Actividad 1: Introducción a NLP tradicional

Teoría

1. ¿Qué es NLP?
   1. El procesamiento del lenguaje natural (NLP) es un subcampo de la lingüística, la informática y la inteligencia artificial que ofrece un conjunto de herramientas y algoritmos necesarios para analizar el gran volumen de datos no estructurados en lenguaje natural. El objetivo es conseguir que una máquina sea capaz de comprender el contenido de los documentos, incluidos los matices contextuales del lenguaje que contienen. De este modo, la tecnología puede extraer con precisión la información y las ideas contenidas en los documentos, así como clasificarlos y organizarlos.
2. ¿Cuáles son los componentes del lenguaje?
   1. Caracteres, palabras, oraciones, párrafos. Lingüística (estudio semántico del lenguaje: contexto, sintaxis, morfemas y lexemas, fonemas). Fonema (unidad sonora). Morfema (combinación de fonemas, pueden ser palabras o prefijos y sufijos). Lexema (variación de morfema, relacionado a su significado… ej: Equipos 🡪 Equipo | Corriendo 🡪 Correr). Sintaxis (reglas para construir gramaticalmente las oraciones). Contexto (da significado a las oraciones). Semántica (significado de oraciones sin contexto). Pragmática (agrega contexto para obtener el significado de las frases)-
3. Enumere y defina los desafíos en NLP
   1. Ambigüedad: Incertidumbre de semántica. La mayoría del lenguaje natural es ambiguo.
   2. Conocimiento común: Conocimiento que disponen en común dos interlocutores. No es nombrado directamente, pero ambos interlocutores son conscientes de que se está hablando de algo en común. Se usa para entender la comunicación.
   3. Creatividad: Lenguaje no definido por reglas. Aplica creatividad a estilos, dialectos, géneros y variaciones.
   4. Diversidad en lenguajes: No hay mapeo entre vocabularios entre dos lenguajes. Soluciones en un lenguaje no funcionan en otro. Difícil construir soluciones agnósticas de lenguaje.
4. Enumere las tareas que existen en NLP, elegir 3 y definirlas.
   1. Language modeling: Predecir la próxima palabra en una oración basado en el historial previo. Aprender la probabilidad de ocurrencia de una secuencia de palabras dado un lenguaje.
   2. Text classification: Clasificar palabras en un set de categorías basado en su contenido.
   3. Information extraction:
   4. Information retrieval: Encontrar documentos relevantes a partir de queries de una gran colección de recursos.
   5. Conversational agent: Construir Dialogue systems para conversar con humanos en lenguaje natural.
   6. Text summarization: Crear resúmenes cortos de documentos grandes. Retener el contenido principal. Preservar el significado del texto.
   7. Question Answering: Construir Question answering system para responder preguntas de manera automática en lenguaje natural.
   8. Machine translation: Convertir un fragmento de texto de un lenguaje a otro.
   9. Topic Modeling: Descubrir tópicos de una gran colección de documentos.
5. ¿Qué es un agente conversacional (Chatbot)?
   1. Construir sistemas de diálogo para conversar con humanos en lenguaje natural. Tareas: Conversión de voz a texto; Análisis semántico y de sentimientos; Responder preguntas; Traducción de máquina. Puede aplicar a dominios de propósito general o aplicado a campos específicos.
6. Enumere y explique cada una de las fases del flujo de trabajo en NLP.
   1. Data Adquisition: Extrae datos de Dataset públicos, Web scraping (descargando páginas html), product intervention (guardar y extraer texto de bases de datos), Data augmentation (tomar datos reales y alterarlos…Cambiamos palabras por sinónimos y hacemos varias combinaciones entre sí, agregamos ruido)
   2. Text cleaning: Se toman htmls, pdfs o se extrae texto de imagen. Se parsea el texto de html o se escanea la palabra. Se limpian los metadatos. Convertir texto al formato esperado y se normaliza. Revisión de ortografía y corrección de errores.
   3. Pre-Processing
      1. Preliminaries
         1. Sentence Segmentation 🡪 Se divide la oración en palabras y caracteres
         2. tokenization (char, grupos de chars o palabras) 🡪 Un token es una estructura de datos que va a tener la información que vayamos obteniendo en todo el proceso
      2. Frequent steps
         1. Stop word removal 🡪 Se eliminan palabras raras que no tienen mucho en común con las demás
         2. Stemming and lemmatization
            1. Stemming: las palabras pueden compartir un mismo stem (primera parte de la palabra). Stemming es un conjunto de técnicas que identifican el stem de la palabra, lo corta y lo extrae para guardarlo en nuestro token en el campo correspondiente.
            2. Lammatization: En este step se hace un análisis sintáctico de la palabra y se trata de obtener su

