Chatbot Prototype for Software Design Pattern Determination

Degree programme : BSc in Computer Science | Specialisation : Digital Business Systems
Thesis advisor : Prof. Rolf Jufer
Expert : Andreas Dürsteler

Chatbots are more relevant than ever, not only in the field of customer service, but also in common everyday situations. Within this project a chatbot was developed that is able to assist users in finding a software design pattern that fits their needs. It is able to show that chatbots can be used to process dynamic user-input and deliver useful results.

Xavier Lucien Frei

Initial situation

Previous work included research about types of chatbots (rule-based vs. corpus-based), evaluation of frameworks that can be used to develop a chatbot and the implementation of a small rule-based scenario with the use of online-frameworks. The results gathered were used as basis for this project.

Goal

The goal of this bachelor thesis was to develop a corpus-based chatbot that is able to assist the user in finding a software design pattern that may fit their needs. Furthermore, it is supposed to have the ability to process the users word-input in combination with a corpus of data about software design patterns and create Word2Vec models out of them. With the help of such models the bot can provide the users with recommendations of software design patterns for their needs and an estimate of how well they could fit.

Implementation & Technologies

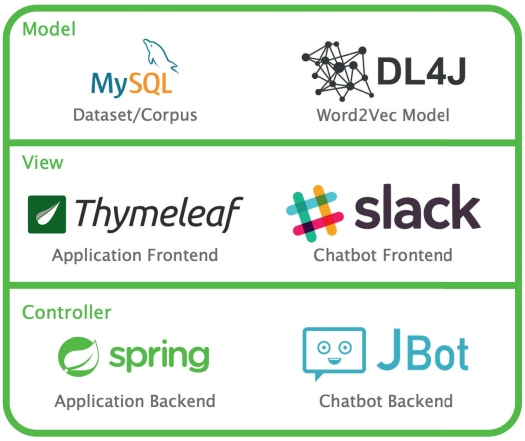
The realisation of this project was split into multiple parts. The first part was the acquisition/building of a dataset/corpus that holds all the information about the software design patterns that are supported by the chatbot. This part was realised by designing a database scheme and implementing it within a Spring application with the help of Lombok annotations. The data for the corpus was mostly taken from the famous book ‘Design Patterns: Elements of Reusable Object-Oriented Software’ by the Gang of Four and was ultimately saved within a MySQL database. The second part consisted of building an administrative frontend to see, add, modify and delete data within the corpus, as well as being able to look at the logs of the chatbot’s conversations. This was accomplished with the use of Thymeleaf. The third part consisted of building a Word2Vec model with the data we gathered, which was done with the use of the Deeplearning4J framework. The final part of the project consisted of actually building the chatbot and combining all the previous components into a whole application that could be deployed on a server. The bot was built with the use of JBot, a Java framework that is based on Spring and can be used to build Slack and Facebook Messenger chatbots.

Conclusion

With the help of various open-source tools it was possible to accomplish the set goal and build a functioning chatbot.
The application includes the following components/features:

- Slack chatbot that can hold simple conversations
- Software design pattern recommendation based on user-input
- Ability for users to rate the results and get examples
- Administrative control panel to see, add, modify or delete all the corpus-data and logs in the background

In the future, this application could easily be extended to provide more user interaction features and the results generated by the bot could be further processed and used to modify and extend the corpus.



Components of the Application