Rate My Tweet: Understanding Comparative Judgement in the Wild

Andy Gray

445348

Submitted to Swansea University in partial fulfilment of the requirements for the Degree of Master of Science

Department of Computer Science

Swansea University

30th September 2021

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... (candidate)

Date ....

Statement 1

This work is the result of my own independent study/investigations, except where otherwise stated. Other sources are clearly acknowledged by giving explicit references. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure of this work and the degree examination as a whole.

Signed .... (candidate)

Date ....

Statement 2

I hereby give my consent for my work, if accepted, to be archived and available for reference use, and for the title and summary to be made available to outside organisations.

Signed .... (candidate)

Date ....

I would like to dedicate this work to . . .

Abstract

Marking and feedback is such an essential part of teaching and learning. For students to improve, they need to receive feedback. However, for the students to receive the feedback, the teachers need to mark it. Marking takes a considerable time for the teacher to complete and creates a significant cognitive load within the process. Therefore an alternative approach to marking called adaptive comparative judgement (ACJ) has been proposed in the educational space. ACJ has derived from the law of comparative judgment (LCJ), a pairwise method that compares and ranks items. While studies suggest that ACJ is highly reliable and accurate while making it quick for the teachers, alternative studies have questioned this claim suggesting that the process can bias the results through its adaptive nature. Additionally, studies have also found out that the ACJ can result in the overall marking process taking longer than a more traditional method of marking. At the same time, the current ACJ applications provide little resources in personalised feedback to individual students.

Therefore, we have proposed a new ranking system that can rank the outcomes from the comparative judgement marking approach. The alternative ranking system was the Elo system. Additionally, aiming to reduce teachers cognitive load, reduce the time required to mark and ultimately provide personalised feedback to the user using NLP techniques. We experimented on Twitter tweets around the topic of Brexit to ask users what tweets they found funnier. The findings found that the Elo system is a suitable system to use for ranking the tweets outcomes. However, the NLP feedback process results provided good building blocks for future experiments that did not have a positive impact as desired.

Acknowledgements

This is an opportunity to acknowledge and thank those who have supported you throughout your studies. Friends and colleagues who you have studied alongside, your families, and your mentors within the department are the usual suspects.

Contents

Chapter 1

Introduction

We have set out to create a tool that can simulate a small scale comparative judgement experiment on what users think about tweets getting compared against each other. This experiment is in light of our stakeholder getting commissioned by the Welsh government to implement a comparative judgement system nationally for all schools in Wales. Comparative judgement is a technique that has been around for almost 100 years [1]. However, while the process can improve results and reduce cognitive loads for teachers and markers, especially at the scale that the stakeholder's implementation will have to work at, it can still require many combinations to be marked and compared. For this experiment, we decided to use tweets based on Brexit to see wat ones people found funnier.

Therefore, we have created a tool that allows users to see a sub-sample of the combinations. Once the users have viewed the varieties, an overall ranking of the results will get created. Two methods got implemented, a more traditional comparative judgement method and an Elo style ranking.

We then aimed to use NLP techniques to extract any insights we could find within the tweets. We intended to extract information on the tweets to see if we could find patterns that would give us insights into what might have impacted the tweets final scores.

The study got broken up into two parts. Part one was a web app to gather user's views on the tweets, and the second part was exploring NLP techniques within a Jupyter Notebook. With our aim to see if we can generate any feedback about the tweet.

1.1 Motivations

For the prior eight years, we have had involvement in some form of an educational environment. Seven of these years involve being a teacher within secondary and sixth form schools. While the focus of teaching is perceived to create lessons for students to learn and grow, we found more and more as the years went on that this wasn't the case. The focus was actually on providing reports about the students, which required data about the students from formal assessments. While having assessments to gauge the level that a student is at is an essential part of education. However, creating, marking, analysing and providing feedback for 30 students or more per class is a time-consuming task. Therefore, this assessment practice takes away the educators' time to do what is essential, creating meaningful lessons tailored for the students.

Therefore, our motivation is to create a tool for educators that will empower them to allow technology to do what it is good at and focus on what they are good at, while aiding teachers with their decision making and allowing them to create and delivering lessons. To shape future generations views.

1.2 Existing Liturature

Within education, teaching and learning have provided assessments to rank students' attainment since 1988s [2]. Due to the students getting assessed, this allowed the teachers to give feedback to learners, allowing them to improve, especially with the introduction of Key Stage (KS) 1, 2 and 3, national curriculums and tests [3, 4]. This newfound focus brought about new areas of tools and techniques for teachers to use. These new tools are called Assessment of Learning (AoL) and Assessment for Learning (AfL) [5, 4, 6]. However, marking and providing feedback can be quite a time-consuming labours task, adding to workload and teacher stress. Especially when school marking policies are in place, and a certain level of marking needs to get done within a specific time frame. Additionally, teachers might implement bias towards students results by basing performance results on how they have done all year, rather than in the face value of the actual assessment.

However, a newfound focus on an approach called Adaptive Comparative Judgement (ACJ) has started to make some traction [7]. ACJ is an altered approach to Louis Leon Thurstone's the Law of Comparative Judgement (LCJ) [8]. The LCJ and ACJ both provide a combination of examples and asks the user to judge which one out of the two is better.

However, ACJ is the method proposed more within education based on its ability to be 'adaptive' in comparing the students work. Instead of every combination getting seen, it can change to make pieces of work that are considered similar get compared more to find out which one is better. ACJ claims to be highly accurate, reduce teachers' workload, and provide meaningful feedback to the students [9]. However, a study found out that the method used within ACJ (rounds) makes the results biased, especially the more rounds there are. This bias demonstrates that being 'adaptive' has no more effectiveness over just having random pairings at all [10]. Some studies also found that the ACJ can take longer than stardard marking using a rubric [11, 12].

Additionally, the feedback it provides is very minimal. Therefore, students do not receive any form of personalised feedback. Instead, they have to rely on their understanding of the task and then extract what they think is important based on their peers' work. As a result, likely to be excluding low ability students from gaining meaningful insights on how to improve.

Therefore, additional avenues get explored. These are regarding other ranking systems and Natural Language Processing (NLP) to provide feedback to the users. The alternative rankings systems, Elo and Glicko [13, 14, 15], are both well-used. Both ranking systems got created to score competitive chess players, with Elo was the first proposed system over the original and then the Glicko system. Both systems look into creating a score that updates on an outcome's results, with the score getting based on the likelihood probability that one entity will win over the other. The main difference between them is the stages required to calculate the score. In comparison, the Glicko system presents improvements over the perceived pitfalls of player manipulation in the Elo system like player rating protection, selective pairing and rating inflation and deflation.

1.3 New Insights

While the comparative judgement technique has many great features, we believe that the concept can still improve. We believe this is especially the case when the comparative judgment system gets expected to get done at a national scale. We believe this because the traditional method would expect all unique pairings to get compared. Additionally, the adaptive comparative judgement that most other systems have adopted still requires time and effort even when the number of individual student work is only around thirty. Therefore, it would be tough to do when needed to get scaled up to a national level. That is why we believe a different ranking system, like an Elo system, could replace the adaptive comparative judgement process and have a more crowed sourced approach. Therefore, reducing cognitive load and the time cost it would take for people to partake.

Furthermore, the current implementations do not provide any meaningful feedback to the students or educators about what makes a piece of work better than the other. Therefore, we think we can look into NLP techniques that can provide some form of feedback. To see if this can become something more meaningful and give some insights. Marking and giving feedback is a crucial role for all educators and the students receiving the feedback.

1.4 Contributions

The main contributions of this work can be seen as follows:

- **A web applicaiton to conduct the comparative judgement**

  We created a web application and hosted it to crowdsource users views on ten tweets based on Brexit. The app provided at random five unique pair comparisons while updating the CJ score and Elo score.

- **A comparion of two different ranking systems**

  Metrics are being stored and calculated based on the two ranking systems, a CJ style and an Elo ranking system. Therefore, the results provide us with a way to compare the effectiveness of the two ranking systems. As a result, they are allowing us to see which one works better in our required situation.

