

INGENIERÍA DE SERVIDORES (2016-2017)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

IBM mainframes Watson Machine Learning

Francisco Javier Morales Piquerasa
Rubén Morales Pérez

21 de mayo de 2017

Índice

| | |
|--|----------|
| 1. Resumen | 3 |
| 2. Memoria | 3 |
| 2.1. Introducción | 3 |
| 2.2. Mainframes IBM z Systems | 3 |
| 2.2.1. Blockchain | 3 |
| 2.3. Watson | 3 |
| 2.3.1. Arquitectura | 4 |
| 2.3.2. Entendiendo las preguntas | 5 |
| 3. Conclusiones | 6 |

Índice de figuras

| | |
|------------------------------------|---|
| 2.2. Watson en Jeopardy! | 4 |
| 2.3. Arquitectura DeepQA | 5 |

Índice de tablas

1. Resumen

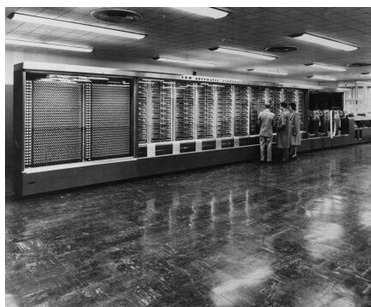
2. Memoria

2.1. Introducción

Hay varios retos que tienen las compañías actualmente para poder mantenerse competitivas en un mundo cada vez más globalizado. El mundo de las tecnologías de la información y la comunicación toma un papel fundamental, cada vez hay corporaciones con página web, aplicaciones u ofreciendo información actualizada a través de redes sociales.

Una vez que tenemos una base tecnológica es recomendable pasar al siguiente nivel, tener información suficiente y de calidad recopilada de forma que podamos obtener un beneficio competitivo con ella. Entonces entra en juego el análisis de datos y mantener esa información segura, por temas de protección de datos. Aquí es donde interviene IBM mainframes [2], grandes ordenadores que nos ofrecen computación en la nube de forma que podremos almacenar los datos en dichos servidores con cierta garantía y a la vez dejar a estos ordenadores el procesamiento pesado de datos.

El primer ordenador digital de propósito general de IBM fue ASCC (Automatic Sequence Controlled Calculator), se desarrolló junto con la Universidad de Harvard. Entre los sistemas más modernos hablaremos de z Systems.



(a) ASCC



(b) Z13

2.2. Mainframes IBM z Systems

2.2.1. Blockchain

2.3. Watson

En 2007 IBM Research se propuso el reto de crear un sistema para competir con los grandes campeones en el juego Jeopardy!. Este sistema, en adelante Watson, entraría dentro de la categoría QA (Question answering), que requiere avances en varias áreas de la ciencia de la computación y la inteligencia artificial. Algunas de estas áreas serían búsqueda y recuperación de información (IR), procesamiento de lenguaje natural (NLP), representación del conocimiento y razonamiento o el machine learning.



Figura 2.2: Watson en Jeopardy!

La comunicación humana es imperfecta, está llena de ambigüedades, polisemia, ironías, la misma información puede darse de varias maneras y a veces la interpretación de una frase no solamente depende del contexto actual sino de conversaciones anteriores entre los interlocutores. Además, dicha comunicación no está estructurada, como en un lenguaje de programación o una base de datos, donde los datos están bien definidos y la información es explícita. A día de hoy la información no estructurada, como es el caso del lenguaje natural, crece más rápido que la estructurada, por ello es buena idea usar análisis profundos del lenguaje natural para hacer inferencia a partir de los datos de los que disponemos.

Desde 2001 hasta 2006 se construyó la base para el reconocimiento de información no estructurada, UIMA (Unstructured Information Management Architecture). UIMA proporciona una plataforma para integrar la información obtenida tras analizar textos, imágenes, etc. Su objetivo es integrar varios programas llamados *anotadores*, que asignan significado semántico a ciertas partes del texto o imagen.

Algunos desarrolladores habían participado anteriormente en Question answering dentro del proyecto PIQUANT [1]. Este sistema tenía un conjunto estático de tipos de respuestas o clases de conceptos que se solicitaban en una pregunta. El nuevo sistema no tendría conexión a Internet, y en unos tres segundos tendría que procesar la pregunta, buscar una respuesta y estimar la probabilidad de que sea correcta. Tras analizar 2,000 partidas de *Jeopardy!* la media de los jugadores que ganaban era una precisión del 85 – 95 % de acierto sobre una media de 40 – 50 % de preguntas respondidas, o 85 % *Precision@40*

2.3.1. Arquitectura

En 2007 se consiguió, usando PIQUANT, un rendimiento 16 % *Precision@70* en las preguntas de *Jeopardy!*. Después se desarrollaron dos técnicas esenciales en el desarrollo de *Watson*, DeepQA y AdaptWatson.