root (Raíz) en la que se deriva la palabra del texto original. El stem y root puede que coincidan o sean diferentes. Es más costoso que el stemming.

* + - 1. Removing digits and punctuation, lowercasing
    1. Other steps
       1. Normalization: convertir 3 en ‘tres’
       2. Language detection: español, inglés, portugués
       3. Code mixing, transliteration: más de un lenguaje en la misma oración.
    2. Advanced processing
       1. PoS tagging, parsing: análisis sintáctico
       2. Coreference resolution: En este caso consiste en encontrar la entidad del texto a que se refiere cuando usamos un pronombre personal.
  1. Feature Engineering:
     1. Domain Expertise (+++) para hacer feature engineering.
     2. Cada palabra debe ser representada para que pueda ser procesada.
     3. Podríamos usar números (Secuencial) pero es mejor usar vectores.
     4. Cada palabra es una dimensión
     5. Text representation
        1. One hot encoding (Representación binaria de palabras en un vector de 1 y 0).
        2. Bag of Words (Grupo de palabras que no tiene en cuenta el orden).
        3. Bag of N-grams (Agrupación de palabras para capturar algo de contexto).
        4. TF - IDF (Índice de importancia de términos en un corpus de texto).
     6. Vectorización de palabras y palabras adyacentes
  2. Modeling
  3. Evaluation
  4. Deployment
  5. Monitoring and Model Updating

1. Explique las diferencias entre un flujo de trabajo clásico y moderno.
   1. Clásico:
      1. Preprocesamiento Manual Extensivo
         1. Involucra una serie de pasos manuales como la tokenización, stemming, lematización, eliminación de stopwords para preparar los datos antes de usarlos en modelos.
      2. Extracción de Características Basada en Reglas
         1. Uso de técnicas tradicionales como bolsas de palabras (Bag of Words) o TF-IDF (Term Frequency-Inverse Document Frequency) para representar el texto como vectores
      3. Modelos Estadísticos Simples
         1. Empleo de modelos como Naive Bayes, SVM (Support Vector Machines), o regresión logística, que requieren un diseño manual de características.
      4. Menor Capacidad de Generalización
         1. Los modelos clásicos suelen tener dificultades para manejar variaciones lingüísticas, ambigüedades y entender el contexto a nivel profundo.
   2. Moderno:
      1. Uso de Representaciones de Texto Avanzadas:
         1. En lugar de técnicas simples, se emplean embeddings como Word2Vec, GloVe, o representaciones contextuales más avanzadas como BERT, GPT, y similares.
      2. Modelos de Aprendizaje Profundo
         1. Implementación de arquitecturas de redes neuronales profundas, como transformers, que pueden manejar grandes cantidades de datos y aprender representaciones complejas del lenguaje.
      3. Entrenamiento de Fin a Fin
         1. El flujo de trabajo moderno tiende a minimizar la necesidad de preprocesamiento manual extensivo, ya que los modelos pueden aprender directamente de los datos en bruto.
      4. Transferencia de Aprendizaje
         1. Uso de modelos preentrenados que se pueden ajustar (fine-tuning) a tareas específicas con menos datos y menos tiempo de entrenamiento
      5. Mejor Capacidad de Generalización y Escalabilidad
         1. Los modelos modernos pueden generalizar mejor a diferentes dominios y lenguajes, y pueden manejar tareas complejas como la traducción automática, la generación de lenguaje natural, y el entendimiento de lenguaje natural

