AI Contract Review

Cycle Two Report

By

Aiden Green

Catherine Bickerton

Jay Scott

Luke Robinson

COMP 4710 Senior Design

Department of Computer Science and Software Engineering

Samuel Ginn College of Engineering, Auburn University

March 25, 2024

Table of Contents

1.0 System Metaphor (Bickerton)..... 3
2.0 Cycle Intent (Bickerton)..... 3
3.0 User Stories (Green)..... 4
3.1 User Stories Defined..... 4
4.0 Design Documentation (Green)..... 7
4.1 Language..... 7
4.2 Architecture..... 7
5.0 Meeting Minutes (Bickerton)..... 8
5.1 Presentation Meeting (02/26/2024)..... 8
5.2 Cycle Intent Meeting with Client (02/27/2024)..... 8
5.3 Status Report Meeting (03/11/2024)..... 9
5.4 Client Meeting (03/12/2024)..... 9
5.4 Status Report Meeting (03/18/2024)..... 9
5.5 Team Meeting (03/19/2024)..... 9
6.0 Management Plan (Scott)..... 10
6.1 Task Assignments..... 10
6.2 Development Schedule..... 10
7.0 Source Code..... 10
8.0 Lessons Learned (Robinson)..... 11
9.0 Test Results (Green)..... 12
10.0 Future Work (Bickerton)..... 14
11.0 Presentation Slides..... 14
12.0 Sponsor's Approval..... 14
13.0 Previous Cycle Docs/Grade Sheets..... 15

1.0 System Metaphor (Bickerton)

The Office of Sponsored Programs at Auburn University reviews all potential external contracts, proposals, and awards to determine whether the university can negotiate or approve them. Staff members are currently required to review each contract manually, which is time-consuming, arduous, and inefficient. Our team's goal is to design an artificial intelligence program that reviews all incoming contracts, flags conflicts within them, and suggests alternative wording in accordance with the university's guidelines. This program will streamline the contract review process for OSP staff members and learn from successful contract negotiations to strengthen its functionality.

2.0 Cycle Intent (Bickerton)

For this cycle, we began by delivering a working implementation of our project to our client for testing and feedback. Upon receiving results from our client, we aimed to make quality improvements based on feedback and work on subcontract audit scanning based on UIDP contract accords. Most feedback focused on either the over or under-flagging of certain terms and phrases, so we added keywords to the Terms and Conditions Contract Matrix to catch previously missed phrases. We visually updated the scanned documents by better differentiating new comments from the documents themselves and highlighting phrases more consistently.

3.0 User Stories (Green)

3.1 User Stories Defined

Document Scanning

- Summary: As a contract reviewer I want a tool that scans agreements to find problematic language to increase the speed at which my department operates.
- Description: This feature should scan an agreement and search for problematic language and clauses that are not acceptable. The scanned agreement should be compared to the FAR Matrix and Contract T&Cs Matrix to flag these issues for review or removal.
- Planned Hours: 12
- Planned this cycle: 12
- Actual: 13
- Actual this cycle: 13
- Coders: Aiden Green
- Testers: Aiden Green
- Reviewers: N/A
- Status: In Progress

Create an Installer

- Summary: As a contract reviewer I want a tool that can be easily installed on my machine without significant technical knowledge.
- Description: This feature should be a basic installer that can easily and quickly install our software package on the user's machine.
- Planned Hours: 10
- Planned this cycle: 0
- Actual: 10
- Actual this cycle: 0
- Coders: Jay Scott
- Testers: N/A
- Reviewers: N/A

- Status: Complete

Sub-Agreement Scanner

- Summary: As a contract reviewer I want a tool that can scan a sub-agreement contract to determine if there have been any findings from an audit.
- Description: This feature should build off of the already-existing GUI and allow the user to submit multiple PDF agreements/questionnaires and return which files need to be reviewed.
- Planned Hours: 15
- Planned this cycle: 5
- Actual: 5
- Actual this cycle: 5
- Coders: Aiden Green
- Testers: N/A
- Reviewers: N/A
- Status: In Progress

Suggest Alternative Language

- Summary: As a contract reviewer I want a tool that can suggest alternative language for the problematic language that has been flagged instead of having to manually search for it.
- Description: This feature should use approved AU alternative language documents to suggest alternative language to a document reviewer when encountering flagged problematic language and clauses.
- Planned Hours: 15
- Planned this cycle: 5
- Actual: 5
- Actual this cycle: 5
- Coders: Aiden Green
- Testers: N/A
- Reviewers: N/A
- Status: Not started

Learning from Negotiations

- Summary: As a contract reviewer I want a tool that can learn from past negotiations and provide better and more accurate suggestions to more quickly and consistently lead to successful negotiations.
- Description: This feature should retain information gained from past negotiations and learn from its successes and failures to improve performance in the future. (There are many instances where the same types of negotiations have to be made with the same companies, this will allow some steps to be skipped, thus saving time.)
- Planned Hours: 30
- Planned this cycle: 0
- Actual: 0
- Actual this cycle: 0
- Coders: N/A
- Testers: N/A
- Reviewers: N/A
- Status: Not started

