Actividad 3: Representación de texto moderna Word Embeddings

Teoría

1. ¿Qué es un Word embeddings?
   1. Técnica avanzada en representación de texto en lenguaje natural. Representan palabras en un espacio vectorial de alta dimensión, donde palabras con significados similares están ubicadas cerca unas de otras.
2. ¿Qué podemos representar en un espacio vectorial?
   1. Podemos encodear información como Caracteres, Palabras, Oraciones, Párrafos y Documentos
3. ¿Qué elementos necesitamos para entrenar / utilizar un modelo de embeddings?
   1. Corpus de texto grande para entrenar
   2. Modelo para el entrenamiento (word2vec, Glove, FastText embeddings)
   3. Window size: ¿Cuántas palabras alrededor del objetivo para capturar el contexto?
   4. Dimensionalidad de los embeddings
   5. Pre procesamiento de datos
4. Enumerar y describir qué mediciones se usan para comparar dos embeddings
   1. Cosine similarity: Mide el ángulo entre dos vectores en un espacio multidimensional, sin considerar la magnitud de los vectores. La similitud del coseno es especialmente útil para comparar embeddings porque se centra en la dirección de los vectores.
5. Explicar cómo se obtienen word embeddings utilizando la técnica Word2Vec (2 enfoques de entrenamiento).
   * 1. Continuous bag of words: Predecir correctamente la palabra central, dado palabras de su contexto. El Language model da probabilidades altas a oraciones buenas (sintáctica y semánticamente correctas) y probabilidades bajas a las oraciones malas (sintáctica o semánticamente incorrectas).
     2. SkipGram: En este caso la tarea es predecir las palabras del

contexto desde la palabra central.

1. Enumere y explique las diferencias de este enfoque y problemas que resuelve con respecto a los anteriores (tradicionales).
   1. Representación Densa: Cada palabra se representa como un vector en un espacio de alta dimensión donde cada dimensión es un valor continuo. Estos vectores son densos, lo que significa que la mayoría de las dimensiones tienen valores distintos de cero.
   2. Captura de semántica: Los embeddings capturan relaciones semánticas entre palabras. Palabras con significados similares están cerca en el espacio vectorial.
   3. Aprendizaje basado en contexto: Los word embeddings se entrenan usando grandes corpus de texto y capturan el contexto en el que aparecen las palabras, lo que les permite aprender las relaciones entre palabras de manera más sofisticada
2. Realizar una tabla comparativa general entre enfoques tradicionales (Word vectorization) y modernos de representación de texto (Word embeddings).

|  |  |  |
| --- | --- | --- |
| **Característica** | **Word Vectorization** | **Word Embeddings** |
| **Representación del Texto** | Vectores dispersos (One Hot, BoW, TF-IDF, etc.) | Vectores densos |
| **Tamaño de los Vectores** | Dependiente del tamaño del vocabulario (puede ser muy grande) | Fijo y bajo, independientemente del tamaño del vocabulario |
| **Sparsity (Dispersión)** | Alta, la mayoría de las dimensiones son cero | Baja, la mayoría de las dimensiones tienen valores distintos de cero |
| **Captura de Similaridad** | No captura similitud semántica entre palabras | Sí captura similitud semántica entre palabras (palabras similares tienen vectores cercanos) |
| **Contexto** | No captura el contexto en que aparecen las palabras | Sí captura el contexto en que aparecen las palabras (basado en co-ocurrencias) |
| **OOV (Out of Vocabulary)** | Problemas significativos con palabras fuera del vocabulario | Mejores manejos de OOV, especialmente con modelos como FastText que usan subpalabras |
| **Normalización** | Requiere técnicas como TF-IDF para ponderar la importancia de las palabras | No requiere normalización adicional, los embeddings se entrenan para ser significativos |
| **Ventajas** | Simplicidad, fácil de implementar, interpretable, no requiere grandes corpus de entrenamiento | Captura relaciones semánticas, eficiente en espacio, transferible a diferentes tareas |
| **Desventajas** | Vectores grandes y dispersos, no captura similitudes semánticas, ineficiente en espacio | Requiere entrenamiento en grandes corpus, puede ser difícil de interpretar |

