

PROBLEM

Current methods for selecting values for parameters used to instantiate Information Retrieval tools for concept location either rely on weak analogies between code and natural language or lack empirical justification.

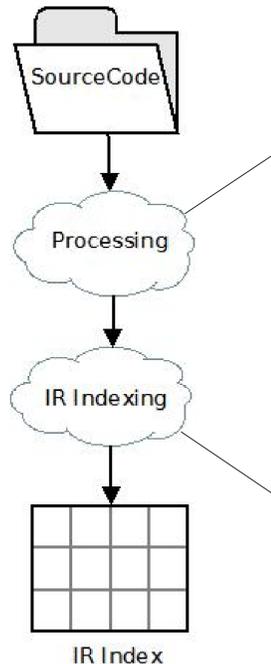
BACKGROUND

How is Information Retrieval used for Concept Location?

- Source code is processed in one of many different ways.
- Processed source code is indexed by one of several IR engines.
- A developer formulates a query from a change request and searches the index.
- Once a starting point for the change is found, Concept Location is over.

How Are Parameter Values Chosen?

- Natural language “default” values.
 - Makes assumptions about code that were made for natural language. (e.g., vocabulary size)
- Use intuition about how parameter effect results.
 - Not backed by empirical evidence.



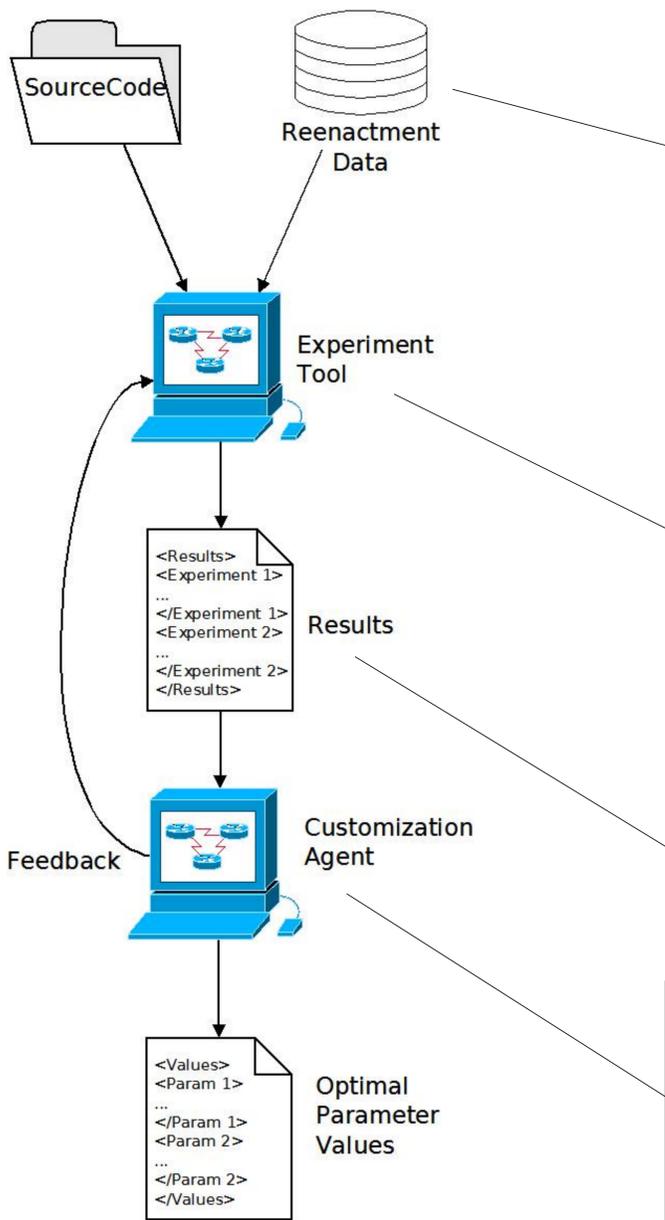
Examples of Processing Parameters

- Remove comments before indexing?
- Filter Numbers?
- Split Identifiers?
(e.g., “numObjects” becomes “num Objects”)
- If identifiers split, keep originals as well?

Examples of Indexing Parameters

- Which document granularity to use (e.g., method or class granularity?)
- Which IR engine to use for indexing and searching?
- Engine specific parameters such as:
 - LSI dimensionality reduction factor.
 - Which similarity metric to use.

OUR PROPOSED APPROACH: THE IR-TUNING TOOL SET



Reenactment Benchmark Data

- Correlates completed change requests with a “Gold Set” of source code artifacts modified in the changes.
- Stored in a relational database for quick and easy access.
- Useful outside of the parameter tuning context to compare novel IR-based concept location techniques.

Experiment Tool

- Uses the reenactment data to run numerous Concept Location experiments.
- Instantiates various IR-based search tools using a variety of parameter values from an initial configuration file.
- Extracts natural language description of change requests from the reenactment benchmark data to use as a query.

Results from the Experiment Tool

- Measure effectiveness and approximate effort by recording the position of the highest ranking member of the “Gold Set” in the search results.
- The effectiveness data that results from all the queries in an experiment can be combined to create a benchmark effectiveness score for that instantiation of the IR-based tools.

Customization Agent

- Too many parameter combinations for brute force.
- Use Customization Agent for feedback on small parameter changes.
- Large effectiveness variance → change parameter at finer level.
- No significant variance → parameter should not change much.
- Optimal parameters emerge after stability is reached.

A process model of the IR-Tuning Tool Set.

ACKNOWLEDGEMENTS AND THANKS

We would like to thank Katrina Ohlemacher for her assistance with visual design and Sonia Haiduc for her technical assistance. This work was supported in part by grants from the US National Science Foundation: CCF-1017263 and CCF-0845706.