- **An exploration into NLP techniques to provide feedback to the user**

  We created a Jupyter notebook exploring NLP information extraction techniques to provide feedback to the user from information extracted from the ten tweets.

1.5 Results Overview

We found that the comparative judgement (CJ) and the Elo scores were positively correlated. Therefore, the Elo score would be an adequate replacement and possibly a better alternative ranking system to use. The Elo system showed more robustness than the CJ system, especially when the CJ system provided tweets that ended up having the same score. The final order ended up getting based on which one came first within the list if two tweets had the same CJ score. However, the Elo score didn't suffer from the same problem. It also allowed and enabled a ranking to be generated, and there was no score the same.

Regarding the NLP information extraction, this ended up being a mixed bag. While it provided good building blocks to build upon, it offered some insights into the tweets to provide feedback. However, the process did not offer anything significant to be used in a more formal setting. For example, within a school and giving feedback to students.

1.6 Overview

We will first look into the background, explaining the need education has for marking, allowing educators to rank students' work, and providing feedback to students to enable them to reflect and improve. We will then look into what comparative judgement is and its different iterations. Additionally, we look into different ranking systems, with both coming from the chess world but get currently implemented in all other scenarios, like e-Sports. We then look into what Natural Language Processing (NLP) is and some techniques to help achieve what we aim to achieve within our implementation. Then finally for this section will look at other applications that aim to implement comparative judgment within them. We will then look at our methodology, explaining the tools and design approaches we decided to use. We then look at the results we found and have a discussion around these. We then finish with a conclusion and suggested further work for this project.

Chapter 2

Lit Review

Education and the sharing of knowledge is a powerful tool. In fact, in our opinion the most important skill anyone can have. As a famous quote said, "give a man a fish, and he will starve, but teach him to fish, and he won't be hungry anymore". However, it wasn't until 1918 that education, as most people in England and Wales have experienced, started to come into effect [16].

Education over the years was very much about just giving the knowledge to the students from the teacher. It wasn't until 1988, under the Education Reforms Act 1988, that assessments got introduced. The introduction was through the introduction of the national curriculum in England and Wales [2].

As the curriculum got rolled out, statutory assessments got introduced to education between 1991 and 1995. Key Stage 1 first, followed by Key Stages 2 and 3, respectively [3, 4]. Only for the core subjects of English, Mathematics and Science had the assessments first introduced. The first assessments in Key Stage 1 were a range of cross-curricular tasks to be delivered in the classroom, known as standardised assessment tasks - hence the common acronym 'SATs'. However, the complexity of the use of these meant more formal assessments quickly replaced them [3, 4]. The assessments in Key Stages 2 and 3 got developed using more traditional tests.

To allow teachers to judge students' attainment, taking tests became the main assessment form in key stage 3. While assessments were the main form, educators were also able to assess their students with other means against the targets set for attainment within the national curriculum [4]. The teacher and assessment outcomes got used on a scale with key learning milestones expected at different ages. A key stage level indicated the result for the students progress. The model was used throughout the next few years until 2005 when the role of tests in KS1 got downgraded to just being an internal support tool to teachers, and in then 2008, the government decided to remove tests in KS3 [4].

This model continued, with minor adjustments to reflect the changing content of the National Curriculum, up to 2004. From 2005, the role of the tests got downplayed at Key Stage 1, with tests being used only internally to support teacher assessment judgements [17]. Further changes came in 2008 when the government announced that testing in Key Stage 3 was to get scrapped altogether [18].

However, with a change of government party, the Conservative party taking power from the Labour party brought about new changes to how education's focuses and pedagogy methods would get conducted. In 2014 the system of attainment levels was removed, creating the educational shift of "Assessing without level" [19]. However, within schools, it was being referred to as 'life after levels'. Especially by our educational colleges and us at the time. Which was the follow up to the changes in the national curriculum in 2013 [19]. The changes within the national curriculum brought a greater focus on more traditional style GCSE academic subjects while reducing the focus on perceived technical labour style jobs. The new curriculum direction created more emphasis on the final exam outcomes at the stages of GCSE and A-Level.

2.1 The Purpose of Assessment, Marking and Feedback in Education

As we have established, assessments became a staple of the UK educational system in 1988. While the term assessments are not usually defined, the word 'assess' is typically associated with measuring, determining, evaluating, and judging [5].

While there can be multiple reasons why educators assess students, assessments aim to serve a purpose to both the teacher and the student in the process. These include: giving feedback to teachers and learners; providing motivation and encouragement; to boost the self-esteem of the pupils; a basis for communication; a method to evaluate a lesson/training method/scheme of work/ curriculum; to entertain [5]. Additionally, the assessment also creates other opportunities to rank students; a method to select and filter students, allocate students a particular pathway or educational direction, or as a way to discriminate or choose between students for a given set reason [5].

2.1.1 Traditional Methods of Assessment and Feedback

There are four main categories of assessment. These are diagnostic, formative, summative, and national assessments [5, 4]. However, it is essential to note that national assessments do not get used within everyday aspects of teaching and learning. This term is the name given to the critical exams like SATS, GCSE and ALevel exams taken nationally. Therefore we will focus on the other three main ones.

Diagnostic assessment is what gets referred to as pre-testing [5]. Educators use this technique to get a base level of knowledge of the students they have inherited. This method is good for showing the progress of attainment over time by having an initial base test. Teachers can then show how well the students have progressed over time with their improvements over the term. This base assessment also provides the teacher with crucial information - the current ability of every student's knowledge. Through knowing this current level of knowledge, teachers can adapt the coming lessons and provide suitable differentiation and scaffolding within the lessons to allow each student to succeed as much as possible. However, we also experienced, within our time as an educator, the technique getting used to create baseline narratives. Teachers were using them to show that the student's knowledge wasn't at the expected level when inherited by the teacher at meetings or performance management reviews. Therefore, being used as a counter-act measure tool by the teacher, if they find themselves being accused of letting the students' performance slip, by trying to counter-act by implying the students were not at the required level in the first place.

The second method, formative assessment, is also known as 'assessment for learning (AFL)' [5, 4]. This method has become one of the main tools for a teacher in terms of assessment and feedback. AFL allows the educator to assess the students' understanding of a topic on the fly during a lesson without a summative assessment. As a result, allowing the teacher to spend more or less time if the students do or don't get the topic, even if they planned more or less time for that topic. Therefore, ensuring that the teaching is not getting carried out for teaching sake. Thus, the emphasis is less on measurements and more on actual learning. AFL can involve using several techniques: teacher assessment - through in-class questions, marking books; to the students assessing their work called self-assessment, or peer assessment - where the students evaluate each other's work [5].

AFL has many values for teachers and students. Within Black and William's paper. 'Inside the black box' [6] discovered that AFL provides massive learning gains, especially with the low attainer groups. Black and William found that AFL and the use of peer assessment raised motivation and self-esteem across the board, but even more so in the low attainers. With the addition of peer assessment being extra valuable to the students. This form of feedback is effective as the feedback will most likely be given back to the students in a manner that they are more familiar with, informs of language and wording. Therefore in a way that makes more sense to them and having the most impact on their learning [20, 6].

The two key ways that teachers can gain insights from AFL is in questioning and marking. Questioning, also referred to as formative questioning, aims to assess what the students in the classroom know about the current topic being discussed or taught to improve learning [5]. However, for this to be effective, students will need an appropriate 'wait time' [21]. A 'wait time' is the term used to ensure that the student, when asked a question, has to be able to formulate their thoughts and answer as the aim is not to catch them out but to gather what they currently understand. Formative questioning is also good when allowing the students to discuss amongst themselves, then answer the teacher. Therefore, allowing them to consolidate with peers to check if they understand the topic before delivering it to the teacher. A student is more likely to say they do not know than give a wrong answer and look silly in front of their peers, known as the technique 'think-pair-share'. Other effective techniques, which do not require students to discuss between themselves, are 'no-hands up', 'show-me board', 'traffic light' systems [22].

