Universidad de Granada

decsai.ugr.es

Modelos de ciencia de datos no numéricos:

Aplicaciones en redes sociales, web y gestión de procesos

DECSAI

Departamento de Ciencias de la Computación e Inteligencia Artificial

Universidad de Granada

decsai.ugr.es

Bloque II: Minería de Texto y de la Web

DECSAI

Departamento de Ciencias de la Computación e Inteligencia Artificial

Universidad de Granada

decsai.ugr.es

Minería de Textos

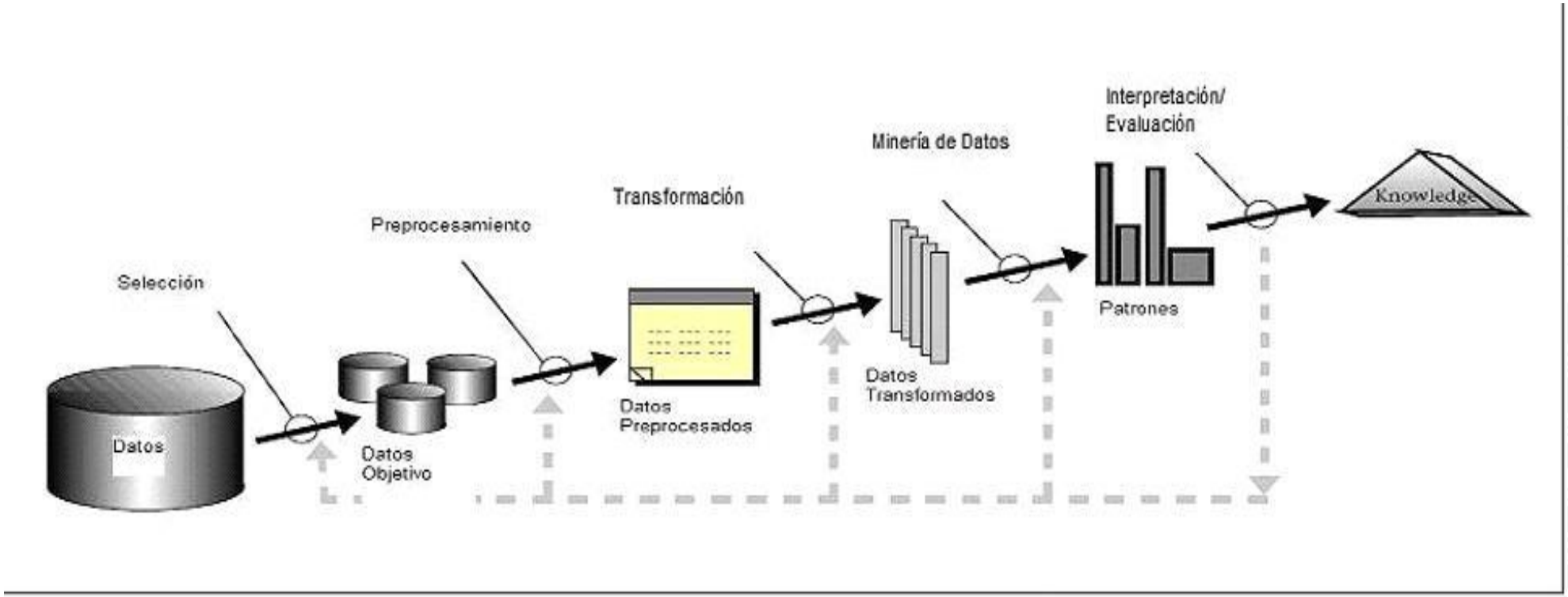
DECSAI

Departamento de Ciencias de la Computación e Inteligencia Artificial

Conceptos previos: KDD

KDD (Knowledge Discovery in Databases)

Proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia comprensibles a partir de los datos.



Conceptos previos: Minería de Datos

Minería de Datos (o Data Mining (DM))

Proceso de descubrimiento eficiente de patrones, desconocidos a priori, en grandes bases de datos.

Relación existente entre KDD y DM:

- DM como fase de KDD
- DM como sinónimo de KDD

De los datos al texto

Motivación

- La gran mayoría de los datos susceptibles de ser procesados en redes sociales, informes de empresa, foros, páginas web, … son textuales.
- Los datos textuales carecen de estructura y homogeneidad. Los datos son simbólicos y los atributos son desconocidos a priori.
- Pueden estar en diferentes lenguajes y fuentes y además estar incompletos, ser irrelevantes o tener ruido.
- Dependen del contexto.

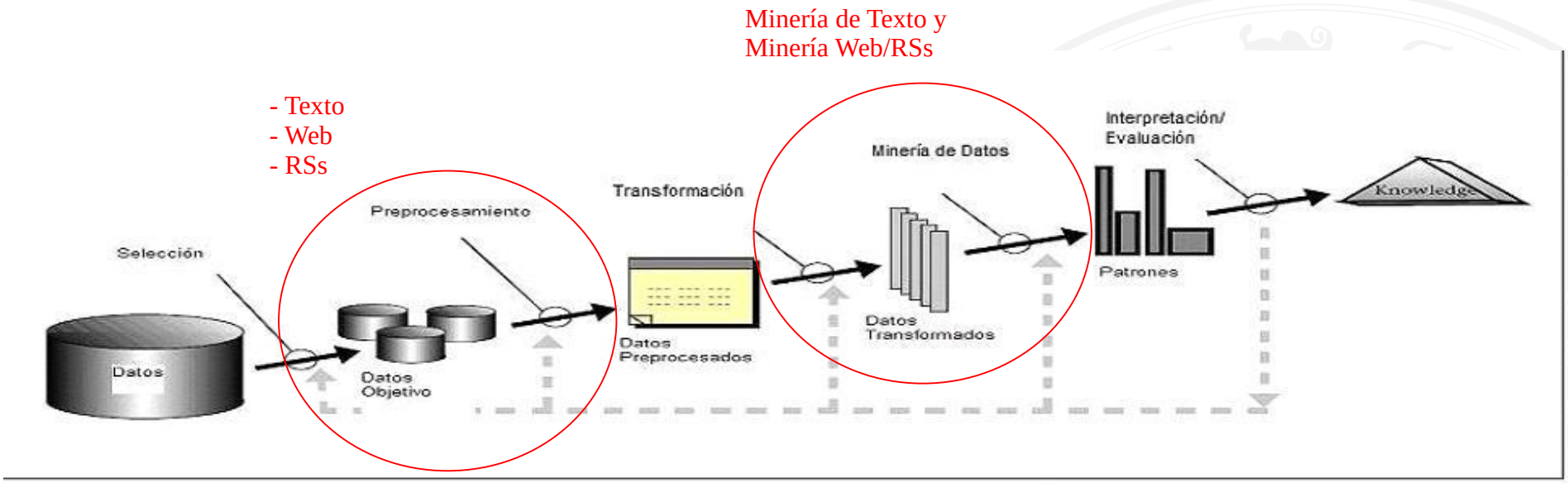
¿TM = KDT?

¿TM = DM con Texto?

KDT

KDT (Knowledge Discovery in Text)

El proceso de descubrir información útil que no está explícitamente en ninguno de los documentos analizados y que aparece cuando los documentos son analizados y relacionados.



KDT versus KDD

KDD

- ❑ Entender el dominio de la aplicación
- ❑ Seleccionar el conjunto de datos objetivo
- ❑ Limpiar, preprocesar y transformar los datos
- ❑ Desarrollo del modelo y construcción de la hipótesis
- ❑ Selección y ejecución de algoritmos de minería de datos
- ❑ Interpretación de resultados
- ❑Visualización

KDT

- ❑ El usuario define conceptos interesantes
- ❑ Los textos se obtienen via RI o manualmente
- ❑ Los textos y conceptos se representan en una Forma Intermedia
- ❑ Identificación de conceptos en la colección de textos
- ❑ Algoritmos de Minería de textos
- ❑ Interpretación de resultados mediante un humano
- ❑ Visualización

KDT versus RI

- La Recuperación de Información (RI) ayuda a KDT para la recuperación de documentos y su preprocesamiento.

- Tienen diferentes objetivos:

  RI: Optimizar la consulta, la búsqueda y la recuperación de documentos.

  KDT: Extraer conocimiento no explícito en los datos.

KDT versus EI

- La Extracción de Información (EI) localiza unidades de texto relevantes para el usuario.
- La EI transforma un documento escrito en Lenguaje Natural en una representación estructurada o basada en slots.
- La EI puede utilizarse en la etapa de pre-procesamiento de KDT.

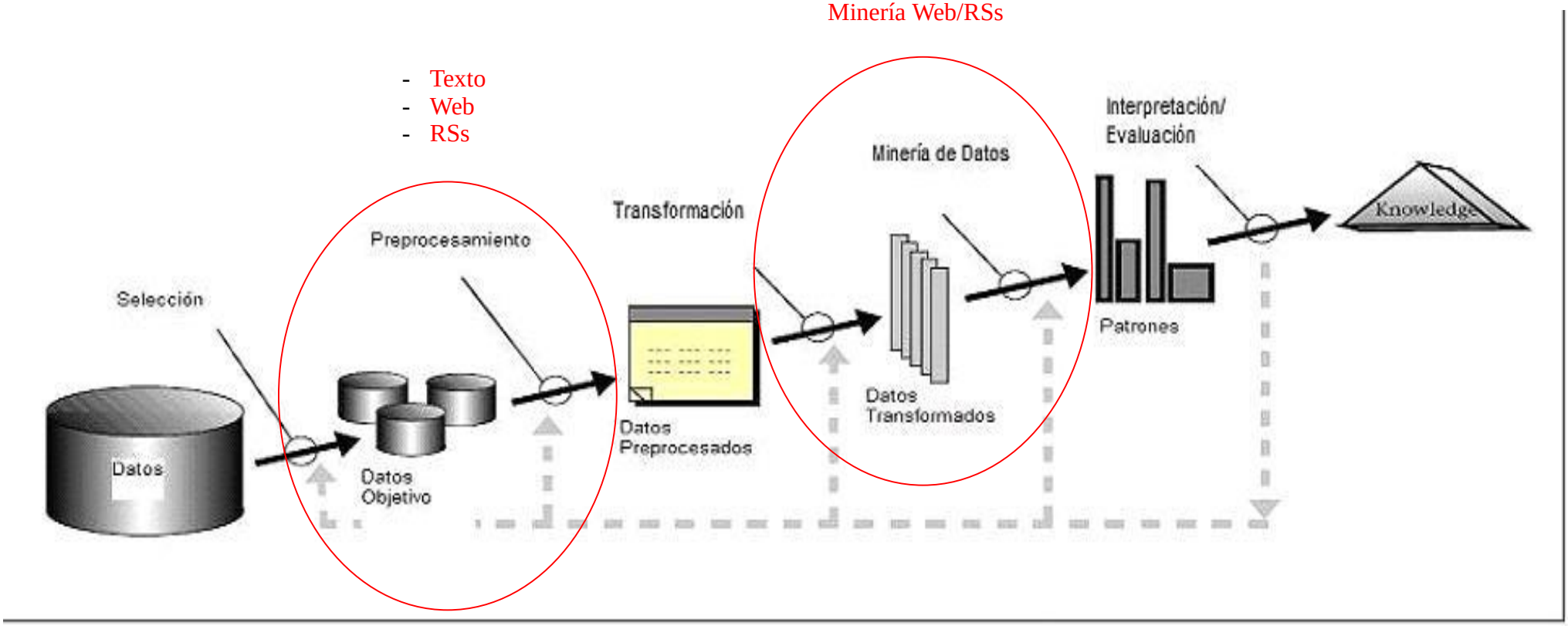
KDT versus PLN

- El Procesamiento de Lenguage Natural (PLN) es una rama de la inteligencia artificial que se enfoca en la interacción entre las computadoras y el lenguaje humano.
- Su objetivo es analizar el lenguaje natural para otros procesos incluyendo el contribuir a procesos que ayuden a la máquina a generear lenguaje natural (IA Generativa).

KDT

Minería de Texto y
Minería Web/RSs

- Texto
- Web
- RSs



Universidad de Granada

decsai.ugr.es

Preprocesamiento

DECSAI

Departamento de Ciencias de la Computación e Inteligencia Artificial

Formas Intermedias

- Una **Forma Intermedia** es un modelo de representación del conocimiento capaz de expresar el contenido implícito del texto de una forma computable mediante un algoritmo o un programa.

