Focal Requirements Document

Students:
Razik Grewal, Peter Meyers, Mitali Potnis

Mentors:
Annette Han, John Stamper

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Version 1.5

1 Introduction

As education continually trends toward online learning, giving students frequent assessments, both formative and summative, is crucial to their development.[1] These assessments often come at a high price to educators however, as they are forced to create a large bank of questions to provide realistic learning opportunities for students of varied skill levels. [2] In addition to the high cost to produce these assessments, educators likewise must spend a significant amount of time providing students with quality feedback. This feedback helps to ensure the assessments students are completing enrich their learning experience, but is time-consuming to produce. Focal intends to reduce the burden of these assessments on educators, thereby saving both their efforts and time. By automating the assessment creation and evaluation pipeline, educators' time will be freed to focus on their various other responsibilities.

As Focal is a product that is intended to ease the burden of assessments on educators, it is vital that it not only shortens the time necessary to create and evaluate questions, but that it does so accurately. Minimal input from domain experts is a primary goal, meaning that the questions generated should be pedagogically sound without the need for additional input. While the ultimate goal is to make Focal an end-to-end solution, having a deployable model is not within the scope of this year-long capstone research. Within the year, our plan is to: acquire and process the data from two new online courses, test methods of key concept extraction, test alternative methods for question generation, identify a methodology for evaluating question quality, and identify a method for evaluating student answer quality. In summation, our plan is to have a well documented end-to-end assessment pipeline that will reside in a repository for future research. In the remainder of this requirements document, we will introduce each of the functional, non-functional, and resource requirements for a deployable Focal while simultaneously identifying those that are within the scope of our research.

2 Intended Users

As discussed earlier, the ultimate goal is that Focal will be a complete end-to-end automatic assessment pipeline. As such, the users of the system would include only educators and learners:

1. **Type A Users: Educators** would have the responsibility of providing the Focal pipeline with the text from a section of their course material. As Focal is intended to be a domain-agnostic solution, educators from a wide range of disciplines should be able to utilize its

functionality. After learning material is input, Focal will return to educators a generated list of pedagogically sound questions and their corresponding answers. Additionally, teachers will receive evaluations of student answers when they are complete.

2. **Type B Users: Students** will interact with Focal by receiving the questions that were produced in the previous step and providing responses to Focal. Focal will then evaluate the similarity between their response and the corresponding accurate solutions to provide the student with feedback.

While the ultimate goal of Focal is to be a tool that is directly employed by educators, that end state is not within the scope of this year-long capstone project, and as such, will be left to future iterations of this research. For this yearlong project, our primary users will be researchers (**Type C Users**) in future iterations of the Focal project. We have the following goals to enable future researchers:

1. Formalize a workflow for Focal. It is imperative that this workflow include proper documentation to ensure code readability and enable further development.
2. Define a better methodology for evaluating the pedagogical soundness of generated questions. Current methodology somewhat naively grades questions on question overlap, and leaves room for improvement. [2]
3. Incorporate an additional subject matter to begin the process of making Focal more domain-agnostic. In previous research, Focal was tested on a single subject matter. [2]
4. Attempt to improve on Focal's ability to produce a broad range of questions, varying in difficulty and style.

3 Functional Requirements

The proposed functionality of Focal is comprised of the following workflow:

1. An educator uploads the material for a section of their course to the Focal user interface.
2. Input course material is processed by Focal's concept extraction model and the proposed key concepts are returned to the educator for review.
3. If the educator approves the list of key concepts, they are sent to Focal's question generation model.
4. If the educator does not approve of the list of key concepts, they are given the opportunity to add or subtract any they disagree with. After these revisions, the list of concepts is sent to Focal's question generation model.
5. Focal's question generation model takes in the list of key concepts and course material and returns to the educator a list of relevant questions.
6. If the educator approves of the list of questions, it is sent to Focal's question answering model as well as to the user interface for student interaction.
7. If the educator does not approve of the list of questions, they are given the opportunity to add or subtract any they disagree with. After these revisions, the list of concepts is sent to Focal's question answering model as well as to the user interface for student interaction.
8. Focal's question answering model creates a list of answers for each question in the approved questions list. These answers are sent to Focal's answer evaluation model.
9. Student's receive questions in the user interface and respond accordingly. Their responses are sent to Focal's answer evaluation model.
10. Focal's answer evaluation model compares the generated answers to student responses and evaluates them for correctness. These evaluations are then sent to the educators and students for review.