Formative marking is the term used when teachers mark students' work and provide some form of feedback, whether it be two starts and a wish or more standard approaches of providing straight-up feedback. The overall aim is to allow the teacher to see where the student is within their knowledge, gain a level of where they are at and then provide feedback of what they have done well but ultimately what they need to improve on. The proving feedback on areas to improve on are essential whether the student is at a C/4 or an A*/9. The constant feedback, no matter the students level, is as an educator always aims to ensure their students can do better. However, it is crucial that the feedback is taken on board and actioned for formative marking to be effective. Otherwise, it is more of a summative action [6, 23]. To combat this, educators would usually allow students times within a lesson, after the feedback gets given, to go back over their work and make changes to their work in a different colour.

The third method is a summative assessment, also known as 'assessment of learning' (AOL) [5]. This type of assessment happens at the end of a teaching unit or topic. It gets used to gain insights into what the students have learnt within the subject covered or the course. Its purpose is to give a student a mark, grade or ranking. Usually, this is the grade that is mainly focused on, as it is the metric that will impact the school the most in terms of league performance tables regarding GCSE and A-level results. From our experience, summative assessments are carried out regularly within schools. This assessment method tends to get used to getting a snapshot of the students and allow the teacher to perform 'what if' moments like, if they were to take the test now, what would they get? By seeing the results, educators can see if students need to attend intervention or if they are performing as expected or even better. With so much riding on these results, for schools and teachers performance management reviews, a lot of emphasis is put into trying to predict the final results for students. We have seen it put a lot of pressure on the teachers and the students and ultimately creates a very stressful environment, which is not the best environment for learning.

2.1.2 The Negative Aspects of Traditional Marking and Feedback Methods

While marking and feedback are essential in a classroom, they also bring about some negative aspects. As debates are happening about who formative assessment is really for [5], are these assessments for the students done to allow the students to be able to improve on their work and knowledge. Or are they more for the schools to predict actually where the students will be, come exam time. Or are they there to show external bodies, like Ofsted, that the school is being rigorous. Or are they for teachers to justify possible results based on results for their performance management reviews?

Additionally, as teachers might have had a KS4 (GCSE) class for two to three years when assessing and doing the summative assessment, the teacher might not see that student's work entirely at face value. The teacher's personal bias might jump in based on how the student has been over the year or even years. For example, if one student has been nice, well behaved and just done the required work, the teacher might provide a higher grade for that student. However, they might give a lower grade score for someone who has been a pain and misbehaved through the year. However, the second student's work might be of better quality, but it is not seen at face value and therefore not accurately marked because of the other factors.

As schools might have multiple teachers teaching a particular subject simultaneously, a process called moderation is required. Moderation aims to make sure that all work being marked and graded is all at the same level. For example, teachers A, B and C's student's work, awarded a Distinction *, are all at the exact agreed and expected quality. However, this can bring about multiple issues. One is that not all teachers might interpret the mark scheme the same as the others and therefore look for different attributes within the students' work. While moderation and standardisation aim is to find out these inconsistencies and get all the teachers on the same page regarding expectations, office politics can also hugely impact it. Imagine the scenario. Five teachers are teaching the same year group and qualification. One teacher is the lead to that subject, so, therefore, would have had all the required training from the exam boards regarding the course, another one is a regular teacher. At the same time, one is an assistant principal, another is a vice principal, and the final one is the head of the faculty. So in the whole school context, the subject lead teacher is higher in the hierarchy than the regular teacher but lower than the other three. However, in the scope of the qualification getting delivered, the lead teacher is at the top. But this can bring about the office politics we were alluding to. Some teachers who are higher up in the school system but not in the qualification scope can throw their weight around say things need to be how they have interpreted the mark scheme. Their interpretation is not always correct, but they push their view for whatever reason, bringing about a few situations. Resulting in, will the lead teacher challenge the more senior figure to say that they are wrong and the exam board expects this, or will they agree not to upset the more senior member of staff? Either way might not end well, and with the tricky world of education, the second option is the more likely choice. However, this brings about issues in regards to inconsistency with work and the awarded mark.

Another drawback to traditional marking is that the requirement of personalised feedback for students. To allow them to develop, students must have personalised areas of where they need to improve. However, in controlled assessments, teachers can give feedback, but it can not be personalised. It has to be generic, but most schools' policies require the feedback to be personalised, creating a conflict between the exam board's requirements and the school's requirements based on Ofsted's expectations. The situation makes a moral and ethical decision. They are likely to be reprimanded by the school if they do not provide the feedback but can be done for malpractice if the exam board catches them for giving the feedback.

When a summative assessment has occurred within a learning sequence, students get usually presented with a grade and feedback. This feedback and mark could be for the end of unit exams or homework, for example. While the teachers want students to focus on the feedback given to help them improve, students focus on the results and will naturally rank order themselves. The UK government has attempted to try and resolve this by removing levels in KS3. However, when KS4 focuses on the final summative assessment, their actual GCSE exams, a provided grade is hard not to offer. Therefore, it is vital to make sure that feedback is acted upon once given.

Finally, a big issue in regards to marking and providing feedback is time. It takes a long time to score a students' work and then give feedback to the students. It is also a very tedious task that a teacher might not do in one sitting. Therefore, with many potential variables in play, the marking of the points award per each exam question, for example, might not be the same. There is also a massive cognitive load that is placed upon the teacher while trying to mark.

Consequently, it is challenging to ensure that consistency and fairness are playing a part in the marking. However, the enormous cognitive load placed upon the teacher can be very draining. It can then affect the quality of the teachers delivery within the lesson, especially with the stress aspects that get placed upon them regarding how quick the feedback needs to get returned to the students.

2.2 Comparative Judgement

2.2.1 What is Comparative Judgement

Comparative judgement is a mathematical way to determine which observation item is better than the other item being observed compared to each other. This method was first proposed in 1927 by Louis Leon Thurstone, a psychologist, under the term "the law of comparative judgement" (LCJ) [8, 1]. In modern-day language, it gets more expressed as a paradigm used to obtain analyses from any pairwise measurement process [24]. Examples of the LCJ are such arrangements as comparing the observed intensity of the weights of objects, comparing the extremity of an attitude expressed within statements, such as statements about capital punishment, and asking what object is more prominent in size. The measurements represent how we perceive things rather than being measurements of actual physical properties [25]. This kind of measurement is the focus of psychometrics and psychophysics [26, 27]

In more technical terms, the LCJ is a mathematical representation of a discriminal process [8]. This process involves a comparison between pairs of a collection of entities concerning multiple magnitudes of attributes. The model's theoretical basis is closely related to item response theory [28] and the Rasch model's theory [29]. These methods are used in psychology and education to analyse data from questionnaires and tests [24, 26].

While comparative judgement is a technique that has been around for almost 100 years, it wasn't until the early 90s that this technique got proposed for use within an educational setting. This first proposal was by Politt and Murry [30], who conducted a study where they tested candidates on their English proficiency within Cambridge's CPE speaking exam. The judges watched 2-minute videos and judged which one out of a pair of videos they deemed better at the requested task in the exam. However, before this, in the 1970s and 80s, comparative judgement was presented as a more theoretical basis for educational assessments [31].

With the momentum of his findings, Politt then presented comparative judgement as a tool for exam boards to use to be able to compare the standards of A-Levels from the different exam boards, replacing the direct judgement of a script that was at the time currently being used [32]. In his papers titled, "Let's Stop Marking Exams" [33], he presents a valid argument for using comparative judgement, with the advantages it brings over some traditional types of marking.

Politt, in 2010, also presented a paper at the Association for Educational Assessment – Europe. It was about How to Assess Writing Reliably and Validly. Politt presented evidence of the extraordinarily high reliability achieved with Comparative Judgement in assessing primary school pupils' skill in first-language English writing [34].

2.2.2 The Logic Behind Comparative Judgement and What it Aims to Do

How comparative judgement works is to present two options to a marker. The marker then gets asked to pick which one of the two options they think is better. The marker will get presented with all possible combinations available, each picking which one they think is better out of the two. An outputted score is then presented based on the method used, providing a preference order of observations.

