Text Mining application in Watson’s DeepQA

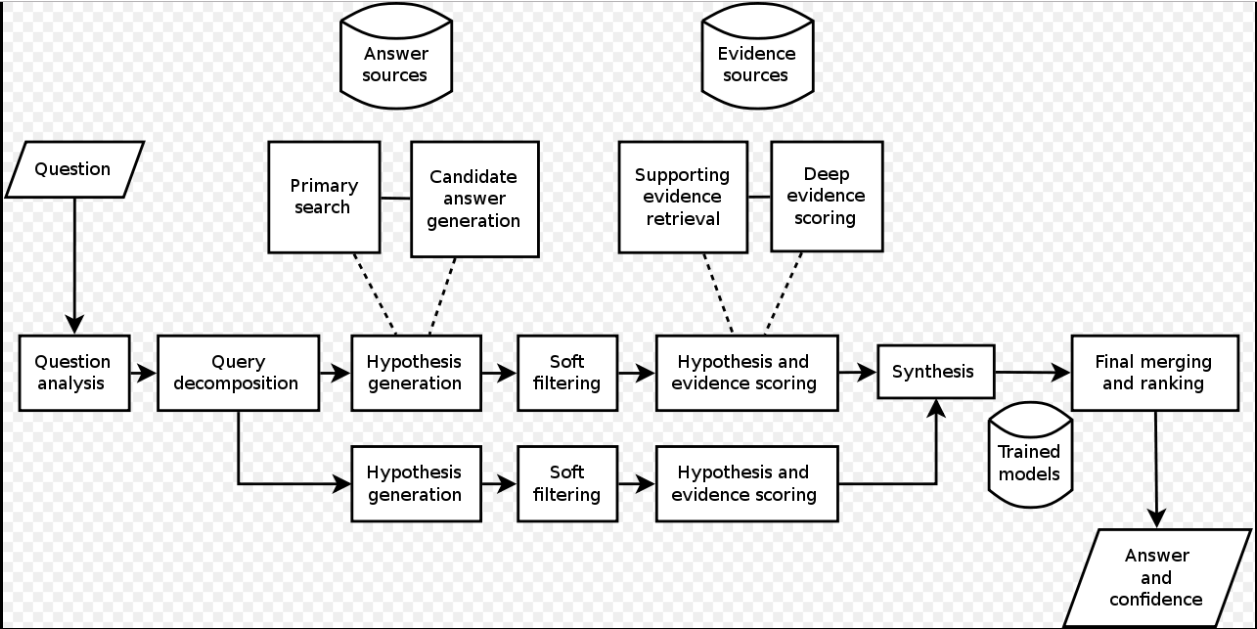
**Abstract**

DeepQA is a software architecture for deep content analysis and evidence-based reasoning that embodies that philosophy. It represents a powerful capability that uses advanced natural language processing, semantic analysis, information retrieval, automated reasoning and machine learning.

IBM Research undertook a challenge to build a computer system that could compete at the human champion level in real time on the American TV quiz show, Jeopardy. The extent of the challenge includes fielding a real-time automatic contestant on the show, not merely a laboratory exercise. The Jeopardy Challenge helped us address requirements that led to the design of the DeepQA architecture and the implementation of Watson. After three years of intense research and development by a core team of about 20 researchers, Watson is performing at human expert levels in terms of precision, confidence, and speed at the Jeopardy quiz show. Our results strongly suggest that DeepQA is an effective and extensible architecture that can be used as a foundation for combining, deploying, evaluating, and advancing a wide range of algorithmic techniques to rapidly advance the field of question answering (QA).

**IBM’s Solution**

The key difference between QA technology and document search is that document search takes a keyword query and returns a list of documents, ranked in order of relevance to the query (often based on popularity and page ranking), while QA technology takes a question expressed in natural language, seeks to understand it in much greater detail, and returns a precise answer to the question.



**Figure 1.1 Model and Techniques used by IBM**

When created, IBM stated that, "more than 100 different techniques are used to analyze natural language, identify sources, find and generate hypotheses, find and score evidence, and merge and rank hypotheses." Watson parses questions into different keywords and sentence fragments in order to find statistically related phrases. Watson's main innovation was not in the creation of a new algorithm for this operation but rather its ability to quickly execute hundreds of proven language analysis algorithms simultaneously. The more algorithms that find the same answer independently the more likely Watson is to be correct. Once Watson has a small number of potential solutions, it is able to check against its database to ascertain whether the solution makes sense or not.

Winning at Jeopardy requires accurately computing confidence in your answers. The questions and content are ambiguous and noisy and none of the individual algorithms are perfect. Therefore, each component must produce a confidence in its output, and individual component confidences must be combined to compute the overall confidence of the final answer. The final confidence is used to determine whether the computer system should risk choosing to answer at all. In Jeopardy parlance, this confidence is used to determine whether the computer will “ring in” or “buzz in” for a question. The confidence must be computed during the time the question is read and before the opportunity to buzz in. This is roughly between 1 and 6 seconds with an average around 3 seconds. Confidence estimation is very critical in shaping overall approach in DeepQA. There is no expectation that any component in the system does a perfect job — all components post features of the computation and associated confidences, and we use a hierarchical machine-learning method to combine all these features and decide whether or not there is enough confidence in the final answer to attempt to buzz in and risk getting the question wrong.

In recent years, the Watson capabilities have been extended and the way in which Watson works has been changed to take advantage of new deployment models (Watson on IBM Cloud) and evolved machine learning capabilities and optimized hardware available to developers and researchers. It is no longer purely a question answering (QA) computing system designed from Q&A pairs but can now 'see', 'hear', 'read', 'talk', 'taste', 'interpret', 'learn' and 'recommend'.

**Comparison with human players**

Watson's basic working principle is to parse keywords in a clue while searching for related terms as responses. This gives Watson some advantages and disadvantages compared with human Jeopardy! players. Watson has deficiencies in understanding the contexts of the clues. As a result, human players usually generate responses faster than Watson, especially to short clues. Watson's programming prevents it from using the popular tactic of buzzing before it is sure of its response. Watson has consistently better reaction time on the buzzer once it has generated a response, and is immune to human players' psychological tactics, such as jumping between categories on every clue.

In a sequence of 20 mock games of Jeopardy, human participants were able to use the average six to seven seconds that Watson needed to hear the clue and decide whether to signal for responding. During that time, Watson also must evaluate the response and determine whether it is sufficiently confident in the result to signal. Part of the system used to win the Jeopardy! contest was the electronic circuitry that receives the "ready" signal and then examined whether Watson's confidence level was great enough to activate the buzzer. Given the speed of this circuitry compared to the speed of human reaction times, Watson's reaction time was faster than the human contestants except when the human anticipated (instead of reacted to) the ready signal. After signaling, Watson speaks with an electronic voice and gives the responses in Jeopardy’s question format. Watson's voice was synthesized from recordings that actor Jeff Woodman made for an IBM text-to-speech program in 2004.

**Suggestions and Improvements**

Domain-specific taxonomic reasoning can be used to evidence correct hypothesis via: concept matching between question and evidence passages, type coercion of answers given the desired answer type, identifying specificity of answer, and equivalent answer merging. Concept term matching: The synonymy and hyponymy encoded in taxonomies may be directly used to enhance term matching within DeepQA. Term matching is used by the DeepQA passage scorers, which attempt to justify hypotheses using unstructured content. DeepQA uses an ensemble of passage scorers with different precision/recall tradeoffs.