MLQDA

Zsolt Takacs – 2472886T

Developing a Machine Learning webapp for Qualitative Data Analysis

Overview

- Overview of presentation
- Background
- Requirements
- Design
- Technologies
- Demonstration of finished system
- Testing
- Evaluation
- Future Work

Background

How do Machine Learning and Qualitative Data Analysis meet?

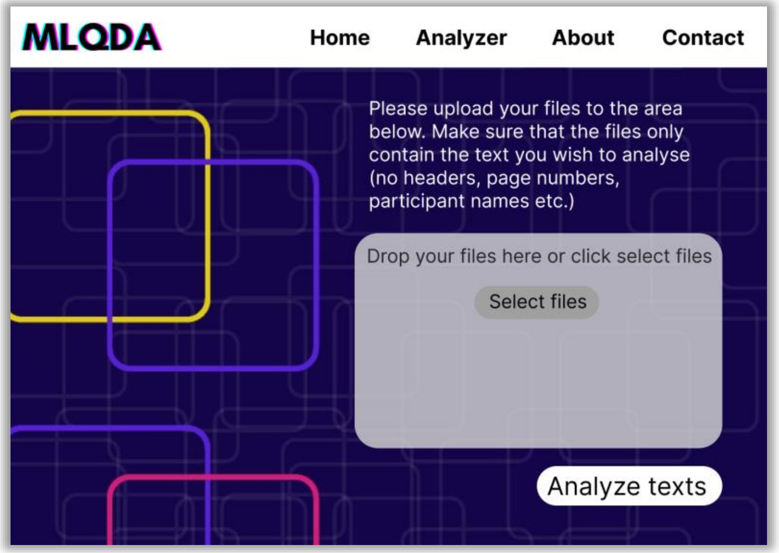
- Qualitative techniques are often described as subjective, unreliable and extremely time consuming
- Machine Learning techniques are becoming accessible rapidly
- Using machine learning to support qualitative data analysis can help overcome the mentioned issues
  - Less subjective coding
  - More reliable and reproducible results
  - Much faster initial coding process
- This has been proposed by multiple researchers previously and have been used in research already
- Most of the prominent applications to support qualitative data analysis offer similar solutions too

Requirements

- Functional requirements prioritised using the **MoSCoW** technique
  - Container requirements:
    - Users must be able to upload a text for processing
    - Users should receive valuable information about how the product processes their texts
    - The entire system could compile a zip file of all the result files to avoid cluttering and
  - ML script requirements:
    - The Machine learning script must be able to work with different file extensions
    - The Machine learning script should be able to automatically calculate the optimal parameters for the machine learning models of their results
    - The Machine learning script could provide visualization of the most important words and their contribution to the topics
- Non-functional requirements

Design

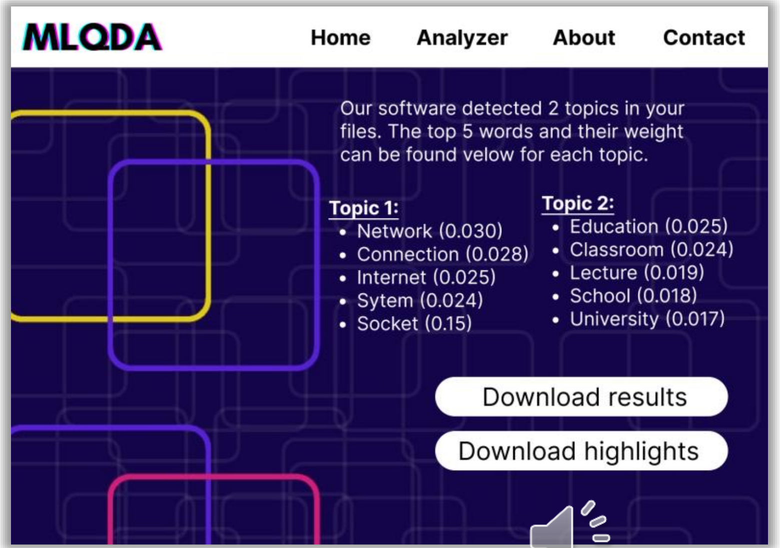
Based on requirements Figma wireframes were developed and converted into a realistic prototype



Please upload your files to the area below. Make sure that the files only contain the text you wish to analyse (no headers, page numbers, participant names etc.)