❑ Términos
❑ Taxonomía de términos
❑ **Conceptos**
❑ **Bolsas de palabras y TF-IDF**
❑ **Grafos de conocimiento** (Knowledge graphs)
❑ **Word Embeddings**
❑ Frases de texto multi-términos
❑ Términos de consultas (Consultas enriquecidas)
❑ Documentos prototipo

https://conceptnet.io/

Ejemplo de formas intermedias

“Los **sistemas difusos** han revolucionado diversos campos de la ingeniería y la inteligencia artificial gracias a su capacidad para manejar **comportamientos difusos** y la imprecisión inherente en muchos procesos del mundo real. La **teoría de conjuntos difusos** permite modelar situaciones que no pueden ser descritas con precisión mediante lógica binaria, utilizando **conjuntos difusos** y **números difusos** para representar incertidumbres. Esta teoría se aplica en **sistemas neuro-difusos**, los cuales combinan redes neuronales con **modelado difuso** para resolver problemas complejos que involucran datos imprecisos. En el contexto de **implementación de sistemas difusos**, se desarrollan **circuitos integrados para el control difuso y neuro-difuso**, lo que permite aplicar estas tecnologías en dispositivos electrónicos y sistemas embebidos.

El **controlador difuso** es un ejemplo claro de cómo los **métodos de razonamiento difusos** se utilizan para controlar procesos en los que las reglas precisas son difíciles de definir, pero se puede manejar la imprecisión a través de inferencias basadas en grados de pertenencia difusa. **Objetos difusos**, tales como **puntos difusos** en espacios de alta dimensionalidad, se emplean para representar entidades cuyas fronteras no son nítidas, y esto resulta esencial en áreas como el procesamiento de señales y la toma de decisiones automatizada. Además, los **sistemas difusos evolutivos** aprovechan algoritmos evolutivos para optimizar las funciones de control difuso, mejorando la adaptabilidad y robustez de los sistemas. Así, el empleo de **sistemas difusos** y **sistemas neuro-difusos** sigue creciendo en importancia, ofreciendo soluciones efectivas para el control y modelado de sistemas complejos y dinámicos.”

Ejemplo de formas Intermedias

- Palabras Clave:

  DIFUSOS, TEORIA DE CONJUNTOS DIFUSOS, SISTEMAS NEURO-DIFUSOS, MODELADO DIFUSO, SISTEMAS DIFUSOS, IMPLEMENTACION SISTEMAS DIFUSOS, CIRCUITOS INTEGRADOS PARA EL CONTROL DIFUSO Y NEURO-DIFUSO, COMPORTAMIENTOS DIFUSOS, CONJUNTOS DIFUSOS, NÚMEROS DIFUSOS, PUNTO DIFUSO, METODOS DE RAZONAMIENTO DIFUSOS, CONTROLADOR DIFUSO, OBJETOS DIFUSOS, SISTEMAS DIFUSOS EVOLUTIVOS.

- Términos:

  controlador, circuitos, números, difusos, sistemas, …

- Conceptos:

  fuzzy, fuzzy-logic, controladores difusos, sistema difusos, ...

Ejemplo de formas Intermedias

- Bolsa de Palabras:

Frecuencia: La bolsa de palabras no tiene en cuenta la frecuencia de las palabras en el texto (si se repiten o no).

No se considera el orden: No importa el orden en el que las palabras aparecen, solo se toma en cuenta qué palabras están presentes.

Tokenización: El texto se ha tokeniza en palabras individuales, eliminando la puntuación.

Posibles pasos adicionales:

Filtrado de palabras vacías: Se podrían eliminar palabras comunes como "la", "de", "y", etc., que no tienen un valor semántico relevante.

Pesado por frecuencia: Se podrían contar cuántas veces aparece cada palabra para construir una matriz de características con los pesos correspondientes.

Ejemplo de formas Intermedias

- Bolsa de Palabras:

sistemas	difusos	han	revolucionado	diversos	campos	ingeniería
inteligencia	artificial	gracias	capacidad	manejar	comportamientos	imprecisión
inherente	muchos	procesos	mundo	real	teoría	conjuntos
permite	modelar	situaciones	pueden	ser	descritas	precisión
mediante	lógica	binaria	utilizando	números	representar	incertidumbres
aplica	neuro-difusos	combinan	redes	neuronales	resolver	problemas
complejos	involucran	datos	imprecisos	contexto	implementación	desarrollan
circuitos	integrados	control	tecnologías	dispositivos	electrónicos	embebidos
controlador	ejemplo	claro	métodos	razonamiento	utilizar	controlar
procesos	reglas	difíciles	definir	manejar	inferencias	grados
pertenencia	objetos	puntos	alta	dimensionalidad	emplean	representar
entidades	fronteras	nítidas	esencial	áreas	procesamiento	señales
toma	decisiones	automatizada	sistemas	evolutivos	aprovechan	algoritmos
optimizar	funciones	adaptabilidad	robustez	sigue	creciendo	importancia
ofreciendo	soluciones	efectivas	dinámicos			

Formas Intermedias: TF-IDF

Pasos para calcular el **TF-IDF (Term Frequency - Inverse Document Frequency)** de un párrafo:

1) Calcular el TF (Frecuencia de Término):

La frecuencia de un término en un documento se calcula como el número de veces que aparece el término en el documento dividido por el número total de términos en el documento.

2) Calcular el IDF (Frecuencia Inversa de Documento):

La frecuencia inversa de documento mide la importancia de un término en relación con todo el corpus. Se calcula usando la fórmula:

$$\text{IDF}(t) = \log\left(\frac{N}{df(t)}\right)$$

donde:

N es el número total de documentos.

df(t) es el número de documentos en los que aparece el término ***t***.

3) Calcular el TF-IDF:

$$TF\text{-}IDF(t) = TF(t) \times IDF(t)$$

Para calcular el TF-IDF del párrafo, se necesita:

El texto completo (que ya lo proporcionaste).

El conjunto de documentos para calcular el IDF.

Ejemplo de Grafos de conocimiento (Knowledge Graphs)

Un Knowledge Graph (KG) es un conjunto de nodos (entidades) y aristas (relaciones) entre esos nodos que representan las relaciones semánticas entre dichos elementos.

- Identificación de entidades: Extraer las entidades clave del párrafo (por ejemplo, términos como "sistemas", "difusos", "control", "teoría").
- Identificación de relaciones: Detectar las relaciones entre las entidades. Por ejemplo, "control" se aplica a "sistemas", o "teoría" se relaciona con "conjuntos difusos".
- Construcción del grafo: Representar las entidades como nodos y las relaciones como aristas (conexiones entre nodos).

Ejemplo de Grafos de conocimiento (Knowledge Graphs)

Paso 1: Identificación de entidades

Al leer el párrafo, podemos identificar las siguientes entidades (nodos):

Sistemas difusos
Comportamientos difusos
Teoría de conjuntos difusos
Modelado difuso
Control difuso
Sistemas neuro-difusos
Objetos difusos
Números difusos
Circuitos integrados
Métodos de razonamiento difusos
Sistemas difusos evolutivos

Ejemplo de Grafos de conocimiento (Knowledge Graphs)

Paso 2: Identificación de relaciones

"Sistemas difusos" -> "utilizan" -> "teoría de conjuntos difusos"

"Sistemas difusos" -> "incluyen" -> "modelado difuso"

"Sistemas neuro-difusos" -> "usados para" -> "control difuso"

"Control difuso" -> "aplica a" -> "sistemas difusos"

"Sistemas difusos evolutivos" -> "optimizan" -> "funciones de control difuso"

"Métodos de razonamiento difusos" -> "se usan en" -> "control difuso"

"Circuitos integrados" -> "para" -> "control difuso"

Ejemplo de Grafos de conocimiento (Knowledge Graphs)

Paso 3: Construcción del grafo

[Sistemas Difusos] -- (utilizan) --> [Teoría de Conjuntos Difusos]

[Sistemas Difusos] -- (incluyen) --> [Modelado Difuso]

[Sistemas Neuro-Difusos] -- (usados para) --> [Control Difuso]

[Control Difuso] -- (aplica a) --> [Sistemas Difusos]

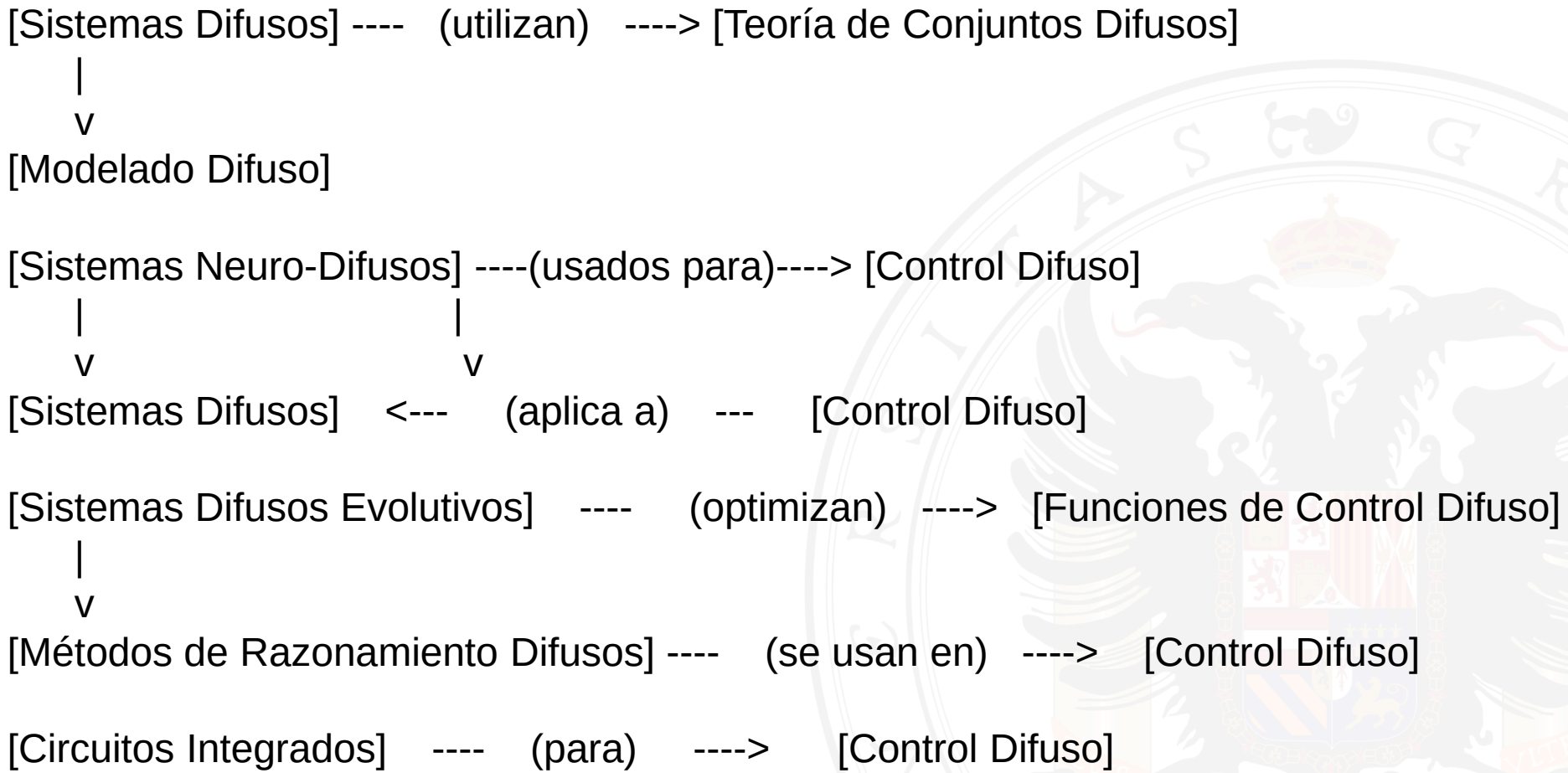
[Sistemas Difusos Evolutivos] -- (optimizan) --> [Funciones de Control Difuso]

[Métodos de Razonamiento Difusos] -- (se usan en) --> [Control Difuso]

[Circuitos Integrados] -- (para) --> [Control Difuso]

Ejemplo de Grafos de conocimiento (Knowledge Graphs)

Paso 4: Visualización del grafo



Formas intermedias: Word Embeddings

Modelos del lenguaje que asignan una representación de palabras en el espacio semántico:

- Para cada palabra, el embedding capta el “significado” de la palabra.
- Cada palabra tiene un único word embedding (o “vector de números reales”), que es una lista de números que representan esa palabra.
- Los word embeddings son multidimensionales; normalmente, para un buen modelo, los embeddings tienen una longitud de entre 50 y 500.
- Palabras similares tienen valores similares de embedding.
- Pasamos de un espacio con una dimensión por palabra a un espacio vectorial continuo con menos dimensiones.

Word Embeddings: Word2Vec

- Word2Vec genera representaciones vectoriales para palabras completas. Utiliza un modelo basado en redes neuronales para aprender un vector de características que representa una palabra.
- Sin embargo, este tipo de modelos (estáticos), mapean cada palabra del lenguaje en un **único vector** del espacio transformado.

Así, por ejemplo, la palabra “banco” tendrá el mismo vector asociado:

“Me voy a sentar en sentar en este banco”.

“Tengo que ir a pagar el recibo al banco”.

Modelos:

Skip-gram: Predice palabras contextuales a partir de una palabra dada.

CBOW (Continuous Bag of Words): Predice una palabra basada en su contexto (las palabras cercanas).