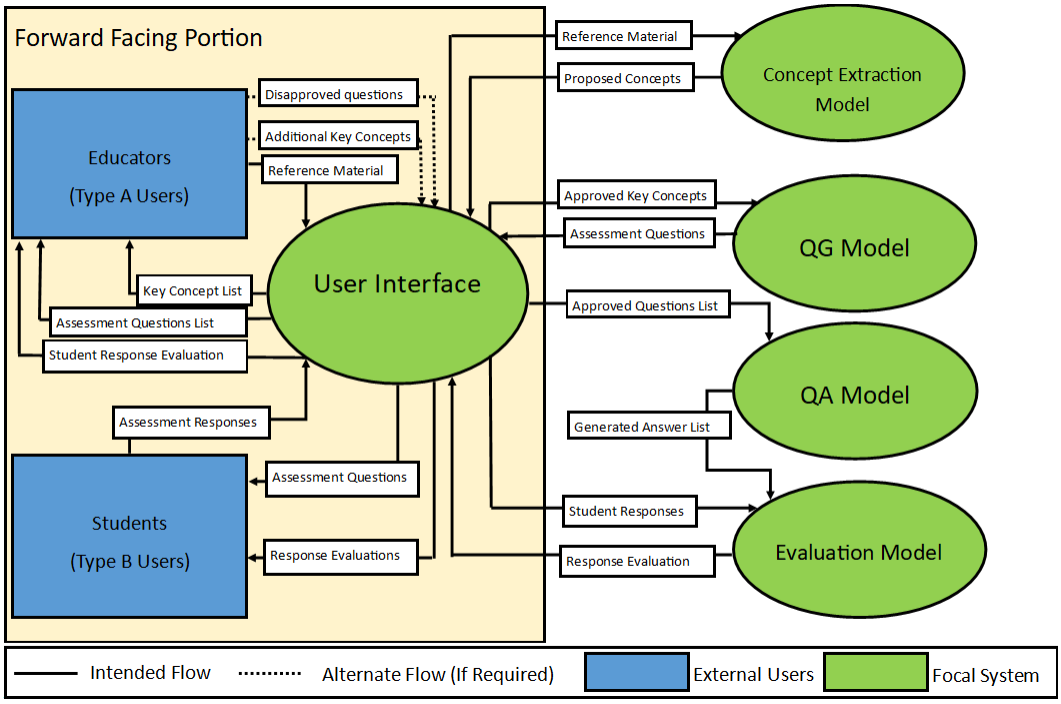

Fig. 1. Context diagram of Focal. This diagram demonstrates Focal's intended end-state as complete assessment pipeline. Ideally, this system should only be interacted with by educators and students.

In order for Focal to be a usable and useful tool for education, it must meet the following functional requirements:

1. Focal must eventually incorporate a user interface that is uncomplicated and usable for even novice users. Many Type A Users will have little or no training in the underlying technologies that enable the Focal pipeline to work, so these complicated tasks must be extracted away for the end-users.
2. Generated questions must be high quality and must be useful in assessments. This include a multitude of factors, and is one of the primary research goals for this year-long project:
   (a) Generated questions must be pedagogically sound. For educators, this means that the questions not only make logical sense, but that they adequately evaluate students on the key concepts of the provided learning material.
   (b) Questions must cover the entire breadth of the learning material. Educators should know that using the Focal pipeline will ensure students are adequately assessed on the entirety of the input learning material.
   (c) Questions should incorporate varying levels of complexity to cater to the needs of students with varying levels of mastery of the subject material.
   (d) Questions should test both "what" and "why". That is, students should be evaluated on their complete understanding of the subject.
   (e) Enhance the caliber of the incorrect options (distractors) for multiple-choice queries.
   (f) Incorporate the content in form of diagrams, images and other non-textual forms for the purpose of question generation context.
3. Generated answers must be robust to the varying possible correct responses for a given question. In an automatic evaluation system that checks for the similarity to a set of reference answers, students should not be punished for a correct response that is left out of the generated answer set.

