Abstract

In this thesis an implementation of an n=gram based language model using the Modified Kneser=Ney smoothing algorithm with the open source Apache Spark large scale data processing engine and the open source document oriented database MongoDB will be presented. This language model will then be used to generate sentences from input sets of keywords.

Contents

Chapter 1

Introduction

Chapter 2

Fundamentals and State of The Art

2.1 State of the Art

2.1.1 TODO some papers

2.2 Interaction Diagrams

2.3 Scenario Testing

Scenario testing, was originally introduced in Kaner [Kan03] and later as Kaner [Kan13]. The author defines scenarios as hypothetical stories, which aid a person in understanding a complex system or problem. Scenario tests are tests, which are based on such scenarios. [Kan13, p. 1] Further, [Kan03, pp. 2–5] defines five characteristics, which make up a good scenario test as follows: A Scenario test must be

- based on a story - based on a description of how the program is being used
- motivating - stakeholders have interest in this test succeeding and would see to it's resolution
- credible - probable to happen in the real world
- complex - complex use, data or environment
- easy to evaluate - it should be easy to tell if the test succeded or failed based on the results

Kaner [Kan13] describes the biggest advantages of scenario testing to be - understanding and learning the product in early stages of development(1), connecting of testing and requirement documentations(2), exposing shortcomings in delivering of desired benefits(3), exploration of expert use of the program(4), expose requirement related issues(5).

2.4 Behavior-Driven Development

Behavior-Driven Development (BDD), pioneered by North [Nor06] is a software development process , that combines principles from Test-Driven Development and Domain-Driven design [EE04].

Its main goal is to specify a system in terms of its functionality (i.e. it's behaviors) with a simple domain-specific language (DSL) making use of English-like sentences. This stimulates collaboration between developers and non-technical stakeholders and further results in a closer connection between acceptance criteria for a given function and matching tests used for its validation.

BDD splits a user story into multiple scenarios, each formulated in the form of *Given*, *When*, *Then* statements, respectively specifying the prerequisite/context, event and outcomes of a scenario.

[TODO] example here

At present ... there based on the division of behavior descriptions and behaviors. Such as Jest/Jasmine combine behavior descriptions and behaviors into one, whereas as Cucumber uses a DSL named Gherkin to specify the behavior descriptions and provides a set of tools to generate behaviors.

2.4.1 the other one(forgot name)

splitting into ...

2.4.2 Gherkin Language

2.5 Vue.js

structure etc. template part code part bindings

2.6 ESLint parser

ESLint [21] is a static code analysis tool for fixing of common code issues and enforcing of consistent code style for ECMAScript/JavaScript with the aim of improving code quality and avoiding bugs.

ESLint uses *rules* to report and fix discovered errors. Rules are on demand plugable extensions, which

... architecture ... rules

.. parse estree and

... vue es lint

ESLINT-vue ESLint is a tool for identifying and reporting on patterns found in ECMAScript/-JavaScript code, with the goal of making code more consistent and avoiding bugs. In many ways, it is similar to JSLint and JSHint with a few exceptions:

ESLint uses Espree for JavaScript parsing. ESLint uses an AST to evaluate patterns in code. ESLint is completely pluggable, every single rule is a plugin and you can add more at runtime.

selectors visitor pattern linting AST reference

So what if there is text here?

Appendix A

Appendix

List of Figures

List of Tables

Bibliography

[Kan03] Cem Kaner. “The Power of’What If... and Nine Ways to Fuel Your Imagination: Cem Kaner on Scenario Testing”. In: *Software Testing and Quality Engineering* 5 (2003), pp. 16–22.

[Kan13] Cem Kaner. “An introduction to scenario testing”. In: *Florida Institute of Technology, Melbourne* (2013), pp. 1–13.

[Nor06] Dan North. “Behavior modification”. In: *Better Software* 8.3 (2006), p. 26. URL: https://dannorth.net/introducing-bdd/.

[EE04] Eric J Evans and Eric Evans. *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley Professional, 2004.

[21] *ESLint - Pluggable JavaScript linter*. [Online; accessed 25. Jan. 2021]. Jan. 2021. URL: https://eslint.org.

[esl21] eslint. *espree*. [Online; accessed 25. Jan. 2021]. Jan. 2021. URL: https://github.com/eslint/espree.