However, an alternative version derived from Louis Leon Thurstone, referred to as the "Pairwise Comparison" [1], will provide an output based on the difference between the quality values is equal to the log of the odds in respect to object-A will be object-B. This formula gets represented as:

$$\log \text{odds}(A \text{ beats } B \mid v_a, v_b) = v_a - v_b \,.$$

Pairwise comparison generally is any process of comparing entities in pairs to judge which of each entity is preferred or has a greater amount of some quantitative property, or whether or not the two entities are identical. The pairwise comparison method get used in the scientific study of preferences, attitudes, voting systems, social choice, public choice, requirements engineering and multiagent AI systems.

Within an educational setting, a different approach of comparative judgement has been proposed. This new adaptation gets referred to as adaptive comparative judgement (ACJ) [7]. It is also the same as the pairwise comparison in concept, just with a different name. ACJ is very similar to the core concept of comparative judgement, as it asks a marker to rate which work is better. However, in this version, the 'scores', the model parameter for each object, get re-estimated after each 'round' of judgements. Resulting in each piece of work being judged one more time on average. During the next round, each piece of work is compared only to another whose is currently estimated to have a similar score. Therefore, comparing each piece of work with a similar score results in an increased amount of statistical information from each judgment to produce the final ranking. As a result, the estimation procedure is more efficient than random pairing or any other pre-determined pairing system like those used in classical comparative judgement applications [7].

2.2.3 What does ACJ aim to achieve and How reliable is it

Multiple studies have shown that ACJ achieves exceptionally high levels of reliability, often considerably higher than the traditional method of marking. It, therefore, offers a radical alternative to the pursuit of reliability through detailed marking schemes [7].

ACJ software estimates a 'measure' for each piece of work getting compared, known as a 'script', and an associated standard error. The process requires several metrics to be measured. These are the true SD, SSR and the index G [10].

The 'true SD' gets calculated for the script by using the formula:

$$(TrueSD)^2 = (ObservedSD)^2 \text{ - } MSE$$

The MSE represents the mean squared standard error across the scripts.

The SSR gets defined like reliability coefficients in traditional test theory, as the ratio of true variance to observed variance with the formula:

$$SSR = (TrueSD)^2/(ObservedSD)^2 \ .$$

Sometimes another separation index G is calculated. Index G represents the ratio of the 'true' spread of the measures to their average error. The formula is:

$$G = (TrueSD)/RMSE$$

The RMSE is the square root of the MSE. Leading to the SSR, as an alternative, to be calculated as:

$$SSR = G^2/(1 + G^2)$$

Studies have found that ACJ has high reliability, even compared to the final results when work is marked more traditional. However, frustration has been prevalent when markers have had to review repetitive work [35]. Additionally, frustration also gets created by the lack of students being able to challenge the final results [35].

When we look at fig: 2.1, we can see that these studies have produced a high SSR score. However, a lot of the studies have used a high resource count to complete the different studies. For example, Pollitt 2012 studies used 54 judges to mark 1000 pieces of scripts, which resulted in 8161 different comparisons getting seen and 16 rounds occurring. In comparison, Whitehouse & Pollit (2012) had 564 scripts to compare and 23 judges. This study took 12 - 13 rounds to get a high SSR score. Therefore, we can see that while ACJ can help with teacher workload in removing a cognitive overload, it takes to create additional workload in the sheer amount of rounds required to get a reliable SSR score.

Additionally, a number of the studies have used 20 - 100 different judges, which is more than most teachers within a single department. Therefore, making it hard for us to see how within a typical set-up of a school it can be used. It brings about questions like, does the requirement needed to produce an accurate judgment outweigh the reduced cognitive load?

Many studies' motivation for using adaptivity in CJ studies is to avoid wasting time and resources by getting judges to make comparisons whose outcome is a foregone conclusion. However, theoretical considerations from the IRT and CAT literature and the simulation study results show that adaptivity produces spurious scale separation reliability, as indicated by values of the SSR coefficient that are considerably biased upwards from their

Study	Adaptive?	What was judged	#scripts	#judges	#comps	%max	#rounds	Av. # comps per script	SSR
Kimbell et al (2009)	Yes	Design & Tech. portfolios	352	28	3067	4.96%		14 or 20 bimodal	0.95
Heldsinger & Humphry (2010)	No	Y1-Y7 narrative texts	30	20	~2000?			~69	0.98
Pollitt (2012)	Yes	2 English essays (9-11 year olds)	1000	54	8161	1.6%	16	~16	0.96
Pollitt (2012)	Yes	English critical writing	110	4	(495)	(8.3%)	9	~9	0.93
Whitehouse & Pollitt (2012)	Yes	15-mark Geography essay	564	23	3519	2.2%	(12-13)	~12.5	0.97
Jones & Alcock (2014)	Yes	Maths question, by peers	168	100,93	1217	8.7%	N/A?	~14.5	0.73 0.86
Jones & Alcock (2014)	Yes	Maths question, by experts	168	11,11	1217	8.7%	N/A?	~14.5	0.93 0.89
Jones & Alcock (2014)	Yes	Maths question, by novices	168	9	1217	8.7%	N/A?	~14.5	0.97
Newhouse (2014)	Yes	Visual Arts portfolio	75	14	?	?	?	13	0.95
Newhouse (2014)	Yes	Design portfolio	82	9	?	?	?	13	0.95
Jones, Swan & Pollitt (2015)	No	Maths GCSE scripts	18	12,11	151,150	100%	N/A	~16.7	0.80 0.93
Jones, Swan & Pollitt (2015)	No	Maths task	18	12,11	173,177	114%	N/A	~19.5	0.85 0.93
McMahon & Jones (2014)	No	Chemistry task	154	5	1550	13.2%		~20	0.87

Figure 2.1: Design features* and SSR reliability results from some published CJ/ACJ studies [10]

true value. The higher the proportion of adaptive rounds, the greater the bias. SSR values above 0.70 and even as high as 0.89 can get obtained from random judgments [10].

Consequently, the conclusion is that the SSR statistic is misleading and worthless as an indicator of scale reliability. Other reliability indicators, such as correlations with measures obtained from comparisons among different judges, or correlations with relevant external variables, should be used instead. Therefore ACJ studies that have used high values of the SSR coefficient alone to justify claims that ACJ is a more reliable system than conventional marking need to be re-evaluated [10].

Additionally, many companies providing CJ tools claim that it only takes 30-seconds to judge a piece of work. However, ultimately the time it will take also depends on the level of the work getting assessed. For example, an A-Level piece of work would take longer than a KS2 assessment. A study where five teachers made 1550 comparisons between them (310 each), and they, on average, took 33 seconds to complete each comparison. Therefore the total marking time was about 2.8 hours per teacher or 14 hours in total. While the two teachers marked the work in a more standard way (using a rubric), taking them 1.5 hours each or 3 hours altogether [11]. Another study claims that CJ requires 17% more marking time than just using a rubric marking system [12]. The results of this comparison of approaches do challenge the efficiency of CJ over standard marking. So if CJ is to become more mainstream within schools, there needs to be a clear benefit for the teachers to adopt this approach. Otherwise, the teachers are less likely to be on board and use the method. As teachers are usually sceptical about new strategies and think they are there to add additional work. However, CJ is recommended for use when the marking is of open-ended exam-style questions [12].

2.2.4 How effective is Comparative Judgement at Providing Feedback?

Multiple studies have got conducted where ACJ has been used to present feedback to the students. The approach gives students insights into how other people have approached a similar situation differently and how peers valued their work [36].

ACJ offers a new way to involve all teachers in summative as well as formative assessment. The model provides robust statistical control to ensure quality assessment for individual students through peer assessment [7]. However, while peer feedback is a good strategy, its effectiveness can be limited by the relative students understanding of both the body of knowledge upon which they are getting asked to provide feedback and the skill set involved in providing good feedback [37].

In contrast, a study showed that when peers were involved in synthesising evidence and feedback, the student's engagement in a double looped system of reflection in action increased performance across assignments. Therefore, it indicates that students were receiving feedback to support them in improving their work. The improvements only came from the ACJ judging process, suggesting that students were critiquing their work relative to the breadth of work presented by their peers. They were also engaged in a critique of the purpose of the design assignments concerning core competency development. In essence, students were developing, responding to, and applying criteria [38].

