LDA VS LSA

Minería de datos

Alumno:

Isaac Perez Amayo

Leonardo Daniel Rosas Garcia

Kevin Raymond Hernandez

Programa educativo:

Ingeniería en Datos e Inteligencia Organizacional.

Presentado a:

Jose Antonio Leon Borges

Cancún, Quintana Roo Marzo del 2022

Índice:

Introducción

En este proyecto haremos una comparación de dos algoritmos de modelado de tópicos, usaremos latent semantic indexing y latent dirichlet allocation, pero primero qué es el Topic modeling?, el topic modeling consiste en descubrir automáticamente los temas de los documentos,estos son algoritmos de análisis de texto no supervisado que se utiliza para encontrar el grupo de palabras del documento dado. nosotros trabajaremos con una colección de documentos que habla sobre los tópicos de machine learning.

Problemática:

En este proyecto resolveremos cual de los dos algoritmos del modelado de tópicos es el más óptimo para el análisis de grandes volúmenes de textos, realizaremos la implementación de ambos algoritmos ya dichos que son LDA y LSA, las problemáticas que tenemos en la realización de este proyecto es que la base de datos con la que debemos trabajar, es texto no pre-procesado así que tenemos que trabajar en pre-procesando la información de manera que todas las palabras de los textos estén limpios para poder utilizar ese texto en los algoritmos de LSA y LDA, también presentamos la problemática que en los algoritmos de minería de texto para poder realizar las métricas solo tenemos dos posibles métricas ya que no hay muchas que podamos usar.

Objetivos:

Nuestros objetivos de este proyecto es determinar qué algoritmo es el más óptimo para determinar la selección de tópicos.

Justificación:
La justificación de este proyecto es que al analizar el conjunto de documentos por sus tópicos podemos identificar automáticamente los tópicos más latentes en el área del machine learning, estos algoritmos nos ayudan mucho dado que lo que logran es un gran ahorro de tiempo y permiten generar más tópicos de los temas más populares, al igual, con esto se logra que los artículos y documentos de machine learning en vez de leer temas que ya no son relevantes pues no tomarlos en cuenta y dedicarse a lo más demandado.

Marco teórico:

LDA : Latent Dirichlet Allocation o conocido como LDA es un modelo generativo que permite que conjuntos puedan ser explicados por grupos no observados que explican porqué algunas partes de los datos son similares.LDA tiene una mezcla de varias características y este usa la distribución categórica tiene una distribución a priori de dirichlet. La característica principal es que LDA sigue la hipótesis de bolsa de palabras es decir que no importa el orden de palabras esta característica es la característica que reconoce la frecuencia en cada palabra y permite que sean intercambiables y permita mayores métodos matemáticos.

LSA: el algoritmo Latent Semantic Analysis también conocido como Latent Semantic Index, LSA se caracteriza por ser un algoritmo que utiliza el modelo Bag of Words, de esta manera se obtiene una matriz que muestra la ocurrencia de los términos en el documento, donde las filas representan términos y las columnas representan los documentos, lo que permite el modelaje de tópicos es que LSA aprende temas latentes realizando una descomposición matricial en la matriz de términos y documentos, la descomposición que se usa es la descomposición de valor singular.
Ahora necesitamos explicar a que se refiere con la descomposición de valor singular, es un método de factorización de matrices que representa una matriz en el producto de dos matrices, para esto usamos una formula muy sencilla:

$$M=U\Sigma V^*$$

- M es una matriz de m × m
- U es una m×n matriz singular izquierda
- Σ es una matriz diagonal n × n con números reales no negativos.
- V es una m×n matriz singular derecha
- V* Es una matriz de n×m, que es la transpuesta de la V.

Marco conceptual:
LSA: Latent Semantic Analysis..
LDA: Latent Dirichlet Allocation.
SVD: Singular Value decomposition.

Desarrollo de proyecto:

Importacion de librerias

```
import pandas as pd
import numpy as np
import re
import wordcloud
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import CountVectorizer
import matplotlib.pyplot
import warnings
from sklearn.decomposition import LatentDirichletAllocation as LDA
from sklearn.decomposition import TruncatedSVD
from sklearn.decomposition import PCA
```

Data frame de papers

```
papers = pd.read_csv("papers.csv")
papers.head()
```

	id	year	title	event_type	pdf_name	abstract	paper_text
0	1	1987	Self-Organization of Associative Database and ...	NaN	1-self-organization-of-associative-database-an...	Abstract Missing	767\n\nSELF-ORGANIZATION OF ASSOCIATIVE DATABA...
1	10	1987	A Mean Field Theory of Layer IV of Visual Cort...	NaN	10-a-mean-field-theory-of-layer-iv-of-visual-c...	Abstract Missing	683\n\nA MEAN FIELD THEORY OF LAYER IV OF VISU...
2	100	1988	Storing Covariance by the Associative Long-Ter...	NaN	100-storing-covariance-by-the-associative-long...	Abstract Missing	394\n\nSTORING COVARIANCE BY THE ASSOCIATIVE\n...
3	1000	1994	Bayesian Query Construction for Neural Network...	NaN	1000-bayesian-query-construction-for-neural-ne...	Abstract Missing	Bayesian Query Construction for Neural\nNetwor...
4	1001	1994	Neural Network Ensembles, Cross Validation, an...	NaN	1001-neural-network-ensembles-cross-validation...	Abstract Missing	Neural Network Ensembles, Cross\nValidation, a...

