Function Assistant: A Tool for NL Querying of APIs
Kyle Richardson and Jonas Kuhn
University of Stuttgart, Germany
kyle@ims.uni-stuttgart.de

Overview: Towards Automated Question-Answering on Software Libraries (APIs)

- Source Libraries: Collections of code with developer documentation (red), sometimes hard to navigate and find functionality:
  - Parallel Data: (Short text description, Function signature)

- API QA: Retrieving code components, e.g., function signatures, from high-level natural language descriptions and queries.
  - Query: How do I create a dependency arc? → target function(s).

- Function Assistant: End-to-end dataset and QA builder for arbitrary APIs, works by training translation model on parallel data.

Dataset Extractor and Translation Pipeline

- Function Assistant Pipeline: Raw (Python) Source Code Library → Parallel Dataset → SMT model → Query Server.
- SMT model has two components (Richardson and Kuhn 2017):
  - Word SMT Model: Generates candidate function signatures from input, uses simple decoding strategy.
  - Discriminative Model: Ranks translation candidates using additional word, phrase and API-level features.
- Implementation: Py/Cython, dependency-injection design, customizable, can be used to implement new models or as baseline.

Query Server Illustration

- API QA: Takes a natural language query, runs through trained SMT model, returns reranked list as possible answers to query, provides pointers to implementation.

Synthetic QA Experiments and Baseline Results

- QA Evaluation: Hard to find real queries, using normal train/test split, treat held out data as synthetic user queries (Deng and Chrupała 2014, Richardson and Kuhn 2017)
  - Data Extractor: API → train set, dev. set, test set (= synthetic queries)

- Measure Accuracy @1, Accuracy @10, MRR, results conform to previous findings.

Experiment: PY27 Dataset

Python

- Ran our pipeline on 27 popular Python open-source projects from github.com, selection of Awesome Python list.
- Goal: Test robustness of extractor, provide new resource for system development and baseline results.

References and Info

- GPL-licensed, data and code can be found via github: https://github.com/yakazimir/Code-Datasets
- Code retrieval prototype: zubr.ims.uni-stuttgart.de
- Supported by the German Research Foundation (DFG), project D2 of SFB 732.

Huijing Deng and Grzegorz Chrupała. 2014. Semantic Approaches to Software Components Retrieval with English Queries. Proceedings of LREC.