

Evaluation

Induction of probabilistic context-free grammars from tree corpora

Praktikum Parsing von natürlichen Sprachen

Tobias Denking

2019-04-30

Abgaben

#	Sprache	LOC	runtime ¹
1	Java	587	1.591 s
2	Python	289	9.463 s
3	Rust	349	0.139 s
R	Rust	474	0.206 s

¹auf meinem Laptop, mit einem Korpus von 2262 Bäumen

Häufige Mängel

- Ausgabe im falschen Format
→ Aufgabenstellung beachten

Häufige Mängel

- Ausgabe im falschen Format
 - Aufgabenstellung beachten
- Parser für S-Expressions nicht robust genug
 - Probleme bei zusätzlichen Leerzeichen, < und .
 - Parsingbibliothek verwenden
 - Randfälle testen

Häufige Mängel

- Ausgabe im falschen Format
 - Aufgabenstellung beachten
- Parser für S-Expressions nicht robust genug
 - Probleme bei zusätzlichen Leerzeichen, < und .
 - Parsingbibliothek verwenden
 - Randfälle testen
- Corpusartefakte im Programm
 - Nichtterminalsymbol ROOT taucht im Programmcode auf
 - Tools sollten nicht unnötig corpuspezifisch sein

Häufige Mängel

- Ausgabe im falschen Format
 - Aufgabenstellung beachten
- Parser für S-Expressions nicht robust genug
 - Probleme bei zusätzlichen Leerzeichen, < und .
 - Parsingbibliothek verwenden
 - Randfälle testen
- Corpusartefakte im Programm
 - Nichtterminalsymbol `ROOT` taucht im Programmcode auf
 - Tools sollten nicht unnötig corpuspezifisch sein
- falsches Verwenden unnötiger Bibliotheken

Häufige Mängel

- Ausgabe im falschen Format
 - Aufgabenstellung beachten
- Parser für S-Expressions nicht robust genug
 - Probleme bei zusätzlichen Leerzeichen, < und .
 - Parsingbibliothek verwenden
 - Randfälle testen
- Corpusartefakte im Programm
 - Nichtterminalsymbol ROOT taucht im Programmcode auf
 - Tools sollten nicht unnötig corpuspezifisch sein
- falsches Verwenden unnötiger Bibliotheken
- ungewöhnlicher Umgang mit Fehlern
 - Fehler dem Nutzer signalisieren
 - auf `stderr`

Häufige Mängel

- Ausgabe im falschen Format
 - Aufgabenstellung beachten
- Parser für S-Expressions nicht robust genug
 - Probleme bei zusätzlichen Leerzeichen, < und .
 - Parsingbibliothek verwenden
 - Randfälle testen
- Corpusartefakte im Programm
 - Nichtterminalsymbol ROOT taucht im Programmcode auf
 - Tools sollten nicht unnötig corpuspezifisch sein
- falsches Verwenden unnötiger Bibliotheken
- ungewöhnlicher Umgang mit Fehlern
 - Fehler dem Nutzer signalisieren
 - auf `stderr`
- unnötiges IO
 - entfernen

Allgemeine Tipps

- auf Robustheit achten
 - Unit-Tests (insbesondere von Randfällen)
 - Integration-Tests mit mehreren Datensätzen
 - Buildskripte testen (z.B. auf anderem Computer oder VM)

Allgemeine Tipps

- auf Robustheit achten
 - Unit-Tests (insbesondere von Randfällen)
 - Integration-Tests mit mehreren Datensätzen
 - Buildskripte testen (z.B. auf anderem Computer oder VM)
- Bibliotheken nutzen
 - für was gebraucht
 - beste Bibliothek finden
 - vorhandene Funktionen ausgiebig recherchieren

Allgemeine Tipps

- auf Robustheit achten
 - Unit-Tests (insbesondere von Randfällen)
 - Integration-Tests mit mehreren Datensätzen
 - Buildskripte testen (z.B. auf anderem Computer oder VM)
- Bibliotheken nutzen
 - für was gebraucht
 - beste Bibliothek finden
 - vorhandene Funktionen ausgiebig recherchieren
- Code soll lesbar sein
 - Hochsprachen sollen *Menschen* das Lesen von Code ermöglichen.
 - Code sollte das *Problem reflektieren*.
 - Code sollte *minimal* sein.