

¿Qué distingue a una aplicación de procesamiento de lenguaje, de una aplicación de datos?

Una aplicación de procesamiento de lenguaje se diferencia de otros sistemas por la forma de entender el lenguaje.

Si pensamos en el comando `wc` de Unix: cuando cuenta bytes o líneas es solo procesamiento de datos, pero cuando cuenta palabras necesita saber qué es una palabra, convirtiéndose así en procesamiento de lenguaje.

Los sistemas más avanzados, como chatbots, traductores automáticos o sistemas de preguntas y respuestas, requieren un entendimiento del lenguaje mucho más profundo:

- Fonética y Fonología: cómo suenan las palabras
- Morfología: las partes que forman las palabras
- Sintaxis: cómo se relacionan las palabras en las oraciones
- Semántica: qué significan las palabras y frases
- Pragmática: qué pretende comunicar realmente el hablante
- Discurso: cómo funciona la comunicación más allá de oraciones individuales

La diferencia clave está en que estos sistemas no solo procesan datos, sino que comprenden el lenguaje humano en sus múltiples dimensiones.

¿Cuáles son las categorías del "conocimiento del lenguaje"?

El conocimiento del lenguaje se divide en seis áreas principales:

- Fonética y Fonología: Entender los sonidos del habla. Un asistente virtual como HAL necesita esto para convertir audio en palabras y viceversa, sabiendo cómo suenan las palabras y cómo producir esos sonidos.
- Morfología: Reconocer las partes que forman las palabras. Esto significa entender que "puertas" es plural o que "corriendo" incluye la acción de correr más la idea de continuidad.
- Sintaxis: Conocer cómo ordenar y relacionar las palabras. HAL necesita esto para construir oraciones que tengan sentido, no solo juntar palabras al azar.
- Semántica: Comprender el significado. Incluye tanto el significado de palabras individuales ("exportar", "seda") como el significado que surge al combinarlas ("Europa Occidental", "finales del siglo XVIII"). También implica entender cómo la estructura de la oración afecta el significado.
- Pragmática: Entender las intenciones del hablante. Esto permite distinguir si alguien está preguntando, pidiendo algo o simplemente comentando, y responder de manera apropiada y cortés.
- Discurso: Manejar conversaciones y textos completos. Cuando alguien pregunta "¿Ese año?", el sistema debe recordar de qué año se habló antes para entender la pregunta.

Básicamente, la mayoría de tareas en procesamiento de lenguaje consisten en resolver ambigüedades en alguno de estos niveles.

Mencione dos razones que expliquen la presencia de ambigüedad en el lenguaje natural.

El lenguaje natural está lleno de ambigüedades por varias razones relacionadas entre sí. La frase "I made her duck" ilustra perfectamente los tipos principales de ambigüedad:

1. Ambigüedad gramatical: Las palabras pueden cumplir diferentes funciones gramaticales:
 - "Duck" puede ser un verbo ("agacharse") o un sustantivo ("pato")
 - "Her" puede indicar "a ella" o "de ella"
2. Ambigüedad semántica: Una misma palabra tiene varios significados:
 - "Make" puede significar "crear" o "cocinar"
3. Ambigüedad sintáctica: Las palabras se pueden agrupar de diferentes maneras:
 - El verbo "make" puede funcionar solo con un objeto ("I made dinner"), con dos objetos ("I made her dinner"), o con un objeto y otro verbo ("I made her duck")

Esto significa que "I made her duck" puede interpretarse como:

"Hice que ella se agachara"

"Le cociné su pato"

"Creé el pato de ella"

Cada interpretación surge de resolver las ambigüedades de manera diferente. Por eso, la mayoría de tareas en procesamiento de lenguaje consisten fundamentalmente en resolver estas ambigüedades para entender qué quiere decir realmente el hablante.

¿Cuáles son las herramientas computacionales y matemáticas para obtener conocimiento lingüístico?

Para enseñar a las computadoras a entender el lenguaje humano, necesitamos herramientas matemáticas y algoritmos específicos. Estas herramientas se organizan en dos categorías principales:

Modelos Formales

1. Máquinas de Estados: Como mapas de ruta con pasos y decisiones
 - Tienen estados conectados por transiciones, como un diagrama de flujo

- Útiles para entender sonidos, estructura de palabras y gramática• Cuando añadimos probabilidades, se vuelven más inteligentes para elegir el camino más probable
2. Sistemas de Reglas: Como gramáticas que aprendimos en la escuela
 - Definen las reglas del lenguaje de forma clara y estructurada
 - Pueden incluir probabilidades para manejar excepciones
 - Perfectos para analizar cómo se forman y combinan las palabras
 3. Lógica: Para capturar el significado real
 - Utiliza matemáticas para representar lo que realmente queremos decir
 - Especialmente importante para entender el significado y las intenciones
 4. Modelos Probabilísticos: El ingrediente secreto para resolver ambigüedades
 - Permiten que el sistema elija la interpretación más probable cuando hay varias opciones
 - Como tener un "sexto sentido" para elegir la interpretación correcta
 5. Modelos de Espacio Vectorial: Matemáticas para buscar información• Usan álgebra lineal para encontrar palabras similares y recuperar información relevante

Algoritmos

1. Búsqueda en Espacios de Estados: Como explorar un laberinto de posibilidades
 - El sistema prueba diferentes interpretaciones hasta encontrar la mejor
2. Aprendizaje Automático: Enseñar a la máquina con ejemplos
 - Clasificadores: Aprenden a categorizar cosas (¿es una pregunta o una orden?)
 - Modelos de Secuencia: Analizan secuencias completas de palabras, no solo palabras aisladas

En resumen, todos estos modelos y algoritmos trabajan juntos para resolver el gran desafío del procesamiento de lenguaje: eliminar las ambigüedades y entender qué quiere decir realmente el ser humano.

¿Con que tesis se formuló el Test de Turing?

Alan Turing se enfrentó a una pregunta imposible: ¿cómo saber si una máquina realmente piensa? El problema era que términos como "máquina" y "pensar" son demasiado vagos y subjetivos para dar una respuesta definitiva.

En lugar de perderse en definiciones filosóficas, Turing tuvo una idea brillante: crear una prueba práctica. Propuso un juego donde lo único que importaría sería cómo usa el lenguaje la computadora.

La lógica era simple pero poderosa: si una máquina puede conversar de manera tan natural y convincente que no podemos distinguirla de un ser humano, entonces para efectos prácticos, esa máquina sería considerada inteligente.

En esencia, Turing cambió una pregunta imposible ("¿piensa la máquina?") por una pregunta verificable ("¿puede la máquina usar el lenguaje como lo haría un humano inteligente?").