DEPARTMENT OF COMPUTER SCIENCE

EJudg: A Tool for Annotating UK House of Lord/Supreme Court Judgements

Samuel Goh Kean Hau

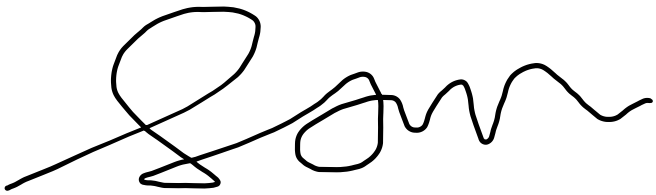
A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering

---

January 20, 2023

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted or otherwise incorporated material, which is the work of others, I have included the source in the references. Any views expressed in the dissertation, other than referenced material, are those of the author.



Samuel Goh Kean Hau, Thursday 19 January 2023

Executive Summary

This report presents new legal analysis tool called eJudg. eJudg is a legal annotation tool designed to aid editors working on an ongoing greater effort to digitize the lengthy history of UK House of Lords/Supreme Court judgements. Developments in this field are intended to allow law students and professionals alike to conduct legal research more easily. Demand for such a tool will only grow over time, as UKHL/UKSC case law history will continue to expand as more judgements are produced by the courts each year [1]. Legal judgements are also complex and lengthy to process in its raw text form, particularly in the jurisdiction of the United Kingdom because each sitting judge on the panel produces their own separate judgements, as opposed to contributing to a single leading judgement as in other jurisdictions.

Focusing on the specific contributions of our tool, this report will discuss how eJudg is intended to be used to improve the pipeline currently in use by the work group that is producing legal analytics systems. eJudg is meant to aid in the processing of the legal corpus document currently in use by the larger work group. The tool offers editors a lightweight and user-friendly interface to modify annotation entries in the corpus. It also addresses a noted shortcoming of prior projects by practicing open-source design principles. This allows ease of integration with other projects and modules produced by future work. Lastly, the tool serves as a testbed for demonstrating the potential for legal analytics tools to be produced from the edited data generated by eJudg.

In short, this project has resulted in a multifunction tool that will be of significant help to a larger organization of legal professionals and academics currently facing an indomitable task. It has resulted in the creation of a lightweight web tool to significantly speed the rate at which digitally annotated cases can be generated. Such results are reflected in the evaluation conducted at the end of this project, which show that eJudg has improved comprehension of legal judgements by 34.1%, and the efficiency of annotating cases by 29.1%.

Acknowledgements

I would like to express my gratitude to the various entities and individuals involved in supporting me throughout the creation of project, without which said project would not have been possible. I'd firstly like to thank the University of Bristol for offering me an opportunity to embark on this journey. I would then like to thank my supervisor, Dr. Oliver Ray, for guiding me through this project and helping me find my own voice and contribution. Then, I would like to thank my friends and family who have provided their unending support throughout my project and allowing me to complete it to the best of my abilities: my parents for their continuous emotional support and never-ending concern for my wellbeing throughout this challenging process; and my siblings for their guidance and experience; Lastly, my friends who have been with me since our undergraduate years and before, for lending their input and criticisms at various stages and being shoulders to hear my worries and concerns.

Thank you to everyone.

Table of Contents

1. Introduction

This report covers a web tool called eJudg – a novel annotation tool developed as part of a larger pipeline being contributed to by many others to digitize the case law library of the UK House of Lords and Supreme Court. In the UK legal jurisdiction, the preservation and understanding of past cases are essential due to the principle of *stare decisis [2]*, whereby decisions on point of law determined in past judgements are binding on future decisions. Hence, since the inception of the institution, a great need to record and analyze judgements made by the courts of the United Kingdom, particular the House of Lords/Supreme Court which serves as the final court of appeal, has always existed. This great need, however, has been paradoxically complicated [3] by the often lengthy and obtuse wording of these legal judgements. Unlike in other jurisdictions, the judgements courts of the UK inherited a systemic quirk from its legacy as a court of common law[4]. Namely, this has taken the form whereby the sitting panel of judges each produce an independent judgement, rather than a single leading judgement. While this feature has been deemed desirable for its preservation of the character and conflicting considerations underpinning the final judgement, often leading to decisions being overturned years or decades later[5], on grounds declared insufficient a generation ago due to the changing times, it has also added an extra layer of complexity of extracting and summarizing the principles of law that was decided in the case in question. Editors and analysts are challenged to read through often conflicting reasonings presented by a panel of the most esteemed practitioners of law in the nation, and somehow reduce the established principles to a short and concise 'in a nutshell' summary.

1.1. Motivations

To answer the dilemma mentioned above, an initiative was launched by professionals within the fields of legal practice and academia to develop a pipeline for digitizing UKHL/UKSC judgements. Beyond merely creating an online archive of judgements, said initiative aims to produce case annotations, to be used in the further development of tools and aids to directly contribute to improving the lives of legal academics and practitioners. This report will be covering some

projects and tools that have been undertaken under this umbrella, specifically those done under the tutelage of our supervisor, Dr. Oliver Ray. One of the many challenges that this initiative currently faces is the production of manually annotated cases from which machine learning tools might be trained. Thus far, manual annotation of judgments and summaries has been a time consuming and expensive process, involving the help of legal experts. This has persistently limited the initiative to a subset of completed annotations produced by Hachey and Grover[6], rather than being able to expand our library of sample cases. Hence, my tool is aimed at working to resolve this existing limitation and serve as a testbed to show how the expansion of our library of annotations might aid more projects to follow.

1.2. Project Aims

Our tool is intended to be integrated within a greater pipeline of work currently pursued by others in the field, allowing them to complete their tasks much more efficiently. As such, the following project aims were decided:

1. Development of a user-friendly annotation tool for editing the SUMO csv corpus currently in use by the other projects within the initiative. This includes creating new case annotations, and the inspection and correction of annotations completed in the past.
2. Take an open-source engineering approach to the project that is easily compatible with expansion. Being aware of the limited development timeframe for our tool, we viewed that it was necessary to future proof our codebase to be built upon by later projects.
3. Demonstrate possible uses of the annotations for developing analytical tools.

1.3. Contributions

By the end of this report, it will be argued that our tool successfully achieved its stated objectives. Evaluation conducted shows that integrating eJudg into the annotation workflow resulted in tasks being completed much more quickly. This included annotations of new and unedited cases, as well as editing of entries for previously annotated cases to fix mistakes. Surveyed users also expressed agreement when asked if the analytical tools demonstrated in eJudg improved their understanding of judgements.

2. Background

2.1. Prior work

Prior to discussing my final product, credit must first be given to the significant preceding work in developing this pipeline. Ongoing attempts to digitize the case law are numerous. The following are the broad types of tools that currently exist on the market.

2.1.1. Online law libraries

Online repositories of past cases exist across numerous jurisdictions. In fact, it may be said that every major jurisdiction each alone possess many such tools to aid practitioners. This can be taken as an indicator of general interest within field for continued work to develop and expand such tools. For the interests of the law of England and Wales, the interest of this project, one may subdivide available tools into those available to the public, and proprietary in-house tools. Lexis Library[7] and Westlaw UK[8] would be common knowledge to law students and legal professionals alike as well maintained and leading libraries that can be accessed via membership or academic access. Such sites are excellent general use tools: Cases available on these sites are often annotated with relevant legislation, case references etc. Often, expert written summaries and comments are provided on top of the raw judgement to aid researchers (see Figure 2.1). However, these sites currently do not offer the relevant annotation function desired in our project.

