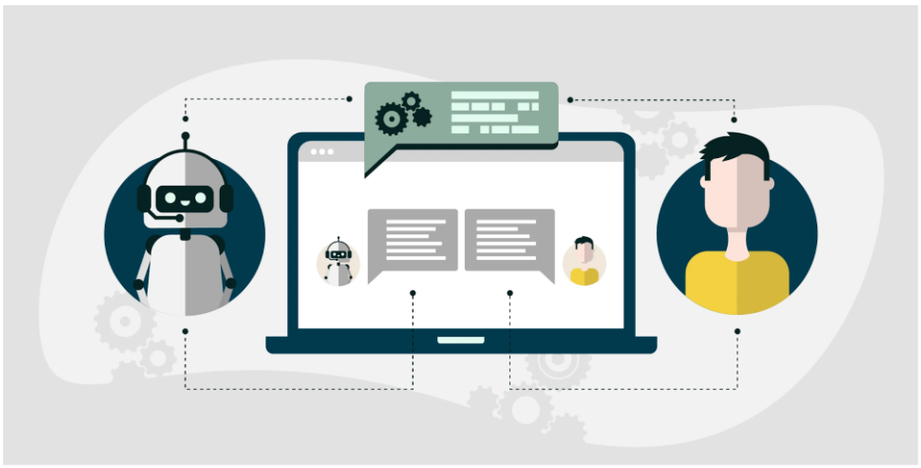
Equipo 1.
Práctica 1: Creación de Diccionario para aplicación de word2vec.
Ávila Pérez José Alberto.
Martínez García Diana Karina.
Vázquez Sánchez Fernando.
Tercero Lopez Alexis Uriel.
Espada López Itzayana
Gutiérrez Palencia Mario.

NATURAL LANGUAGE PROCESSING (NLP).

Definición: El lenguaje natural surgió por la necesidad de comunicar al usuario con la computadora. Por lo que podemos decir que el procesamiento del lenguaje natural (NLP) es el enfoque computarizado para analizar texto, que se basa tanto en un conjunto de teorías como en un conjunto de tecnologías, dedicado a que las computadoras comprendan declaraciones o palabras escritas en lenguajes humanos.

Historia. La historia de la NLP generalmente comienza en 1950, Alan Turing publicó un artículo titulado "Máquina e inteligencia" que anunciaba lo que ahora se llama la prueba de Turing como un subcampo de inteligencia. Algunos sistemas de lenguaje natural beneficiosos y exitosos que se desarrollaron en 1960 fueron SHRDLU, un sistema de lenguaje natural que funciona en "blocks world" usando un conjunto reducido de palabras que se escribió entre 1964 y 1966.



Encadenamiento de tareas analíticas de NLP.

Cualquier tarea práctica de NLP debe realizar varias subtareas. Por ejemplo, todas las tareas de bajo nivel de la sección de subproblemas de NLP deben ejecutarse secuencialmente, antes de que puedan comenzar las tareas del nivel superior.

Hay 5 fases involucradas en el procesamiento del lenguaje natural.

1. Análisis morfológico y léxico.
2. Análisis sintáctico.
3. Análisis semántico.
4. Integración del discurso.
5. Análisis pragmático.

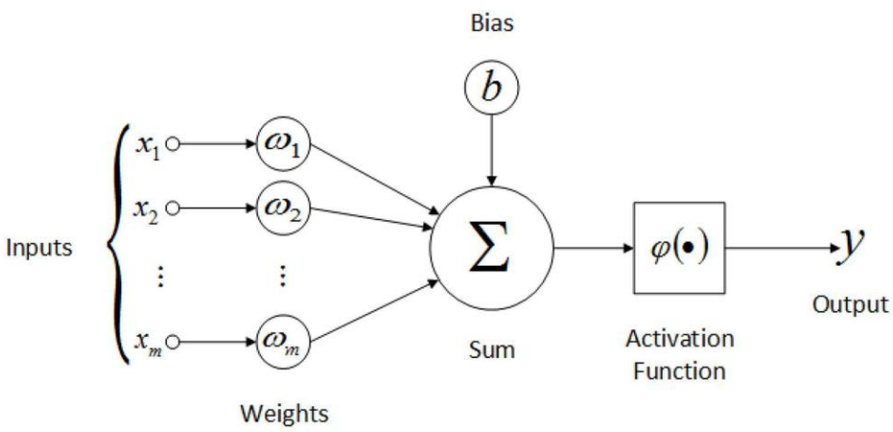
Actualidad. Muchas personas se involucran con el procesamiento del lenguaje natural a diario, Siri, Alexa o Google Assistant son un ejemplo de interacción con la NLP. Otros usos de la NLP incluyen la detección de spam, el análisis de opiniones, la respuesta a preguntas y la identificación del habla.

