1. DISTANCIA DE LEVENSHTEIN

jueves, 21 de septiembre de 2023 10:06 p. m.

También conocida como distancia de edición, propuesto por Vladimir Levenshtein en 1965, es una medida que se utiliza para calcular la similitud entre dos cadenas de texto. Se basa en el número mínimo de operaciones necesarias para transformar una cadena de caracteres en otra, donde las operaciones permitidas son:

1. Inserción: Agregar un carácter.
2. Eliminación: Eliminar un carácter.
3. Sustitución: Remplazar un carácter.

La distancia se utiliza para medir cuánto se parecen dos cadenas de caracteres, y el valor resultantes es un número entero no negativo. A menor el número, mayor similitud.

EJEMPLO DE DISTANCIA DE LEVENSHTEIN CON MATRIZ

		p	a	b	l	o
	0	1	2	3	4	5
b	1	1	2	2	3	4
r	2	2	2	3	3	4
a	3	3	2	3	4	4
u	4	4	3	3	4	5
l	5	5	4	5	3	4
i	6	6	5	5	4	4
o	7	7	6	6	5	4

Operaciones	
abajo	borrado
alado	inserción
diagonal	sustitución
diagonal	nada

Los número se obtienen: el mínimo de los números de alado + 1(diferente) o + 0 (igual)

En este caso el número mínimo de operaciones entre las cadenas son 4.

PSEUDOCÓDIGO

```
Función distanciaLevenshtein(s1, s2):
    M = longitud(s1)
    N = longitud(s2)

    #  Crear matriz (M+1)x(N+1) distancias parciales.
    Mat = nuevaMatriz(M+1, N+1)

    # Inicializar la primera fila y la primera columna.
    Para i de 0 a M:
        Mat[i][0] = i
    Fin Para

    Para j de 0 a N:
        Mat[0][j] = j
    Fin Para

    # Calcular la distancia.
    Para i de 1 a M:
        Para j de 1 N:
            # Calcular los costos
            Costos = 0
            Si s1[i-1] != s2[j-1]:
                Costos = 1
            Fin Si

            Calcular el mínimo entre las operaciones.
            Mat[i][j] = min(Mat[i-1][j] + 1, Mat[i][j-1] + 1, Mat[i-1]
            [j-1] + Costos)
```

```
        Fin Para
    Fin Para

    # La distancia final es el de la esquina inferior izquierda.
    Distancia = Mat[m][n]

    Devolver Distancia
```

VENTAJAS

- Versatilidad.
- Precisión.
- Adaptabilidad.
- Implementación sencilla.
- Amplio soporte.

DESVENTAJAS

- Costo computacional.
- Memoria.
- Sensibilidad al tamaño de entrada.
- No tiene en cuenta contexto.
- No es adecuado para todas las aplicaciones.

2. DISTANCIA DE DAMERAU-LEVENSHTEIN

lunes, 25 de septiembre de 2023 10:39 p. m.

Es una variante de la distancia de Levenshtein. Las operaciones permitidas son:

- Inserción.
- Eliminación.
- Sustitución.
- Transposición: Intercambiar dos caracteres adyacentes en la cadena.

Es especialmente útil en aplicaciones donde se espera que las transposiciones sean comunes, como en la corrección ortográfica y la detección de errores de escritura.

Las ventajas y desventajas son similares a las de Levenshtein.

PSEUDOCÓDIGO

```
Función distanciaLevenshtein(s1, s2):
    M = longitud(s1)
    N = longitud(s2)

    #  Crear matriz (M+1)x(N+1) distancias parciales.
    Mat = nuevaMatriz(M+1, N+1)

    # Inicializar la primera fila y la primera columna.
    Para i de 0 a M:
        Mat[i][0] = i
    Fin Para

    Para j de 0 a N:
        Mat[0][j] = j
    Fin Para

    # Calcular la distancia.
    Para i de 1 a M:
        Para j de 1 N:
            # Calcular los costos
            Costos = 0
            Si s1[i-1] != s2[j-1]:
                Costos = 1
            Fin Si

            Calcular el mínimo entre las operaciones.
            Mat[i][j] = min(Mat[i-1][j] + 1, Mat[i][j-1] + 1, Mat[i-1]
            [j-1] + Costos)

            # Transposición
            Si i > 1 and j > 1 and s1[i - 1] == s2[j - 2] and s1[i - 2]
            == s2[j -1 ]:
                Mat[i][j] =  min(Mat[i][j], Mat[i - 2][j - 2] + Costos)
            Fin Si

        Fin Para
    Fin Para

    # La distancia final es el de la esquina inferior izquierda.
```

```
Distancia = Mat[m][n]

Devolver Distancia
```

3. AUTOCORRECTORES

jueves, 21 de septiembre de 2023 10:13 p. m.