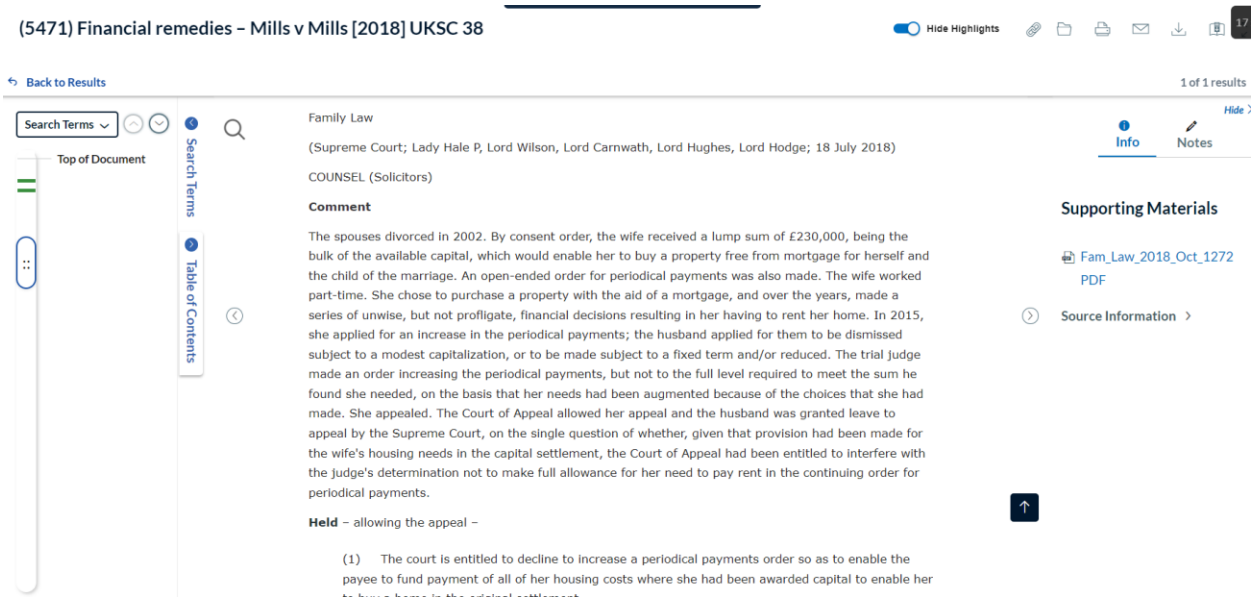


Figure 2.1: Case view sample from Lexis Library

2.1.2. Annotation/Editing tools

Further from the prior mentions that directly intersected with the path my tool was intended to fill, a broad assessment was also conducted to survey the general landscape of tools and products available on the market that might be used to help process our SUMO data. Particular attention was given to document editors, given that our pipeline currently uses a common csv. Our survey indicated that most of the market options centralize around shared document annotation tools. [9][10][11] Ultimately, these tools were only marginally relevant to our research, mostly to provide references for interface design.

2.1.3. SUMO

Of all ongoing work in our field of interests, the most relevant to our project consists of a collection of projects under the supervision of Dr. Oliver Ray. To provide a summary of the ongoing work within our work group:

SUM – The roots of the work group was automated legal summarization project conducted by Hachey and Grover. Their system utilized machine learning to attempt to generate legal summaries using two classifiers: relevance and rhetorical role. The former attempted to map the original case document to manually annotated summaries, to extract the most relevant

sentences. The rhetorical role classifier, on the other hand, attempted to map the argumentative structure of the cases, using rhetorical classifiers such as BACKGROUND, DISPOSAL, FACTS etc. Together, the product of Hachey and Grover's system constituted the initial 69 annotated judgements that later projects discussed below would expand on.

ASMO – The next major expansion to the library of annotated judgements was conducted by Valvoda et al [12]. In their research, they majorly expanded the scope of the legal annotation initiative in two ways. Firstly, he produced a corpus of 300 expert annotated and summarized judgement, by incorporating Hachey and Grover's judgements into a new list of judgements scraped from the House of Lords' website. However, an arguably even more important contribution was the formation of the annotation schema known as 'ASMO'. In Valvoda's annotation scheme, several key elements of a judgement were highlighted by legal experts to train machine learning tools: AS, or agreement statements, denoted statements of agreement, whether on facts or on points of law, were expressed between the panel of a particular judgement in their respective arguments. MO, or majority opinion, denoted the intended outcome of Valvoda's machine learning pipeline, whereby the relationships between the sitting judges derived from the classification of their agreements, would be used to determine the majority opinion of the analyzed judgement. Overall, Valvoda's work is seminal to all successive projects to be discussed below.

UKHL corpus/SUMO - The latest additional to work on the corpus came from Conroy and Imansyah[13], who sought to merge Hachey and Grover's summarization (SUM) with Valvola's work on ASMO. There were two key objectives of these projects: The first was to merge the annotations of the two systems into one complete 300 judgement document, known as the 'UKHL corpus'. Valvoda, in the process of their research, had split their annotations into three separate 100 case files for training. Next, Conroy sought to merge the summarization classifiers created by Hachey and Grover's pipeline with Valvoda's ASMO annotations, to create a superior summarization (SUMO) method. Aside from achieving said product, it is also important to note that her work created the complete corpus now in use by all parties as the gold standard.

JudgeViz[15] – Lastly, JudgeViz was a tool developed by Daniels', a prior MSc student on this program, that utilized the data generated by Conroy's SUMO project to create visual aid tools for legal research. Daniels' achieved this by creating a pipeline to visualize the relationships between judges in the form of a graph. in contrast to the work on the corpus, it is argued that Daniels' work was more directly inspirational to our tool, given it was the first iteration of a visual aid tool intended to present the data from the corpus to be read by users. Furthermore, it is conceded that Daniels' work explores the technical aspects of the technical aspects of the relationship map in more detail than is pursued in our tool. However, a limitation to her tool was its opacity to editors, as JudgeViz only accessed the UKHL corpus in a read-only fashion and had no editing capability.

judgeViz

A new approach to improving information retrieval from UKHL judgments.

By

Miranda Lily Eliza Daniel

Department of Computer Science
University of Bristol

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering.

September 2022

Word count: 20060

University of BRISTOL

DEPARTMENT OF COMPUTER SCIENCE

SUMO: A System for Automatically Summarising UK House of Lords (UKHL) Judgments with Majority Opinion

Amy Eileen Conroy

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering.

September 13, 2020

Figure 2.2: Past MSc projects from Daniels and Conroy

3. Project Overview

In envisioning the scope and intended outcome of our project, two significant considerations were kept ever present. One is the greater eventual pipeline that our tool is finally intended to be integrated with. Awareness of this helped narrow the focus of development and provide a clear intended objective. Also greatly relevant were prior completed projects by peers under Dr Oliver, both to draw inspiration for functionality to test, and to work to distinguish my project outcome as distinct and intended to work in concert with them.

In providing the breakdown of the functionality presented by our tool in our report, the following narrative structure was used: *Premise* sets out an overview of the function to be implemented. Next, o*bjectives* state the design goals set for implementing said functionality. Lastly, *execution* provides the detailed breakdown as to how the functionality was implemented.

3.1. Programming Languages and modules

In selecting a design environment to build our tool, several different languages were tested. Ultimately, we decided on developing a web tool, and correspondingly would be using the three cornerstone languages of web design: HTML for the site markup, Javascript for the scripting, and CSS for styling. This option was selected for several reasons: the first is the easy access to learning tools. Sites such as GeeksforGeeks.org[16] and w3schools.com[17] provided easy to pick up tutorials, and using a common environment allowed us to tap into the vast experience of other developers, such as via StackExchange.com[18] for troubleshooting help and suggestions. The second consideration was the lightweight nature of a web development, which brought several benefits: Firstly, no proprietary IDE was needed to work on these three languages, only a simple code editor (Visual Studio Code[19] was the editor selected). This made it easy to work portably and across several machines. Finally, the decision was made in consideration of the usability of the final tool for future developers and editors. A web tool would be easy to pass between users and be worked on with low overhead, making it a sensible choice.