- **Limitación:** Si la palabra nunca se ha visto en el conjunto de datos de entrenamiento, no tiene representación.

Mikolov, Tomas (2013). “Efficient Estimation of Word Representations in Vector Space”

Ejemplo Word2Vec

Para transformar el párrafo anterior, habría que:

- **Tokenizar**: Convertir el párrafo en una lista de palabras (tokens).
- **Entrenar un modelo Word2Vec**: Utilizar un modelo Word2Vec sobre el texto tokenizado. Esto requiere un corpus o conjunto de documentos. Si no tienes un corpus grande, puedes entrenar el modelo con el texto proporcionado, pero generalmente se recomienda entrenar Word2Vec en un corpus más grande.
- **Obtener los vectores de palabras**: Una vez entrenado el modelo, se pueden obtener los vectores de las palabras. Cada vector tendrá 100 dimensiones que representan a cada palabra en el espacio semántico, y cada uno de esos vectores captura información contextual sobre cómo se usa la palabra dentro del párrafo.
- **Por ejemplo,** un vector para la palabra "sistemas" podría verse así (solo un fragmento):

Vector de "***sistemas***":

- [0.00570329 -0.00496022 -0.02229911 ...] # Vector de 100 dimensiones

Word Embeddings: FastText

- **FastText** mejora Word2Vec al descomponer las palabras en subunidades llamadas **n-gramas**. Estas son secuencias de caracteres dentro de las palabras, lo que permite que FastText genere representaciones no solo para palabras completas, sino también para los fragmentos dentro de esas palabras.
- **FastText** crea un vector para cada n-grama (subcadena de la palabra) y luego, **para cada palabra, la representación se calcula como la suma de los vectores de sus n-gramas.** Esto permite que FastText tenga representaciones más robustas para palabras que nunca han aparecido en el conjunto de datos de entrenamiento, ya que puede generalizar a partir de los n-gramas que componen una palabra.
- **Ventajas:**

• Mejor rendimiento en el manejo de palabras raras o desconocidas (*out-of-vocabulary*).
• Mejora en lenguajes morfológicamente ricos (como el alemán o el finlandés) porque entiende la estructura interna de las palabras.
• FastText es particularmente útil para lenguajes que combinan muchas raíces y sufijos, ya que los n-gramas pueden capturar estas variaciones de manera más eficiente.

Word Embeddings: ELMo

ELMo (Embeddings from language Models)

(Allen Institute for AI, 2017)

- Asigna un vector a cada palabra o token en función de toda la secuencia que contiene a la palabra.

- Usa una arquitectura para modelado de lenguaje con dos capas de **LSTM bidireccionales.**

- Las redes LSTM (del inglés Long Short-Term Memory) **son un tipo de red neuronal recurrente que se utiliza para procesar y modelar secuencias de datos**.

- A diferencia de FastText, ELMo produce representaciones de palabras que dependen del contexto completo de la oración. Esto significa que la misma palabra tendrá diferentes vectores si aparece en diferentes contextos.

- ELMo es capaz de capturar relaciones semánticas complejas y dependencias a largo plazo dentro de un contexto textual, lo que lo hace más flexible y potente para tareas como análisis de sentimientos, traducción, etc.

Word Embeddings

ELMo (Embeddings from language Models):

- En la LSTM, la transformación forward captura el contexto anterior de la palabra y la backward el contexto posterior y la concatenación de ambas forma el **vector intermedio.**

- La representación final de la palabra es la **suma ponderada de la entrada y los dos vectores intermedios,** de forma que combina información de todas las capas y el contexto de la palabra.

- Estas representaciones se pueden **añadir a cualquier modelo al entrenarse de forma no supervisada** y mejoran el rendimiento de tareas como preguntas-respuestas, análisis de sentimiento, etc.

- **Desventaja:** las representaciones en cada dirección se entrenan de **manera independiente** y las representaciones de cada palabra se integran en el modelo.

FastText versus ELMo

FastText:

Preentrenado en grandes corpus de texto: FastText se entrena en grandes corpus de texto de manera **no supervisada**, aprendiendo las representaciones de palabras basadas en n-gramas. Una vez entrenado, puede ser utilizado directamente para representar las palabras en cualquier nuevo corpus.

Entrenamiento y eficiencia: FastText es relativamente más ligero y rápido en términos de entrenamiento y uso, ya que es un modelo de *word embeddings* que no depende de un modelo complejo y pesado.

ELMo:

Preentrenado en grandes corpus de texto y luego afinado: ELMo también es entrenado en grandes corpus (como Wikipedia, libros, etc.) usando un *Language Model* basado en LSTM bidireccional, y luego puede ser afinado para tareas específicas como clasificación de texto, traducción, etc.

Modelo pesado: ELMo es más complejo y pesado en comparación con FastText, ya que requiere entrenamiento de redes neuronales profundas (bidireccionales) y una mayor capacidad computacional.

Word Embeddings: BERT

BERT (Bidirectional Encoder Representations from Transformers)

- Es un modelo que crea representaciones usando el contexto anterior y posterior de cada palabra y que, una vez entrenado previamente, se puede ajustar (*fine-tuning*) para una tarea específica posterior.
- Tiene un vocabulario pre-entrenado de unas 30.000 palabras con un vector asociado de dimension 768.
- Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018).
- BERT: Pre-training of deep bidirectional transformers for language understanding.

Word Embeddings: BERT

BERT (Bidirectional Encoder Representations from Transformers)

- Usa un modelo de múltiples capas de **transformadores (módulos de self-attention)**. Usando estos módulos, en lugar de LSTMs, se consigue aprender los pesos de atención de cada palabra del contexto anterior y posterior. Es decir, la representación de cada palabra se basa en todo el contexto de la oración en lugar de solo la parte anterior y posterior.

- El modelo se **pre-entrena** en dos tareas no-supervisadas. La primera, **ocultando aleatoriamente** un porcentaje de los tokens (palabras) de entrada y aprendiendo a predecirlos. La segunda, eligiendo dos frases, que el modelo debe **predecir si son consecutivas o no.**

- Una vez que el modelo se ha pre-entrenado, se puede ajustar en una tarea posterior y **afinar (fine-tuning) los parámetros** para dicha tarea.

ELMo versus BERT

ELMo:

Aunque ELMo ofrece mejoras significativas en las representaciones de palabras en comparación con los modelos tradicionales, como Word2Vec, su rendimiento en tareas complejas de NLP generalmente es inferior al de BERT aunque es más rápido. ELMo es más adecuado para tareas que no requieren un alto nivel de comprensión contextual.

BERT:

BERT ha establecido nuevos **récords de rendimiento** en una gran cantidad de tareas de NLP, como **clasificación de texto**, **preguntas y respuestas**, **traducción automática** y **resolución de ambigüedad**. Su capacidad para generar representaciones altamente contextuales de palabras y frases lo hace más adecuado para tareas complejas.

Word Embeddings: RoBERTa

RoBERTa (Robustly Optimized BERT pre-training Approach):

- Basado en BERT. Tiene la misma arquitectura pero usa un tokenizador a nivel de byte y no de carácter (BPE, el mismo que GPT-2) y un sistema de pre-entrenamiento diferente.
- Bert tiene un enmascaramiento de tokens fijo de 10 y se realiza solo una vez al principio, por lo que en tiempo de entrenamiento el modelo solo vé 10 variaciones de cada frase.
- Roberta sin embargo realiza el enmascaramiento durante el entrenamiento sin limitaciones en las variaciones y esto mejora el comportamiento.

-- RoBERTa: A Robustly Optimized BERT Pretraining Approach by Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov (2018)

Modelos Transformers versus Large Languages (LLMs)

Modelos Transformers es un término que describe la arquitectura subyacente, es decir, el tipo de red neuronal que utilizan estos modelos.

LLMs se refiere más al tamaño y capacidad del modelo. Aunque muchos de los modelos basados en transformers, como GPT-3, son LLMs debido a su tamaño masivo, no todos los transformers son LLMs. Por ejemplo, BERT o RoBERTa pueden ser considerados transformers, pero no siempre se les clasifica como LLMs debido a su tamaño comparativamente más pequeño (en términos de parámetros y datos de entrenamiento).

Large Language Models (LLMs) recientes

ALBERT (A Lite BERT) 2019 (Google) - Una versión más ligera y eficiente de BERT.

T5 (Text-to-Text Transfer Transformer) 2020 (Google) - Un enfoque unificado para todas las tareas de NLP.

ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) 2020 (Google) - Más eficiente en términos de entrenamiento, con un enfoque diferente en el preentrenamiento.

Longformer 2020 (Allen Institute for AI) - Optimizado para manejar secuencias de texto muy largas.

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) 2021 (Microsoft) - Mejora la atención y el decodificado de la información para capturar relaciones semánticas más detalladas.

GPT-4 (Generative Pre-trained Transformer 4) 2023 (OpenAI) - Potente modelo generativo con capacidades excepcionales de razonamiento y aprendizaje con pocos disparos.

Fases del pre-procesamiento

Recolección de documentos

Estandarización de documentos

Toquenización

Lematización

Lema	Categoría
sal	nombre
salir	verbo

Ejemplo:
http://www.molinolabs.com/lematizador.html

¿stemming?

Diccionario de la colección

Stemming

- Análisis morfológico (inflectional stemming): Eliminar diferentes formas de la misma palabra.
- Por ejemplo:

  - Plural y Singular: libro/libros

  - Tiempos verbales: ayudo, ayudaré, ayudamos, …
- En otros idiomas no es tan sencillo debido a:

  - Irregularidades en los verbos: seek/sought (buscar/buscado en inglés), ageben/agegeben (declarar/declarado en alemán)

Stemming

Palabra	Raíz
abaco	abac
abajo	abaj
abandera	abander
abandona	abandon
abandonada	abandon
abandonadas	abandon
abandonado	abandon
abandonados	abandon
abandonamos	abandon
abandonan	abandon
abandonar	abandon
abandonarlo	abandon
abandonaron	abandon
abandono	abandon

Ejemplo Algoritmo Porter en español online: https://tartarus.org/martin/PorterStemmer/

Ejemplo Algoritmo Porter en inglés online: https://9ol.es/porter_js_demo.html

Diccionario

Documentos /Términos	Término 1	Término 2	Término 3	…	Término M
Documento 1	p_{11}	p_{12}	p_{13}	…	p_{1m}
Documento 2	p_{21}	p_{22}	p_{23}	…	p_{2m}
Documento 3	p_{31}	p_{32}	p_{33}	…	p_{3m}
…					
Documento N	p_{n1}	p_{n2}	p_{n3}	…	p_{nm}

p_{ij} es el peso del término en el documento y puede estar basado en:

- Esquemas binarios de presencia/ausencia
- Frecuencias normalizadas
- tf-idf (frecuencia del término/frecuencia inversa del documento)

$$tfidf(t_i) = tf(t_i) * idf(t_i)$$

$$idf(t_i) = \log(\frac{N}{df(t_i)})$$

Diccionario

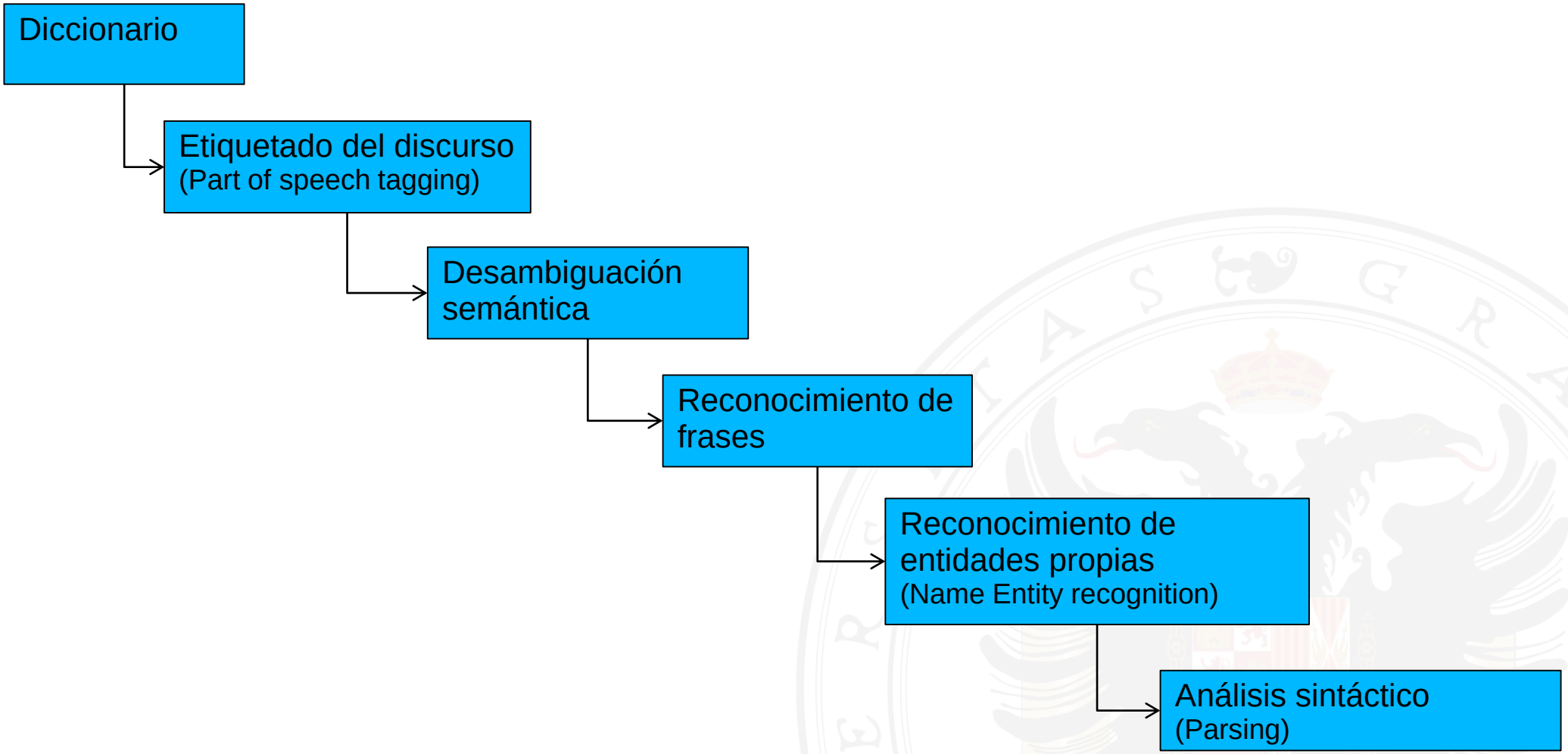
Técnicas de reducción de diccionario:

- Diccionario local
- Palabras de parada (stopwords)
- Palabras frecuentes
- Selección de características (hacer diccionarios locales a una categoría)
- Reducción de tokenes: lematización, sinónimos.