Risk Assessment

- Summary: As a contract reviewer I want a tool that will give me an overall risk rating for a contract so I can make appropriate risk assessments during a negotiation.
- Description: This feature should take into account all risk factors for a given contract and provide past relevant information to assist reviewers in a manual risk assessment. This will be useful when the negotiations do not follow a typical format.
- Planned Hours: 40
- Planned this cycle: 0
- Actual: 0
- Actual this cycle: 0
- Coders: N/A
- Testers: N/A
- Reviewers: N/A

- Status: Not started

Gathering Documentation

- Summary: As a contract reviewer I want a tool that has a basic understanding of negotiation and contract approval.
- Description: In order to train the artificial intelligence to properly flag and negotiate contract terms, the team is gathering both basic public documents and OSP-specific contracts to give the AI a basic understanding of the contract negotiation process.
- Planned Hours: 5
- Planned this cycle: 0
- Actual: 5
- Actual this cycle: 0
- Coders: N/A
- Testers: N/A
- Reviewers: Catherine Bickerton
- Status: Complete

4.0 Design Documentation (Green)

4.1 Language

We are the second semester of students assigned to this project and all of the existing code-base is written in Python. We have all decided that it will be easiest to remain with this language, as we are all confident in our Python knowledge. Python also has many readily available machine learning libraries, such as scikit-learn, that we can make use of for the more complicated portions of this project.

4.2 Architecture

This project will take in the text of a given document and process it in many different ways to speed up the contract review process. Currently (for a simplified layout description), there is a simple GUI where the user can select a desired

document, save it, and then scan it. When the user selects the ‘scan’ option the main operation of the script begins, which is as follows: the contract is converted to a .txt file (this makes the document easier to deal with), the script to flag problem language is then ran (using the FAR (Federal Acquisition Regulation) Clause Matrix and T&Cs Matrix that load on GUI open), the scanned contract is then converted back to a docx file, and then the contract is annotated when back in its original state.

Towards the end of this cycle we have added another feature to this tool. Our customer needed something to scan lengthy subcontracting agreements to see if there are any findings from an audit. We will use the same GUI as the other scanning feature, but just add another option where the user can enter multiple PDFs and have them scanned for specific questions related to the audit.

5.0 Meeting Minutes (Bickerton)

5.1 Presentation Meeting (02/26/2024)

- Both teams met with Dr. Dozier and the other teams under his supervision to present cycle one deliverables and future plans for cycle 2 while gaining feedback from the professor.
- Attendees: All members (orange and blue)
- Time: 8:00 - 9:00 (1 hour)

5.2 Cycle Intent Meeting with Client (02/27/2024)

- Both teams met with Darren May to discuss cycle intent and plan a deliverable software implementation for OSP staff to test during the week of spring break. The next day, we provided the office with a working version of our project.
- Attendees: CB, CB, LR, TW, DM
- Time: 11:00 - 11:15 (1 hour)

5.3 Status Report Meeting (03/11/2024)

- Both teams met with Dr. Dozier and the other teams under his supervision to give status report updates and gain feedback from the professor.
- Attendees: All members (orange and blue)
- Time: 8:00 - 9:00 (1 hour)

5.4 Client Meeting (03/12/2024)

- Both teams met with Darren May to discuss feedback concerning the delivered implementation of the project.
- Attendees: JS, EB, TW, DM
- Time: 8:00 - 9:00 (1 hour)

5.4 Status Report Meeting (03/18/2024)

- Both teams met with Dr. Dozier and the other teams under his supervision to give status report updates and gain feedback from the professor.
- Attendees: All members (orange and blue)
- Time: 8:00 - 9:00 (1 hour)

5.5 Team Meeting (03/19/2024)

- The orange team stayed in close contact throughout the day to assess the feedback from the OSP and implement quality of life improvements. We also began the construction of the subagreement audit scanner, creating a framework that allows for these documents to be scanned and evaluated.
- Attendees: AG, CB, JS, LR
- Time: 10:30 - 6:10 (7hr 40min on/off)

6.0 Management Plan (Scott)

6.1 Task Assignments

For this cycle, Catherine Bickerton continued to work on documentation for the project and deliverables for the class, ensured the team stays on track to accomplish our goals this cycle within the given time frame, and resolved several missing keyword flagging issues based on the feedback provided by the client. Aiden Green started work on the new audit subagreement user story ensuring that going into cycle 3 we would have a head start and be able to look towards working on the learning from negotiations, and potentially the risk assessment user stories. Jay Scott continued to work on the installer to ensure that we are able to provide demo products to the client for feedback, as well as starting to update the user and developer manuals based on the work we have done so far. Luke Robinson worked on improving the clarity of the text highlighting which was an issue based on the client's feedback regarding the ease of use when reading through the document to differentiate between new and original text.