3.2. Annotation

Premise - From the inception of the project, it was decided that annotation of the SUMO corpus would be the primary intended function of this project. This direction was due to this functionality being the most pressing shortage noted in discussions with Dr. Oliver Ray. As we have discussed prior, much of further work being conducted within the work group were reliant on the SUMO corpus. Hence at present, a major area of limitation for existing projects, particularly those seeking to train machine learning tools, was the relatively limited set of annotated cases. The past method of annotation conducted by Hachey and Grover, namely manual annotation directly from the raw legal text by legal experts, is a time consuming and expensive process, a task that has proven difficult to be repeated on an infinitely expanding library of UKHL/UKSC judgements. Hence, it was determined that a such a task, if simplified and made more approachable, would be highly beneficial to other ongoing projects while being an achievable aim within the scope of this project.

Objectives - We envisioned the design of a simple and user-friendly annotation tool that would enable easy editing of the SUMO corpus. Said annotation function should enable an editor to easily navigate entries in the .csv and view its annotation tags. Said editor should then be easily able to add to and correct annotations. Finally, the tool should be able to output the changes back into a .csv file.

Execution - Our tool achieved this functionality in the form of the “Case” page of our site. The page may be described as follows: the central element of the page is the view of a breakdown of each entry in the case by sentence. By default, the page is set to “View” mode, meant to allow a reader to view important information about each sentence of a case. However, by means of a toggle switch fixed to the bottom right of the screen, the reader can access “Edit” Mode. In Edit Mode, a permanent fixed “Edit” column appears on the on the right sidebar. Said column presents the annotation entries relevant to the selected line in the case, except that the annotation values are now editable. The Edit panel also includes ‘previous’ and ‘next’ buttons to facilitate an editor’s quick navigation between lines. Lastly, the footer of the Edit panel provides

buttons for the editor to save and export their suggested changes as a .csv file. These changes can then be assessed and merged with the original corpus.

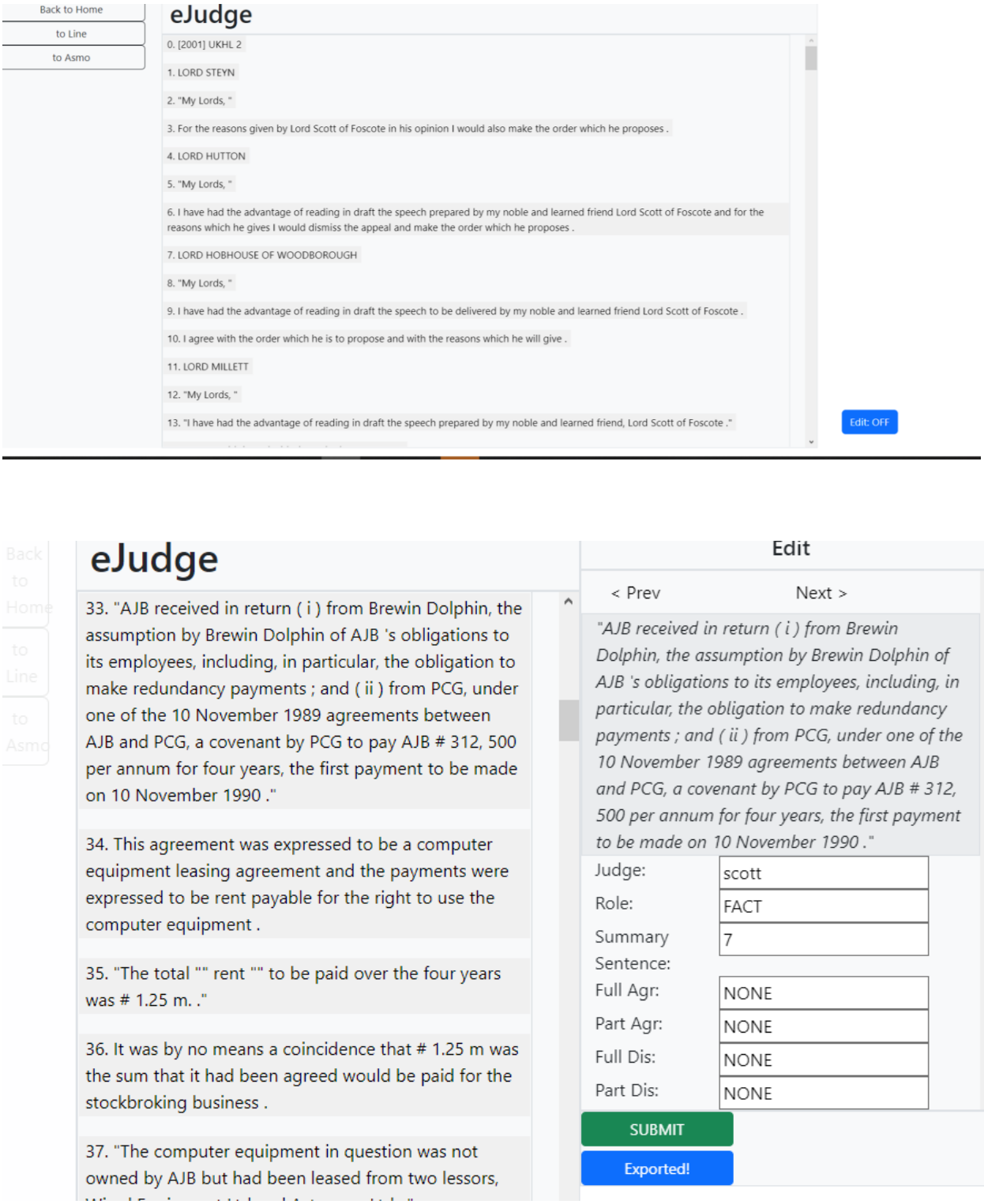


Figure 3.1 and 3.2: 'Case' page view

3.3. Open-Source Design

Premise - An area of concern that was identified in our review of prior projects within the work group, particularly those conducted by prior MSc students under Dr. Oliver, was the lack of integration between the projects developed by each student. Each project was built and operated independently, where there might have been significant time-savings and gains from building upon one another. This complication has meant that, while each individual project on their own were well built and performed excellent in their own spaces, they were evolutionary dead ends for the purpose of future expansion. This has also meant some degree of redundancy in each's functions, as each student had to replicate functions already created prior. For instance, prior to developing their independent improvements to the SUMO pipeline, Imansyah and Conroy had to respectively reconstruct Hachey and Grover's older pipelines from scratch, which wasted valuable time.

Objectives – We envisioned pursuing a software engineering approach that emphasized open-source design principles, such as modding and ease of integration with other tools. Rather than introducing a discrete functionality with this segment, the goal of was to apply software engineering techniques to our codebase to ensure that future designers will easily be able to not only use our tool but modify it to their needs as the SUMO pipeline develops.

Execution - This functionality was a substantial challenge given the constraints of the MSc project. We had to weigh the time commitment to refactoring existing code to improve modularity, versus expanding the uses of our tool. Ultimately, approximately three weeks were spent learning better coding practices and rebuilding our site to be more futureproof.

