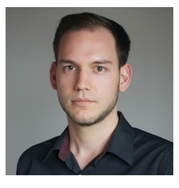
End-to-End Non-Factoid Question Answering with an Interactive Visualization of Neural Attention Weights




Andreas Rücklé and Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP)
Department of Computer Science, Technische Universität Darmstadt
https://www.ukp.tu-darmstadt.de

1 Summary

A modular, extensible framework to visualize and compare attention mechanisms in (non-factoid) question answering

Non-Factoid Question Answering

- Questions that do *not necessarily* ask about facts; can often be answered with opinions, experiences, descriptions...
- We search for *existing* content on the web that can answer the question (e.g. by searching in CQA platforms)

Task: Answer Selection

- Given a question and a set of candidate answers, we rank the candidates for relevancy according to the question
- Attention-based models achieve state-of-the-art results

Challenge

- Understanding strengths and weaknesses of particular attention mechanisms is important
- Researchers usually plot attention weights for a number of predefined Q/A pairs within a dataset
- This is not interactive and makes the direct comparison of different approaches time-consuming

Our Contributions

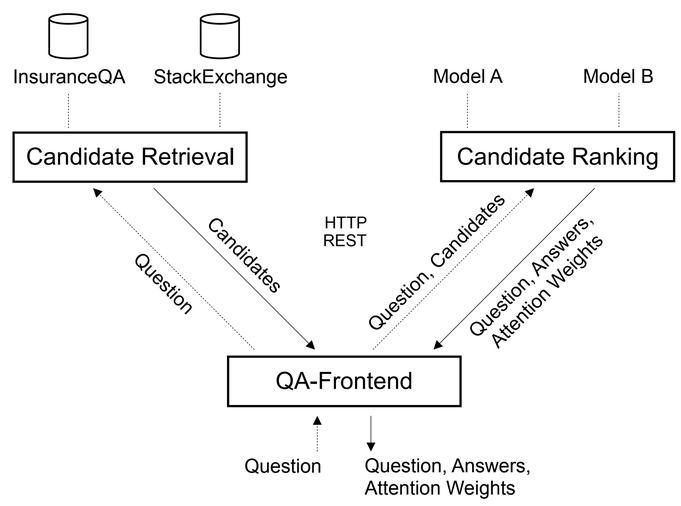
1. We present a framework for the **interactive exploration** of different attention-based models
   - By transforming models to end-to-end QA systems
   - Supports one-way-, two-way-, and self-attention
2. Our UI allows **comparing** different approaches side by side

Software Release

- The source code of our framework is **publicly available on GitHub**
- Includes a short documentation and an API reference

https://github.com/UKPLab/acl2017-non-factoid-qa

2 System Architecture



3 Candidate Retrieval

Approach

- We use ElasticSearch to index all answers of a dataset
- For an input question we query this index to obtain a set of candidate answers

Extensibility

- Our service implementation can easily be extended with new datasets through public interfaces
- We include readers for InsuranceQA and StackExchange

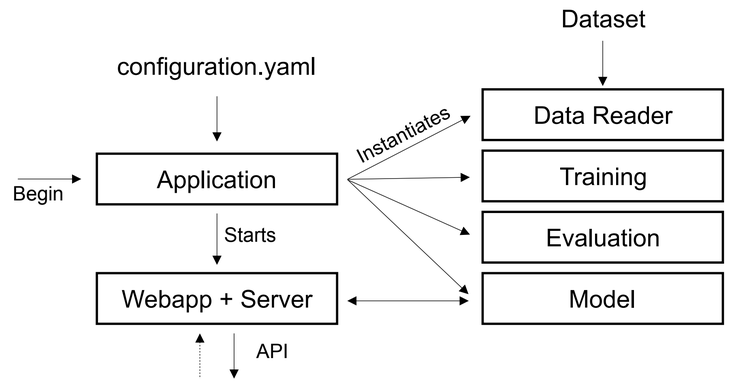
4 Candidate Ranking

Approach

- We re-rank all candidates with (attention-based) NNs
- The results include attention weights for *every Q/A pair*
  → Question attention can be different for each answer