4. Type B Users should receive worthwhile feedback in a timely manner. Incorporating assessments into online learning can be critical to success, but their implementation must not place undue burden on learners. Students should quickly see whether their response is evaluated as correct and, if not, should be given appropriate details as to why.

4 Non-Functional Requirements

Non-functional requirements describe the attributes and characteristics of a software system that are not related to its functionality, but rather its performance, usability, reliability, maintainability, and other qualities. Following are the Non-functional requirements our project must adhere to in order to ensure that our software system meets the desired levels of quality and performance and is able to provide a satisfactory user experience.

1. **Performance**:
   (a) The system must generate questions and answers within 5 minutes of receiving the learning material from educators.
   (b) The answer comparison process must take less than 2 seconds to evaluate the student's response.
2. **Accuracy**:
   (a) The system must produce high-quality questions that are pedagogically sound and cover the entirety of the learning material.
   (b) Questions must vary in complexity and style to cater to students with different levels of mastery of the subject.
   (c) The system must be able to produce answers that are robust to the varying correct responses for a given question.
3. **Usability**:
   (a) The user interface must be simple and easy to use, even for Type A and B users who have little or no training in the underlying technologies.
   (b) The system should be designed to reduce the burden of assessments on educators and learners by automating the assessment creation and evaluation pipeline.
4. **Scalability**:
   (a) The system should be designed to work with educators from a wide range of disciplines, making it generalized, domain-agnostic, and adaptable to different subject matters.
5. **Maintainability**:
   (a) The system must have proper documentation to ensure code readability and enable further development.
6. **Security**:
   (a) The system must be designed with security measures to protect sensitive student data.

These non-functional requirements are important for ensuring that Focal meets the needs of its intended users and provides an effective solution to their assessment needs.

5 Resource Requirements

1. **Hardware**:
   (a) The system requires a server or cloud-based infrastructure to run the machine learning models that generate questions and answers.

2. **Software**:
    (a) The system requires software tools for natural language processing and machine learning, such as GPT-3, SQuAD, and T5. [3] [4]
    (b) The system must have a user interface, which could be developed using a variety of programming languages and frameworks.
3. **Data**:
    (a) The system requires a large dataset of educational materials, such as MOOCCubeX, which contains text from various educational disciplines. [5]
    (b) The system also requires a dataset of high-quality questions and answers to evaluate the pedagogical soundness of the questions it generates.
4. **Human resources**:
    (a) The project requires a team of developers, data scientists, and domain experts to design, develop, and test the system.
    (b) The project also requires educators and students to test and provide feedback on the system.
5. **Time**:
    (a) The project requires a year-long capstone timeline to complete the research goals and formalize a workflow for Focal.
    (b) The project will also require additional time for future research iterations to fully develop Focal into a complete assessment pipeline.

6 Project Scope

The scope of our project for the year long capstone is to create a viable end-to-end assessment pipeline. This means that we will have evaluated different approaches to generate questions, generate answers, and evaluate correctness. Additionally, we will have identified an appropriate solution for evaluating the quality of generated questions. We also will have improved the question generation system in the sense of considering non-textual information as its context and improving the quality of the distractors for multiple-choice questions. Notably, we do not expect to deploy a solution by the end of this capstone research. Our implementation will reside in a repository for future research.

As we do not expect to deploy a solution in this research iteration, it is imperative that we formalize and document thoroughly the implementation of the Focal pipeline for future Type C users. This project is expected to take multiple iterations before reaching the point of deploying a solution for use by Type A and B users, so it is essential that our documentation enables this further research. In order to achieve a viable end-to-end assessment pipeline within this year long research project, we have the following priorities of work:

1. Acquire and process the data from two online courses (one chemistry course and one data science course). This data will be the basis for all of our models and will help with the goal of creating a more domain-agnostic pipeline.
2. Document the existing Focal Pipeline from previous iterations. While the pipeline exists to generate questions, it is not formally documented to allow for extension into answer generation and evaluation. [2]
3. Determine a better method for evaluating question quality. The existing method is rather naive, and could be made more robust.[2]
4. Conduct tests to identify the best methods for automatically generating answers to the questions generated by focal.

