Similarity of Programming Languages

C/C++, C#, Go and Python

https://bit.ly/3GKybaq

Group 03

Kevin Innerebner - Tokenizer, Model Training, Visualization
Philip Loibl - Codefetcher
Manuell Schöller - Plots, Codefetcher
Tobias Wolf - Merging datasets, Analysis

TU Graz - Advanced Information Retrieval WS 22/23

Introduction / Motivation

- We like different languages for different reasons and for various tasks
- Use what we learned to see how “similar” they are
- We expected C++ and Go to be most similar, followed by C# and then Python
- Only a superficial analysis of semantics
- No regard for actual syntax, deeper keyword meaning, paradigms etc

Dataset and IR Methods

- Need complete, valid code files for language parsing using ANTLR4
- Github Rest API for random code acquisition; ~2000 files per language
- Tokenize code using four ANTLR4 parsers
  - Replace variables, literals with fixed tokens for better semantics
- Train a single FastText (W2V extension) model supporting all languages with Gensim
  - Prefix some identical keywords with language to learn different semantics
- 2D Representation of language tokens of multidimensional vectors (100) using UMAP

Results & Analysis

✅ C++, Go and C# indeed similar

✅ Python noticeably an outlier

❌ Cannot generate syntactically correct code

❌ Large dataset of good data is time consuming

- Core set of tokens appear in most files (import, return, …)
- Vocabulary shrinks very fast when increasing min_count

Conclusion

- Output is only as good as the input (i.e. the dataset)
- Syntax similarities
- Superficial semantic similarities possible
- Clear difference between static and dynamically typed languages
  - Domain might also have impact e.g. Python and Data Science
- Word2Vec / FastText is not good at code completion (as to be expected)