AUTOCORRECT

Funciona utilizando el algoritmo de Levenshtein para sugerir correcciones de palabras mal escritas en función de las palabras que se encuentran en su diccionario incorporado.

1. Construcción del Diccionario: Lista de palabras correctas que se utilizan como referencia para buscar sugerencias de corrección.
2. Entrada del usuario: Cuando un usuario ingresa un texto, la biblioteca toma la entrada y analiza cada palabra en busca de posibles errores.
3. Búsqueda de coincidencias: Si la palabra se encuentra en el diccionario, se encuentra correcta y se pasa a la siguiente palabra. Si no, se asume que está mal escrita.
4. Cálculo de Distancia de Levenshtein: Se ordena las palabras en función de similitud con la palabra mal escrita, eligiendo aquellas que tiene una distancia de Levenshtein menor.
5. Sugerencia de correcciones.

PYSPELLERCHEKER

1. Configuración y carga del diccionario.
2. Análisis del texto.
3. Verificación de ortografía.
4. Sugerencias de corrección.

TEXTBLOB

1. Instalación y configuración
2. Creación de un objeto Textblob.
3. Corrección ortográfica.

TextBlob utiliza un diccionario incorporado y un modelo estadístico para realizar la corrección ortográfica. También tiene la capacidad de manejar otros idiomas además del inglés, lo que lo hace versátil para muchas tareas de procesamiento de lenguaje natural.

Además de la corrección ortográfica, TextBlob ofrece una amplia gama de otras funcionalidades de procesamiento de lenguaje natural que pueden ser útiles en diversas aplicaciones, como análisis de sentimientos, extracción de información y más.

HUNSPELL

Es una librería de chequeo de escritura usado en Python. Provine de la librería "pshunspell".

Hace uso de un diccionario de Hunspell, generalmente los archivos de estos diccionarios hacen uso de terminaciones ".dic" o ".aff".

4. NIVELES

martes, 26 de septiembre de 2023 11:10 a. m.

- **Nivel Fonético:** ¿Cómo las palabras son pronunciadas?
- **Nivel Morfológico:** Estudia la estructura de las palabras para clasificarlas y delimitarlas.
- **Nivel Sintáctico:** Realiza un análisis de la sintaxis el cual incluye la acción de dividir una oración en cada uno de sus componentes.
- **Nivel Semántico:** Complemento del anterior y busca entender el significado de la oración. (Contexto de la oración).
- **Nivel Discursivo:** Examina el significado de la oración en relación a otra oración en el texto o párrafo del mimo documento.
- **Nivel pragmático:** Análisis de oraciones y como se usan en diferentes situaciones.

5. BIBLIOTECAS DE NLP

martes, 26 de septiembre de 2023 11:26 a. m.

NLTK

Es una biblioteca de Python usada en NLP y lingüística computacional. Proporciona una plataforma robusta y flexible para trabajar con texto y datos de lenguaje natural. Los aspectos importantes son:

- **Amplia gama de recursos:** tiene varios corpora en varios idiomas que útiles para tareas de PLN, análisis de sentimientos, tokenización, etiquetado, etc.
- **Facilidad de uso:** Es fácil de aprender, ya que ofrece una sintaxis clara y coherente.
- **Compatibilidad con recursos externos:** Se puede utilizar recursos lingüísticos externos con NLTK.
- **Librería de Aprendizaje Automático:** Incluye módulos de M.L. y se pude integrar con otras librerías como scikit-learn.