1. Realizar una tabla comparativa con todos los modelos de la familia Word2Vec para representar texto (Word2Vec, Doc2Vec, GloVe y FastText).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Característica** | **Word2Vec** | **Doc2Vec** | **GloVe** | **FastText** |
| **Descripción** | Modelo para aprender representaciones vectoriales de palabras | Extensión de Word2Vec para aprender representaciones de documentos | Modelo basado en descomposición matricial para representar palabras | Modelo que mejora Word2Vec al usar subpalabras y morfología |
| **Tipo de Representación** | Vectores densos para palabras | Vectores densos para palabras y documentos | Vectores densos para palabras | Vectores densos para palabras (incluye subpalabras) |
| **Dimensionalidad** | Fija y baja (generalmente 100-300 dimensiones) | Fija y baja (similar a Word2Vec, pero incluye documentos) | Fija y baja (generalmente 100-300 dimensiones) | Fija y baja (generalmente 100-300 dimensiones) |
| **Técnica de Entrenamiento** | CBOW (Continuous Bag of Words) y Skip-Gram | Similar a Word2Vec, con enfoque adicional en etiquetas de documentos | Descomposición matricial basada en co-ocurrencias | Skip-Gram, similar a Word2Vec, pero considerando subpalabras |
| **Captura de Semántica** | Sí, palabras con significados similares están cerca en el espacio | Sí, tanto palabras como documentos capturan relaciones semánticas | Sí, captura relaciones semánticas a través de co-ocurrencias globales | Sí, captura relaciones semánticas y morfológicas entre palabras |
| **Manejo de OOV (Out of Vocabulary)** | Limitado, no maneja bien palabras fuera del vocabulario | Limitado, similar a Word2Vec | Limitado, no maneja bien palabras fuera del vocabulario | Mejora significativa, puede generar vectores para palabras OOV usando subpalabras |
| **Captura de Morfología** | No captura morfología | No captura morfología | No captura morfología | Captura morfología al considerar subpalabras y caracteres |
| **Contexto Considerado** | Considera contexto local (ventana de palabras) | Considera contexto local (ventana de palabras y documentos) | Considera contexto global de co-ocurrencias | Considera contexto local y estructura interna de palabras |
| **Normalización** | No requiere normalización adicional | No requiere normalización adicional | No requiere normalización adicional | No requiere normalización adicional |
| **Ventajas** | - Captura relaciones semánticas - Eficiente para tareas de NLP comunes | - Captura tanto palabras como documentos - Útil para tareas que requieren contexto de documentos | - Captura relaciones semánticas a nivel global - Basado en co-ocurrencias, útil para grandes corpus | - Captura relaciones semánticas y morfológicas - Eficaz en manejo de palabras OOV |
| **Desventajas** | - No captura morfología - Problemas con palabras OOV | - Similar a Word2Vec, pero más complejo de entrenar | - Menos eficiente en corpus pequeños | - Más complejo y computacionalmente costoso - Puede requerir más recursos para entrenar |
| **Aplicaciones Típicas** | Análisis de sentimiento, clasificación de texto | Análisis de documentos, búsqueda de información en documentos | Tareas de NLP en gran escala, análisis semántico | Tareas de NLP que requieren robustez en vocabulario, análisis morfológico |
| **Ejemplos de Uso** | Traducción automática, modelos de clasificación | Análisis de textos largos, extracción de características en documentos | Análisis semántico, modelos de recomendación | Aplicaciones multilingües, procesamiento de lenguas con alta inflexión |

Práctica

En esta actividad práctica aplicar los conceptos vistos en la teoría. Para ello realizar una copia, ejecutar y analizar los colabs 06 - Modelo Word2Vec preentrenado.ipynb y 10 - Modelo GloVe preentrenado.ipynb utilizando la biblioteca Gensim para responder las siguientes preguntas a modo de reflexión:

1. ¿Por qué se utiliza un benchmark de analogías para evaluar los resultados obtenidos del entrenamiento de embeddings?
   1. El entrenamiento de word2vec es una tarea no supervisada. La evaluación depende del objetivo de la aplicación. Por ende, no hay forma objetiva de evaluar bien el resultado.
2. ¿En qué consiste la mejora que introdujo Gensim en 2016 para evaluar la similaridad de palabras de los modelos entrenados?
   1. Se utiliza un dataset académico por defecto (pero se puede crear un dataset propio basado en un negocio particular). Contiene pares de palabras junto con juicios de similitud asignados por humanos. Mide la relación o coocurrencia de dos palabras.
3. ¿Cómo se puede hacer finetuning de un modelo de Gensim con nuestras palabras adicionales?
   1. Se carga el archivo temporal del modelo pre entrenado
   2. Se definen nuevas oraciones
   3. Se actualiza el vocabulario
   4. Se entrena el modelo con el vocabulario actualizado
   5. Se limpia el archivo temporal para guardar el modelo
4. Enumere y explique las diferentes funciones exploradas en el colab de GloVe para trabajar con similaridad entre palabras.
   1. most\_similar: encuentra el top N de palabras más similares
   2. most\_similar\_cosmul: igual que la anterior, pero utiliza el objetivo de combinación multiplicativa
   3. n\_similarity: computa la similitud de coseno entre dos conjuntos de palabras
   4. similar\_by\_vector: encuentra el top N de palabras más similares por vector
   5. similar\_by\_word: encuentra el top N de palabras más similares
   6. similarity: computa la similitud de coseno entre dos palabras
5. Investigar las técnicas PCA (Principal Components análisis) y TSNE e indicar para que se utilizaron en el colab de GloVe
   1. PCA: es una técnica estadística utilizada para la reducción de dimensionalidad en conjuntos de datos. PCA transforma un conjunto de variables posiblemente correlacionadas en un conjunto de variables no correlacionadas, llamadas componentes principales.
   2. TSNE: es una técnica de reducción de dimensionalidad y visualización de datos, especialmente útil para datos de alta dimensión. Se centra en preservar las relaciones locales entre puntos de datos. Esto significa que los puntos que están más cercanos en el espacio original seguirán siendo cercanos en el espacio reducido. Intenta mantener las relaciones de vecindad, por lo que es particularmente eficaz para detectar grupos o clústeres en los datos.
6. ¿Qué usos útiles puede encontrar aplicar similaridad entre documentos de texto?
   1. Puede servirme para encontrar temas comunes, ampliar información, detectar plagios, generar resúmenes, analizar tendencias, clasificar documentos, facilitar la recuperación de la información