Google está utilizando NLP para muchos casos de uso, como su herramienta de traducción, pero una con la que quizás esté menos familiarizado es la detección de spam en Gmail. Gracias a NLP, Gmail es capaz de filtrar los mensajes de spam de su bandeja de entrada.

REDES NEURONALES.

Definición. De manera sencilla una **ANN** es la pieza de un sistema informático diseñado para simular la forma en que el cerebro humano analiza y procesa la información.Las **ANN** tienen capacidades de autoaprendizaje que les permiten producir mejores resultados a medida que se dispone de más datos.

Como sabemos hoy en día existen muchos algoritmos de Inteligencia Artificial, las redes neuronales pueden realizar lo que se denomina **Deep Learning - (Aprendizaje Profundo).**



Entonces podemos concluir que una red Neuronal es un algoritmo del aprendizaje automático - (Machine Learning). Esto significa que permite que un programa aprenda algo analizando muestras de entrenamiento con datos etiquetados o no etiquetados.

En el fondo de los algoritmos de Machine Learning convencionales presentan formas estadísticas ordenadas de predecir algunos resultados, ya sea el teorema de Naïve Bayes o los famosos árboles de decisión , las redes neuronales intentan reconocer patrones y aprender una tarea emulando nuestros cerebros.

Historia. El primer trabajo informado en el campo de las redes neuronales comenzó en 1940 , con Warren McCulloch y Walter Pitts intentando crear una red neuronal simple con circuitos eléctricos.

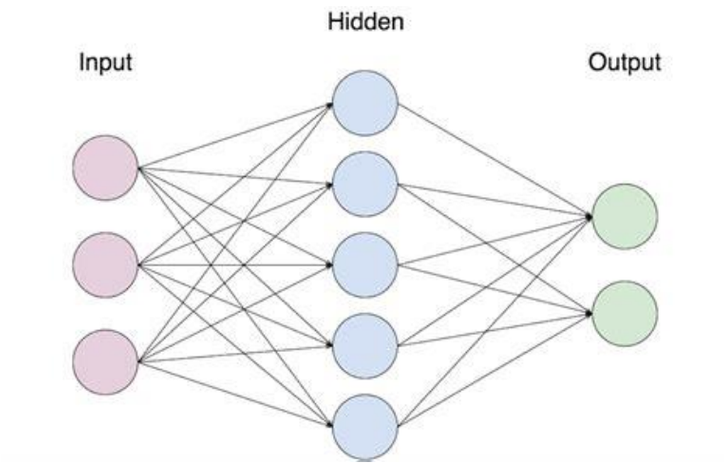
La investigación de Warren & Walter provocó que la investigación se dividiera en dos enfoques. Un enfoque se centró en la aplicación de procesos biológicos, mientras que el otro se centró en la aplicación de redes neuronales a la inteligencia artificial.

Estructura de una ANN. Hay principalmente tres capas en las redes neuronales artificiales:

1. Capa de Entrada / Input Layer. La capa de entrada es la que contiene neuronas que son responsables de las entradas de características, es decir de nuestros datos. Además también hay una neurona para el sesgo, recordemos que es sesgo es responsable de la transferencia de la línea o curva desde el origen. En total hay n+1 neuronas en la capa de entrada.

2. Hidden Layer / Capa oculta. Las capas ocultas son las capas que se encuentran entre las capas de entrada y salida. El número de capas ocultas puede variar según la aplicación y la necesidad**.**

3. Output Layer / Capa de salida. La capa de salida contiene neuronas responsables de la salida del problema de clasificación o predicción.



Aplicaciones de una red neuronal artificial.

1. Clasificación y categorización de textos. La clasificación de texto es una parte esencial en muchas aplicaciones, como la búsqueda web , el filtrado de información, la identificación de idiomas, el análisis de sentimientos etc..

2. Reconocimiento de voz. El reconocimiento de voz tiene muchas aplicaciones, como robótica, telefonía móvil, asistencia virtual, videojuegos, entre otros.

3. Detección de objetos. La detección de objetos a partir de imágenes se usa ampliamente para detectar cualquier objeto y clasificar la imagen en base a eso. Necesita un gran conjunto de datos de entrenamiento con todas las coordenadas del objeto de interés claramente especificadas.