However, all these examples allow students to gain feedback in a ranked method, of how well they have scored against others with the addition of seeing other students work modelled to them. But at no point are the students getting any truly personalised feedback on what has worked well and what needs improving. Additionally, it relies heavily on students to self-assess and provide their internal improvements, relying on them genuinely understanding the requirements, which would be a meagre chance for less confident, low-achieving students. Therefore, it is a more superficial process and lacks any try impact for methods required in a secondary or sixth form classroom. So we believe that the CJ, while it does remove cognitive loads, actually adds more work for the teacher to provide the basic required information they would need in their classroom to present to the students.

Therefore, questions are produced on CJ's effectiveness if it takes longer than standard marking and doesn't provide any form of personalised feedback to the students, resulting in the teacher having to do more work to remove the cognitive workload off the teacher. Is this a trade-off worth making? We find the current methods on offer hard to justify the trade when teachers time is already limited. However, it does have a lot of potentials.

2.3 Other Rating Systems

While comparative judgment has proven to be a suitable method of ranking pairwise matches of students work over the years, it has its limitations. For example, comparative judgment requires every combination to be compared against, which means for a class of thirty students, accounting for 435 different combinations. Take into account a subject like English, which every student will have to take. A typical school year could have 120, which would mean 7,140 different combinations. That is a lot of time and comparisons that would be required. Therefore, to truly take the cognitive load off a marker or teacher, it would be better to try and have different people sub-sample the work. Then, from the scoring of the sub-samples, use this to generate an overall ranking. In essence, it is creating a competitive scoring system against each other. Two suitable systems to achieve this would be an Elo or Glicko rating system.

2.3.1 Elo Ranking System

The Elo ranking got first introduced into competitive chess in the 1980s [39]. However, it got created in the 1960s by Arpad Elo as a replacement for the Harkness System. The Harkness System got used by the United States Chess Federation (USCF) at that time [13]. Additionally, the Elo system has gets used as a ranking system for football, American football, basketball as well as eSports like Counter-Strike: Global Offensive and League of Legends [40, 41].

The Elo system looks at the difference in two players ratings, then serves as a predictor for the match's outcome. The players Elo rating is depicted as a number and will change over time depending on the games' outcomes, with the winners taking points from the losers. However, how many points get awarded is decided upon the difference in ranking between the players. If the higher ranked player wins, only a few rating points get taken from the lower rank player. However, if an 'upset win' occurs, when the considerably lower rank player beats the higher rank player, then a much greater number of points will be gained to the winner and deducted from the loser. Ultimately, even when 'upset wins' happen, the ranking of the players will reflect the valid scores over time [42].

However, there are ways that players who know how the system works can cheat it. These methods include protecting one's rating, selective pairing and ratings inflation and deflation.

Players protecting one's rating discourages game activity for players wanting to preserve their score. In essence, this situation gets created when players are not playing any more games once they are at a high score [43]. A method against this behaviour is to award an activity bonus combined with the ranking score [44].

Selective pairing is when players choose their opponents, which results in players choosing opponents that the player has the minimal risk of losing. Additions like a k-factor got added, but these do not solve the problem completely [44]. Additional implementations have got added, like auto-pairing, which are based on random pairings but have a winner stays on context [44].

Inflation is when a score means less over time. For example, a player has a score of 2500 and gets ranked 5, but later, another is ranked 15. It shows that the player's ability is decreasing over time. When deflation happens, this indicates that advancement is happening. Deflation is when a score of 2500, got a player ranking of 7, but at a later date, the score is then put ranked the player 2. Therefore, we must consider when using ratings to compare players between different eras. The ranking gets made more difficult when inflation or deflation are present [45].

The Elo system has a flaw in that it is almost certainly not distributed as a normal distribution. As a result, weaker players have greater winning chances than Elo's model predicts [39]. However, the Elo ratings still provide a valuable mechanism for rating based on the opponent's rating.

2.3.2 Glicko Ranking System

The Glicko rating system [14] and Glicko-2 rating system [15] are methods for assessing a player's strength in games of skill, such as chess and Go. Mark Glickman invented it to improve the Elo rating system and initially intended it for primary use as a chess rating system [14]. Glickman's principal contribution to measurement is "rating reliability", called RD, for rating deviation [14].

Both the Glicko and Glicko-2 rating systems are under the public domain. It is said that these systems can get found used on game servers online [45]. Additionally, the formulas used for the systems are available on Mark Glickman's website [46].

The RD measures the accuracy of a player's rating, with one RD being equal to one standard deviation. Then the RD is added and subtracted from their rating to calculate this range [15]. Once a game has been completed, the amount the rating changes depends on the RD. The changes are smaller when the player's RD is low, as the player's rating is already well known. Similarly, when the opponent's RD is high, due to the factor that the opponent's rating is not well known at this point [15]. The RD itself decreases after playing a game, but it will increase slowly over time of inactivity [15].

2.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a subfield of AI that aims to understand natural language through trying to process and analyse it [47, 48]. Ultimately NLP is teaching computers how to understand humans in natural language. However, this is not straightforward, as language is a complex, ever-changing form even for humans. There are three main categories that NLP problems fall into, heuristics, machine learning, and deep learning [48]. The nature of ML algorithms gets designed to work with unknown datasets, allowing data scientists to learn how to use language [47]. While this will bring us a vast amount of insights, as mentioned before, the ever caning landscape of language does not mean that it is perfect and, once made, doesn't need revision. Therefore generating and understanding natural language are the most promising but most challenging tasks in NLP [47, 48].

To understand the complexities of machines attempting to understand language, we must first know what we mean when we state 'what is language'. Language is a structured communication system, which involves many combinations of its fundamental components of varying complexities. For example, some of these components are characters, words and sentences to name a few [48].

Human language gets constructed of four major building blocks, and are phonemes, morphemes, lexemes and syntax, and context [48]. To make an effective NLP app, we need to ensure our application has these different building blocks used within its foundations (see fig: 2.2). However, knowing these building blocks does not entail we can do what we like within NLP. NLP has a lot of challenges that involve ambiguity, common knowledge, creativity and diversity across languages [48].

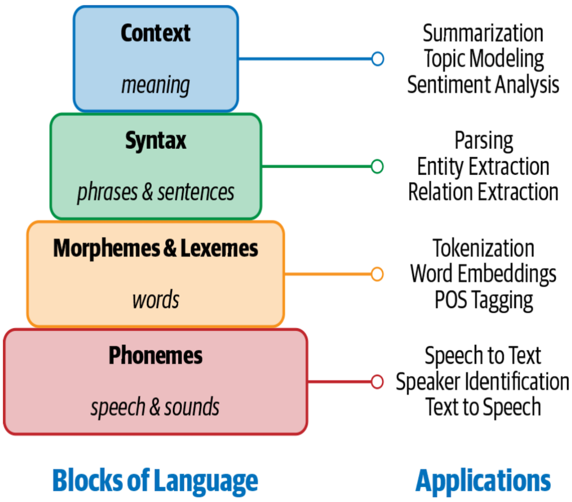


Figure 2.2: The building blocks and their tools for understanding language [48].

2.5 Related Work

While comparative judgment is not a new concept, only a few current systems implement a version of it as a tool for marking. These current CJ projects have a slightly different take on the CJ process but have very similar fundamentals. The current offerings are created or provided by RM Compare, a consortium of universities called D-PAC and No More Marking.

RM Compare is probably the version with the most prominent presence.