One example of this in practice is the evolution of the functions found in **data.js** file of our site. This script includes all the data processing functions used across our site, including the reading and writing of the corpus .csv (readFile()), and the storing of temp changes to the browser (getCaseData()). For instance, our initial implementation of data extraction of the CSV was hardcoded to the existing header structure, which was viewed as a source of parsing errors in the future.

```
[..]
for(var i=0;i<details.length;i++){
        var e = createDiv("row bg-transparent text-start", "");
        var text = "";
        if (i == 3){ // Judge
            text = "Judge: " + details[5].toUpperCase();
        }else if(i == 4)){ // Sentence
            e.style.fontStyle = "italic";
            text = details[getHeaderIndex("text")];
        }else if(i == 5)// Role
            text = "Role: " + details[i].toUpperCase();
        }else if(i == 6) // Summary alignment
            text = "Aligns with sentence " + details[i] + " of summary.";
        }
    }
}
```

```
function getHeaderIndex(str){
    var headers = JSON.parse(localStorage.getItem("Headers"));
    return headers.indexOf(str);
}
function getHeaderTag(index){
    var headers = JSON.parse(localStorage.getItem("Headers"));
    return headers[index];
}

[..]
for(var i=0;i<details.length;i++){
```

```
        var e = createDiv("row bg-transparent text-start", "");
        var text = "";
        if (i == getHeaderIndex("judge")){ // Judge
            text = "Judge: " + details[5].toUpperCase();
        }else if(i == getHeaderIndex("text")){ // Sentence
            e.style.fontStyle = "italic";
            text = details[getHeaderIndex("text")];
        }else if(i == getHeaderIndex("role")){ // Role
            text = "Role: " + details[i].toUpperCase();
        }else if(i == getHeaderIndex("align")){ // Summary alignment
            text = "Aligns with sentence " + details[i] + " of summary.";
        }
```

Figure 3.3: Original vs. Revised Codebase excerpt

Upon refactor, said CSV parser instead dynamically initialized a list consisting of the corpus headers upon load the file, and saves it to the browser (csvtoArray()). Paired with this change is getHeader() functions to query the temp header list saved. Refactors, such as those described above, would allow our tool to work with future developments to the SUMO pipeline.

3.4. Analytical tools/Learning Aids

Premise - Daniels' JudgeViz demonstrated that the data from the SUMO corpus can be used to create visual aids to allow a more user-friendly reading experience and extended analysis of cases. The proliferation of legal research tools across jurisdictions also points to a demand in the industry for such a functionality. As our tool is already used to extract data from the SUMO corpus, it was sensible to mirror Daniels' work and use said data in the development of learning tools.

Objectives - Building on Daniels' prior work, we aimed to expand on the processing of the annotated data from our editor to develop visual tools for readers.

Execution - This implementation took the form of a "ASMO" page on our site, that extracted data from the stored case to generate visual aids. In our demonstration, two simple and useful graphs were generated. The first was a measurement of the proportion at which each judge constituted the whole sum of the judgement. This, it is argued, is a useful pattern to be identified by the reader as it points to the argument structure and degree of conflict between the panel of judges. A judgement with a relatively skewed distribution, where few judges contribute most of the decision, point to a relatively uncontested decision with the remaining judges mostly voicing agreement. On the other hand, a mixed distribution with multiple substantive judgements would suggest a contested decision, pointing out that multiple judges saw the need to voice their respective arguments.

The second graph form that was tested was the "Majority Opinion" Graph, that mapped the flow of argument between the judges in more detail than the simpler distribution graph mentioned above. Said graph consisted of a concentric doughnut shape. The major outer ring denoted all

the judges involved in the judgement, and their respective color tag. For each judge, the four layers of inner rings then denote the possible relationships with other judges: Full Agreement, Partial Agreement, Full Disagreement and Partial Disagreement. Upon initiation, each of these layers are blank. When a relationship is identified within the corpus, said judge's colour is then filled into the respective inner ring. At a glance, this visual aid allows a user to see a relatively complex map of the arguments present in the judgement. The map of an uncontested decision, for instance, would likely show blank disagreement ring layers, and only one or two colors in the agreement rings, denoting the majority opinion. On the other hand, a contested decision would see all four rings heavily coloured, and the user would be guided to inspect each judge for their respective contributions.

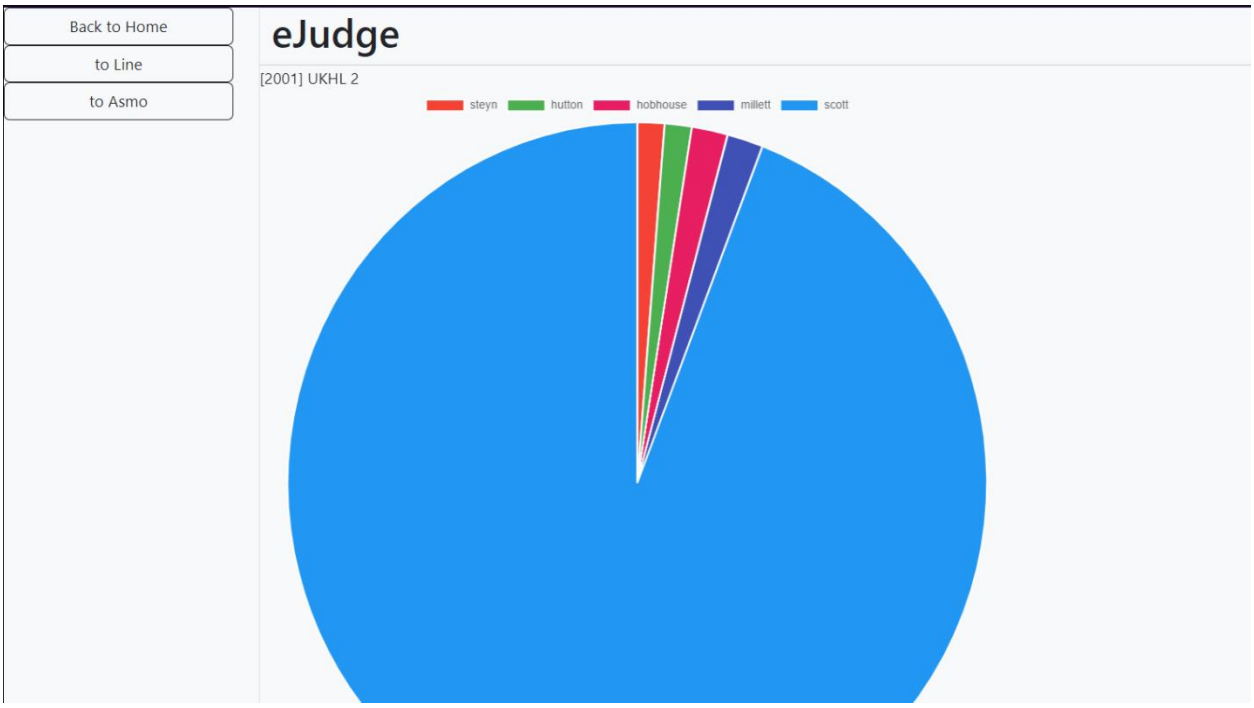


Figure 3.4: Breakdown Graph

4. Evaluation

This section outlines the approach taken to evaluate the efficacy of our tool in achieving its stated objectives.

4.1. Conduct of Evaluation

Our primary method of evaluation took the form of a user survey and exercise conducted online and in person over the span of two weeks. All in all, we managed to secure the participation of 10 volunteers. To gauge the accessibility of our tool, the decision was made to not limit the survey participants to individuals familiar with the UK legal system. Instead, participants were asked to disclose their familiarity with the UK legal system, of which 3 of the participants described themselves as possessing “familiar” and “very familiar” levels of understanding. There were a few grounds for this decision. Firstly, we decided that given that our tool is intended to improve the accessibility of the annotation workflow for all users, it was worth including non-legal experts in the pool of participating candidates to see if familiarity with the legal system would still constitute a substantial factor in the candidates’ ability to complete the tasks given during the survey. Another consideration was the limited time and resource available to conduct our evaluation. Within the limited window of opportunity allocated, we judged it unlikely that we would be able to secure a large enough group of participants strictly with experience with the UK legal system for our survey. Despite the reasonings provided above, we, however, would like to acknowledge the risks involved with involving non-legal experts in our evaluation of a tool intended for use in a pipeline thus far conducted by legal academics and professionals. We were aware that this qualification might potential lower the quality of the results of our survey than had we restricted the participants. However, it will be argued that thankfully, our evaluation results did not show this to be the case.

