Design

To be written, chapter overview

0.1 Research questions

The research questions is what will guide the study, with the help of the set research questions we will explain how we are going to find the answers for each RQ. The following subsections will walk-through the thought process and methods for finding the answers.

Need to define RQ in background/theory or introduction

Continuation on our approach to solve the RQs, the design of the experimentation i.e. the constant parts vs the dynamic part

RQ 1. Is it possible to map software architectural concerns in java source code through machine learning?

RQ 2. Which input is most suitable for creating a classifier?

RQ 3. How large dataset is needed to create a satisfactory classifier?

RQ 4. Which of the chosen classifiers is the best for the task?

0.1.1 RQ:1

RQ:1 Is it possible to map software architectural concerns in java source code through machine learning? This is answered straightforwardly by implementing a model that can map concerns in java coded systems through machine learning, and answering the following RQs will result in the need to implement said classier which in turn answers the RQ:1.

0.1.2 RQ:2

RQ:2 Which input is most suitable for creating a classifier?

Work in progress

To find the most suitable input for a classifier, we split the experiment into three parts, the experimentation, comparison and representation.

The experimentation revolves around testing different pre-processing to find models with satisfactory performance.

0.1.3 RQ:3

When training a machine learning model, oftentimes, the models are trained with a large set of data so that the model becomes well adapted for various types of situations. However, in our situation, the total dataset becomes limited by the number of files that the system carries. Not only is the dataset limited, but also the number of files specified for training. We want to use as little training data as possible but still archive high performance in successful mapping. When using a small training set, a realistic scenario occurs, in which we only know the subset of a system, and we want to map the rest with the help of the subset.

To test how the classifiers perform with various training sizes, we decide to create

two different ways of splitting the data between training and testing. The first test is based on the ratio between concerns in the system. That is, splitting data into training and test samples based on the ratio. This approach does not represent a realistic scenario, as often, information about the ratio between concerns are unknown. Even if the split is not realistic, the results can still be used to see how well the classifiers perform under specific training sizes.

In the second approach, we base our training data on an absolute value instead of the ratio between concerns. As a subset of the system is known, we use this subset to train the classifier. To simulate this, we extract x number of files from each concern. As this approach does not take the ratio into account, another problem arises: when the number of files x is greater than the total number of files that the concern carries. We have decided to set a threshold value; if the number of files does not exceed the threshold value, the concern will be dropped and not included when performing the tests.

0.1.4 RQ:4

The classifiers tested in this experiment are Naive-Bayes, Support Vector Machine(=SVM) and Maximum entropy. We test Naive-Bayes because of its simplicity and previously achieved performance when **?** used Naive-Bayes to map architectural concerns. Maximum entropy has shown to be a great candidate alongside Naive-Bayes. In **?** maximum entropy performs better than Naive-Bayes in 2 out of 3 data sets, which motivates us to investigate further if maximum entropy can outperform Naive-Bayes in our study. Alongside Naive-Bayes and Maximum entropy, which both has the property of being probabilistically built, we decided to test SVM. This robust machine-learning algorithm has shown excellent results in topics regarding text classifications **?**.

When trying to answer the question: which classifier is the best for the task, we base the results by answering question RQ2 and RQ3. That means we try to find an optimal classifier with respect to the pre-processing and size of the training set. In this study, we

have chosen to not experiment with the classifiers hyperparameters; instead, we define a presetting of hyperparameters for each classifier, see section **??**.

When comparing the results from each classifier, we compare metrics such as accuracy, recall, precision, F1 score and prediction probabilities. The metrics will be graphically represented in section **??** and conclusions will be drawn as which.

0.2 Subject systems

The three open-sourced systems that are the subject for this study was provided by the supervisor. The biggest reason for choosing these systems as the subjects of this study is that mapping the concerns was done prior to this study and was provided along with the source files. This saves time from finding and correctly mapping the concerns and shifts the focus of the study to the implementation and study of the machine learning models.

The systems that were provided is jabref, ProM and TeamMates. JabRef is an acronym for Java Alver Batada Reference. It is a cross-platform citation, and reference management software that uses BibTeX and BibLatex. ProM is an extensible framework for Process Mining which is from where the name stems. The framework provides means for monitoring and analysis of real-life processes. TeamMates is an online peer feedback system for student team projects.

Need to reference the jabref site? And the other system sites?

Table 1: Subject systems

System name	Version #	Lines of code	Lines of comments	# of java files	# of concerns
Jabref	3.7	88,562	17,187	845	6
ProM	6.9	69,492	22,763	867	15
TeamMates	5.110	102,072	12,514	812	6

Table 1 shows the size of the subject systems used in the study by specifying the lines of code, lines of comments, the number of java source files and the number of architectural concerns.