TextBlob

Librería de NLP para Python que simplifica muchas de las tareas de Procesamiento de texto. TextBlob proporciona una API simple y fácil de usar para extracción de entidades, análisis de sentimientos, traducción de texto y más. Los aspectos importantes son:

- **Facilidad de uso.**
- **Funcionalidad integrada:** Proporciona funcionalidades integradas para tareas comunes de NLP. Esto facilita el desarrollo rápido de aplicaciones de procesamiento de texto.
- **Facilidad de extensión:** Permite la creación de analizadores y clasificadores personalizados.
- **Integración de modelos externos:** Usa modelos de NLTK y Pattern.

Stanford Core NLP

Suite de herramientas de NLP desarrollado por el Grupo de Procesamiento del Lenguaje Natural de la Universidad de Standford. Algunos aspectos importantes:

- **Análisis de sentencias complejas.**
- **Soporte para múltiples idiomas.**
- **Entidades nombradas**: Es capaz de indentificar y clasificar entidades nombradas en el texto.
- **Procesamiento en Batch o en Tiempo Real:** Permite analiza texto en lote o en tiempo real, lo que lo hace versátil.
- **Integración con Java:** Está diseñado para Java, pero cuenta con interfaces y wrappers para otros lenguajes de programación, como Python, por medio de librerías de terceros.
- **Personalización y extensiones:** Además de tareas de NLP, es posible personalizar y extender sus funcionalidades.

Spacy

Librería de NLP para Python, se destaca por su eficiencia, velocidad y facilidad de uso. Aspectis clave sobre spaCy:

- **Eficiencia y velocidad:** Es rápido y escalable, lo que lo hace adecuado para el procesamiento de grandes datos.
- **Modelos Pre-entrenados:** No hay necesidad de entrenar modelos desde cero.
- **Soporte Multilingüe.**
- **Facilidad de uso.**

- **Tokenización eficiente.**
- **Extensibilidad**: Es extensible y permite personalizar y ampliar sus funcionalidades según necesidades del proyecto.
- **Uso en diversas aplicaciones.**

Textacy

Librería de NLP en Python que proporciona herramientas adicionales y funcionalidades para el análisis de texto avanzado y la minería de texto. Se centra en simplificar las tareas. Se construyes sobre spaCy. Algunos aspectos importantes son:

- **NLP para la Web Social.**
- **Textos en múltiples idiomas.**

Gensim

Librería de NLP para Python que se especializa en modelización de temas y la representación de texto en forma de vectores denso. Se centra en la recuperación de información, análisis de texto, construcción de sistemas de recomendación:

- **Representaciones vectoriales.**
- **Implementación eficiente.**
- **Algoritmos de vectorización.**
- **Uso eficiente de memoria.**

pyLDAvis

Librería de Python para la visualización de modelos de asignación de temas generados por Latent Dirichket Allocation. Aspectos importantes:

- **Visualización interactiva.**
- **Integración con Gensim y scikit-learn.**
- **Personalización:**

6. ANÁLISIS MORFOLÓGICO

miércoles, 27 de septiembre de 2023 10:46 a. m.

Consiste en determinar la categoría gramatical de cada palabra que forma una oración. Para analizar morfológicamente una oración, se toma cada palabra y se determina a qué categoría pertenece.

CLASES DE PALABRAS ENE EL ANÁLISIS MORFOLÓGICO

Determinantes

Son palabras que acompañan al sustantivo para aportar información sobre el mismo. Las clases son

Sustantivos

Son palabras que nombran a cualquier cosa.

Verbos

Expresan acción, existencia, estado o condición del sujeto.

Pronombre

Sustituyen al nombre.

7. TreebankWordTokenizer

viernes, 29 de septiembre de 2023 07:43 a. m.

El tokinizador de Treebank usa expresiones regulares para torkenizar textos como en Penn Treebank. Este es un método que es invocado por *word_tokenize()*. Asume que el texto ya ha sido segmentado en oraciones, usando *sent_tokenize()*.

MÉTODOS

- ~~Span_tokenize(s): Identifica tokens usando "integer offsets" (start_i, end_i), donde s[start_i:end_i] es el token correspondiente.~~
- ~~Span_tokenize_sents(strings): Aplica self.span_tokenize() a cada elemento de los strings.~~
- ~~Tokenize(text).~~
- ~~Tokenize_sents(strings): Aplica self.tokenize() a cada elemento de los strings.~~

TREEBANK

Es un corpus lingüístico en el que cada frase ha sido parseada, es decir anotada con su estructura sintáctica. La estructura sintáctica se ha representado generalmente como una estructura arbórea que recibe la denominación TreeBank.