7 Terminology, Definitions, Acronyms, and Abbreviations

- **Accuracy**: Refers to the extent to which a system's output reflects reality.
- **GPT-3**: Generative Pre-Trained Transformer 3, this product is a large language model that has proven valuable at tasks vital to this project, such as question answering and text generation.
- **Maintainability**: Refers to the ability for a system to be updated and improved.
- **MOOCCubeX**: A large dataset of educational materials, in addition to a pipeline that is adept at automatically extracting key educational concepts from text.[5]
- **NLP**: Natural language processing, this is the branch of machine learning that focuses on the processing and generation of natural language.
- **Performance**: Refers to metrics that evaluate the speed of a system in the context of this project.
- **QA**: Question Answering, this is the process of using NLP models to automatically generate answers to given questions.
- **QG**: Question Generation, this is the process of using NLP models to automatically create questions based on text suppied to the model.
- **Scalability**: Scalability is a measure of the ability for a system to increase or decrease in capacity to meet the needs of the end-user.
- **Security**: Security in the context of this research refers to the protection and encryption of user's personal information.
- **SQuAD**: Stanford Question Answering Dataset, this is a large dataset made of questions on Wikipedia articles and their corresponding answers.[4]
- **T5**: Test-to-Text Transfer Transformer created by Google, considered a state-of-the-art model in many NLP tasks.[3]
- **Usability**: Usability refers to the extent to which users understand and can take advantage of the functionalities offered by a system.

8 References

[1] Yong Zhao, Jing Lei, Bo Yan Chun Lai, and Hueyshan Sophia Tan. What makes the difference? a practical analysis of research on the effectiveness of distance education. *Teachers College Record*, 107(8):1836–1884, 2005.

[2] Huy A. Nguyen, Shravya Bhat, Steven Moore, Norman Bier, and John Stamper. Towards generalized methods for automatic question generation in educational domains. In *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption*, pages 272–284, Cham, 2022.

[3] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. In *North American Chapter of the Association for Computational Linguistics*, 2020.

[4] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*, 2016.

[5] Jifan Yu, Yuquan Wang, Qingyang Zhong, Gan Luo, Yiming Mao, Kai Sun, Wenzheng Feng, Wei-Hao Xu, Shulin Cao, Kaisheng Zeng, Zijun Yao, Lei Hou, Yankai Lin, Peng Li, Jie Zhou, Bingsheng Xu, Juan-Zi Li, Jie Tang, and Maosong Sun. Mooccubex: A large knowledge-centered repository for adaptive learning in moocs. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.

9 Reflection

Throughout the process of writing this requirements document, we were really able to gain a better understanding of what the scope of our project is. We discussed all the relevant end goals of the project with our mentor as well as within our group to help guide the process. After understanding what the end goals for Focal were, we were then able to backtrack and decide what goals were reasonable within the time frame of our capstone. As we go forward, we will continue to reference this requirements document as a basis for what we will prioritize. If we could redo this requirements document, however, we would likely choose to spend a little less time producing the document itself and more time reviewing literature surrounding our topic. While we have reviewed a fairly large number of research papers regarding our topic, there is still room for us to improve in this regard. Also, we would think about focusing on the differences between a research based project and a project for industry from the start. This would have helped us to understand better the specifics needed in this requirements document.

10 Changes to Previous Deliverables

There are no changes to previous deliverables to report at this time.

11 Change Log

Table 1. Change Log and Versioning

Version	Date	Changelog
V1.0	2023.02.20	Added Introduction, Project Scope, Functional Requirements
V1.1	2023.02.22	Added Non-functional Requirements, Resource Requirements
V1.2	2023.02.23	Modified Functional, Resource Requirements and Intended Users
V1.3	2023.02.24	Added References,Terminology
V1.4	2023.02.26	Labelled Intended Users, Modified Context Diagram, Added prioritized goals in the Scope Section
V1.5	2023.02.26	Updated and Added Alternative flows in the Functional Requirements