Estas solo son algunas aplicaciones, pero hay muchísimas más, como la detección de tumores, o predecir los salarios de un empleado, verificar la calidad de los productos de la producción a gran escala, al momento de traducir un texto entre otro.

WORD EMBEDDING.

Definición. Los Word Embeddings son funciones que nos permiten mapear palabras a un vector n-dimensional, de números reales, partiendo del supuesto que palabras que se encuentran en un espacio semejante deben tener algún tipo de relación.

Conceptualmente implica el encaje matemático de un espacio con una dimensión por palabra a un espacio vectorial continuo con menos dimensiones.

Apunta para cuantificar y categorizar las semejanzas semánticas entre elementos lingüísticos basándose en sus propiedades distribucionales en muestras grandes de

datos de lengua. Con ese supuesto, se han creado modelos que intentan capturar la mayor cantidad de información posible del entorno en una palabra, llegando a poder contener incluso información semántica, y sintáctica.

Aplicaciones Words Embedding. Al ser una técnica de la NLP se le utiliza para generar una gran gama de algoritmos y propuestas que han mejorado el comportamiento de muchas tareas de Procesamiento de Lenguaje Natural, cómo es la identificación de nombres, identificación de idiomas, traducción automática, etc. En este caso daremos un caso práctico para poder usarlo de manera sencilla en nuestros programas.

Algoritmos de word Embedding

1. Embedding Layer. Es una word embedding que se aprende junto con modelo de red neuronal en una tarea específica de Natural Language Processing, como el modelado de lenguaje o clasificación de documentos.

2. Word2vect. Es un método eficiente para aprender Word Embedding de una manera eficiente, independiente del corpus del texto.

Fue desarrollado por Tomas Mikolov, et al. en Google en 2013 como una respuesta para hacer que el entrenamiento basado en redes neuronales de la incrustación sea más eficiente y desde entonces se ha convertido en el estándar para desarrollar incrustaciones de palabras previamente entrenadas.

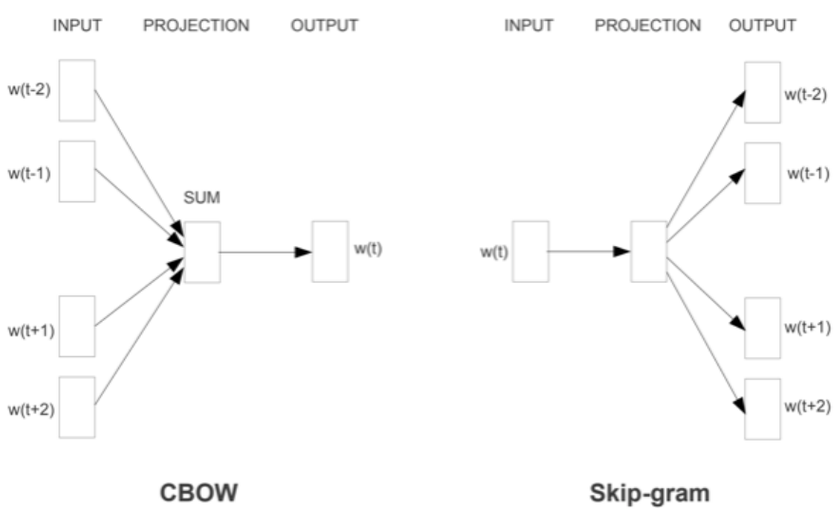
Además, el trabajo involucró el análisis de los vectores aprendidos y la exploración de la matemática vectorial en las representaciones de palabras.

Se introdujeron dos modelos de aprendizaje diferentes que se pueden utilizar como parte del enfoque de word2vec para aprender la palabra incrustación; son:

- Continuous Bag-of-Words, or CBOW model (Bolsa de palabras continua o modelo CBOW)
- Continuous Skip-Gram Model. (Modelo continuo de omisión de gramo)

El modelo CBOW aprende la incrustación al predecir la palabra actual en función de su contexto. El modelo Continuous Skip-Gram aprende prediciendo las palabras circundantes dada una palabra actual.

El modelo Continuous Skip-Gram aprende prediciendo las palabras circundantes dada una palabra actual.



Ambos modelos están enfocados en aprender sobre palabras dado su uso local, donde el contexto se define por una ventana de palabras vecinas. Esta ventana es un parámetro configurable del modelo.