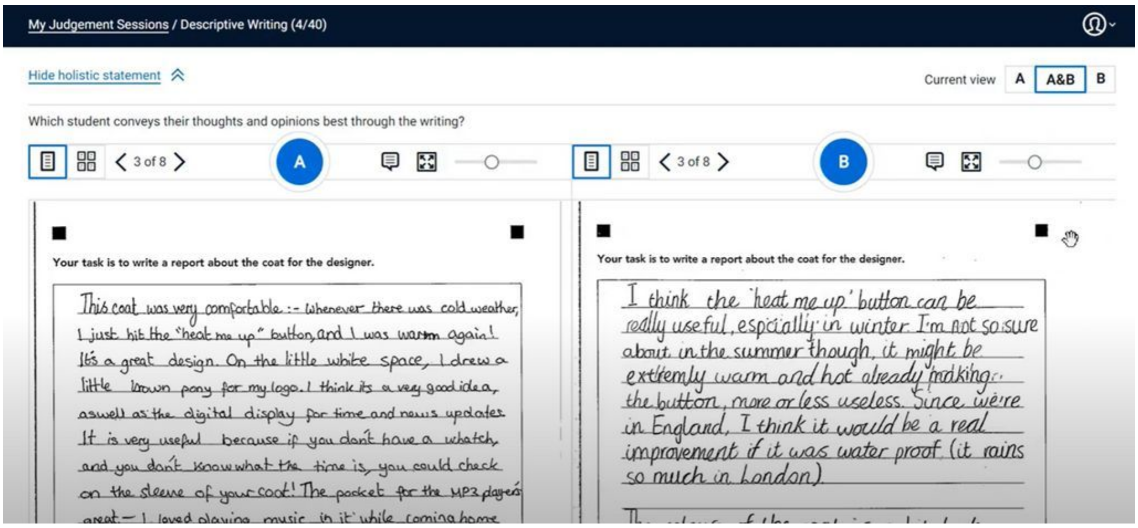


Figure 2.3: RM Compare's ADJ System.

RM Compare uses ACJ, based on The Law of Comparative Judgement. The assessor (a teacher, lecturer, examiner or student) is presented with two anonymised pieces of work in a side-by-side pairwise comparison and asked to use their professional judgement to select which of the two is better at meeting the assessment criteria (see fig: 2.3).

RM Compare says that through repeated pairwise comparisons, optimised by an iterative, adaptive algorithm, a highly reliable scale or rank order is created through consensus over what 'good', 'better', and 'best' looks [9].

RM Compare empowers users across educational organisations to collaborate on assessments and is proven to increase student attainment. It also reduces the cognitive load from teachers, which gets achieved through the very nature of the comparative judgment process. It also has a straightforward and effective UI for the user to interact with [9].

However, it still has an extensive workload as for it to be effective, the markers (known as judges) need to go through several rounds [9]. Multiple examples online were stating 16 rounds. RM Compare states that these numerous rounds are required to reduce the error uncertainty rate. The algorithm's adaptiveness will ensure that pairs closely matched to each other get checked more to confirm the order is correct, reducing the algorithm's error rate calculation. A high level of uncertainty will get compared more often to check the consensus between the judges [9].

An issue with the application is that it doesn't provide any real form of meaningful feedback. RM Compare suggests that the students gain feedback from the system is for the students to compare their peer's work through the system [9]. Once this comparison by the students gets completed, the students' peered work ranking results will get compared against the teachers [9]. Which then, in turn, gets used as a point of discussion [9]. Therefore, in our opinion, not providing any meaningful form of feedback. While RM Compare that the process has a considerable impact on students attainment, this claim feels more like a marketing gimmick. While we agree that this process can generate insights into students' expectations, it does not provide meaningful, personalised feedback. Therefore, not allowing them to know what they need to do to improve.

No More Marking is another CJ platform that offers the features of assessing primary writing, improving secondary writing and assessing GCSE English. The company consists of Daisy Christodoulou, who is an influential person within education. She has also received an MBE. Additionally, Dr Christopher Wheadon, Dr Ian Jones, Dr Patrick Barmby, Mr Brian Henderson, Mr Neil Defty.

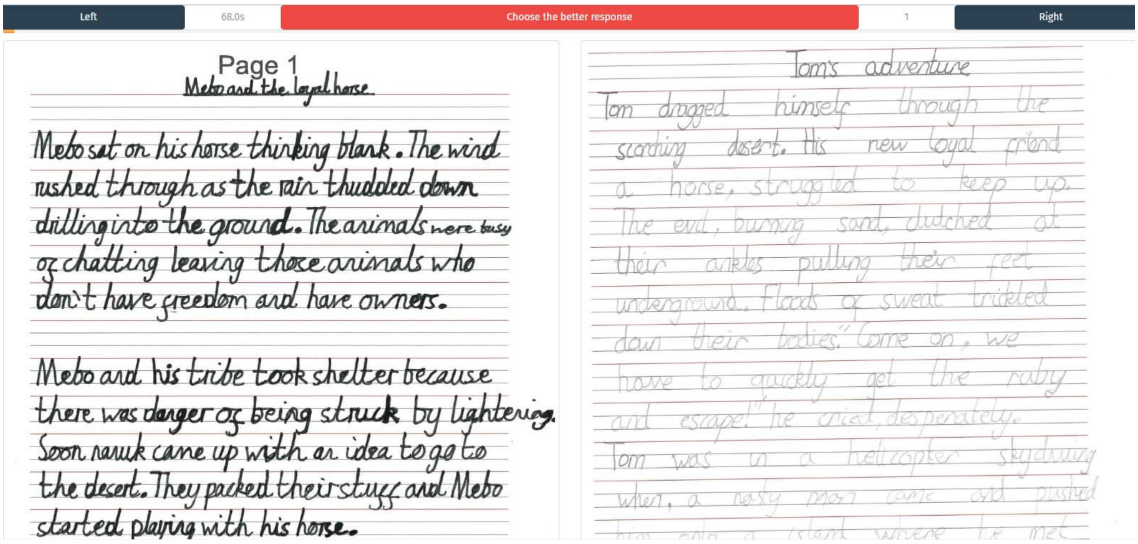


Figure 2.4: No More Marking's ADJ System.

No More Marking states that their system uses comparative judgement. 'Which is a process where judges compare two responses and decide which is better. Following repeated comparisons, the resulting data is statistically modelled, and responses placed on a scale of relative quality' [49]. The No More marking team also claim that 'research has shown the process to be as reliable as double marking, but much quicker [49]'. However, liturature has show that this is not necessarily true.

The No More Marking system (see fig: 2.4) has a very similar layout and design to the RM Compare's version, but we believe with slightly better characteristics. The system is again backed up with research to claim how effective CJ is at marking and how much quicker it can speed up the marking process, which No More Marking have linked to on their website. Additionally, they claim the process is highly reliable. So overall, the system works and acts very similar to RM Compares. As well as claiming a high accuracy and reliability both backed up by research.

However, just like RM Compare's system, No More Marking has the same underlying issues, in our opinion, as they are very similar and are using the same fundamental technology. Additionally, No More Marking's approach to providing feedback allows the students to do their CJ on peer's work. As discussed in the literature, it has many flaws in this approach, especially as it does not provide any personalised feedback to the student on how to improve.

Figure 2.5: D-Pac's ADJ System.

D-PAC has a slightly different focus compared to RM Compare and No More Marking. While D-PAC provides an application (see fig: 2.5), its main focus is to provide the ACJ algorithm [50, 51].

D-PAC decided to open-source their algorithms following a meeting with the team developing the Digital Platform for the Assessment of Competences (D-PAC). The D-PAC project is a consortium of Antwerp University, iMinds and Ghent University funded by the Flemish government [50]. The D-PAC consortium had become disappointed with the lack of products to support researchers and assessment practitioners in CJ. Therefore, D-PAC decided to produce an open-source solution for Comparative Judgement that will support their research program and support the growth of research in this field more generally [50].

Therefore, in comparison, D-PAC has provided the ACJ algorithm that powers No More Marking's platform.

2.6 Overall Aim

Comparative judgement is a power tool. It can remove a lot of cognitive load from the teacher. It also eliminates the teacher's bias in the marking process, especially when the teacher knows whose student they are marking. Teachers can consider how the student has performed over the year instead of how they did in that final piece of work. Potentially takings away merits of the students performance in the moment of the exam.

However, the current process of adaptive comparative judgment can reduce the cognitive load with the teacher marking and lessen the potential for bias from the teacher. Current implementations do have their limitations and still create a lengthy process. With some systems still having markers to mark student's work up to, some examples have 16 rounds of marking, which is still very time-consuming. Which, if you want to expand this to a national level, wouldn't be very effective.