A tener en cuenta: los diccionarios reducidos basados en este tipo de técnicas mejoran las técnicas y procesos que los utilizan.

También se pueden considerar conjuntos de términos multi-palabra (n-gramas).

Fases de pre-preprocesamiento (II)



Part of Speech Tagging (POS)

Ejemplo:

Esto es un ejemplo para no aburriros

Esto: Artículo
Es: Verbo
Un: Artículo
Ejemplo: Sustantivo
Para: Preposición
No: Partícula negativa
Aburriros: Verbo

Desambiguación semántica

- Basada en conocimiento:
  - Diccionarios
  - Tesauros
  - Ontologías
  - Word Embeddings
- Basados en corpus:
  - Etiquetados
  - Gran cantidad de ejemplos

Name Entity Recognition

Encontrar y clasificar nombres en un texto:

La Guardia Civil ha detenido en Diezma a Juan López y María Clavero de Bollullos, con antecedentes policiales en 2024, como presuntos autores de cuatro robos en vehículos llevados a cabo en estaciones de servicio

Persona: Juan López, María Clavero

Fecha: 2024

Localización: Diezma, Bollullos

Organización: Guardia Civil

Técnicas de pre-procesamiento

- Análisis de Texto completo
- Categorización
- Técnicas de PLN (Procesamiento de Lenguaje Natural)
  - Etiquetado de parte del discurso
  - Tokenización
  - Lematización
- Técnicas de EI
  - Categorización
  - Adquisición de patrones léxicos-sintácticos
  - Extracción automática de términos
  - Localización de trozos de texto
- Técnicas de RI
  - Indexación

Formas Intermedias

- La técnica para conseguir la forma intermedia determina el tipo de información a conseguir en el proceso de descubrimiento

- Por ejemplo:

Pre-procesamiento	Representación	Descubrimiento
Categorización	Vector de términos representativos	Relaciones entre los términos
Análisis de Textos completos	Secuencias de palabras	Patrones del lenguaje
Extracción de información	Tabla de base de datos	Relaciones entre entidades

Preprocesamiento - Forma Intermedia

PREPROCESAMIENTO	FORMA INTERMEDIA	DESCUBRIMIENTO
Técnicas de Procesamiento de LN	Bolsa de palabras	Reglas de Asociación
Taxonomía de términos	Términos	Generalización de Reglas de Asociación
Episodios	Episodios	Reglas de Episodios
Taxonomía de términos	Jerarquías de Conceptos	Reglas de Asociación
- Tokenización - Etiquetado de parte del discurso - Lemmatizaciones	Direct Acydic Graph	
	Conceptos o términos	Reglas de Asociación cualitativas
	Representación Basada en Modelos de Recuperación de Información	- Clustering / Visualización de Documentos - Reglas de Asociación - Modelado predictivo (Modelos de Clasificación)
Indexación	Estructura Indexada (conjunto de palabras clave)	Reglas de Asociación
- Etiquetado de parte del discurso - Extracción de términos	Documentos Prototípicos	Clustering de Términos

Preprocesamiento – Ejemplo

Se desea realizar data warehousing en una base de datos médica

¿Cómo procesar campos de texto en una base de datos relacional?

Texto original ⇨Forma intermedia de representación

Preprocesamiento – Ejemplo

Análisis de registros multifrase (identificar separadores)

Limpiar registros de palabras vacías

Substituir acrónimos

Construir diccionario de datos

Calcular frecuencias de términos

Calcular itemsets frecuentes

Obtener forma intermedia

Minería de Texto y Ontologías

Utilizando Minería de Texto podemos obtener información muy útil a partir de grandes cantidades de texto.

Sin embargo, para poder obtener mejores resultados es necesario comprender la **semántica** del texto procesado.

Las **ontologías** nos permiten representar y organizar la información **semántica** de cara a poder emplearla en procesos de minería de texto.

Introducción a las Ontologías I

- Las ontologías describen **conceptos** y **relaciones** existentes en algún dominio, de forma compartida y consensuada.
- Generalmente representan una jerarquía de conceptos junto a relaciones entre conceptos que pueden directas, transitivas, reflexivas, etc.
- Esta conceptualización debe ser representada de una manera **formal**, **legible** y utilizable mediante un **procesamiento automático**.
- Las ontologías no sólo permiten la **estructuración** del conocimiento, sino que también permiten realizar un **razonamiento** sobre las afirmaciones que modelan.
- Mediante el uso de razonadores, se puede validar una ontología o realizar tareas tales como clasificación de instancias y clases.

Introducción a las Ontologías I

Formalmente, una ontología está formada por:

- Clases. Son los conceptos del dominio.
- Propiedades. Pueden ser de dos tipos:
  - Relaciones: enlaza dos clases de la ontología.
  - Atributos: son las características propias de una clase.
- Individuos. Son las instancias concretas de una clase.
- Axiomas. Son restricciones impuestas a los elementos de la ontología

Introducción a la Ontologías II

Existen diferentes tipos de ontologías según el tipo de conocimiento que modelan

- **Ontología de Alto Nivel:** Describen conceptos muy generales como espacio, tiempo, eventos, que son independientes de un problema o dominio particular. Lo más razonable es tener ontologías de alto nivel comunes para grandes comunidades de usuarios.

- **Ontología de Dominio:** Describe el vocabulario asociado a un dominio genérico, especializando los conceptos introducidos en la ontología de alto nivel.

- **Ontología de Tarea:** Describe el vocabulario asociado a una actividad o tarea genérica especializando una ontología de alto nivel.

- **Ontologías de Aplicación:** Son las ontologías mas específicas. Los conceptos de estas ontologías suelen corresponderse con los roles desempeñados por las entidades de dominio cuando se realiza una cierta actividad.

Introducción a la Ontologías III

- Las Ontologías formalizan la parte **intensional** del conocimiento *(estructura)* sobre un dominio mientras la parte **extensional** *(datos)* la proporciona una Base de Conocimiento (Knowledge Base, KB), que contiene aserciones sobre las instancias de los conceptos y relaciones de la ontología.

- Podríamos decir que el conocimiento en una KB se rige por una conceptualización explícita o implícita. Dicha conceptualización es la ontología.

- Herramienta de gestión de ontologías: Protège

- Buscadores de ontologías:

  http://www.ontobee.org/

- https://bioportal.bioontology.org/ontologies

Ontologías versus Knowledge Graphs

Característica	Ontología	Grafo de Conocimiento
Propósito	Formalizar y estructurar conocimiento.	Representar relaciones entre entidades de forma flexible.
Estructura	Jerárquica y formal (clases, subclases, relaciones).	Gráfica y más flexible (nodos y aristas).
Flexibilidad	Menos flexible, ya que requiere reglas formales.	Más flexible y dinámico, se puede agregar información fácilmente.
Relaciones	Basadas en reglas formales, puede ser lógica.	Relacionadas de manera más directa, con más libertad.
Tamaño	Normalmente más pequeña, estructurada y definida.	Generalmente más grande, menos estructurada y más abierta.
Aplicaciones	Razonamiento lógico, web semántica, IA formal.	Motores de búsqueda, sistemas de recomendación, chatbots.

Lenguajes de Representación de Ontologías

El World Wide Web Consortium (W3C) es un consorcio internacional compuesto por organizaciones e investigadores, cuyo objetivo es desarrollar estándares para la web.

En el caso de representación de ontologías los principales estándares son:

- RDF (Resource Description Framework)
- RDFS (RDF Schema)
- OWL/OWL2 (Web Ontology Language)

Web Ontology Language (OWL 2) I

OWL 2 es un lenguaje de para la **definición de ontologías** en la Web Semántica.

Las ontologías OWL 2 representan **clases**, **propiedades**, **individuos**, y **valores de datos**.

OWL 2 puede expresarse usando **varias sintaxis**, la única reconocida como obligatoria es la RDF/XML.
Las ontologías OWL 2 pueden verse como grafos RDF.

Es posible **razonar** de forma automática sobre OWL 2.

Introducción básica
http://www.w3.org/TR/owl2-overview/

Guía de referencia rápida del lenguaje
http://www.w3.org/TR/owl2-quick-reference/

Web Ontology Language (OWL 2) II

La **semántica** asociada a las estructuras del lenguaje OWL 2 puede asignarse según dos criterios:

- **Semántica Directa (OWL 2 DL):** Se corresponde con la Lógica Descriptiva (DL) SROIQ, pero limita el uso de algunas estructuras y propiedades, para garantizar que sea decidible.

- **Semántica RDF (OWL 2 Full):** Se aplica la semántica que se usa en RDF, tiene todo el poder expresivo de representación, pero limita las capacidades de razonamiento al no ser siempre decidible.

Web Ontology Language (OWL 2)

Atendiendo a su capacidad expresiva se definen varios perfiles:

- **OWL 2 EL:** Apropiado para aplicaciones que usan grandes ontologías, y donde se puede sacrificar la capacidad expresiva para garantizar el rendimiento.
- **OWL 2 QL:** Apropiado para aplicaciones con ontologías ligeras (tamaño pequeño y poca complejidad) y con un alto número de instancias, que se organizan para poder usar consultas relacionales.
- **OWL 2 RL:** Apropiado para aplicaciones en las que ontologías relativamente ligeras se usan para organizar gran cantidad de individuos y es necesario operar sobre los datos como triples RDF.

Las ontologías de OWL 1, son por definición ontologías OWL 2 válidas.

Minería de Texto y Ontologías I

Atendiendo al uso que se puede hacer de las ontologías en procesos de minería de texto podemos hablar de ontologías comunes y ontologías de dominio.

- **Ontologías comunes:** Donde se representan relaciones de objetos generales. Tienen el inconveniente de que carecen en muchos casos de lenguaje específico técnico de un dominio particular.
  - Un ejemplo son los diccionarios semánticos tales como WordNet.
- **Ontologías de dominio:** Representan un vocabulario y un conjunto de relaciones más reducido sobre un dominio específico. Se usan para representar vocabulario específico de un dominio concreto que no es de uso común.
  - Suelen realizarse por expertos, de forma manual o semi-automática, y por tanto, su construcción es costosa.

Integración de Ontologías y Lexicons

Para poder establecer una comunicación y realizar un intercambio de información es necesario compartir el mismo conjunto de palabras (lexicon) y conocer el modelo subyacente a éste.

Este modelo puede representarse como una ontología, cuya función es agrupar conceptos similares, definir sus relaciones mutuas, y dar soporte a la herencia de propiedades y el razonamiento.

El lexicon y la ontología capturan diferentes tipos de información

- **Lexicon:** Información sintáctica específica del lenguaje e información morfológica.
- **Ontología:** Es independiente del lenguaje y captura el significado formal y las interrelaciones entre conceptos que no se reflejan en el lexicon.

Minería de Texto y Ontologías II

Las ontologías en Minería de Texto desempeñan diferentes roles atendiendo a su uso, de forma general podemos distinguir:

- **Ontologías como recurso semántico.**
- **Ontologías como producto de un proceso de Minería de Texto:**
  - Aprendizaje de Ontologías
  - Representación mediante Ontologías

Ontologías como Recurso Semántico

Para poder conocer el significado del texto que se procesa es necesario acudir a fuentes de información que determinen la semántica de los términos procesados.