Práctica

Hacer una copia, ejecutar y analizar el colab 02 - Introduccion a NLP.ipynb en donde se realiza un flujo de trabajo clásico utilizando la biblioteca NLTK y responder las siguientes preguntas a modo de reflexión

1. ¿Qué problema se busca solucionar en el colab? Definir alcance y meta
   1. Aplicar tecnicas de NLP simbolicas y estadísticas para hacer predicciones binarias en textos. Vamos a ver el dataset [Twitter Sentiment Analysis](https://www.google.com/url?q=https%3A%2F%2Fhuggingface.co%2Fdatasets%2Fcarblacac%2Ftwitter-sentiment-analysis) que es un dataset armado por la universidad de Michigan y kaggle a partir de un corpus de tweets proporcionado por Niek Sanders. Este dataset se aplica clasificacion binaria en la que etiqueta como positivo (1) o negativo (0).
2. ¿Porque se revisa la cantidad de palabras únicas?
   1. Cada palabra es una dimensión, por lo que se cuentan las palabras únicas para ver cuantas dimensiones tendría un array de palabras.
3. ¿Por qué se busca reducir esta cantidad de palabras únicas?
   1. Para reducir la dimensión al momento de hacer el análisis.
4. Enumere y explique los pasos de preprocesamiento utilizados. Indicar por qué se aplicaron.
   1. Tokenización: se utiliza para tokenizar palabras y así poder eliminar algunas de estas sin tener que aplicar reglas.
   2. Eliminación de stopwords: se detectan las palabras más comunes, ya que no suelen proporcionar información útil
   3. Eliminación de símbolos: no aportan información del texto, así que se los incluye como stopwords para eliminarlos
   4. Aplicar stemming y lemmatization: simplifica el vocabulario reduciendo las palabras a su raíz.
5. ¿Qué son las stopwords y por qué se eliminan?
   1. No aportan valor. Y deben retirarse de los problemas de NLP.
6. ¿Cómo se representó el texto para trabajar en NLP?
   1. Se representó utilizando One Hot encoding.
7. ¿Qué ventajas y desventajas observa que podemos tener al usar el tipo de representación de texto?
   1. Ventajas de One Hot Encoding
      1. Simplicidad: es fácil de entender e implementar. Cada palabra se convierte en un vector simple que es directo y sin ambigüedades.
      2. No Hay Asunciones Sobre la Relación Entre Palabras: Como cada palabra es representada de manera independiente, no se hacen suposiciones sobre la relación entre las palabras.
      3. Determinístico: La representación es completamente determinista; es decir, el mismo texto siempre se codifica en el mismo vector, lo que facilita la reproducibilidad.
   2. Desventajas de One Hot Encoding
      1. Alta dimensionalidad: Cada palabra en el vocabulario se representa como un vector que tiene una longitud igual al tamaño del vocabulario.
      2. Pérdida de Información Semántica: One-Hot Encoding no captura ninguna relación semántica entre palabras. Palabras con significados similares o sinónimos tendrán representaciones completamente diferentes y ortogonales.
      3. Ineficiencia computacional: Dado que los vectores son esparcidos y grandes, los cálculos como la similitud de palabras o las operaciones de distancia entre vectores son ineficientes.
      4. Escalabilidad limitada: En problemas con vocabularios extensos o con texto de longitud variable, la dimensionalidad de los vectores de One-Hot Encoding puede volverse inmanejable.
      5. Incapacidad de capturar contexto: One-Hot Encoding representa palabras de manera aislada, sin considerar su contexto en una oración o documento. Esto limita su capacidad para modelar la ambigüedad y el significado contextual.
8. ¿Qué pasa si encontramos una palabra que esté fuera del vocabulario representado inicialmente con la representación usada?
   1. Se producirá un error en el modelo, dado a que todas las palabras ocupan una dimensión. Una palabra nueva requerirá usar una dimensión más, lo que significa que se debe redefinir todo.

