

brmson (YodaQA)

A DeepQA-style Question Answering Pipeline

Petr Baudiš [⟨pasky@ucw.cz⟩](mailto:pasky@ucw.cz)

FEE CTU Prague; brmlab hackerspace

European Fall 2015

Who here knows IBM Watson?



Won over two human Jeopardy! champions in 2011.
IBM's flagship in "Cognitive Computing".

brmson: Question Answering



A Question Answering system inspired by **IBM Watson**
and its DeepQA pipeline architecture.
A bit Do It Yourself style, but serious effort!

What questions do we look at?

Hi!

What's the time?

Do *you* dream of electric sheep?

Can you make me a program that prints all primes?

Can entropy ever be reversed?

How do you work?

誰があなたを作成しましたか？

What's the highest mountain in the world?

Only **knowledge** (“trivia”, “factoid”) questions.

Where to get the answer?

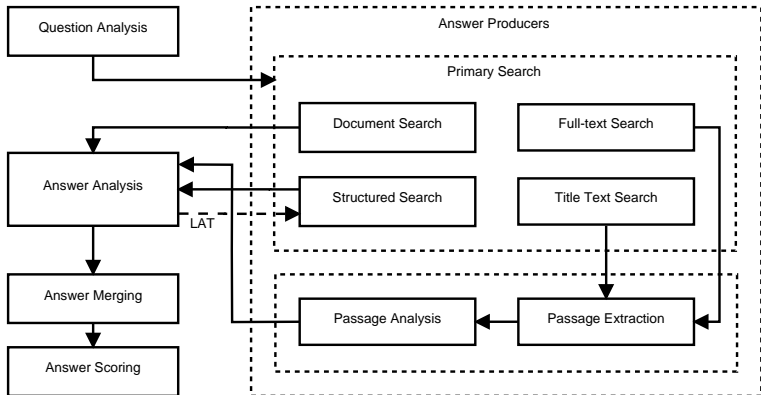
Unstructured knowledge bases (Wikipedia):

- Information Retrieval problem (fulltext search)
- Information Extraction problem
- Type checking

Structured knowledge bases (linked data):

- DBpedia, Freebase
- SPARQL query on RDF store

How Does It Work?

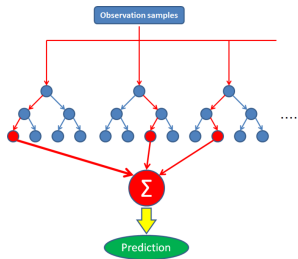
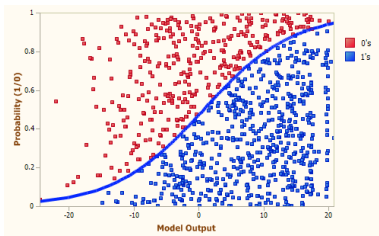


How Does It Work!

- Question is analyzed, **clues** and **lexical answer type** extracted
- Fulltext search for clues (we use English Wikipedia)
- *Hundreds* of candidate answers are generated from matching passages, introduction passages and document titles
- Candidate answers are **scored** based on various features
- Important features: **Type coercion**
 - “Is the answer a color?”
 - “Is the answer an inventor, or at least a person?”
- Top scored answer is yielded

Machine Learning Basics

- **Gold standard** (training / testing set) — few hundred questions with correct answers
- Each answer is decorated by many features
- Logistic Regression: We look for the right combination of feature weights
- Decision Forest: Many decision trees for specific feature combinations



brmson: YodaQA Implementation

- **YodaQA:** “Yet anOther Deep Answering pipeline”
- Designed and implemented from scratch
- Java, UIMA framework
- Architecture based on simplified **IBM DeepQA** (as published)
- NLP analysis: Third-party UIMA annotators via **DKPro**
- **Open Source!** Everything is on github.com/brmson, including documentation
- **Looking for contributors, collaborators, commercial ideas...**

Current State

Current performance (TREC):

32.6% accuracy-at-one

79.3% recall

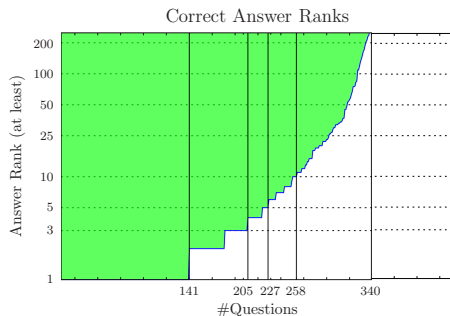
Current performance (Movies):

45% accuracy-at-one

75.5% recall

Work in progress:

Advanced semantic methods,
multi-constraint questions.



Conclusion

- Practical, open source QA system
- Clean architecture, very modular system
- Reasonably documented!
- Long term:
 - Closed domain QA with powerful user interface
 - Bleeding edge NLP research (PhD)
 - Startup aims

pasky@ucw.cz

petr.baudis@gmail.com

Thank you for your attention!