Suggested Upper Merged Ontology (SUMO)

Yet Another Great Ontology (YAGO)

Systematized Nomenclature of Medicine (SNOMED)

Suggested Upper Merged Ontology (SUMO)

Es una de las mayores ontologías formales existentes.

- Ontología gratuita propiedad de IEEE (Diciembre 2000).
- Alrededor de 25.000 términos y 80.000 axiomas.
- Mapeada a todos los lexicon de WordNet.
- Escrita en el lenguaje SUO-KIF (Standard Upper Ontology *Knowledge Interchange Format*).
- Se emplea en investigación y aplicaciones sobre búsqueda, lingüística y razonamiento.
- Las ontologías que extienden SUMO deben estar disponibles bajo la licencia GNU General Public License.

Web SUMO: http://www.ontologyportal.org/
SumoBrowser:
https://sigma.ontologyportal.org:8443/sigma/Browse.jsp?kb=SUMO

Yet Another Great Ontology (YAGO)

Ontología extraída de forma automática a partir de Wikipedia, WordNet y Geonames:

- Desarrollada en el Instituto Max Planck (2008).
- Contiene 10 millones de entidades y 120 millones de hechos sobre esas entidades
- Yago combina la taxonomía de Wordnet enriquecida con el sistema de categorización de la Wikipedia, asignando las entidades a más de 350.000 clases.
- YAGO está enlazada a SUMO y la Ontología de DBPedia.
- Se utilizó en el sistema Watson de IBM.
- Se distribuye con licencia Creative Commons.

https://yago-knowledge.org/

Systematized Nomenclature of Medicine (SNOMED)

Es la mayor recopilación de terminología médica multilingüe del mundo.

- Contiene términos sobre anatomía, enfermedades, procedimientos, microorganismos, etc...
- Originalmente desarrollada en 1973, se ha ido ampliando con información proporcionada por el NHS inglés.
- Traducida a numerosos idiomas

Explorador: http://browser.ihtsdotools.org/

Aprendizaje de Ontologías I

Consiste en la generación automática o semi-automática de ontologías utilizando técnicas de aprendizaje automático (ML) o de procesamiento de lenguaje natural (NLP).

Las primeras referencias al término (Ontology Learning), podemos encontrarlas en [Madche and Staab, 2001] donde se describe en términos de la adquisición de un modelo de un domino a través de los datos.

Cuando el aprendizaje de ontologías se realiza sobre fuentes textuales no estructuradas, es cuando hablamos de aprendizaje de ontologías a partir de texto.

Aprendizaje de Ontologías II

El proceso de aprendizaje de ontologías puede verse como un proceso de ingeniería inversa, pero presenta los siguientes inconvenientes:

- En el proceso de creación de un texto sólo refleja el dominio de conocimiento del autor de forma parcial, por lo que el proceso de ingeniería inversa como mucho podrá reconstruir parcialmente dicho modelo de conocimiento
- El conocimiento del mundo raramente se menciona de forma explícita

Aprendizaje de Ontologías III

No existe un consenso entre la comunidad investigadora en cuanto a cuales son las tareas concretas que deben realizarse en el proceso de aprendizaje.
En nuestro caso nos ceñiremos a la definición de subtareas para el desarrollo de ontologías definida en [Cimiano, 2006].

$\forall x(país(x) \rightarrow \exists y\ capital\_de(y,x) \wedge \forall z(capital\_de(z,x) \rightarrow y=z))$

disjuntos(río, montaña)

$capital\_de \leq_R localizado\_en$

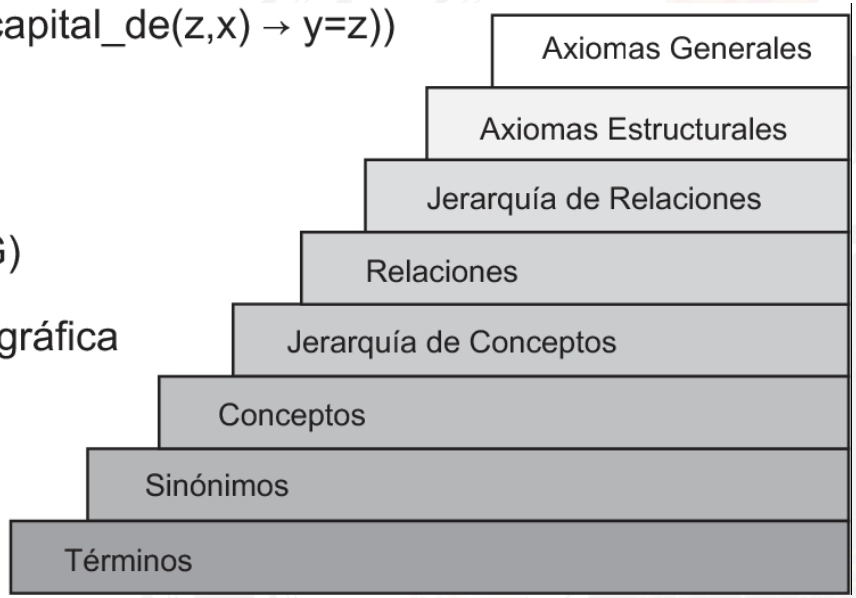
fluye_a_través(dominio:río, rango:ZG)

$capital \leq_c ciudad$, $ciudad \leq_c$ zona geográfica

$c:=$ país $:= \langle i(c), ||c||, Ref_c(c) \rangle$

[país, nación]

río, país, nación, ciudad, capital ...



Aprendizaje de Ontologías IV

Para realizar cada una de las subtareas se suelen utilizar distintas técnicas:

Términos – frecuencia, TFIDF, entropía, estadísticas de un corpus de referencia

Sinónimos – diccionarios electrónicos (implica WSD)

Conceptos - diccionarios electrónicos

Jerarquía de conceptos – clustering, patrones de Hearst

Relaciones – reglas de asociación

Jerarquía de relaciones - clustering

Axiomas estructurales

Axiomas generales

Tareas

Las ontologías en Minería de Texto, se pueden emplear para diversas tareas:

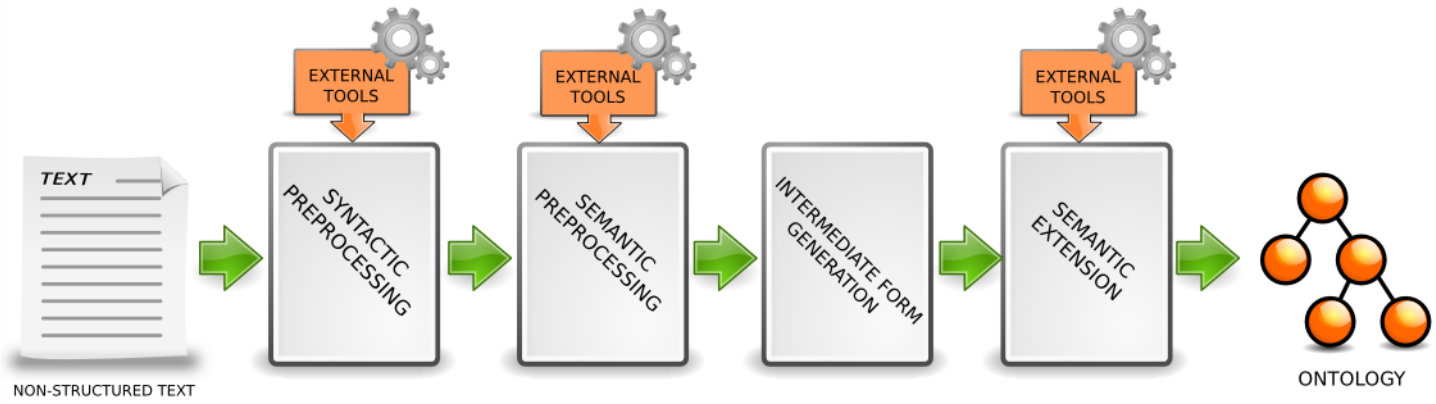
- Extracción de Información (IE)
- Recuperación de Información (IR)
- Aprendizaje de Ontologías (OL)
- Población de Ontologías (OP)
- Detección de eventos (NED)

Aplicaciones

Existen diversas aplicaciones en las que se emplean ontologías para Minería de Texto:

- Clasificación de páginas web en Directorios Web
- Agrupamiento de Documentos Médicos Multilingües
- Creación de ontologías de dominio a partir de Texto

Ejemplo: Generación de Ontologías I

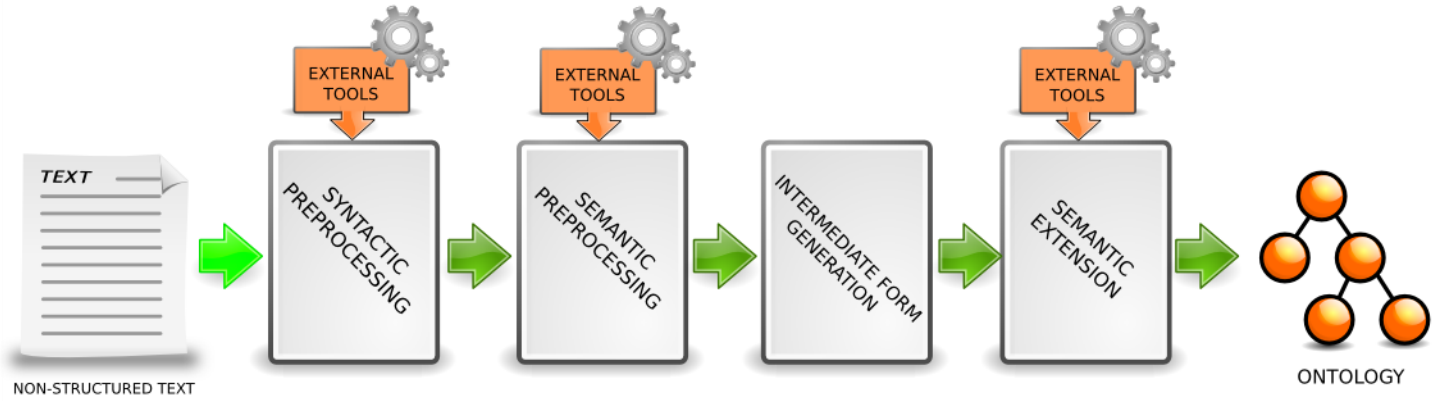


16616 FREDDI: A FUZZY RELATIONAL DEDUCTIVE DATABASE INTERFACE
21186 DATA SUMMARIZATION IN RELATIONAL DATABASES THROUGH FUZZY DEPENDENCIES
164017 GEFRED - A GENERALIZED-MODEL OF FUZZY RELATIONAL DATABASES

El objetivo es la generación de una ontología que represente los principales temas de los que tratan los textos.

Se emplea una metodología genérica que puede instanciarse con distintas herramientas en diferentes etapas.

Ejemplo: Generación de Ontologías II

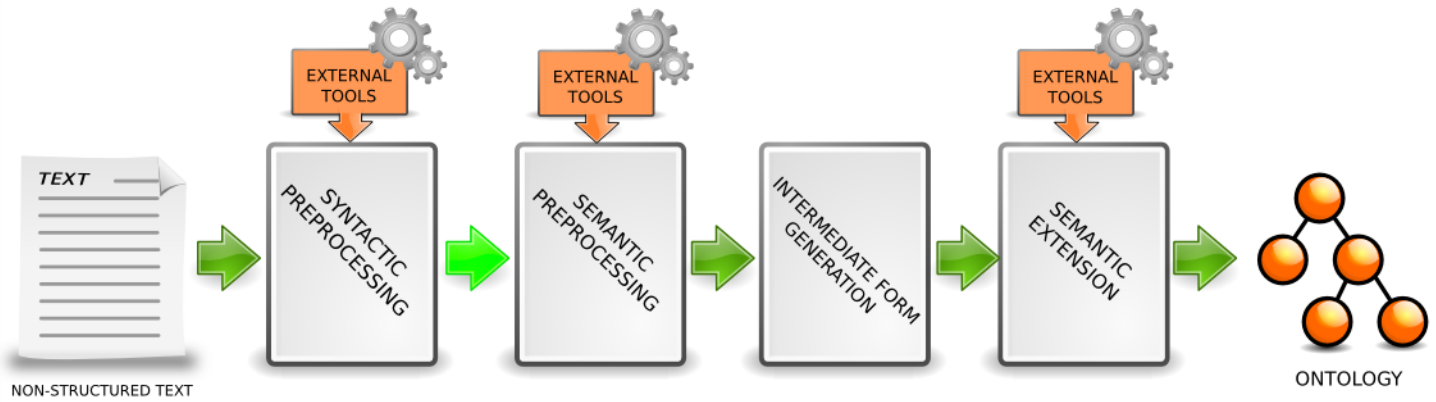


16616 FREDDI**:** **A** FUZZY RELATIONAL DEDUCTIVE DATABASE INTERFACE
21186 DATA SUMMARIZATION **IN** RELATIONAL DATABASE**S** **THROUGH** FUZZY DEPENDENC**IES**
164017 GEFRED **-** **A** GENERALIZED**-**MODEL **OF** FUZZY RELATIONAL DATABASE**S**

Preprocesamiento sintáctico:

- Tokenización
- Eliminación de palabras vacías
- Etc...