Biblioteca spaCy: Funcionalidades básicas y lingüísticas

Hacer una copia y ejecutar el colab 03 - NLP spaCy.ipynb y responder las siguientes preguntas a modo de reflexión:

1. ¿Qué elementos guarda un token en spaCy?
   1. Texto original
   2. Lemmatización del texto
   3. Parte del discurso (adjetivo, verbo, sustantivo)
   4. Morfosintáctica: Tiempo verbal, género
   5. Dependencia sintáctica: relación con otros tokens
   6. Cabeza sintáctica: el token del cual depende el token actual
   7. Forma: longitud, mayúsculas y minúsculas
   8. Alfabético: indica si lo componen caracteres alfabéticos
   9. Dígito: indica si es dígito
   10. Puntuación: indica si es punto
   11. Entidad nominada: indica si es persona, organización o ubicación
   12. Índice: la posición en el documento
   13. Probabilidad de formar parte del discurso
   14. Lengua: idioma detectado del token
2. ¿Cómo se pueden explicar los términos de spaCy que queremos conocer?
   1. Se puede utilizar el método explain para indicar la tokenización. Y cada uno de los atributos se los puede invocar llamándolo. Ejemplo: a la lemmatización la podemos llamar como token.lemma\_
3. ¿En qué consiste tarea Named Entity Recognition y con qué fin podemos combinarlas con tareas NLP?
   1. Puede utilizarse para relacionar tareas con entidades, lo que permite ver las recepciones o emociones positivas y negativas asociadas a una persona o lugar. Asociar categorías semánticas con personas.
4. ¿Con qué fin utilizamos Rule-based matching en NLP con spaCy?
   1. Permite encontrar una regla que permita reconocer texto sin tener sin tener que entrenar todo un modelo de machine learning para solucionarlo. Aunque es difícil de leer, debugear y es propensa a errores.
5. Explicar las aplicaciones de word embeddings que se hizo en el colab proporcionado.
   1. Los word embeddings permiten representar palabras de manera que las relaciones semánticas entre ellas se reflejan en la geometría de los vectores. Los word embeddings son representaciones densas, lo que significa que un vector tiene valores continuos en muchas dimensiones. Mientras que el one-hot encoding crea un vector de longitud igual al tamaño del vocabulario (potencialmente miles o millones de dimensiones), los word embeddings típicamente tienen entre 50 y 300 dimensiones. Los word embeddings están diseñados para capturar la semántica de las palabras. Un aspecto destacado de los word embeddings es que ciertas operaciones vectoriales pueden capturar relaciones semánticas. En el colab por ejemplo, se usa para verificar la relación de entidades similares (Google y Microsoft) con otras poco relacionadas (Google y Cataratas del Iguazú)
6. Investigar qué es Semantic Parsing y qué función cumple en chatbots. Indicar qué funciones lingüísticas vistas se utilizan y con qué fin se emplean.
   1. Se trata de analizar una oración o fragmento de texto y transformarlo en una forma que pueda ser fácilmente entendida y utilizada por una máquina, como un gráfico semántico, una estructura de datos o una consulta lógica.
   2. El semantic parsing ayuda al chatbot a entender la intención detrás de una consulta.
   3. Un aspecto clave del semantic parsing es la identificación y extracción de entidades (como nombres de personas, lugares, fechas, etc.) y sus relaciones.
   4. La estructura semántica generada por el parsing puede ser convertida en una consulta que el chatbot utiliza para recuperar información de una base de datos o realizar una acción específica
   5. Los chatbots necesitan manejar un lenguaje natural que a menudo es ambiguo o complejo.
   6. El semantic parsing permite al chatbot mantener el contexto y seguir el flujo lógico de la conversación.