8. LEMATIZACIÓN

viernes, 29 de septiembre de 2023 08:11 a. m.

La lematización relaciona una palabra flexionada o derivada con su forma canónica o lema. Y un lema no es otra cosa que la forma que tienen las palabras cuando las buscas en el diccionario.

Como el proceso de lematización toma en consideración probable clase de palabra (llamada POS) podemos usar dicha información para filtrar la lista de lemas.

La lematización es un proceso clave en muchas tareas práctica de NLP pero tiene dos costos:

- Consume recursos (sobre todo de tiempo).
- Suele ser probabilística (En algunos casos obtendremos resultados inesperados).

9. STEMMING

viernes, 29 de septiembre de 2023 08:16 a. m.

Convertir palabras en raíces. Las raíces son la parte invariable de palabreas relacionadas sobre todo por su forma. De cierta forma se parece a la lematización, pero los resultados (las raíces) no tienen por qué ser un idioma.

El stemming es mucho más rápido desde el punto de vista del procesamiento que la lematización. Tiene como ventaja que reconoce relaciones entre palabras de distinta clase. El stemming puede reducir el número de elementos que forman el texto.

La desventaja es que los algoritmos son más simples que los de lematización. Pueden "recortar" demasiado la raíz y encontrar relaciones entre palabras que realmente no existen (overstemming). Las raíces pueden ser demasiadas extensas o específicas, y que tengamos más bien un déficit de raíces (understemming), en cuyo caso palabras que deberían convertirse en una mismas raíz no lo hacen.

10. PORTER STEMMING

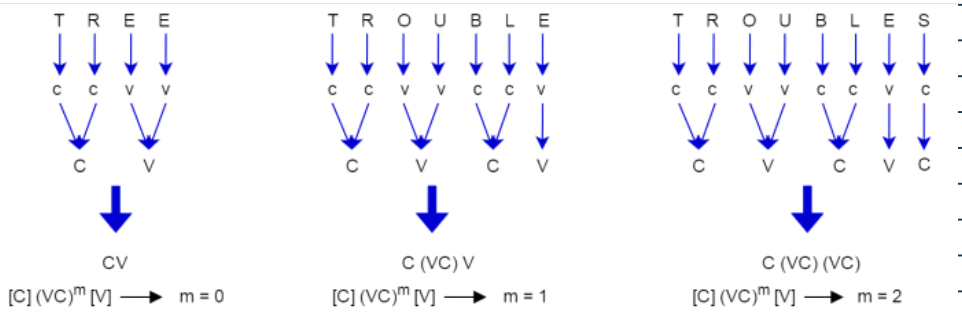
domingo, 1 de octubre de 2023 12:24 p. m.

Desarrollado en 1980 por Martin F. Porter en 1980, es un algoritmo de derivación léxica.

El proceso implica una serie de pasos que se aplican a una palabra para reducirla a su forma raíz.
Está basado en el idioma inglés. El resumen simplificado es el siguiente:

1. **Eliminación de Sufijos Comunes:**
   Elimina sufijos comunes como "s", "es", "ed", "ing", etc.
2. **Eliminación de Sufijos de Plural:**
   Elimina sufijos que indican pluralidad, como "s" y "es".
3. **Eliminación de Sufijos de Participio Pasado:**
   Elimina sufijos que indican participio pasado, como "ed" y "ing".
4. **Eliminación de Sufijos de Gerundio:**
   Elimina sufijos que indican gerundio, como "ing".
5. **Eliminación de Sufijos Derivados:**
   Elimina sufijos derivados, como "al", "ance", "ence", "er", "ic", etc.
6. **Eliminación de Prefijos Comunes:**
   Elimina ciertos prefijos, como "un", "dis", "re", etc.

Está diseñado para el idioma inglés.



Para saber más:

https://vijinimallawaarachchi.com/2017/05/09/porter-stemming-algorithm/

Gramáticas Independientes de Contexto.

viernes, 6 de octubre de 2023 11:04 a. m.