Drop your files here or click select files

Analyser start page



Our software detected 2 topics in your files. The top 5 words and their weight can be found velow for each topic.

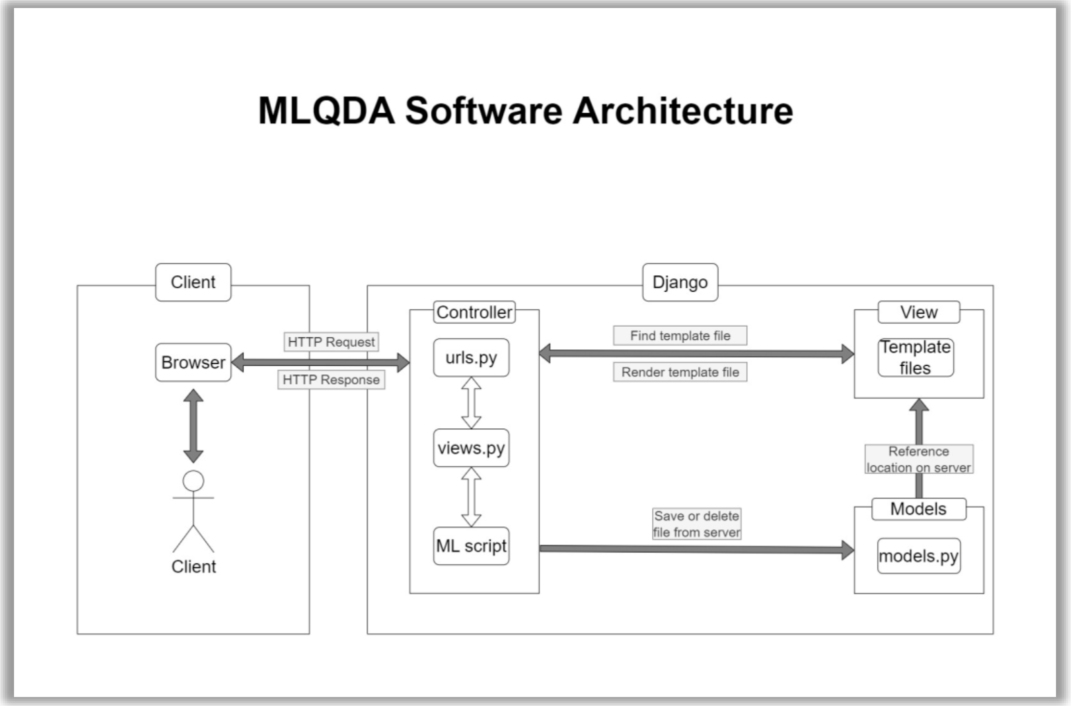
Topic 1:
- Network (0.030)
- Connection (0.028)
- Internet (0.025)
- Sytem (0.024)
- Socket (0.15)

Topic 2:
- Education (0.025)
- Classroom (0.024)
- Lecture (0.019)
- School (0.018)
- University (0.017)

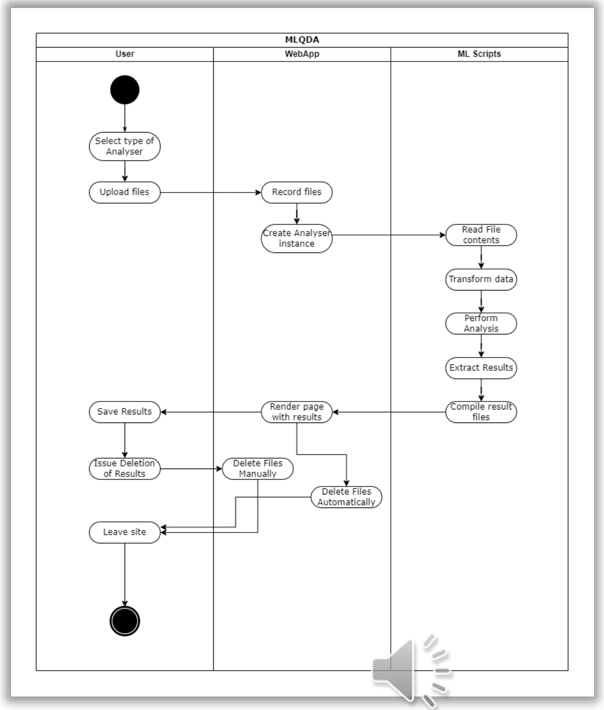
Analyser results page

Design

Based on requirements a system architecture diagram and a user activity diagram was sketched out to represent the inner logic of the proposed system.