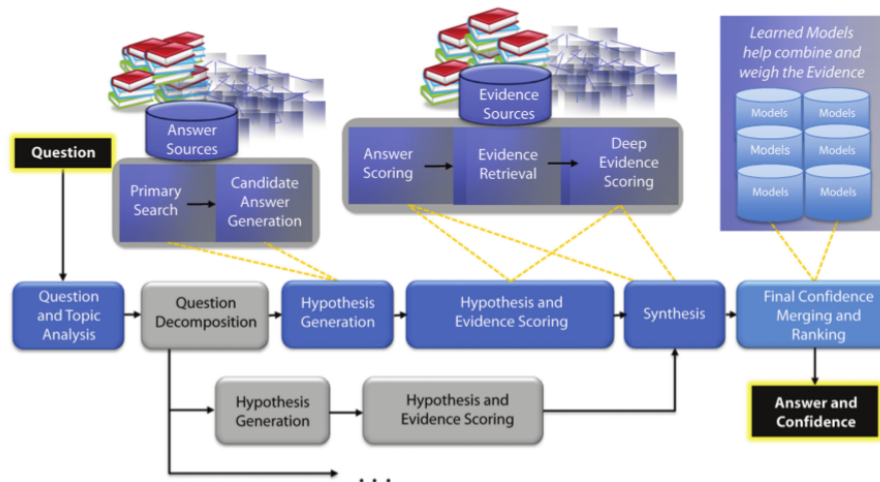


Figura 2.3: Arquitectura DeepQA

DeepQA define varias etapas del análisis, cada una tiene diferentes implementaciones que se ejecutan en paralelo. Nunca se admite que se ha entendido perfectamente una pregunta, de forma que pueda verse directamente la respuesta en una base de datos. Lo que se hace es buscar varias posibles preguntas candidatas, ponderar la probabilidad de que cada una sea la correcta, buscar varias posibles respuestas a cada pregunta y usar diferentes fuentes para ponderar la probabilidad de cada respuesta. En dicho proceso se le da valor al hecho de que una respuesta concuerde con el tipo que se espera, por ejemplo un monumento histórico, una persona, etc. También se tienen en cuenta fechas, geografía, la fiabilidad de las fuentes, etc. Este análisis produce cientos de valores o características, cada uno indicando el grado de evidencia que se ha encontrado de cada tipo que refuerza una respuesta concreta. Todas esas características deben ser combinados para obtener un único valor representando la probabilidad de que la respuesta sea correcta. Para dar una ponderación a cada característica de DeepQA se entrenó con un modelo estadístico usando machine learning a partir de preguntas y respuestas. El resultado final es una lista de respuestas candidatas, cada una con un valor que nos indica si es buena respuesta o no, en función de la evidencia.

El rendimiento del sistema no era demasiado alto, hasta que consiguieron integrar diferentes técnicas algorítmicas usando una metodología llamada **Adapt Watson**. Estos componentes actúan sobre la arquitectura de DeepQA con el objetivo de entender las preguntas, buscar respuestas candidatas y ponderar la confianza en dichas respuestas. El equipo documentó más de 8,000 experimentos independientes en el ciclo de vida de Watson, cada uno con 10 – 20GB de datos. A partir de estos datos se buscaban fallos y sus posibles causas. De esta forma se mejoró hasta un 85 % Precision@70.

2.3.2. Entendiendo las preguntas

Una pregunta tiene un tipo de respuesta potencial que llamaremos LAT (lexical answer type), además cada pregunta en *Jeopardy!* tiene una categoría que ayudará a identificar el tema sobre el que se pregunta. Para entender las preguntas formuladas se usa un parser ESG (English Slot Grammar) y un generador PAS (predicate-argument structure).

- ESG identifica partes de la frase y roles sintácticos como el sujeto y relaciones entre las diferentes partes de la frase.

- El generador PAS es usado para producir una representación abstracta, que representa la interfaz con la parte analítica (otra capa de DeepQA de la resolución del problema).

En cuanto a las **respuestas y fuentes de evidencia** al principio se usaba añadieron enciclopedias y libros de referencia. Después se desarrolló un proceso semiautomático para hacer crecer el conocimiento de Watson, no se puede añadir contenido sin parar ya que afecta al rendimiento del sistema. Tras analizar el contenido que se añade se crea PRISMATIC, que extrae información y usa la estadística para deducir lo que es conocimiento base, esto ayudará en la parte de generación de respuestas candidatas y calcular sus probabilidades.

Cada una de las consultas usan información estructurada y no estructurada usando varios mecanismos de búsqueda complementarios. Una pareja de consulta y respuesta representan una hipótesis. Una métrica que se usa para medir la bondad de una respuesta es el porcentaje de preguntas en las que dicha respuesta es generada como candidata. Esta medida ayuda a combinar los resultados, sobre todo cuando tenemos varias interpretaciones diferentes de lo que se está preguntando. De esta forma se determina la probabilidad final de que una respuesta sea correcta para la pregunta inicial. De las experiencias de los desarrolladores con el sistema PIQUANT sabían que no era una estrategia adecuada intentar anticipar todos los tipos de respuesta y crear algoritmos que busquen solamente instancias de dichas clases. Esto es debido en parte a la complejidad de algunas preguntas, ya que todo el sistema se apoya en haber reconocido correctamente el tipo de respuesta necesaria, y que en el conocimiento del sistema dicha respuesta esté bien clasificada. En la siguiente pregunta de la categoría *decoración* el tipo de respuesta buscada es una *dirección*:

Si usted está de pie, es la dirección que debe buscar para ver el revestimiento.

Respuesta: Abajo

3. Conclusiones

Referencias

- [1] IBM T.J. Watson Research Center. Ibm's piquant in trec2003.
- [2] IBM. Ibm mainframes. https://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_intro.html.