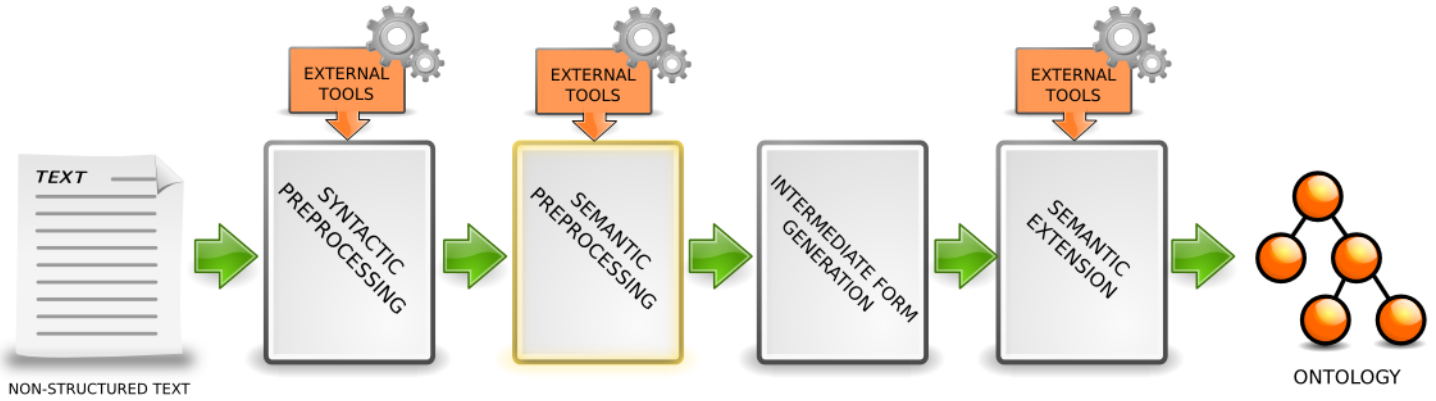
Ejemplo: Generación de Ontologías III



Preprocesamiento semántico:

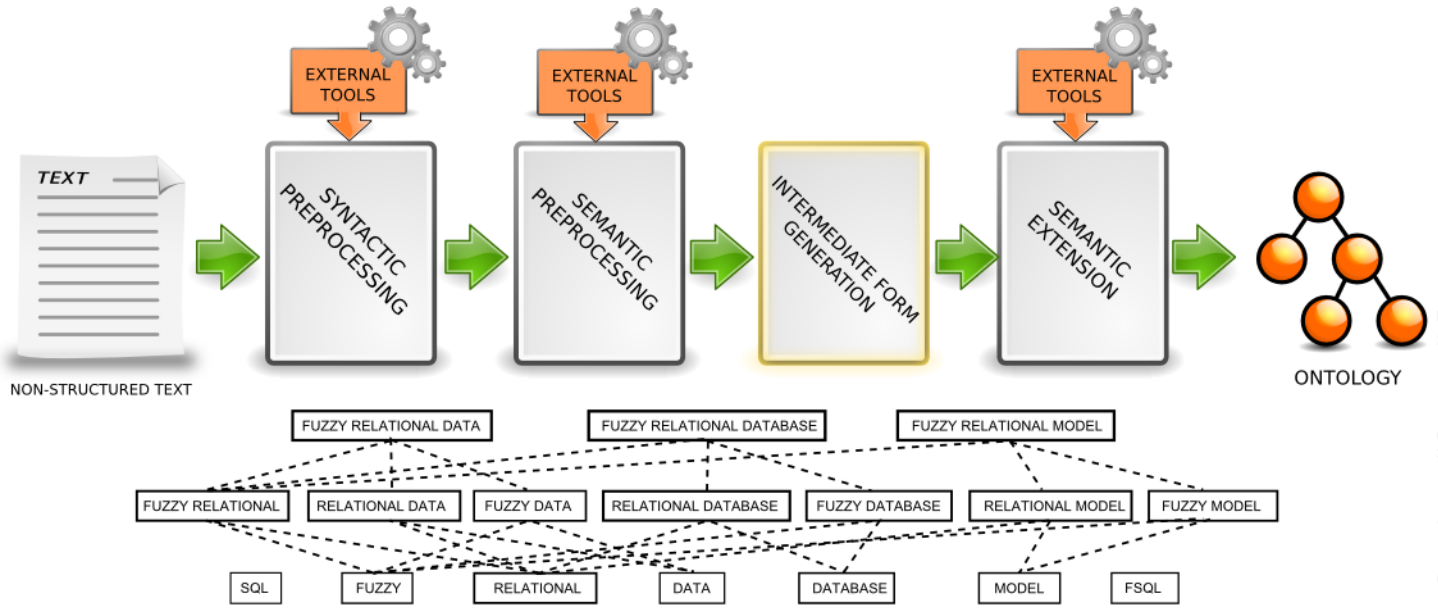
- **Unifica** los términos
- Determina el conjunto del sinónimos del término
  - Categoría gramatical (POS)
  - Desambiguación (WSD)

Ejemplo: Generación de Ontologías IV



- Determinar el **representante canónico** de cada conjunto de sinónimos
- Sustituir los términos por el representante canónico del conjunto de sinónimos.

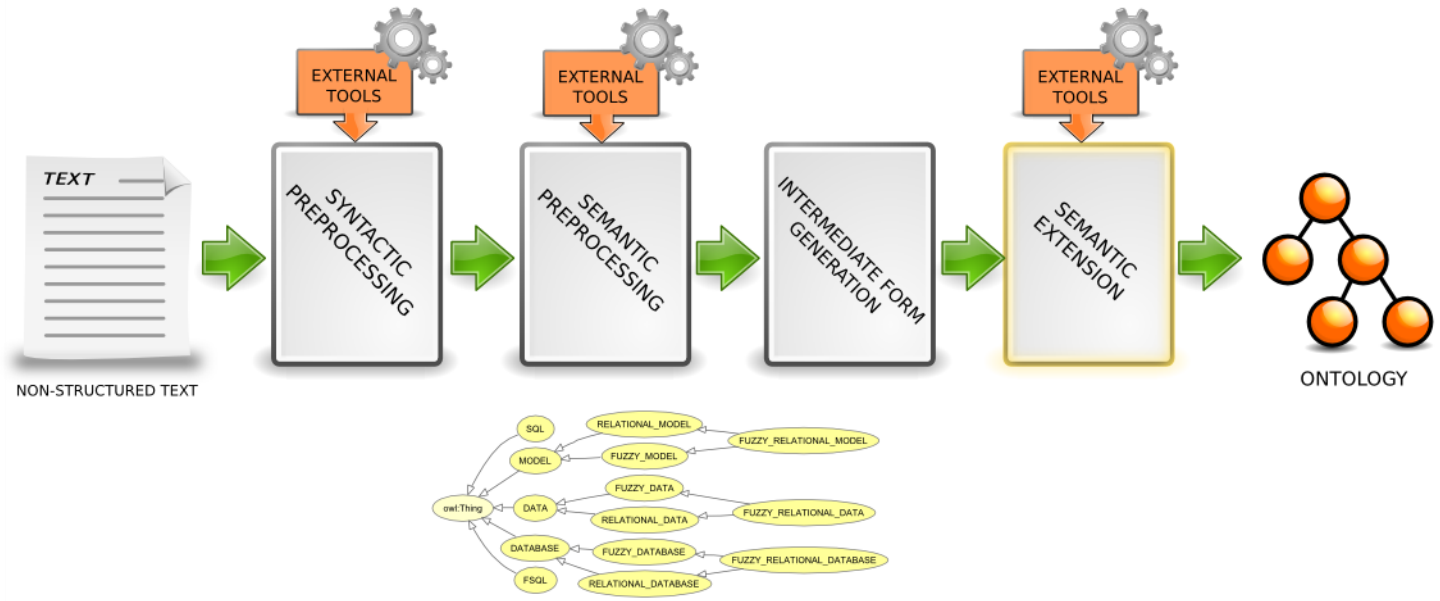
Ejemplo: Generación de Ontologías V



Se genera la forma intermedia:

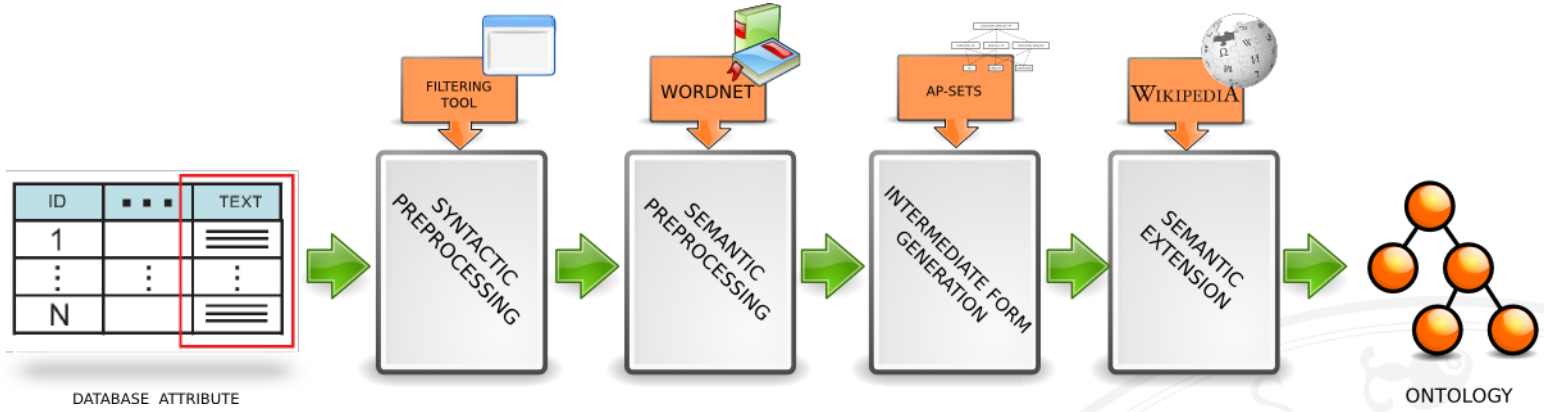
- Bolsa de palabras (BOW)
- Representación estructurada

Ejemplo: Generación de Ontologías VI



- Se seleccionan los términos apropiados de la representación intermedia
- Se obtiene información adicional de una ontología de referencia o herramienta externa.

Ejemplo: Generación de Ontologías VIII



El origen de los datos puede ser documentos de texto, o tuplas de una base de datos.

La representación intermedia va a determinar la forma de procesar los datos.

La instanciación se puede hacer con herramientas tales como WordNet o Wikipedia.

Otras técnicas que usan Semántica

La amplia difusión que ha tenido el uso de ontologías en procesos de Minería de Texto, se debe a la importancia de contar con herramientas que permitan gestionar la semántica de los datos textuales.

Existen otras técnicas que tratan de obtener dicha semántica no a través del uso de fuentes de conocimiento externas, sino a través del descubrimiento de la semántica latente del texto, a través de técnicas estadísticas.

Entre estas técnicas destacan:

- Análisis de Semántica Lantente (Latent Semantic Analysis / **LSA**)
- Latent Dirichlet Allocation (**LDA**)

Análisis de Semántica Latente

Latent Semantic Analysis (LSA) [Deerwester et al. 1990], es una técnica algebraica de análisis factorial que premite reducir la dimensionalidad de una matriz de términos-documentos, capturando la mayor parte de la varianza de un corpus textual.

LSA parte de la hipótesis de que palabras con significados similares ocurrirán en contextos similares.

Cuando comparamos documentos, lo hacemos utilizando los términos. LSA permite comparar los documentos a un nivel más general, a nivel de concepto o características/factores (*features*).

Utilizando descomposición en valores singulares (Singular Value Decomposition SVD) podemos extraer esas características de los documentos.

Análisis de Semántica Latente

Mediante SVD la matriz de documentos-términos A se separa en 3 matrices.

$$A = U \sum V^T$$

- U : Relaciona cada término con los nuevos conceptos encontrados.
- $\sum$ o S : Matriz diagonal que contiene los valores singulares de A representados en orden descendente.
- V^T : Relaciona los documentos con los conceptos.