Proposed Software Architecture



Activity Diagram

Technologies

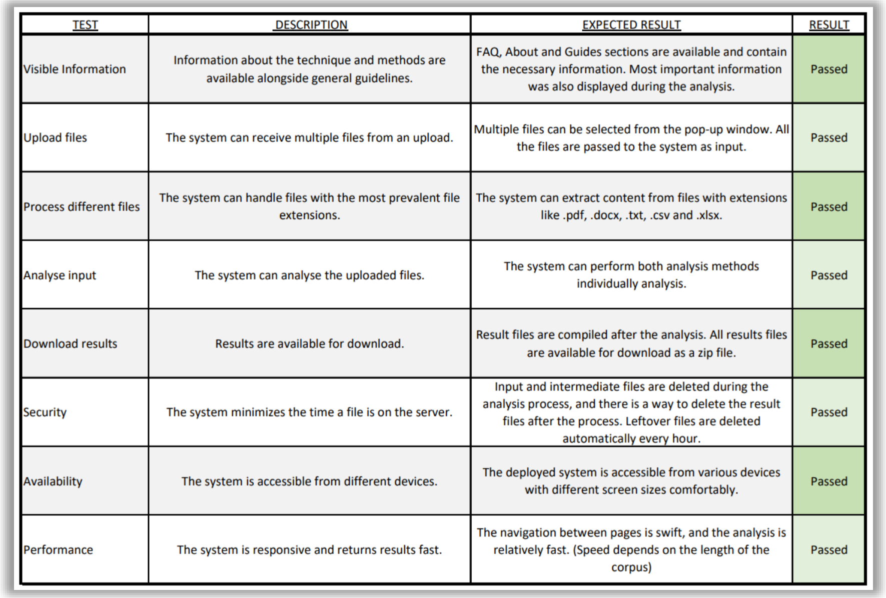
pythonanywhere
by ANACONDA
SQLite
django
spaCy
NLTK
GENSIM
topic modelling for humans

Demonstration of the finished system

Testing

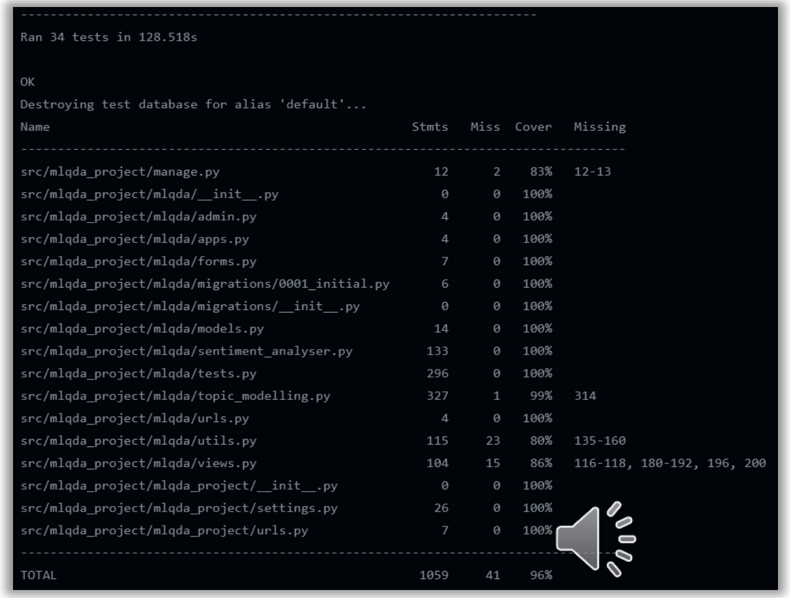
Two major type of testing – Manual Testing and Unit Testing:

1. Manual Testing table filled out after every deployment
2. Unit tests developed concurrently to code base and run as part of the CI pipeline



TEST	DESCRIPTION	EXPECTED RESULT	RESULT
Visible Information	Information about the technique and methods are available alongside general guidelines.	FAQ, About and Guides sections are available and contain the necessary information. Most important information was also displayed during the analysis.	Passed
Upload files	The system can receive multiple files from an upload.	Multiple files can be selected from the pop-up window. All the files are passed to the system as input.	Passed
Process different files	The system can handle files with the most prevalent file extensions.	The system can extract content from files with extensions like .pdf, .docx, .txt, .csv and .xlsx.	Passed
Analyse input	The system can analyse the uploaded files.	The system can perform both analysis methods individually analysis.	Passed
Download results	Results are available for download.	Result files are compiled after the analysis. All results files are available for download as a zip file.	Passed
Security	The system minimizes the time a file is on the server.	Input and intermediate files are deleted during the analysis process, and there is a way to delete the result files after the process. Leftover files are deleted automatically every hour.	Passed
Availability	The system is accessible from different devices.	The deployed system is accessible from various devices with different screen sizes comfortably.	Passed
Performance	The system is responsive and returns results fast.	The navigation between pages is swift, and the analysis is relatively fast. (Speed depends on the length of the corpus)	Passed