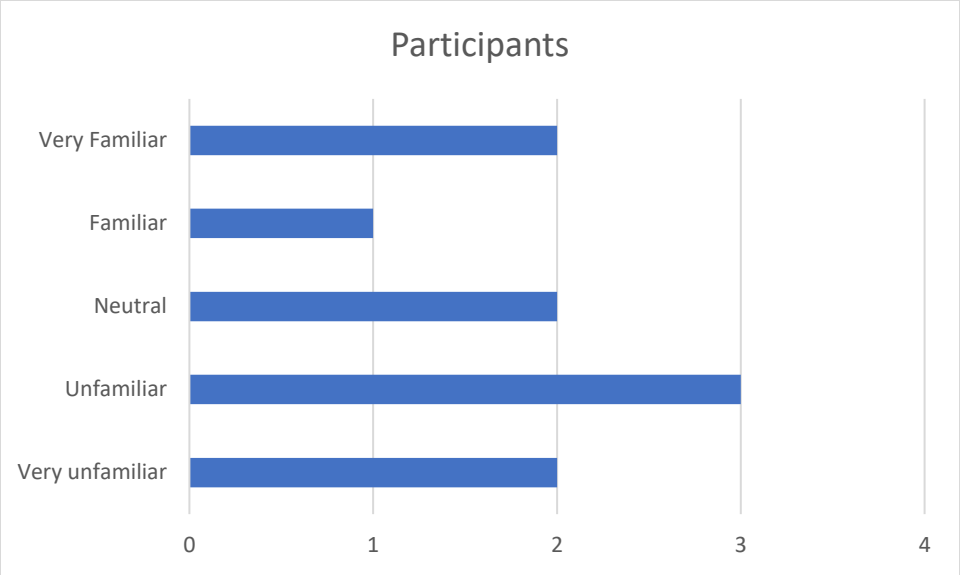


Figure 4.1: Survey of participants' legal knowledge

Another major omission in our evaluation that must be mentioned is the lack of evaluation of the second design element of our tool, namely its open-source design. The primary factor that contributed to this omission was the challenge of designing an exercise to assess this factor within the context of a short user survey. It is envisioned that the evaluation of this design element would have had to take the form of a software engineering exercise, where participants would be surveyed based on their ability to build some simple tools with the help of eJudg, as opposed to without. Ultimately, it was judged that such a test would be impossible to conduct within the given timeframe and with a general group of survey participants.

4.2. Annotation

Having qualified the scope of our survey, we would proceed to describe how the evaluation was conducted. To test the annotation function, Participants were provided a set of tasks to complete involving the manipulation of entries within the corpus. For each task, the participants were given a test portion of the corpus, followed by tasks to complete within a timed period. The types of tasks were as follows:

1. Editing of data entries: Participants were given a work sheet of entries to edit and were asked to complete correctly edit as many edits on the test corpus as possible.

2. Inspection for mistakes: Participants were given access to a reference sheet showing the 'correct' entries for the data in the test corpus and were then asked to spot as many errors in the test corpus as possible.

The exercise would be conducted in two rounds. In the first round, participants would edit the test corpus directly in Microsoft Excel. They would then be asked to complete the same task on eJudg by loading the test corpus into the tool's editor. Several steps were taken to ensure the integrity of the results collected. Firstly, we prepared two sets of test corpus to be used in both rounds of the test. Each participant was then randomly provided either corpus for their tasks. Lastly, the participants were not informed of the timing window for which they would be assessed. This allowed them to gain a level of familiarity with the tools before being assessed.

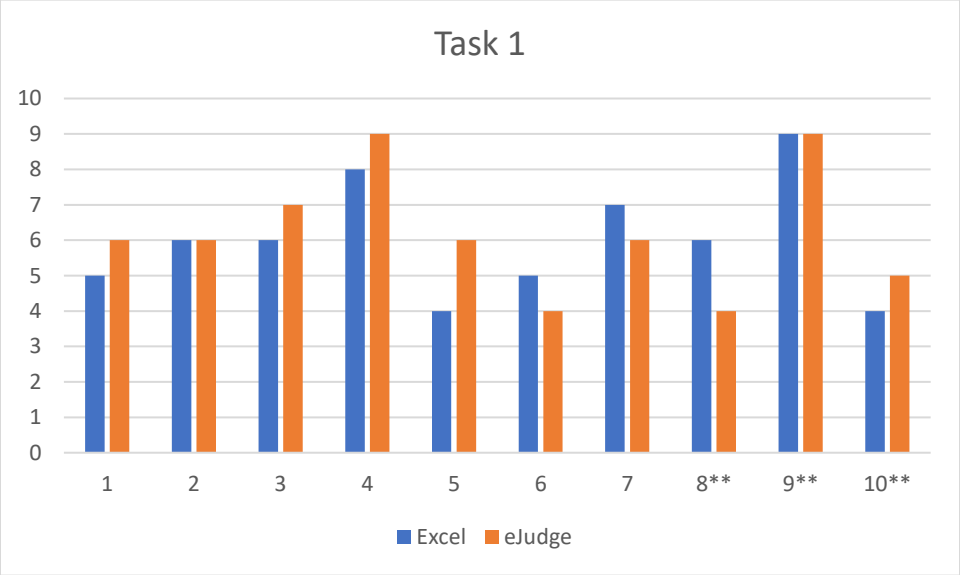


Figure 4.2: Task 1 Results

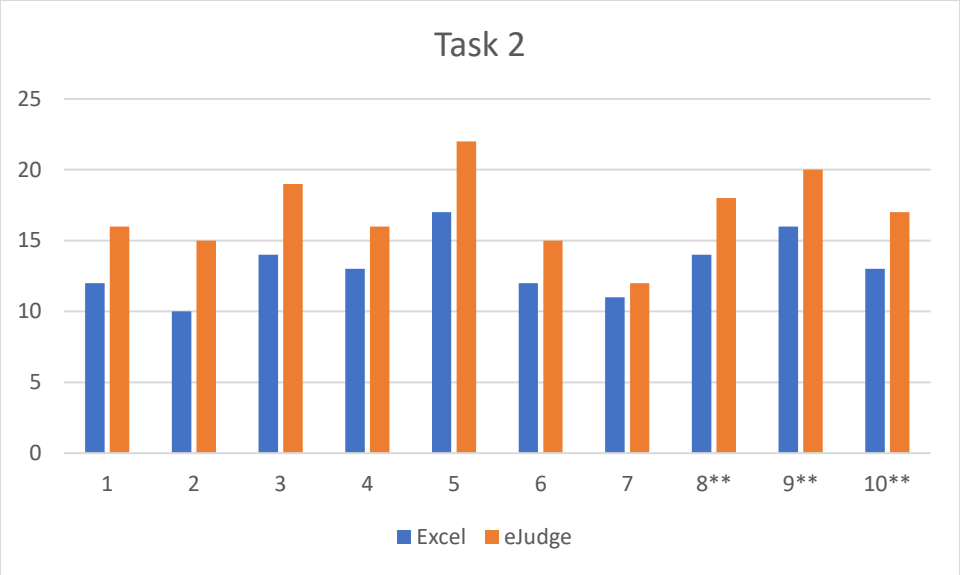


Figure 4.3: Task 2 Results

Figure 4.2 and 4.3 shows that using eJudg improved the work speed of participants on average across all tasks. On average on task 1, participant showed an improvement of output by 5.6%. However, some participants did not demonstrate any difference in performance between editing the csv directly, and when using eJudg. Participants 3, 4 and 5 were even less efficient when using eJudg, being 20%, 14.2%, and 33% slower respectively. It may be argued that this result was due to the speed of navigating each entry corpus via eJudg, whereas a quick user can navigate the corpus via Microsoft Excel rapidly, which can save work time. On the other hand, the improvement when using the provided tool for task 2 appears to be more dramatic: all participants showed a positive improvement of an average of 29% at spotting errors, with participant 3 showing the highest improvement of 35.7%.

