wordcount de más alta frecuencia.
  - Necesita 2 pasadas
- HashingTF
  - Representa el texto usando una técnica de *hashing*, donde cada palabra se asigna a una columna mediante una función hash. No necesita construir un vocabulario explícito.
  - **Funcionamiento**:
    - Utiliza una función hash para mapear palabras a índices en una matriz de tamaño fijo.
    - Las colisiones (dos palabras asignadas al mismo índice) son posibles, pero se controlan estadísticamente.
    - Genera una matriz dispersa con los conteos de palabras asignados según el hash.
  - Define una una función de mapeo en un espacio hash donde todas las frecuencias son mapeadas.
  - Ejemplo, size = 1000.

UNIVERSIDAD EAFIT

Escuela de
Ciencias Aplicadas
e Ingeniería

Embeddings (incrustaciones) como mecanismo de reducción de dimensionalidad

- próxima clase -

Resumen

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- Tokenización: n-gram
- Remoción de palabras de parada (SWR) -> la mayoría de las veces igual, algunos casos de modelo de lenguaje que quiero conservar como Analisis de Sentimientos con Modelo de léxico.
- Catacterísticas (features):
  - BoW / TF-IDF
- Doc & word Representation
- Reducción de dimensionalidad
  - PCA, CountVectorizer, HashingTF, Embeddings, etc.

Resumen

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- La representación de texto determina que clase de algoritmos de minería pueden ser aplicados
- Hay múltiples formas: string, palabras, reglas sintácticas, etc.
- Podemos combinar algunas
- Este curso esta enfocado en la representación a nivel de palabra

Contenido

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

1. Intro NLP
2. Representación del texto
3. Preparación de texto
4. Características y Representación de documentos
5. Modelos de recuperación

Cual es el reto de la recuperación

- Tener cientos o miles de documentos o elementos de texto, y extraer un subconjunto de ellos.
- Generalmente no es exacta la recuperación, y dependiendo del algoritmo o modelo genera unos u otros resultados.
- Se debe utilizar alguna función de ranquin.

Modelos de recuperación

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Definir la estructura sobre la que se buscará:
  - Indice Invertido
  - Indice Directo (vector storage)
- Modelos booleanos
  - And, or, not sobre el índice invertido
  - Coincidencia de terminos
  - Documentos binarios resultantes (relevante o no)
- Modelos de espacios vectoriales
  - Documentos y consultas como vectores
  - Funciones de similitud principalmente.
- Modelo probabilístico
  - Asocia una probabilidad de relevancia con la consulta
  - Puede ranquear
- Modelo de Lenguaje
  - Calcula la probabilidad de que un documento sea relevante dado un Modelo de Lenguaje
  - Considera la probabilidad de generación del documento a partir del modelo de lenguaje

Métodos basados en Similaridad

Elementos de las funciones de ranquin

- Bag of Words (BoW)
- TF -> Term Frecuency.
  - tf(term, d) -> cuantas veces aparece el término ‘term’ en el documento d
- DF -> Document Frecuency
  - df(term,C) -> cuentas veces aparece el término ‘term’ en toda la colección C
- DL -> Document Lenght
  - |d| = l, longitud del documento.

ALGUNAS METRICAS DE SIMILARIDAD

- Euclidiana
- Coseno
- Manhattan
- Minkowski
- Hamming
- Jackard
- Mahalanobis

Proceso de Recuperación con Indice Invertido

- Se ingresa la consulta -> se vectoriza
- Se utiliza el Índice Invertido para identificar los documentos Relevantes para la consulta
- Se utilizan modelos de recuperación para Ranquear
- Los documentos se ordenan
- Los documentos se muestran y se espera una retroalimentación implícita o explicita sobre la calidad de la consulta y el ranking.

Proceso de Recuperación con LLM y Vector Storage

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Se ingresa la consulta -> se genera el embedding
- Se utiliza el Vector Store para identificar los Vectores Relevantes
  - Se utilizan modelos de lenguaje para recuperación y ranquear
- Los documentos se ordenan
- Los documentos se muestran y se espera una retroalimentación implícita o explicita sobre la calidad de la consulta y el ranking.

Introducción al almacenamiento de vectores

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- En general, los documentos en texto desarrollaron una fundamentación computacional para almacenar de forma persistente el Indice Invertido y lograr así una búsqueda más rápida.
- En el caso de estos motores de texto, en general es natural buscar por una o varias palabras (tokens) en el indice invertido, con estructuras como hash tables, la recuperación es muy eficiente.
- PERO, otros tipos de información como imágenes, sonidos, habla, etc, NO CUENTAN con esas palabras claves para realizar la búsqueda, y utilizan técnicas más asociadas a QBE (Query by Example).

Introducción al almacenamiento de vectores

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Además, para muchos procesos de Minería o Aprendizaje de máquina, podria representar miles de documentos, imágenes, habla, etc como vectores que quisiera almacenar en una Base de datos optimizada para este tipo de información.
- Además, los mecanismos de embeddins que están profilefando hoy en día require estas bases de datos, sobre todo con LLM (Large Language Models).

Introducción al almacenamiento de vectores

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- Contar con una BD que almacene, recupere y procese los datos vectoriales, tiende a ser una necesidad marcada hoy en día, para las nuevas Apps de IA.
- Aplicaciones como:
  - Sistemas de recomendación
  - Búsqueda de imagenes, texto, sonido, habla, etc
  - Detección de fraude
  - Reconocimiento de habla
  - Procesamiento natural del lenguaje
  - Etc
- Bases de datos Vectoriales como:
  - Milvus, Redis, Pinecone, ElasticSearch, AWS OpenSearch, GCP Vertex AI, GCP Pinecone, Azure Cognitive Search, Chroma, Weaviate

Ejemplo con CHROMA Vector DB

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- https://www.trychroma.com/
- Colab Demo:
  - https://colab.research.google.com/drive/181Kummxd8yOyRqFu8I0aqjs2aqnOy4Fu?usp=sharing
  - Cargar datos
    - `dataset = load_dataset("sciq", split="train")`
  - Crear una colección y adicionar datos (vectores)
    - `client = chromadb.Client()`
    - `collection = client.create_collection("sciq_supports")`

```
collection.add(
ids=[str(i) for i in range(0, 100)], # IDs are just strings
documents=dataset["support"][:100],
metadatas=[{"type": "support"} for _ in range(0, 100)
],
)
```

Ejemplo con CHROMA Vector DB

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- https://www.trychroma.com/
- Colab Demo:
  - https://colab.research.google.com/drive/181Kummxd8yOyRqFu8l0aqjs2aqnOy4Fu?usp=sharing
  - Consultar datos (vectores)
    - `results = collection.query(`
    - `query_texts=dataset["question"][:10],`
    - `n_results=1)`
  - Imprimir resultados
    - `for i, q in enumerate(dataset['question'][:10]):`
    - `print(f"Question: {q}")`
    - `print(f"Retrieved support: {results['documents'][i][0]}")`
    - `print()],)`

Contenido

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

1. Intro NLP
2. Representación del texto
3. Preparación de texto
4. Características y Representación de documentos
5. Modelos de recuperación

UNIVERSIDAD EAFIT

Escuela de
Ciencias Aplicadas e Ingeniería

Gracias!