Therefore, we want to look into different methods of ranking students that could get used to allow a crowdsourced way of marking in a CJ style, to be implemented. Suggested alternatives are an Elo system ranking. Additionally, we want to create NLP tools that will allow us to interrogate the data and see if there are any patterns within the data and the end rankings. Allowing us to suggest what aspects of the data makes the content get perceived as good.

Chapter 3

Methodology

In order to apply any ML and NLP to the tweet dataset, to see if we could do any information extraction and statistical analysis, we first needed to be able to generate a ranking of the ten tweets we had obtained. We sourced the tweets themed around Brexit on Twitter, and then a pipeline (see fig: 3.1) for sourcing peoples preferences of the tweets was created. The pipeline created was handled by the web app. The web app allowed the user to create an account and then compare the tweets. The resulting decision updated the Elo rating for each tweet and the more simplified traditional CJ method. Each user gets only presented five different combinations, ensuring that a single tweet was only seen by the user once.

Figure 3.1: A visual representation of the processes pipline.

3.1 Overview of Application

3.1.1 Web Application

The application has two main sections. The first section is a web application. This web application aims to rank the ten Twitter tweets by presenting users with two tweets and asking them which one is better. In essence, the web application is a tool to crowdsource data on peoples views based on the tweets that they get presented. The web app then creates two ranking systems. One ranking system uses an Elo system, and one the users a more pairwise CJ style. The pairwise CJ score gets calculated by the total wins getting subtracted by the total losses.

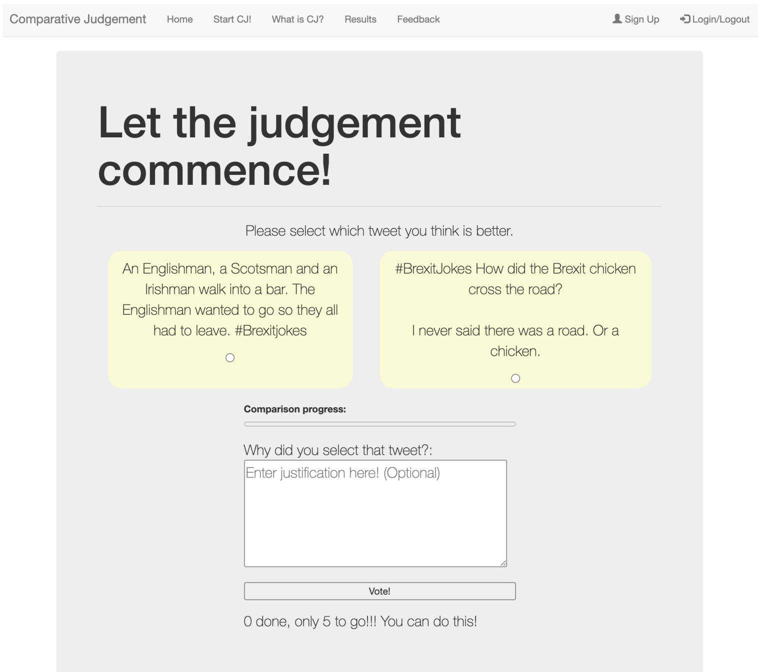


Figure 3.2: The web page users were presented to capture their judgememnts. To see all the web pages see appendix: A

3.1.2 NLP Information Extraction Notebook

The second section is an exploratory Python notebook looking into NLP tasks on the tweets. We carry out sentiment analysis and information extraction on the tweets to see if any patterns within the tweets match their ranking's place. For example, positive sentiment tweets getting a higher ranking with a particular theme, other than Brexit possibly showing. The ultimate aim is to create feedback based on the results and the information.

3.1.3 NLP Information Extraction

Information extraction is the process of extracting relevant information from text. Some of this information could be calendar events and names of people, to list a few [48]. We, as humans, do this all the time. We extracted the information from multiple sources, like reading documents or conversations. However, for computers, this is not such a straightforward task. Due to the ambiguous nature of natural language, information can mean multiple things depending on the context in which it is getting used.

Due to its complex nature, information extraction relies on several separate takes, which, when used together, generates information. These steps include keyphrase extraction, named entity recognition, named entity disambiguation, and linking and relationship extraction [48].

Next, we will look into the different building blocks that can extract information from our text to provide feedback to the user. We will look into part of speech tagging, named entity recognition, feature extraction, sentiment analysis, text similarity, utterance pattern matching, text similarity scoring and word sequence pattern recognition.

3.1.3.1 Part of Speech Tagging

Part of Speach (POS) tagging has the hidden Markov model (HMM) underpinning it [48]. The HMM is a statistical model that assumes an underlying, unobservable process with hidden states [52]. POS tagging ultimate aim is to identify the nouns, verbs, and other key parts of speech [47].

We decided to implement POS tagging on the tweets to see if any insights would help provide any feedback to the user. While it might not give us many insights on its own, it can get used as an additional tool that, when paired with other methods, can help provide some insights. We also felt that when the POS tagging got visualised, this would help create a clear picture of the structure of the tweet.

3.1.3.2 Named Entity Recognition

Named Entity Recognition (NER) is the task of identifying entities in a document for information extraction [48]. Entities usually are made up of names of persons, locations, organisations, money expressions and dates, to list a few [53]. NER is an important step within the pipeline of information extraction [48].

As this is a crucial stage in information extraction, we decided to implement it and use it in its pre-trained form from the libraries offerings. We decided to use this method due to the time restrictions of the project and to see how well it performs and if it can help generate feedback to the user.

3.1.3.3 Feature Extraction

Feature extraction aims to transform tokens into features. An excellent technique to achieve this is a bag of words (BOW). This technique will count the occurrences of a particle token within our text. Therefore, for each token, we will have a feature column. This feature column gets referred to as text vectorisation. However, using a standard BOW will lose the word order, and the counters can not be normalised [53].

In order to preserve some order, we can count the tokens as pairs or triplets, for example. This technique gets also referred to as n-grams. The n refers to the number of tokens to get referenced. Some examples are 1-grams for tokens and 2-grams for token pairs. However, this has its problems as it can create too many features [48]. A solution to this problem is to remove some n-grams from the feature set. This solution can get achieved by using the metric based on the frequency of their occurrence [48].

The n-grams that we would want to remove based on their frequency are high and low-frequency n-grams. High-frequency grams get usually referred to as stop words, and low-frequency grams are rare words or typos [53]. We especially want to remove the low-frequency n-grams as they can create overfitting. Ultimately, we ideally want the medium frequency words.

A technique we can use to find the medium frequency n-grams is term frequency-inverse document frequency (TF-IDF). TF-IDF has two main stages, the term frequency (TF) and the inverse document frequency (IDF). The TF ($tf(t,d)$) looks for the frequency of the n-gram (term) t in the document d [54]. While IDF takes the total number of documents in the corpus ($N = |D|$) and the number of documents where the term t appears ($|\{d \in D : t \in d\}|$) [54]. So the IDF gets represented as $idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$ [54]. TF-IDF ($tdidf(t,d,D) = tf(t,d) \cdot idf(t,D)$) achieves a high weight by a high-term frequency, within a given document, and a low document frequency of the term in the whole collection of documents [54].

Through using TF-IDF, we can replace counters within our BOW with the TF-IDF value. We can then normalise the result row-wise by dividing by $L_2 - norm$. Through this method, important features will have a relatively high value. Through this method, we are then able to display the key features within our documents.

3.1.3.4 Sentiment Analysis

Sentiment analysis, which can also be known as opinion mining or emotion AI, uses NLP, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis gets widely applied to materials such as reviews and survey responses, online and social media, and healthcare materials for applications. It aims to find out if a perceived text has got classed as positive or negative and, in some instances, neutral [55, 56].

Through aiming to gain an insight into if a tweet is positive or negative can provide some insights into possible patterns emerging. This feature, we believe, could be a helpful tool in providing feedback to the user, especially if there is a clear pattern in terms of a tweets sentiment and its final ranking.