Ejemplo LSA

Tomamos un conjunto de documentos y sus términos correspondientes, dado que sólo se tienen en cuenta propiedades estadísticas, en el ejemplo se usan letras para representar términos.

```
d1: c a a b c b c
d2: a b c a b c c
d3: d e f f d
d4: f d e d f
```

Creamos la matriz de términos-documentos donde las filas representan términos únicos y las columnas documentos. El contenido de la celda en este caso será la frecuencia absoluta, pero podía usarse un esquema de pesos como TF-IDF.

	d1	d2	d3	d4
a	2	2	0	0
b	2	2	0	0
c	3	3	0	0
d	0	0	2	2
e	0	0	1	1
f	0	0	2	2

- Ejemplo LSA: http://matpalm.com/lsa_via_svd/index.html

Ejemplo LSA

Aplicamos SVD y obtenemos las siguiente matrices:

A = U x S x Vt

	d1	d2	d3	d4
a	2	2	0	0
b	2	2	0	0
c	3	3	0	0
d	0	0	2	2
e	0	0	1	1
f	0	0	2	2

	f1	f2	f3	f4
a	0.48	0	0	0
b	0.48	0	0	0
c	0.72	0	0	0
d	0	-0.66	0	0
e	0	-0.33	0	0
f	0	-0.66	0	0

	f1	f2	f3	f4
f1	5.83	0	0	0
f2	0	4.24	0	0
f3	0	0	0	0
f4	0	0	0	0

	d1	d2	d3	d4
f1	0.70	0.38	0	0
f2	0	0	-0.70	-0.70
f3	0	0	0	0
f4	0	0	0	0

En **S** podemos ver la fuerza relativa de los dos conceptos/características en f1 y f2

En **U** vemos como se relaciona cada término con los conceptos encontrados

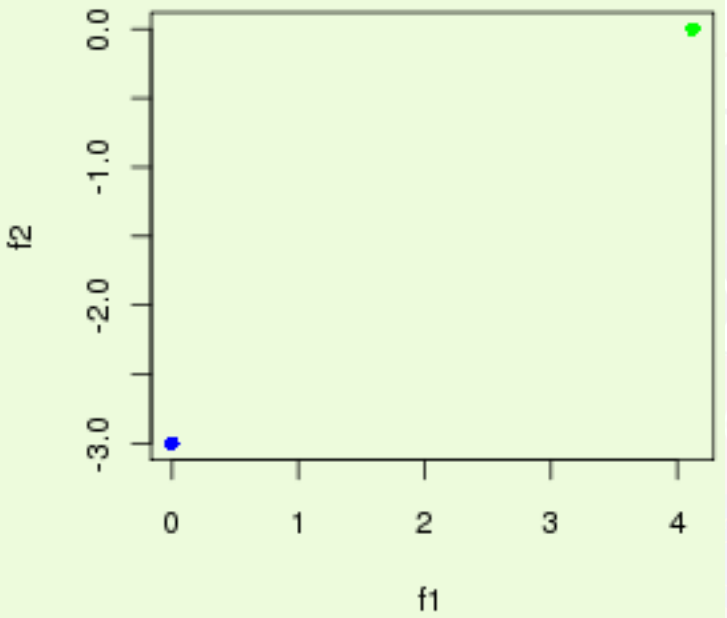
En $\mathbf{V}^{T}$ vemos la relación de los documentos con los conceptos.

Ejemplo LSA

Si multiplicamos S y VT obtenemos la relación entre los documentos y los conceptos.

Representando en una gráfica f1 y f2 vemos como en efecto existe una separación evidente entre los documentos d1 y d2, respecto a d3 y d4.

	f1	f2	f3	f4
d1	4.123	0.000	0.000	0.000
d2	4.123	0.000	0.000	0.000
d3	0.000	-3.000	0.000	0.000
d4	0.000	-3.000	0.000	0.000



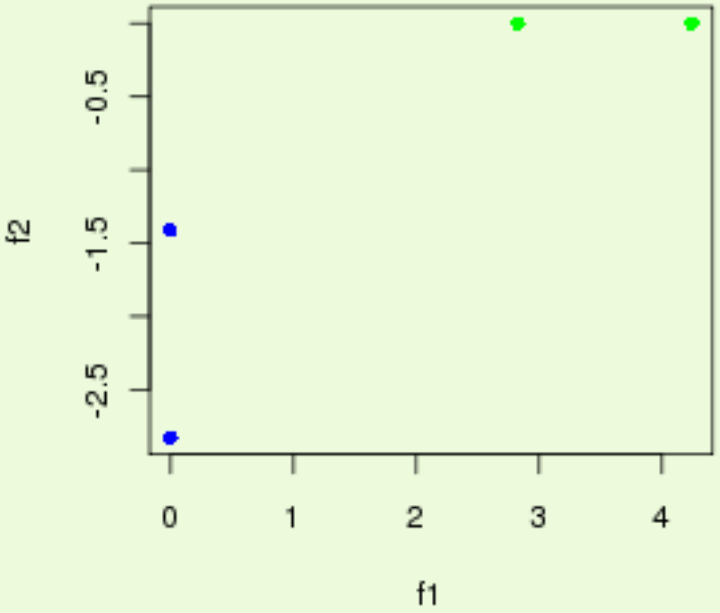
Ejemplo LSA

Si multiplicamos las matrices U y S, obtendremos la relación entre los términos y los conceptos.

Los términos *a*, *b* y *c* se alinean con el concepto 1, mientras que d, e y f lo hacen con el concepto 2.

También vemos como c tiene una asociación más fuerte con el concepto 1 que a o b (porque tiene mayor frecuencia), al igual que e tiene una menor asociación con el concepto 2, que d o e (por su menor frecuencia).

	f1	f2	f3	f4
a	2.82	0	0	0
b	2.82	0	0	0
c	4.24	0	0	0
d	0	-2.82	0	0
e	0	-1.41	0	0
f	0	-2.82	0	0



Análisis de Semántica Latente

Ventajas:

- Permite reducir la dimensionalidad de la matriz de datos
- Al ser un enfoque puramente numérico se puede emplear con cualquier idioma.

Limitaciones:

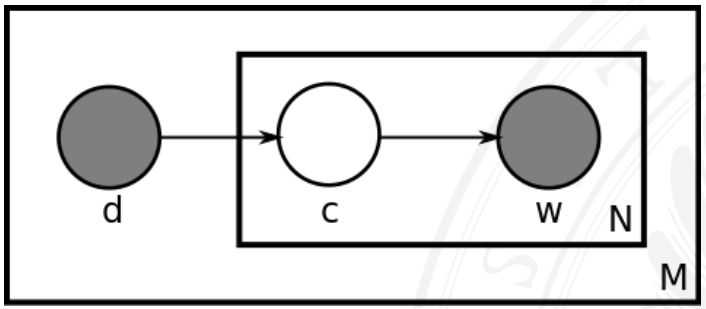
- Dada la alta dimensionalidad de los datos textuales, calcular SVD es muy costoso.
- No tiene en cuenta el orden de las palabras
- El nuevo espacio de conceptos/características es muy difícil de interpretar al ser una combinación lineal de un conjunto de palabras del espacio original.
- No se puede generalizar para incluir información adicional (fechas, autores...).

Probabilistic LSA (pLSA)

También conocido como *aspect model*, es una alternativa a LSA.

Es un modelo de variable latente que asocia una variable de clase no observada *c* (aspecto/concepto), con cada documento de la colección *d* y representa cada aspecto como una distribución de palabras con una determinada probabilidad *P(w|c)*.

$$P(w,d) = \sum_c P(c)P(d|c)P(w|c) = P(d)\sum_c P(c|d)P(w|c)$$



http://commons.wikimedia.org/wiki/File:Plsi_1.svg

d es la variable del documento, c es un concepto al que pertenece una palabra obtenida mediante la distribución P(c|d), w es una palabra obtenida de la distribución de palabras del concepto al que pertenece esa palabra P(w|c). Las variables d y son variables observadas, el concepto c es una variable latente.

Latent Dirichlet Allocation (LDA)

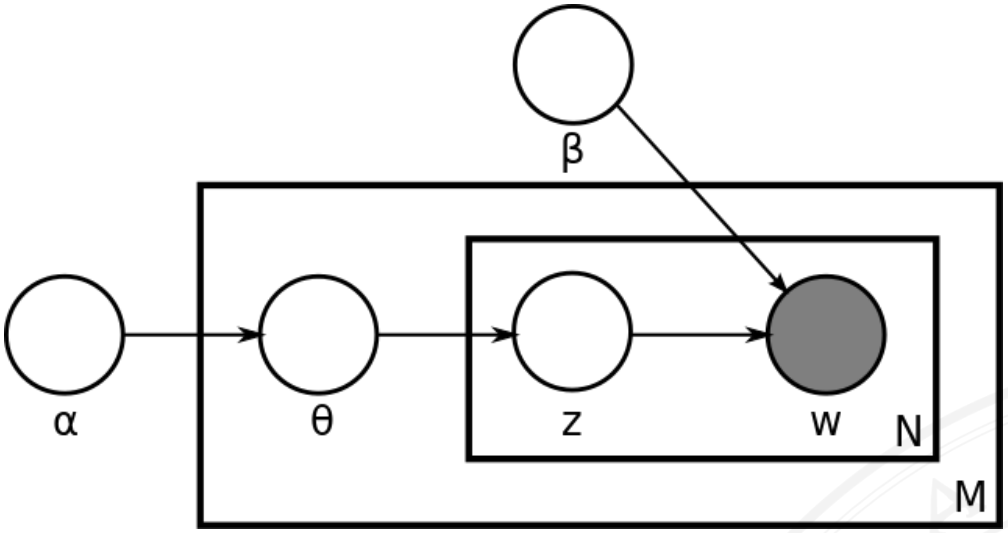
Es un modelo generador probabilístico de un corpus de documentos. La idea es que que cada documento está representado por una mezcla de temas, donde cada tema es una variable latente caracterizada por una distribución sobre un vocabulario fijo de palabras.

La completitud del proceso generador para un documento se consigue considerando a priori una distribución de Dirichlet en el documento sobre los temas y en los temas sobre las palabras.

La estructura del modelo LDA permite la interacción de las palabras observadas en documentos con las distribuciones estructuradas de un modelo de variables oculto.

Este método se ha aplicado para encontrar estructuras útiles en distintos tipos de documentos, tales como emails, literatura científica, libros en librerías digitales y archivos de noticias.

Latent Dirichlet Allocation (LDA)



http://commons.wikimedia.org/wiki/File:Latent_Dirichlet_allocation.svg

α : Es el parámetro de la distribución Dirichlet para las distribuciones de temas por documento.

β: Es el parámetro de la distribución Dirichlet para las distribuciones de palabras por tema.

θ: Es la distribución de temas por documento i

z: Es el tema para la palabra w en el documento i

w : Es una palabra concreta.

Solo w es una variable observable, el resto son ocultas.

Online LDA

Existe una variante Online de LDA para analizar flujos de texto [AlSumait et al. 2008].

El modelo OLDA considera la ordenación temporal de la información y asume que los documentos van llegando en intervalos de tiempo discretos.

En cada intervalo de tiempo de un tamaño determinado (hora, día, año...) un flujo de documentos de tamaño variable se recibe para ser procesado.

Un documentos recibido se representa como un vector de palabras.

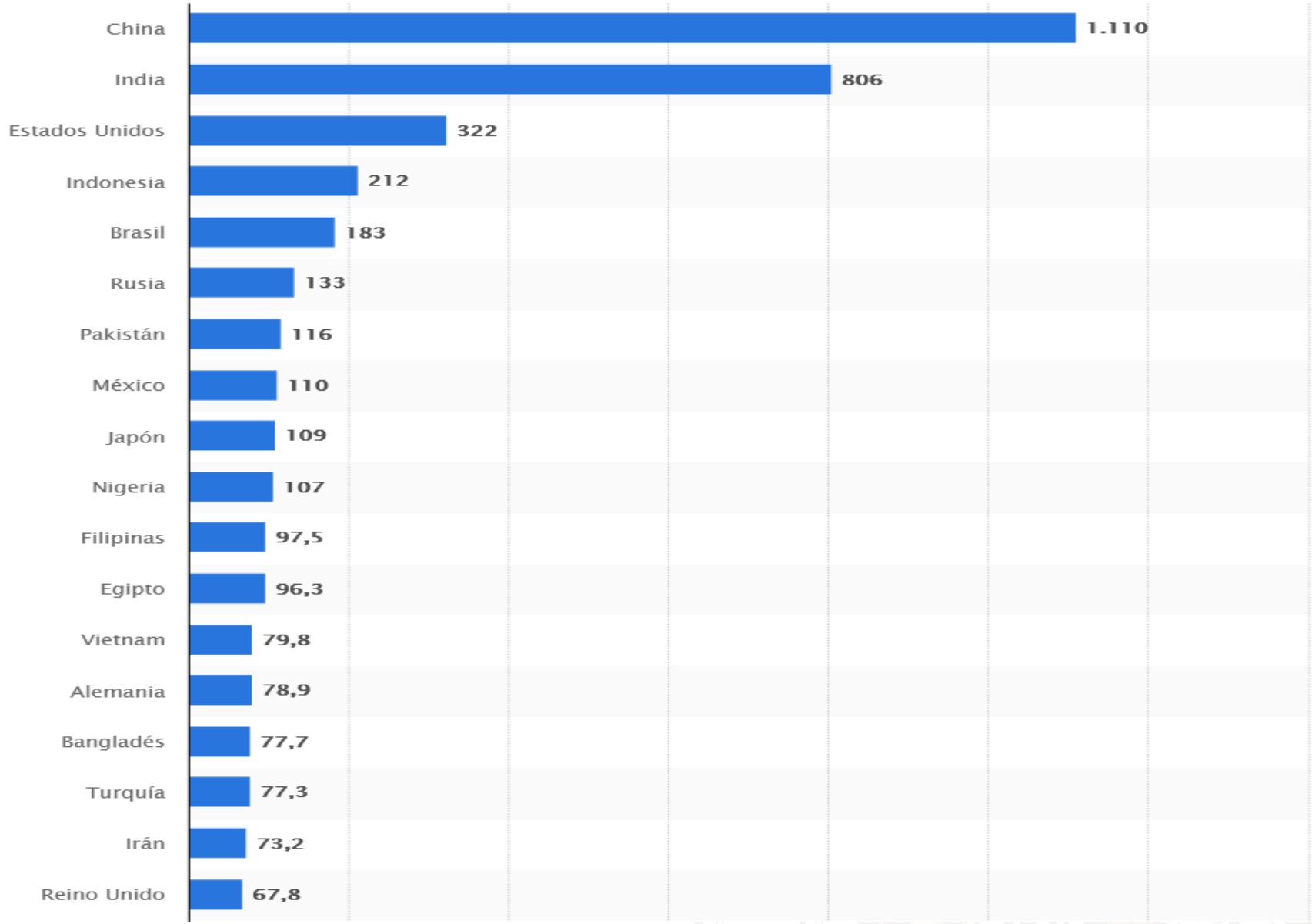
Entonces se usa LDA para modelar los documentos recibidos. El modelo generado en un instante determinado, se usa como distribución a priori en el siguiente intervalo de tiempo, cuando lleguen nuevos documentos.