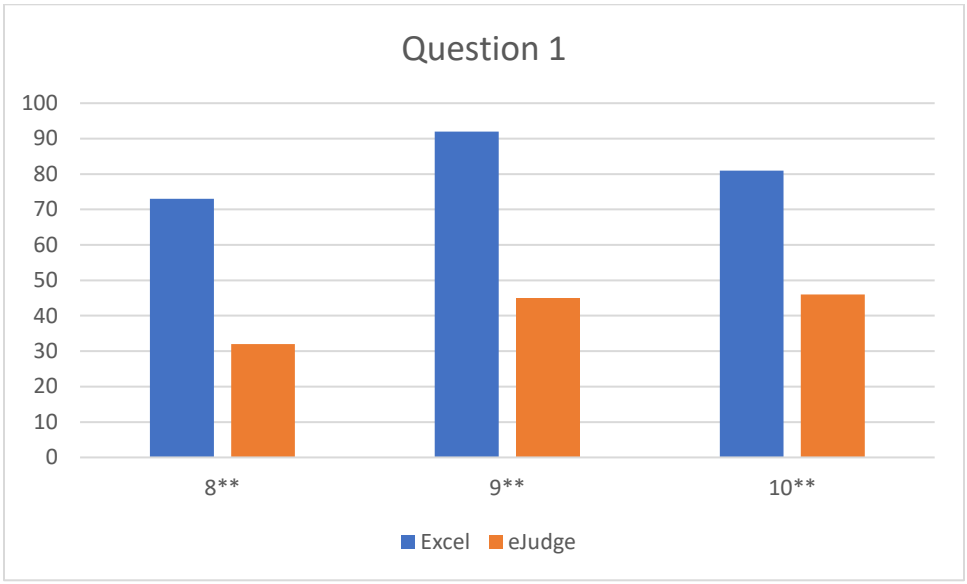
4.3. Judgement Analysis

To evaluate the analytical tools, participants were given the raw text of a legal judgment and given a questionnaire to answer regarding its contents. The time taken for them to complete the questionnaire was recorded. Questions provided were as follows:

Q: What was the majority opinion of the case?

Q: Which judges were in favor of the final judgement? Which dissented?

This test was then repeated with the users allowed to use eJudg to assist their analysis. To improve the user experience, it was judged necessary to directly instruct users on the utility of the graphs shown in eJudg. For instance, it was explained how the judgment breakdown graph showed which judged presented the bulk of the reasoning, and how the MO graph showed the relationships between each judge.



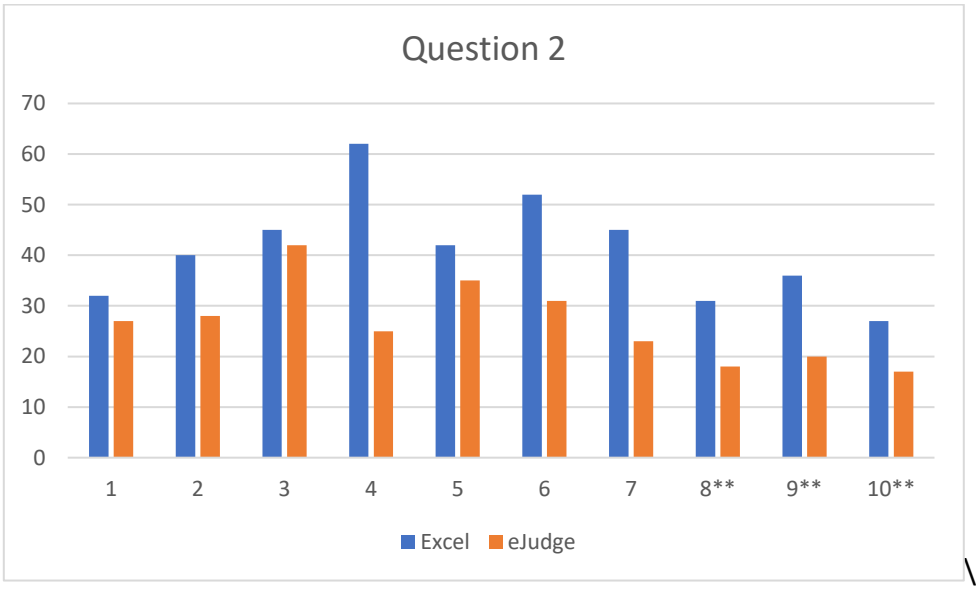


Figure 4.3 and 4.4: Judgement Analysis Results

Figure 4.3 and 4.4, the results of the exercise, are grouped according to the participants' familiarity with the UK legal system. The results clearly show that all users were significantly faster at answering the questions once given eJudg, averaging an improvement of 34.1%. One modification was made to the questionnaire mid trial that must be mentioned: We decided after several tests to modify the list of questions for participants not familiar with the legal judgments (Participants 1-7). Specifically, we ultimately decided to remove question regarding the majority opinion of the judgement as we found users were spending a lot of time struggling to answer this question, needing substantial guidance. This modification was judged as reasonable and to not significant change the quality of the results of the questionnaire.

4.4. User Feedback

Lastly, the participants were asked to complete a questionnaire rating their experiences on their survey. This questionnaire consisted of four questions, listed below:

Q1: After using our tool(eJudg), did you found it helpful for completing the tasks provided?

Q2: After using our tool(eJudg), did you find it helpful for improving your understanding of legal judgments?

Q3: After using our tool(eJudg), did you find it easy to learn?

(Optional) Q4: Please provide us with any additional feedback regarding your experience. Participants are welcome to comment on their general experience with the exercise, or specific to their experience using eJudg.

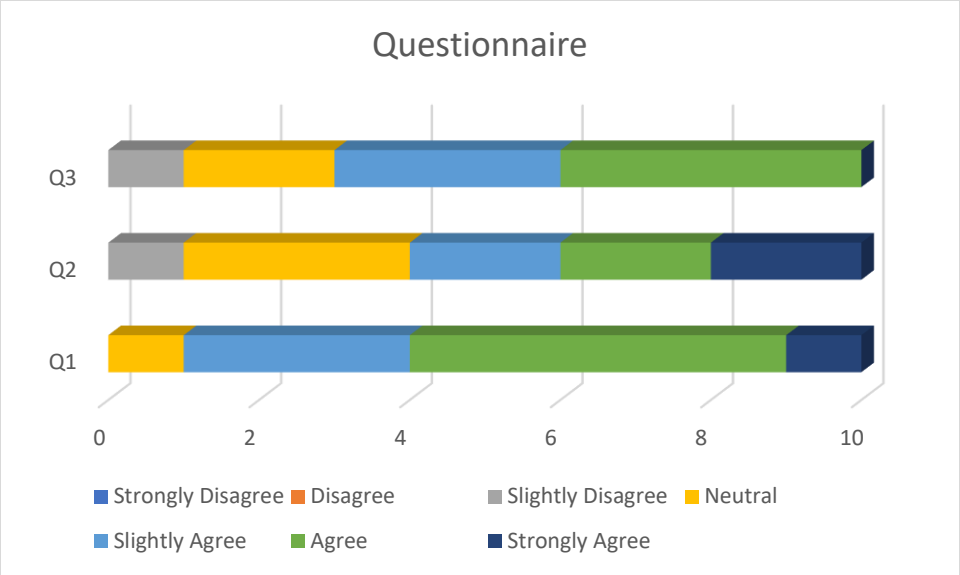


Figure 4.5: Questionnaire Results

For questions 1-3, participants were asked to provide their answer on a seven-point Likert [20] scale ranging from "Strongly disagree" to "Strongly agree". Question 4 was a freeform prompt, where participants were given the chance to leave any additional comments that did not fall within the three rated questions above. The results of our questionnaire showed a strong indicator that eJudg was viewed positively by our participants, with 70% agreeing (either somewhat agree, agree, or fully agree) that our tool helped them with their tasks and improved their understanding of cases. Of note was that of our 3 users who expressed familiarity with the UK legal system, 2 answered neutrally or negatively to the second question. This suggests that the analytical functions of eJudg may not be as particularly useful to legal experts at improving their understanding of judgments. It is posited that this indicates an area for future development.