Extensibility

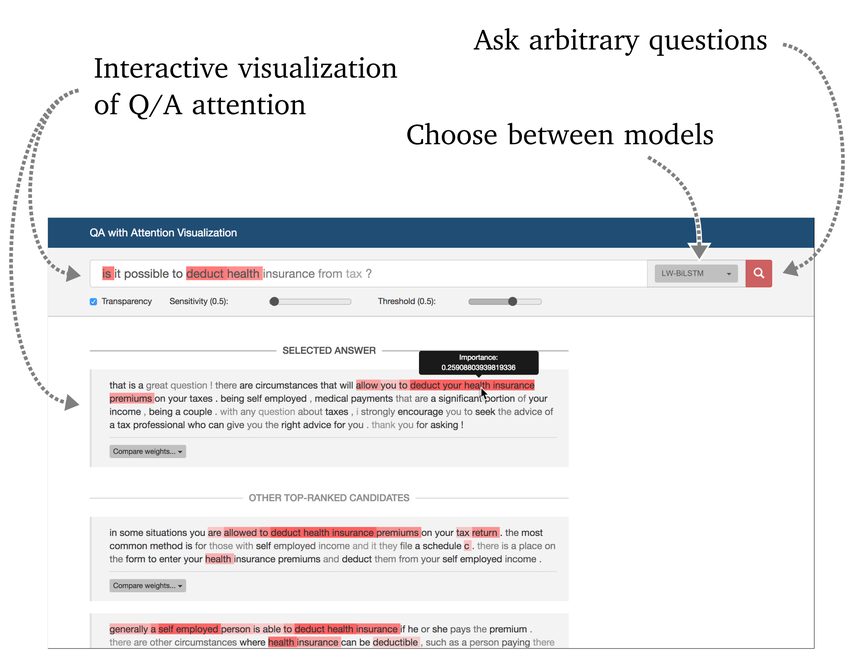
- The service implementation includes an extensible answer selection framework based on TensorFlow
- All components can freely be swapped and combined (dataset, model, training, evaluation)
- The framework is highly configurable through external YAML files (see the example on the right)

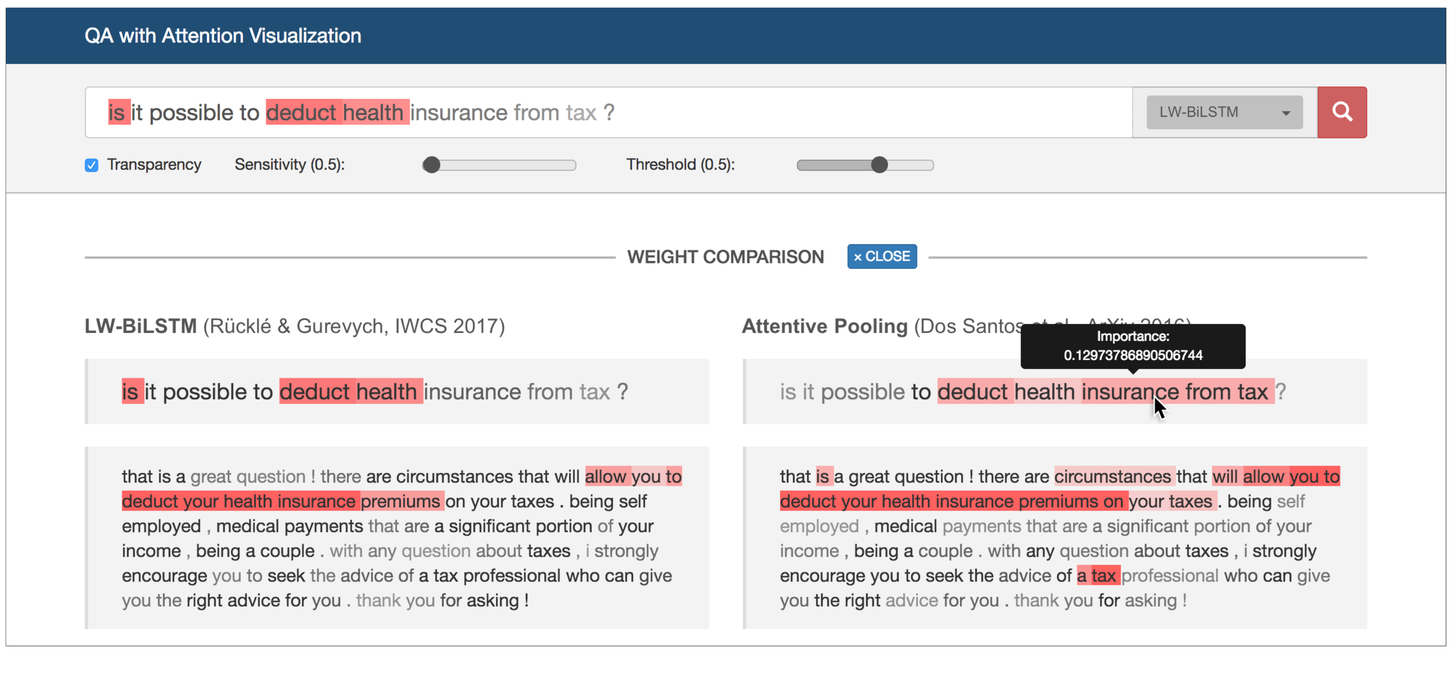


YAML Configuration:

```yaml
data-module: data.insuranceqa.v2
model-module: model.ap_lstm
training-module: training.dynamic
evaluation-module: evaluation.default

data:
  embeddings: data/glove.6B.100d.txt
  insuranceqa: data/insuranceQA
  lowercased: true
  ⋮

model:
  lstm_cell_size: 141
  margin: 0.2
  trainable_embeddings: true
  ⋮

training:
  batchsize: 20
  epochs: 100
  save_folder: checkpoints/ap_lstm
  dropout: 0.3
  optimizer: adam
  initial_learning_rate: 0.001
  scorer: accuracy
  ⋮

evaluation:
  skip: true
```

5 QA-Frontend





This work has been supported by the German Research Foundation as part of the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1).
We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.