NLP2API: Query Reformulation for Code Search using Crowdsourced Knowledge and Extra-Large Data Analytics

Mohammad Masudur Rahman Chanchal K. Roy
Department of Computer Science, University of Saskatchewan, Canada
{masud.rahman, chanchal.roy}@usask.ca

***Abstract*—Software developers frequently issue generic natural language (NL) queries for code search. Unfortunately, such queries often do not lead to any relevant results with contemporary code (or web) search engines due to vocabulary mismatch problems. In our technical research paper (accepted at ICSME 2018), we propose a technique–NLP2API–that reformulates such NL queries using crowdsourced knowledge and extra-large data analytics derived from Stack Overflow Q & A site. In this paper, we discuss all the artifacts produced by our work, and provide necessary details for downloading and verifying them.**

I. NLP2API: Artifact Details

Our technique, NLP2API [2], takes a generic natural language query as an input and then reformulates the query for code search using appropriate API classes mined from Stack Overflow. Our algorithm design and empirical evaluation of the technique produced different sets of artifacts. These artifacts can be divided into three major groups as follows:

A. Algorithm Design and Tool Implementation

- **`nlp2api-runner`** is our prototype. It takes a generic natural language query as input and returns a ranked list of Java API classes relevant to the query.
- **`README`** provides detailed instructions and example commands for running and evaluating our prototype. A screenshot of the prototype's run is also attached.
- **`data/`** contains Java programming keywords and stop words used for the natural language pre-processing.
- **`dataset/`** contains texts and code segments from 656K Stack Overflow Q & A threads, and the *Lucene index* developed from them. Compressed files should be decompressed in the same directory.
- **`fastText/`** contains our implementation of `fastText` [1], and our skip-gram model trained on the Stack Overflow Q & A threads. It provides word-embeddings for both a natural language keyword and an API class which are then used to determine their semantic proximity.
- **`jdk-fasttext-checker`** contains necessary commands to check `fastText` and JDK 8 installations.
- **`LICENSE`** outlines the licensing details of our replication package and experimental data.

B. Tool Evaluation: API Suggestion Performance

- **`NL-Query+GroundTruth`** contains 310 natural language queries and ground truth Java API classes for them. They are carefully collected from the Q & A threads of four tutorial sites–*KodeJava*, *JavaDB*, *Java2s* and *CodeJava*. From each Q & A thread, the question title is captured as a query, the code example is considered as a ground truth for code search, and the API classes discussed in the accompanying prose are captured as the ground truth API classes [2]. **`oracle-310`** is a duplicate file required for the prototype's internal use.
- **`candidate/`** contains the candidate API classes for 310 NL queries before their ranking with the heuristics.
- **`NLP2API-Results-Borda`** contains the API classes suggested by our technique when only *Borda score* heuristic is employed.
- **`NLP2API-Results-Q-A-Proximity`** contains the API classes suggested by our technique when only *semantic proximity* heuristic is employed.
- **`NLP2API-Results`** contains the API classes for 310 NL queries suggested by our technique.

C. Tool Evaluation: Query Reformulation Performance

- **`code-ext-index`** is the corpus containing 4,170 code segments including the ground truth. The corpus is indexed and used by *Lucene* for performing code search.
- **`search-engine-resx`** contains original search results and our improved results for 310 NL queries.

II. Implications of our Artifacts

- **Benchmarking:** Our dataset and results can be used as the *benchmark* for future tools and techniques.
- **Reproducibility & Kick-starting:** Our prototype and intermediate data offer replication and possible reproducibility. Our source code can kick-start the next tool.
- **Reusability:** Our skip-gram models, 1.3 million Q & A threads and code segments from Stack Overflow can be reused for various other purposes (e.g., code completion).

III. NLP2API: Download Link

The artifacts are uploaded in a Google Drive, and can be downloaded from: **`https://goo.gl/Meujmx`**

References

[1] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.
[2] M. M. Rahman and C. K. Roy. Effective reformulation of query for code search using crowdsourced knowledge and extra-large data analytics. In *Proc. ICSME*, page 12, 2018.