Of the 10 participants of the survey, four opted to leave additional comments on their experience. These comments pointed to several areas of improvements, such as the new user experience of the tool ("*I did not understand how to use your tool until you explained it")* and the general aesthetics ("*The layout is confusing. Some color [and instructions] would be better")*

4.5. Summary

In summary, our evaluation strongly indicates that our tool is effective at its purpose of aiding users with the relevant annotation and analytical tasks. The results showed that most users preferred using of eJudg when asked with the tasks provided. It is particularly interesting to note that both users with and without legal experience offered positive sentiments regarding their use of the project tool. However, our survey also highlighted several criticisms with the current iteration of the tool. Note should be made to criticism of the user experience, as many users suggested ways to improve the user interface. Such remarks point to necessary further work to improve the user experience for the site.

5. Conclusion

5.1. Limitations

As mentioned in prior contributions, the primary limitation of this project was the limit of our dataset to the prexisting annotated cases in the SUMO corpus. While undoubtedly a substantial effort by prior projects and manual editors, the corpus represents a minute portion of UKHL/UKSC judgements produced over the lifetime of the institution. This would mean that no guarantee may be made that eJudg's annotation and analysis functions would be compatible with the entire archive of UKHL/UKSC caselaw. Indeed, one recent change in the structure of UK Supreme Court judgements may be commented on. Recently, the Supreme Court has moved to allow judges to reduce their position to a statement of agreement with another judge on a judgement. This is a departure from the past structure in a manner that would affect our existing qualifiers, as thus far our corpus has assumed a one-to-one relationship between an judgement and the judge. This would be a challenge for our existing analytical functions, for instance the Majority Opinion graph, which would need to be adjusted to account for corpus entries being associated with multiple judges.

On a similar note, we would argue that our analytical tools sufficiently tested only a small subset of the data present within our corpus document. In its current iteration, our analysis of the corpus data has been limited very simple patterns, i.e the proportion of judge's contributions, and the relationships between each judge. Furthermore, as prior work by Conroy shows, there remains a wealth of information that remains unannotated in the current iteration of the corpus. It was hoped that some experimentation. However, we were unable to implement this function into our tool within the time allowed.

5.2. Contribution

Our project, it is argued, contributed a small but significant role to the ongoing work: a functional editing tool. Following positive observations from our user evaluation, we are confident that our tool represents a major quality-of-life (QoL) improvement in the ongoing work for annotators and software engineers alike. Further, we argue that our tool successfully builds

upon the work by Daniel in demonstrating the front-end use of the corpus pipeline for generating learning and research tools.

5.3. Future work

As covered within our section on limitations, we identified room for expansion of the annotation functionality of eJudg to allow new annotation categories. While it must be stressed that even the existing qualifiers remain underutilized, additional functionality to facilitate modification to the schema of the corpus would be a great expansion to the usefulness of our tool for annotators.

The positive reception to the learning aids demonstrated on eJudg also suggest this as an area for improvement and investment. With our tool enabling easier editing of annotations, and the introduction of a modular design language, we argue that our tool has potential to be used on future projects. Existing online legal aid tools, such as those covered prior in our Backgrounds sections, are limited in their suite of analysis tools. We foresee that eJudg may be used to address this shortcoming, were it developed further.

The last area for further work to be mentioned is support for, a database. As covered in earlier, working from a local Excel document does pose its benefits, such as ease of data transfer between members of the project group. Nonetheless, it is suggested that a logical next step would be the integration of a permanent data storage system shared between all users. This feature is further desired in the context of our eJudg given that it is meant to provide annotation functionality to the group. The editing process currently would rapidly provide divergent versions of the corpus due with each edit. A central repository would allow. With permanent data storage, another potential expansion of our tool's functionality would be to review and edit submitted changes for correction.

Bibliography

1. *J. Alder, Dissents in courts of last resort: tragic choices?, Oxford Journal of Legal Studies, 20 (2000), pp. 221–246.*
2. *https://www.law.cornell.edu/wex/stare_decisis*
3. *B. Kerr, Dissenting judgments–self-indulgence or self-sacrifice, The Birkenhead Lecture, London, 8 (2012), p. 10.*
4. *C. Law: THE COMMON LAW AND CIVIL LAW TRADITIONS. Available: https://www.law.berkeley.edu/wp-content/uploads/2017/11/CommonLawCivilLawTraditions.pdf*
5. *W Leech, Revisionism in the House of Lords: The Bastion of Rigid Stare Decisis Falls, Harvard Law Review Vol. 80, No. 4 (Feb., 1967), pp. 797-803 (7 pages)*
6. *B. Hachey and C. Grover, Extractive summarisation of legal texts, Artificial Intelligence and Law, 14 (2006), pp. 305–345.*
7. *LexisNexis: https://www.lexisnexis.com/en-us/gateway.page*
8. *Westlaw UK: https://legalsolutions.thomsonreuters.co.uk/en/products-services/westlaw-uk.html*
9. *Google Docs: https://www.google.com/docs/about/*
10. *Zoho: https://www.zoho.com/writer/*
11. *Bit.ai writer: https://bit.ai*
12. *J. Valvoda, Annotation guide 2.0, 2018.*
13. *O. Ray, A. Conroy, and R. Imansyah, Summarisation with majority opinion., in JURIX, 2020, pp. 247–250.*
14. *M.Daniels, judgeViz A new approach to improving information retrieval from UKHL judgments. September 2022.*
15. *https://www.geeksforgeeks.org/javascript/*
16. *https://www.w3schools.com/html/*
17. *https://codereview.stackexchange.com/questions/tagged/javascript*
18. *Visual Studio Code: https://code.visualstudio.com*

19. *Recent judgements, e.g. On Tower UK Ltd (formerly known as Arqiva Services Ltd) (Appellant) v AP Wireless II (UK) Ltd (Respondent). Available: https://www.supremecourt.uk/cases/docs/uksc-2019-0225-judgment.pdf*