Manual Testing Table



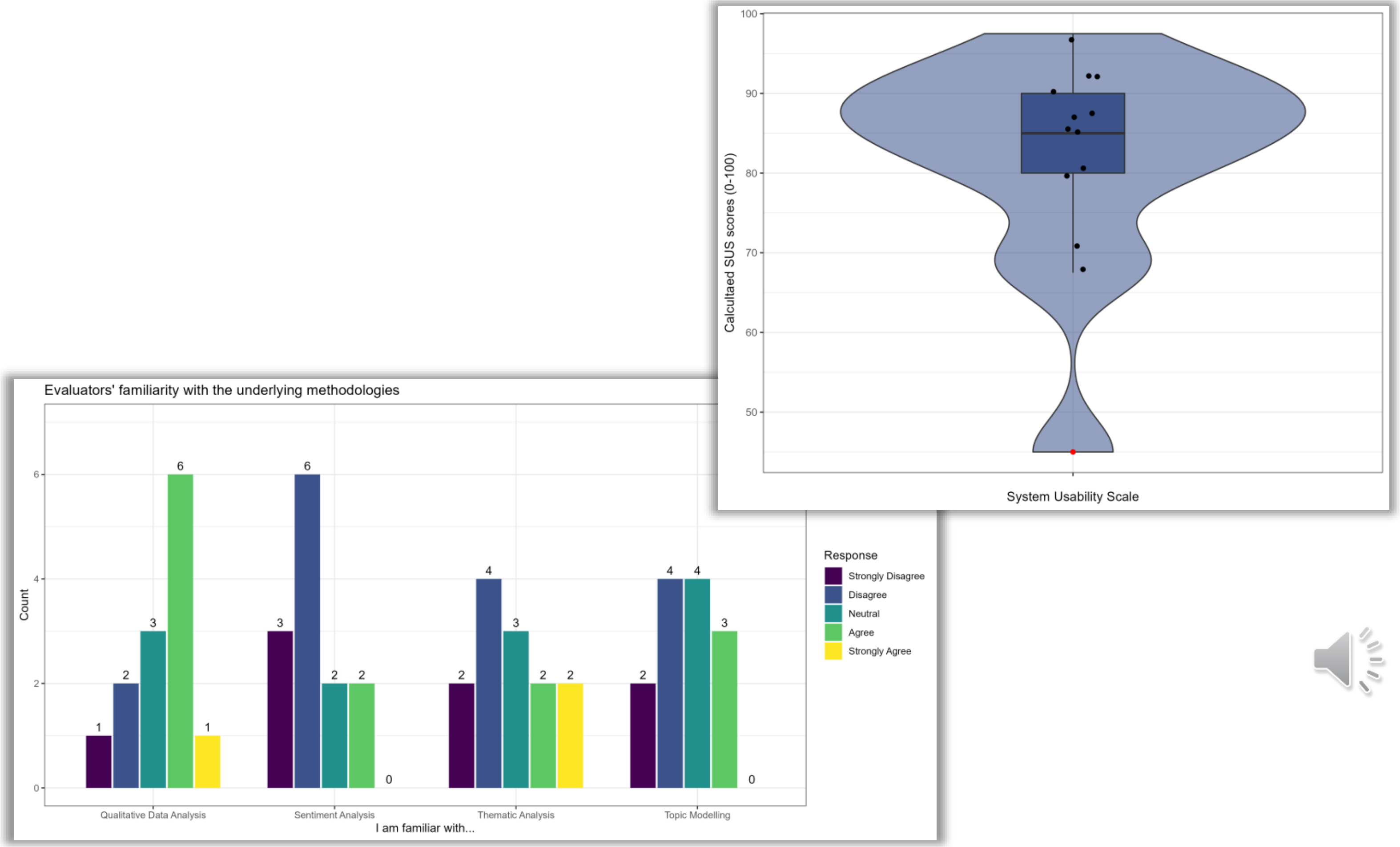
```
----------------------------------------------------------------------
Ran 34 tests in 128.518s

OK
Destroying test database for alias 'default'...
Name                                                Stmts   Miss  Cover   Missing
---------------------------------------------------------------------------------
src/mlqda_project/manage.py                            12      2    83%   12-13
src/mlqda_project/mlqda/__init__.py                     0      0   100%
src/mlqda_project/mlqda/admin.py                        4      0   100%
src/mlqda_project/mlqda/apps.py                         4      0   100%
src/mlqda_project/mlqda/forms.py                        7      0   100%
src/mlqda_project/mlqda/migrations/0001_initial.py      6      0   100%
src/mlqda_project/mlqda/migrations/__init__.py          0      0   100%
src/mlqda_project/mlqda/models.py                      14      0   100%
src/mlqda_project/mlqda/sentiment_analyser.py         133      0   100%
src/mlqda_project/mlqda/tests.py                      296      0   100%
src/mlqda_project/mlqda/topic_modelling.py            327      1    99%   314
src/mlqda_project/mlqda/urls.py                         4      0   100%
src/mlqda_project/mlqda/utils.py                      115     23    80%   135-160
src/mlqda_project/mlqda/views.py                      104     15    86%   116-118, 180-192, 196, 200
src/mlqda_project/mlqda_project/__init__.py             0      0   100%
src/mlqda_project/mlqda_project/settings.py            26      0   100%
src/mlqda_project/mlqda_project/urls.py                 7      0   100%
---------------------------------------------------------------------------------
TOTAL                                                1059     41    96%
```

Unit Test Coverage Report

Evaluation

The evaluation of the system took place in two steps

1. Pilot user evaluation by an qualitative analysis expert to gather general feedback
2. Open user evaluation to quantify system usability and gather more feedback



Future Work

- Include alternative machine learning techniques
- Improve Topic Modelling visualisation by updating pyLDAvis
- Scale up Topic Modelling to allow a more dynamic way of hyperparameter usage
- Automate highlighting after manual changes in topic words
- Include more instructions to support first-time users

References

- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative research in psychology, 3(2), 77-101.
- Chen, N. C., Drouhard, M., Kocielnik, R., Suh, J., & Aragon, C. R. (2018). Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity. ACM Transactions on Interactive Intelligent Systems (TiiS), 8(2), 1-20
- Crowston, K., Allen, E. E., & Heckman, R. (2012). Using natural language processing technology for qualitative data analysis. International Journal of Social Research Methodology, 15(6), 523-543.
- Janasik, N., Honkela, T., & Bruun, H. (2009). Text mining in qualitative research: Application of an unsupervised learning method. Organizational Research Methods, 12(3), 436- 460.
- Parks, L., & Peters, W. (2022). Natural Language Processing in Mixed-methods Text Analysis: A Workflow Approach. International Journal of Social Research Methodology, 1-13.