Limpieza de datos

Borraremos los datos que no usaremos

```
papers = papers.drop(columns=['id','event_type','pdf_name'])
papers.head()
```

	year	title	abstract	paper_text
0	1987	Self-Organization of Associative Database and ...	Abstract Missing	767\n\nSELF-ORGANIZATION OF ASSOCIATIVE DATABA...
1	1987	A Mean Field Theory of Layer IV of Visual Cort...	Abstract Missing	683\n\nA MEAN FIELD THEORY OF LAYER IV OF VISU...
2	1988	Storing Covariance by the Associative Long-Ter...	Abstract Missing	394\n\nSTORING COVARIANCE BY THE ASSOCIATIVE\n...
3	1994	Bayesian Query Construction for Neural Network...	Abstract Missing	Bayesian Query Construction for Neural\nNetwor...
4	1994	Neural Network Ensembles, Cross Validation, an...	Abstract Missing	Neural Network Ensembles, Cross\nValidation, a...

Titulos limpieza

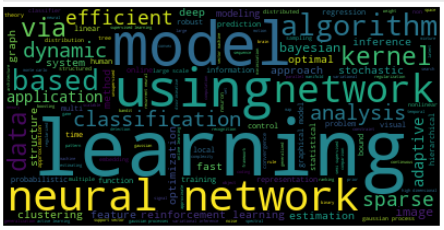
Usaremos la limpieza de los titulos para poder crear modelos de prediccion ya que nesesitamos dejar solo las palabras que se ocuparan para crear el modelo de prediccion y eliminando los simbolos y las mayusculas para evitar la repeticion de errores de palabras

```
print(papers['title'].head())
papers['title_processed'] = papers['title'].map(lambda x: re.sub('[,\.!?]', '', x))
papers['title_processed'] = papers['title_processed'].map(lambda x: x.lower())
papers['title_processed'].head()
```

Wordcloud de titulos

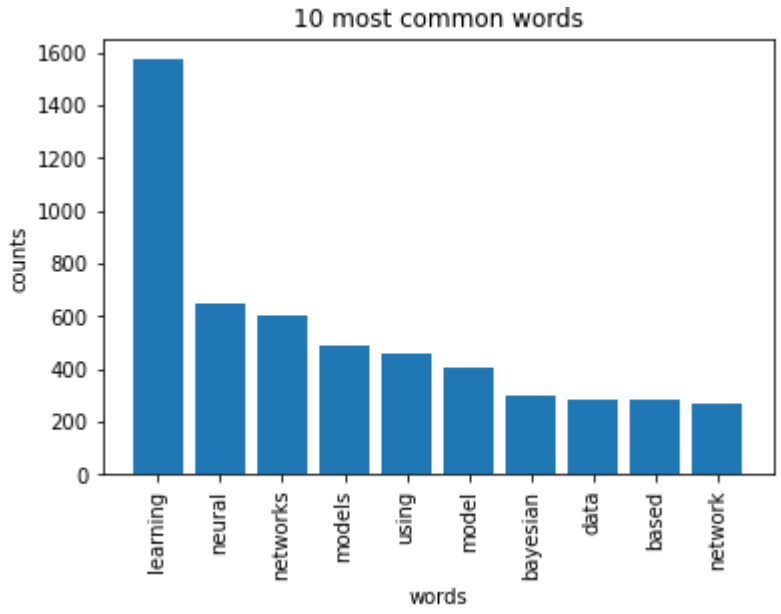
Crearemos un wordcloud para ver cuales palabras son las más comunes de manera grafica de manera vizual

```
long_string = ' '.join(papers['title_processed'])
wordcloud = wordcloud.WordCloud()
wordcloud.generate(long_string)
wordcloud.to_image()
```



Palabras mas comunes en titulos

```
def plot_10_most_common_words(count_data, count_vectorizer):
    import matplotlib.pyplot as plt
    words = count_vectorizer.get_feature_names()
    total_counts = np.zeros(len(words))
    for t in count_data:
        total_counts+=t.toarray()[0]

    count_dict = (zip(words, total_counts))
    count_dict = sorted(count_dict, key=lambda x:x[1], reverse=True)[0:10]
    words = [w[0] for w in count_dict]
    counts = [w[1] for w in count_dict]
    x_pos = np.arange(len(words))

    plt.bar(x_pos, counts,align='center')
    plt.xticks(x_pos, words, rotation=90)
    plt.xlabel('words')
    plt.ylabel('counts')
    plt.title('10 most common words')
    plt.show()

count_vectorizer = CountVectorizer(stop_words='english')

count_data = count_vectorizer.fit_transform(papers['title_processed'])

plot_10_most_common_words(count_data, count_vectorizer)
```



Resultados:

Conclusiones:

Referencias bibliográficas:

- Latent Dirichlet Allocation. (2020, 18 de marzo). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 03:27, marzo 28, 2022 desde https://es.wikipedia.org/w/index.php?title=Latent_Dirichlet_Allocation&oldid=124355588.
- Stochastic Variational Inference", Matthew D. Hoffman, David M. Blei, Chong Wang, John Paisley, 2013
- Cvitanic, T., Lee, B., Song, H. I., Fu, K., & Rosen, D.. *LDA v. LSA: A Comparison of Two Computational Text Analysis Tools for the Functional Categorization of Patents*.

International Conference on Case-Based Reasoning, (). Retrieved from https://par.nsf.gov/biblio/10055536.

-