Minería de Texto Multilingüe

- Los contenidos de las páginas web se encuentran en diversos idiomas.
- En ocasiones, la información relevante para un usuario no se encuentra en su propio idioma.
- Los Recursos Léxicos sólo existen para algunos idiomas.

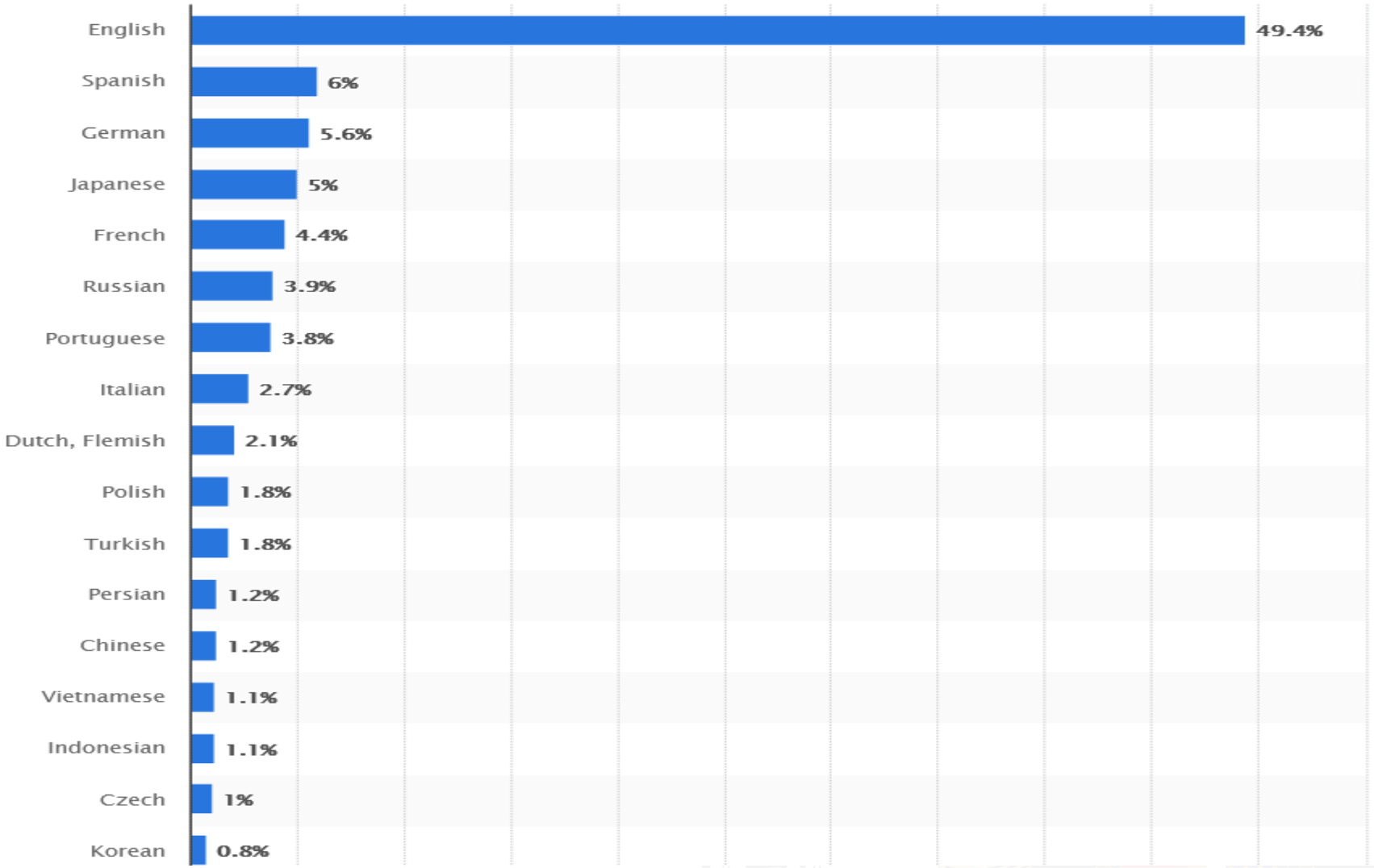
Usuarios de Internet en el Mundo (Enero2025)

Fuente:Statista.com



Lenguajes más usados para contenido web en Internet (Febrero 2025)

Fuente: Stata.com



Fuente: www.statista.com

Minería de Texto Multilingüe

La gran cantidad de información disponible en diversos idiomas, ha generado la necesidad de adaptar las técnicas de minería de texto a contextos multilingües.

Algunas de las aplicaciones desarrolladas son:

- Traducción automática (Machine Translation).
- Recuperación de contenido multilingüe.
- Extracción de conocimiento entre lenguajes (cross-language).
- Minería de opiniones entre lenguajes.
- Categorización de textos.
- Resumen de textos.

Cada una de estas aplicaciones presenta una serie de retos diferentes.

Resumen de Textos

El resumen de textos multilingües consiste en crear un resumen de un conjunto de documentos similares, incluyendo las frases más relevantes de dichos documentos en el resumen.

Dado un conjunto de documentos (cluster), los pasos a seguir para realizar un resumen son:

- Calcular el vector medio del cluster.
- Ordenar los documentos según su similitud al vector medio.
- Seleccionar sentencias candidatas a partir de pesos que incluyen su relevancia y posición en el documento.
- Comprobar la similitud de la sentencia con otras presentes en el resumen.
- Añadir la sentencia candidata si no es redundante.
- Añadir sentencias hasta alcanzar el tamaño de resumen deseado.

El origen multilingüe de los textos obliga a seleccionar un idioma objetivo y traducir los resultados a dicho idioma.

LSA Multilingüe

Se necesita un corpus multi-paralelo en el que los mismos textos estén traducidos a distintos idiomas (se suelen usar traducciones de la biblia y el corán, así como documentos de patentes).

Cuando se usan múltiples lenguajes, se apilan las matrices de términos-documentos. Existen diversas aproximaciones para realizar estos apilamientos:

- **Básica** [Chew and Abdelali (2007)]: Se colocan las matrices unas a continuación de otras. Las filas corresponden a términos en todos los lenguajes.

- **Tucker1** [Kolda and Bader (2009); Tucker (1966)]: Se colocan las matrices formando una tercera dimensión.

$$X_k \approx U_k S_k V_k \qquad k = 1, \ldots, K.$$

- **PARAFAC2** [Harshman (1972)]: Similar a Tucker1, incluye una matriz H diagonal densa.

$$X_k \approx U_k H S_k V_k^T \qquad k = 1, \ldots, K.$$

Minería de Texto Multilingüe

El análisis de texto multilingüe aporta el beneficio de capturar información complementaria del mismo evento a través de distintos lenguajes.

- Sobre Contenido
- Sobre Opiniones y Sentimientos

- Uno de los grandes desafíos es desarrollar **herramientas**, **recursos** y **aplicaciones** de minería de texto multilingüe con el menor coste de desarrollo y tiempo posible.

Propuestas para Desarrollo de Herramientas

- Unicode.
- Teclados Virtuales.
- Modularidad.
- Clases de Token compartidas.
- Estructuras de entrada y salida uniformes.
- Simplicidad en las reglas y el lexicon.
- Compartir recursos entre lenguajes (lexica, gazetteers, grammar rules).
- Usar teoría de gramática.
- Usar Aprendizaje Automático (Machine Learning).
- No especificar en exceso.
- Minimizar el uso de herramientas específicas para un lenguaje.
- Evitar herramientas específicas para un lenguaje.

- Ralf Steinberger

Recursos Multilingües I

WordNet

- Se comenzó a desarrollar en 1985 en el Laboratorio de Ciencia Cognitiva de Princeton.
- Base de Datos léxica que contiene información sobre **nombres**, **verbos**, **adjetivos** y **adverbios** en Inglés.
- Para buscar en WordNet debemos conocer el **lema** de la palabra y su **categoría gramatical** (part-of-speech/POS)
- Se organiza en **Synsets**, conjuntos de palabras sinónimas.
- Los Synsets están relacionados a través de relaciones **conceptuales**, **semánticas** y **léxicas**, estableciendo una red.
- https://wordnet.princeton.edu/

Recursos Multilingües II

- Algunas de estas relaciones son:
  - **Hiponimia (Hyponym)**
    - El término específico usado para designar un miembro de una clase. X es un hipónimo de Y, si X es algún tipo de Y.
  - **Hiperonimia (Hypernym)**
    - El término general empleado para designar una clase completa de instancias específicas. Y es un hiperónimo de X, si X es algún tipo de Y.
  - **Meronimia (Meronym)**
    - El nombre de un constituyente de parte de, la sustancia de, el miembro de algo. X es un merónimo de Y, si X es parte de Y.
  - **Holonimia (Holonym)**
    - El nombre del todo al que hacen referencia los merónimos. Y es un holónimo de X, si X es parte de Y.
  - **Antonimia, Troponimia, Similitud, ...**

Recursos Multilingües III

EuroWordNet [Vossen, 2001]

- Base de datos que almacena WordNets en distintos idiomas.
- Proyecto Europeo para integrar WNs de 8 lenguajes europeos: ES, IT, EN, NL, FR, DE, CS, ET.
- Los distintos WNs se conectan a través de un índice Inter-Lingual-Index (ILI) que permite relacionar synsets en diferentes idiomas.
- https://archive.illc.uva.nl/EuroWordNet/

Recursos Multilingües IV

Multilingual Central Repository (MCR)

- Basado en EuroWordNet, integra en éste distintas versiones del WordNet de Princeton, y WordNets para *Castellano*, *Euskera*, *Gallego* y *Catalán*.

- Además incluye:

  - *EuroWordNet Top Concept Ontology* – *Enlazando* 64 conceptos definidos en la ontología con los ILI.

  - *WordNet Domains* – Extiende WN con etiquetas de dominio para cada synset, seleccionadas de un conjunto de 200 etiquetas estructuradas de forma jerárquica.

  - *Nuevas relaciones obtenidas de forma automática.*

    - https://www.cs.upc.edu/~nlp/meaning/demo/demo.html

Recursos Multilingües V

Listados de WordNets disponibles y el tipo de licencia que soportan:

Open Multilingual Wordnet

https://omwn.org/

Wordnets in the World

http://globalwordnet.org/wordnets-in-the-world/

Recursos Multilingües VI

BabelNet [Navigli – Ponzetto, 2012]

- Red semántica y ontología multilingüe lexicalizada.
- Desarrollada por Linguistic Computing Laboratory de la Universidad de la Sapienza (Roma).
- Conecta **Wikipedia** y **WordNet**, además de otros recursos como **WikiData**, **Wiktionary**, **OmegaWiki** y **Open Multilingual WordNet**.
- Proporciona conceptos y entidades lexicalizadas en diversos idiomas, así como sus relaciones semánticas.
- Está disponible una **API en Java** para acceder a los datos a través de un servicio *HTTP RESTful*.
- Cada **Babel synset** representa un sentido y contiene sinónimos en diferentes lenguajes.
- http://babelnet.org/