6.2 Development Schedule

The start of cycle 2 was focused on producing a demo for the project to give to our client for initial feedback on the progress we have made thus far as well as beginning work on the new audit subagreement user story. After some coordination between the orange and blue teams, an installer for the demo was successfully created and delivered to the client to test and collect feedback over the spring break. Following the spring break, the client provided some valuable feedback regarding some over-flagging with problematic language, some common phrases that were not being flagged, and some difficulty differentiating some of the flagged text versus the original document. After receiving this feedback, the orange team was able to successfully resolve what seems to be most issues that were reported by the client, as well as get a good initial start on the new audit subagreement user story.

7.0 Source Code

https://github.com/Ktmking111/AI_Contract_Review_Spring2024

8.0 Lessons Learned (Robinson)

This cycle taught us further about what the development in the real world demands as we delivered the first product to our client, received feedback based on their real world testing, and got to work on resolving these issues to be able to deliver the software they are in need of.

With the initial release of our software to our clients, we felt confident in its performance and were eager to share this with our clients and move forward into developing new features to deliver. However, the feedback we received told us that our work so far has been good, but still needs some revision for it to be worthwhile for our users. Our scanning tool, while it was giving good insight into the potential concerns with the contract, was also found to be highlighting some sections that didn't necessarily need to be, and also missing some phrases that really should be highlighted. This was something that was difficult for us to identify in our own testing as we don't have the knowledge that these professionals do on exactly what should and should not be highlighted. This taught us the importance of having real world testing in our development cycle as it allows our software to be tested against those who are already great at completing the task our software is attempting to.

With all of this, communication continued to prove itself to be very important. On top of the feedback on the performance of our software that was much needed, we needed to communicate about how expectations might shift moving forward. As we reworked some of these flaws, we had to shift time and effort away from features we had initially hoped to put more time into this cycle. This meant communicating with our client about what they feel our time would be best spent on not only this cycle, but moving forward through the rest of the semester – we needed to make sure we knew which features are most essential to the software's usability.

Further, this cycle proved even more how important maintaining a remote repository with branches intentionally separated is. With our code base expanding, and each developer working on different tasks, we found the use of different branches for each person and feature to be beneficial to our organization in development.

As we move into the last cycle of the semester, the lessons we have learned so far will surely prove to be essential in our workflow while we continue to push to deliver the best product we can to meet our clients needs.

9.0 Test Results (Green)

In the most recent release of last semester's code there had been an accumulation of problems which eventually led to some of the Problem Language highlighting not working as well as it did in the previous versions. We went back to a previous version and it was working better but still not great. Using Jaccard Similarity to flag Problem Language produced a lot of potential problem language, but with a very low similarity (most around 15%). I didn't believe this was very helpful, so I found another way to do it with Sklearn's TF-IDF (Term Frequency–Inverse Document Frequency) Vectorizer. This is essentially a weighting system that determines how "important" a word is to a document. With this method I seemed to get a higher confidence on some of the things that seem like they should be caught. The issue is we still don't have an unmarked and fully marked document to test on so we know for sure what we need to catch.

Jaccard:

[POTENTIAL PROBLEMATIC LANGUAGE DETECTED]

Notice.

Problem Category: Insurance

Common Problems: Notice of Cancellation

Preferred Language: []

Why: None

1st response to Sponsor: Auburn University does not issue 30 days' notice of cancellation.

Confidence: 0.3333333333333333

TF-IDF with Cosine-Similarity:

[POTENTIAL PROBLEMATIC LANGUAGE DETECTED]

Notice.

Problem Category: Insurance

Common Problems: Notice of Cancellation

Preferred Language: []

Why: None

1st response to Sponsor: Auburn University does not issue 30 days' notice of cancellation.

Confidence: 0.5797386715376658

[END POTENTIAL PROBLEMATIC LANGUAGE]

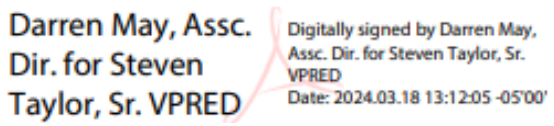
10.0 Future Work (Bickerton)

As cycle progression has gone smoothly, our plans for the semester continue. Unfortunately we pushed back the Learning from Negotiations user story from this cycle to cycle 3 because of the immediate revisions we made after getting feedback from the OSP. While the audit scanner is not completed, we have made great progress towards checking whether or not certain institutions are low-risk auditees. We firmly believe that the audit scanner can be worked on and completed by the end of cycle 3, and we also desire to make significant headway on negotiation learning based off of both AU specific documents and publicly available contract databases.

11.0 Presentation Slides

AI Contract Cycle 2 Presentation

12.0 Sponsor's Approval



13.0 Previous Cycle Docs/Grade Sheets

AI Contract Spike Report

Spike Presentation

AI Contract Cycle 1 Report

AI Contract Cycle 1 Presentation