

# YodaQA on BioASQ

## Prototype Notes

Petr Baudiš `<baudipet@fel.cvut.cz.cz>`  
Jan Šedivý

3C Group, FEE CTU Prague

CLEF 2015

# YodaQA

A Question Answering system inspired by **IBM Watson** and its DeepQA pipeline architecture.

**Primary goals:**

- Practicality
- Extensible design
- Academic reusability
- Minimum hand-crafted rules

**Current status:** Open-domain factoid questions (TREC QA), replicating the DeepQA scheme with 80% answer production recall, 35% prec@1, 0.44 MRR.

# YodaQA on BioASQ

**BioASQ Goal:** Measuring performance of open-domain QA with only minimal domain adaptation.

**Phase B Only:** No IR performed, input is the question + relevant snippets. We consider exact answers only.

*Which genes are mutated in Gray platelet syndrome patients?*

*The genetic defect responsible for gray platelet syndrome was recently identified in biallelic mutations in the **NBEAL2** gene.*

*Linkage analysis revealed a 63 cM region on the X chromosome between markers G10578 and DXS6797, which segregated with the platelet phenotype and included the **GATA1** gene.*

*We identified a family with gray platelet syndrome (GPS) segregating as a sex-linked trait.*

# Question Analysis

- Full dependency parse
- **Focus** generation (hand-crafted dependency, pos rules)
  - What was the first **book** written by Terry Pratchett?
  - The **actor** starring in Moon?
  - **BioASQ Specific**: Deal with imperatives (“Name the gene that ...”)
- **LAT** (Lexical Answer Type) generation (from focus)
  - **Where** is Mount Olympus? **location**
- **Clues** (search keywords, keyphrases) generation:
  - POS and constituent token whitelist
  - Named entities
  - Focus and the NSUBJ constituent
  - **Concepts**: enwiki article titles (entity linking)

**Outcome:** Question representation

# Full-text Answer Production

Given a passage that might bear an answer ...

- Full dependency parse
- All **noun phrases** are candidate answers
- Stock OpenNLP NER: All **named entities** are candidate answers
- **CRF chunker** for generating candidate answers that includes q/a parse tree alignment features (inspired by Jacana)

**Outcome:** Set of candidate answers

# Answer Analysis

- Each answer is POS-tagged and has dependency tree, Focus generated (dependency root)
- **LAT generation** — named entity type, DBpedia concept type, WordNet instance-of relation, rule for CD POS
  - **BioASQ Specific** — GeneOntology answer search
- **Type coercion** of question + answer LAT: *Unspecificity* is path length in the **WordNet** (*hypernymy, hyponymy*) graph
- Answer features (help determine trustworthiness) for:
  - Clue overlaps, in-sentence proximity
  - Generated LATs, type coercion
  - **81 features** in total  
(some depend on IR, unused in BioASQ)
- Logistic regression generates answer confidences

**Outcome:** Ordered set of Answers

# Question Types

- **Factoid:** Return top-scored answer
- **List:** Return top five answers
- **Yes/no:** Not done; always return yes (strong bias in dataset)
- **Summary:** Not done
  
- Only exact answers; ideal answers not generated

# Open Domain Benchmarks

## TREC2001, 2002 test:

System	Precision@1	F <sub>1</sub>	MRR
LLCpass03 (hand-crafted system)	68.5%	—	—
OpenEphyra (hand-crafted OSS)	“above 25%”	—	—
JacanaIR (modern fully-learned OSS)	—	23.1%	—
<b>YodaQA v1.1</b>	26.4%	26.4%	0.325

## Benchmark results on the **WebQuestions** test:

System	F <sub>1</sub> @1	F <sub>1</sub> (Berant)
Sempre	35.7%	35.7%
JacanaFB	35.4%	33.0%
<b>YodaQA v1.1</b>	34.3%	—
STAGG (summer 2015, state-of-art)	—	52.5%

# BioASQ Performance

**Hold-out** experiments (on a split of the official train data):

Pipeline	AP Rcl.	Prec@1	MRR
<b>BioASQ final</b>	33.0%	10.0%	0.132
w/o G.O.	33.0%	8.0%	0.120
w/o G.O., CRF	33.0%	5.5%	0.114
w/o yes/no q.	43.5%	10.1%	0.148

**Official** results (test batch #5):

- **Factoid MRR:** 3/8 (out of unique systems)  
oaa 0.27; fdu 0.25; YodaQA **0.20**; fa1 0.15; ...
- **List F1:** 2/7 (out of unique systems)  
oaa 0.19; YodaQA **0.16**; fdu 0.134; ...

# Conclusion

- **YodaQA:** “Yet anOther Deep Answering pipeline”
- Designed and implemented from scratch
- Java, UIMA framework
- NLP analysis: Third-party UIMA annotators via **DKPro**
  
- Open-domain systems can exhibit good **BioASQ performance** even with little domain adaptation
  
- **Open Source!** Everything is on [github.com/brmson](https://github.com/brmson), including documentation
- **Looking for contributors, collaborators, commercial ideas...**

baudipet@fel.cvut.cz

**Thank you for your attention!**