3. GloVe. Global Vectors for Word Representation (algoritmo de Vectores Globales para Representación de Palabras), es una extensión del método word2vec para el aprendizaje eficiente de vectores de palabras.

Las representaciones clásicas del modelo de espacio vectorial de palabras se desarrollaron utilizando técnicas de factorización matricial como Latent Semantic Analysis (LSA), que hacen un buen trabajo al usar estadísticas de texto global pero no son tan buenos como los métodos aprendidos como word2vec para capturar el significado y demostrarlo en las tareas, cómo calcular analogías.

GloVe es un enfoque para combinar las estadísticas globales de las técnicas de factorización matricial como LSA con el aprendizaje local basado en el contexto en word2vec.

En lugar de utilizar una ventana para definir el contexto local, GloVe construye una matriz explícita de palabra-contexto o co-ocurrencia de palabras utilizando estadísticas en todo el corpus de texto. El resultado es un modelo de aprendizaje que puede resultar en una mejor inserción de palabras en general.

GloVe, es un nuevo modelo de regresión log-bilineal global para el aprendizaje no supervisado de representaciones de palabras que supera a otros modelos en analogía de palabras, similitud de palabras y tareas de reconocimiento de entidades nombradas.

Bibliografía.

1. Brwleen, J. (2019, Agosto 07). Algorithms Word Embedding. What Are Word Embeddings for Text?, https://machinelearningmastery.com/what-are-word-embeddings/, fecha de consulta: Octubre 05, 2020.

2. Word embedding. (2018, octubre 23). Qué son los word embedding y cómo usarlos,http://eenube.com/index.php/ldp/machine-learning/137-que-son-los-word-embeddings-y-como-usarlos, fecha de consulta: Octubre 04, 2020.

3. Abhimanyu Chopra, Abhinav Prashar, Chandresh Sain, (2013). Natural Language Processing,http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.407.6907&rep=rep1&type=pdf, fecha de consulta: Octubre 03, 2020.

4. Prakash M Nadkarni, Lucila Ohno-Machado, Wendy W Chapman, (2011, Septiembre 01), Natural language processing: an introduction, https://academic.oup.com/jamia/article/18/5/544/829676#31768955, fecha de consulta: Octubre 03, 2020.

5. Sayan De, (2019, Agosto 27), What is Artificial Neural Networks(ANN)?, https://tec4tric.com/ml/nn/what-is-artificial-neural-networks, Fecha de Consulta: Octubre 3, 2020.

6. sanya4, (2020, Septiembre 29), A Quick History of Neural Networks, https://www.analyticsvidhya.com/blog/2020/09/quick-history-neural-networks/, fecha de consulta: Octubre3, 2020.

7. Data Monsters, (2017, Agosto 17), 10 Applications of Artificial NeuralNetworks in Natural Language Processing, https://medium.com/@datamonsters/artificial-neural-networks-in-natural-language-processing-bcf62aa9151a, fecha de consulta: Octubre 3, 2020.

8. Andrew Ng, sin fecha, IA para todos, https://www.coursera.org/learn/ai-for-everyone, fecha de consulta: Octubre 3, 2020.

9. Sanket Doshi, (2019, Marzo 16), Skip-Gram: NLP context words prediction algorithm, https://towardsdatascience.com/skip-gram-nlp-context-words-prediction-algorithm-5bbf34f84e0c, fecha de consulta: Octubre 5, 2020.

10. Dario Radečić, (2020, Junio 18), Softmax Activation Function Explained, https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60, fecha de consulta: Octubre 5, 2020.

11. Mikolov, Tomas, Sutskever, Ilya, Chen,Kai, Corrado, Greg, Dean, Jeffrey, (Sin fecha),Distributed Representations of Words and Phrasesand their Compositionality, https://sea.acatlan.unam.mx/pluginfile.php/184625/mod_folder/content/0/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf?forcedownload=1, fecha de consulta: Octubre 5, 2020.

12. Derek Chia, (2018, Diciembre 5), An implementation guide to Word2Vec using NumPy and Google Sheets, https://towardsdatascience.com/an-implementation-guide-to-word2vec-using-numpy-and-google-sheets-13445eebd281, fecha de consulta: Octubre 5, 2020.

13. Chris McCormick, (2017, Enero 11), Word2Vec Tutorial Part 2-Negative Sampling, http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/, fecha de consulta: Octubre 5, 2020.

14. Michael U. Gutmann, Aapo Hyvärinen, (2013), Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics, https://www.jmlr.org/papers/v13/gutmann12a.html, fecha de consulta: Octubre 5, 2020.