Appendix A: HTML Code Sample

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
lang="en-GB" xml:lang="en-GB">
<head>
<meta charset="UTF-8"/>
<title>eJudg(WIP) - Home</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2J10vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
<script src="./JS/data.js"></script>
<script src="./JS/main_search.js"></script>
<script src="./JS/main_edit.js"></script>
<script src="./JS/main_nav.js"></script>
</head>
<body>
<main>
    <div class="container text-left d-grid gap-2">
        <div class="row border">
            <div class="col themed-grid-col"><h1>eJudg</h1></div>
        </div>
        <div class ="row">Choose Corpus: </div>
        <div class="row"> <input type="file" id="CorpusField"
accept=".csv"></div>
        <div class="row">
            <div class="col-xs">
                <button id="SearchButton">Search</button>
                <button id="EditButton">Edit</button>
            </div>
        </div>
```

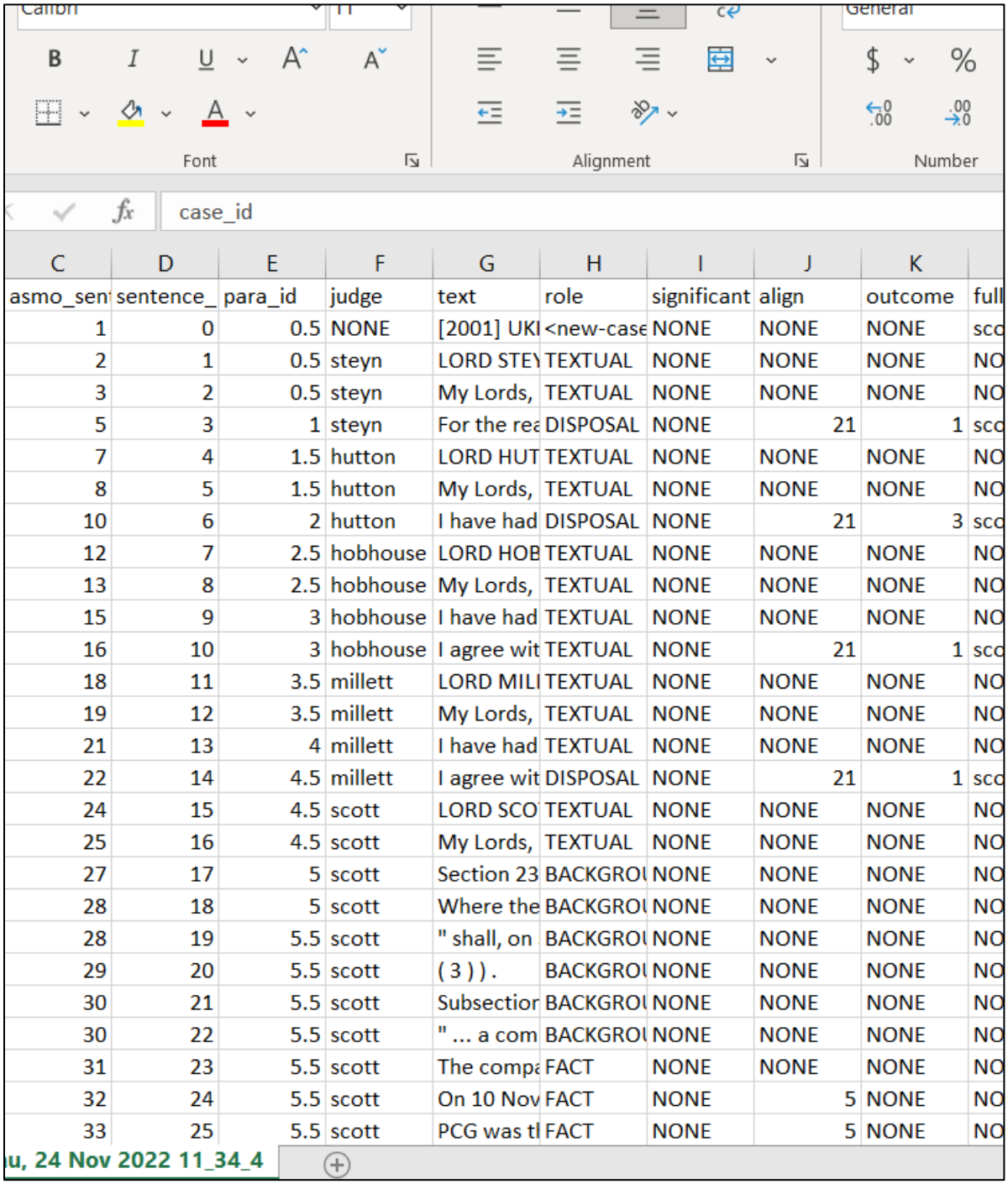
Appendix B: CSS Code Sample

```
/*MAIN*/

.main {
    overflow:hidden;
}

.btn {
    max-width: 15vw;
}

/*LINE*/
#main #sidebar #edit{
    min-height: 100vh;
    transition:.5s;
}

#mainbody{
    overflow-y:scroll;
    height:90vh;
}

#edit {
    position:fixed;
    top:0;
    right:0;
    opacity:0%;
    transition:.4s;
}

#toggle {
    position:fixed;
    max-width:10vw;
    bottom:5vh;
    right: 5vw;
}

#editbody{
```

Appendix C: Javascript Code Sample

```
"use strict"

var debug = 1;

function csvtoArray(str){
    if(debug){
        console.log("DEBUG: Init csvtoArray()");
    }

    var headers = str.slice(0, str.indexOf("\n")).split(",");
    localStorage.setItem("Headers", JSON.stringify(headers));

    var rows = str.slice(str.indexOf("\n") + 1).split("\n");
    rows = splitSentence(splitRowElements(rows));

    const cases = splitCases(rows);
    return cases;
}

function readFile(field) {
    return new Promise(function(resolve) {
        if (debug){
            console.log("DEBUG: Init readFile()");
        }

        const file = document.getElementById(field).files[0];
        if(!file){
            alert("ERROR: Null file received!")
        }else if(file.type != "text/csv"){
            alert("ERROR: Invalid file received!")
        }else{
            const fileLoaded = new Promise((resolve) =>{
                const reader = new FileReader();
                reader.addEventListener("load", function(e){
                    var text = e.target.result;
                    text = csvtoArray(text);
                    resolve(text);
```

Appendix D: SUMO Corpus Excerpt



Font Alignment Number

case_id

C	D	E	F	G	H	I	J	K	
asmo_sen	sentence_	para_id	judge	text	role	significant	align	outcome	full
1	0	0.5	NONE	[2001] UKI	<new-case	NONE	NONE	NONE	sco
2	1	0.5	steyn	LORD STEY	TEXTUAL	NONE	NONE	NONE	NO
3	2	0.5	steyn	My Lords,	TEXTUAL	NONE	NONE	NONE	NO
5	3	1	steyn	For the rea	DISPOSAL	NONE	21	1	sco
7	4	1.5	hutton	LORD HUT	TEXTUAL	NONE	NONE	NONE	NO
8	5	1.5	hutton	My Lords,	TEXTUAL	NONE	NONE	NONE	NO
10	6	2	hutton	I have had	DISPOSAL	NONE	21	3	sco
12	7	2.5	hobhouse	LORD HOB	TEXTUAL	NONE	NONE	NONE	NO
13	8	2.5	hobhouse	My Lords,	TEXTUAL	NONE	NONE	NONE	NO
15	9	3	hobhouse	I have had	TEXTUAL	NONE	NONE	NONE	NO
16	10	3	hobhouse	I agree wit	TEXTUAL	NONE	21	1	sco
18	11	3.5	millett	LORD MILL	TEXTUAL	NONE	NONE	NONE	NO
19	12	3.5	millett	My Lords,	TEXTUAL	NONE	NONE	NONE	NO
21	13	4	millett	I have had	TEXTUAL	NONE	NONE	NONE	NO
22	14	4.5	millett	I agree wit	DISPOSAL	NONE	21	1	sco
24	15	4.5	scott	LORD SCO	TEXTUAL	NONE	NONE	NONE	NO
25	16	4.5	scott	My Lords,	TEXTUAL	NONE	NONE	NONE	NO
27	17	5	scott	Section 23	BACKGROU	NONE	NONE	NONE	NO
28	18	5	scott	Where the	BACKGROU	NONE	NONE	NONE	NO
28	19	5.5	scott	" shall, on	BACKGROU	NONE	NONE	NONE	NO
29	20	5.5	scott	(3)) .	BACKGROU	NONE	NONE	NONE	NO
30	21	5.5	scott	Subsectior	BACKGROU	NONE	NONE	NONE	NO
30	22	5.5	scott	" ... a com	BACKGROU	NONE	NONE	NONE	NO
31	23	5.5	scott	The compa	FACT	NONE	NONE	NONE	NO
32	24	5.5	scott	On 10 Nov	FACT	NONE	5	NONE	NO
33	25	5.5	scott	PCG was tl	FACT	NONE	5	NONE	NO

u, 24 Nov 2022 11_34_4

21.