3.1.4 Text Similarity

Text similarity scoring aims to analyse and measure how close two entities of text are to each other [54]. We can compare two objects. By comparing these objects, it is then possible to predict how similar they are. We can use docs, spans, tokens or Lexeme to calculate the similarity score [57]. To measure the similarity scores between text entities, we can use two main types of methods, term and document similarity [54].

Predicting similarity helps build recommendation systems or flag duplicates. For example, it allows for the system to suggest user content that's similar to what they are currently looking at or label a support ticket as a duplicate if it is very similar to an already existing one [57]. Additionally, similarity measures are an excellent way to take the noisy text data and group the text together. It allows us to see what text gets considered similar to each other by using unsupervised clustering techniques [54].

As the dataset we are dealing with are Twitter tweets, we decided to do this through entire document similarity and spans of named entities to see if the results provide us with any insights in terms of providing any feedback to the user.

3.1.4.1 Utterence Pattern Matching

Utterances are usually anything a user has said, which could be in the form of speech or text. For example, "Can I have pizza" or "how big is the Eifel tower?". Therefore the main aim of utterance pattern matching is for the NLP model to extract the actions that the users want to execute [48].

In most cases, intents can be identified by looking for verbs in the dialogues of the users. However, sometimes the complete sentence is used to determine the intent of it [56]. In the given sentence, the user wants to place an order for a pizza. Now that we know the intent, we can trigger a secondary action, in this example, ordering food. Nevertheless, our goal is to see if it spots any patterns that might be noteworthy for presenting as feedback or get used to triggering a secondary action for feedback generation.

3.1.4.2 Finding Word Sequence Patterns

Word sequence patterns is an assuring trade-off between more traditional approaches of NLP and ML for information extraction [58]. Word sequence patterns aim to learn linguistic assets such as lexicons or patterns automatically [59]. One of the earliest works around this topic presents a means to get linguistic patterns from plain texts [60].

Therefore, this technique aims to present the tweets to the algorithm and see if it can spot any patterns. If it can, we want to be able to present this to the user. Hence, allowing the user to receive some insightful feedback about the tweets.

3.1.4.3 Key Phrases

The Key phrases method aims to take a document object and find the word or phrase with the most information. This technique is effective, especially when creating a chatbot. Key phrases allow the computer to determine what the user, who is interacting with the chatbot, is talking about. A single word in the question can sometimes be enough, but we might need to look at phrases. Key phrases work well with dependency parsing [47].

We decided to experiment with this feature to see if we could extract the key phrases from the tweets and see if they could provide us with any insights and present them to the user as feedback.

3.2 Tools

To create the web application and insights from the tweets, we required to use several tools. It is a requirement that we develop a full-stack web application with a user UI, an area to input the user's judgements on the tweet, store the results using a database, and extract information from the tweets using NLP techniques. Several factors within the final application needed to be satisfied for the tools to be appropriate for use.

Figure 3.3: An example of a Trello Kanban board [61].

We will be using Trello for the kanban tools (see fig: 3.3). "Kanban" is the Japanese word for "visual signal" [62]. Using Kanban boards allows us to keep our work visible. Using Kanban boards allows others to see what is going on and what is needed to get done. Ultimately it allows everyone to see the complete picture.

3.2.1 Programming Language

While many programming languages can handle creating a full-stack application and conducting ML, for example, Java [63], Php [64] and JavaScript [65]. We decided to use the Python language [66]. We decided upon Python due to our familiarity with it over the other main languages and its versatility. We made this decision because Python can make full-stack applications with the use of additional libraries and handle most NLP ML tasks using libraries like NLTK [67], SpaCy [68], Sci-Kit Learn [69], and TensorFlow [70].

3.2.2 Libraries

While we use the Python programming language to create the web application and the NLP information extraction, we require significantly different libraries to complete each task. We will look into the potential web libraries available to us and the NLP focused libraries. We will then present the libraries that we decided upon for each of the parts.

3.2.2.1 Web Application

For creating the web application, there were two main libraries available. These were Django and Flask.

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source [71].

While Flask is a small framework by most standards, which is referred to as a "micro-framework," and small enough that once you become familiar with it, you will likely be able to read and understand all of its source code [72].

After experimenting with the two frameworks, we decided upon Flask. Flask got decided upon because of the short time frame to put the project together. Additionally, the lightweight nature of the framework also played a fact. As this will be just an initial prototype, Django's other requirements would be unessential additionals to the project. Therefore, taking focus away from what we believe is the main focus.

3.2.2.2 NLP Tasks

There are several NLP library packages already available within Python, all having pros and cons. The most popular and influential libraries are Natural Language Toolkit (NLTK) [56], Gensim [73], CoreNLP [74], spaCy [68], TextBlob [75].

Although NLTK, TextBlob was used in some experimenting, we decided to use spaCy as the main NLP library. However, NLTK was used on the side (especially with their stop words). As one of the key things we wanted to do was extract information from the tweets, spaCy allowed us to do this and prepare the data for deep learning. While we did not need a very deep Recurrent Neural Network (RNN), we did implement one to complete the sentiment analysis on the tweets. We used an RNN with two things in mind, to see how well it could perform on small amounts of text, like a tweet, and with the future thoughts of it being able to handle large amounts of text, like someone's essay in an exam. The RNN got constructed by using TensorFlow.

3.2.3 IDE

While many great IDEs are available like Pycharm, Jupyter Lab, Atom and Sublime, we decided to use VS Code. The decision behind this was that it allowed us to explore code within interactive python notebooks (ipynb) and standard python scripts. Additionally, it allowed us to create HTML, CSS, and Javascript files within the same IDE.

3.3 Ranking System

As discussed in the literature review, along with a more traditional pairwise comparative judgment algorithm, we could choose either an Elo or Glicko system. While each has advantages and disadvantages, we decided to use the Elo system. We decided to use this system as we felt it would be more robust for how we intend to be calculating the tweet scores, as we will be taking random pairings of tweets that will only be seen once by the user. Only seeing the tweet appear once removes any opportunity for a user to underrate a tweet because it has been seen multiple times without losing its impact.

Due to this reason, the Elo system, with its probability aspect to the scoring, helped determine outcomes on potential unseen tweet combos. While not considering if a tweet gets seen more than any others, this would have a massive impact on the CJ pairwise comparison method.

$$\text{Prob A Wins} = 1/1 + 10^{(B-A/400)}$$

Figure 3.4: To calculate the expected score for a tweet.

$$\text{new score} = \text{rating} + 32 * \text{score} - \text{expected score}$$

Figure 3.5: Formula to calculate the new Elo score for a tweet.

3.4 Data Set

There were two datasets used within this study. The primary dataset was the ten tweets gathered from Twitter, with a theme of being a joke based on Brexit. The other dataset was the IMDB sentiment analysis dataset. This dataset got used to train and test our RNN model before using our tweets on it.

3.4.1 Data Capture Method

Twitter's developer API got used to allow for the tweets to get extracted. Additionally, the library Tweepy [76] got also used. The tweets were then uploaded to the Firebase database [77] through a Python Notebook for the main web app to access them. Having the tweets in the database also allowed us to be then able to create a notebook to then access the data to then do the NLP investigating within.

3.4.2 Pre-Processing

Regarding the data pre-processing within the web app, the only processing occurred was removing the _b characters and replacing them with
 tags. We did this to allow the tweets to have the same layout as they did within Twitter. We decided that a few tweets, especially the Q+A style ones, lost their impact if they were not displayed correctly. Therefore, doing this allowed us to keep the integrity of the tweet and its comedy delivery.

3.4.3 T-Rating Score

The T-Rating (Twitter Rating) score is a metric that we created to use as a baseline comparison for the ranking methods we use within our application. The T-rating gets decided by adding up a tweets likes and retweets, then divided by the total number of followers the author of the tweet has.

We decided to normalise the data by using the number of followers a tweeter has. An assumption got made that an author with more followers is likely to have more retweets and likes due to more people being likely to see the tweet in the first place. Therefore this is, in essence, a weighted sum model for comparison [78]. While there are approaches that look to create a Tweet-ranking system using sentiment scores and popularity measures [79]. While we are aware that this approach would create a better ranking of Twitter tweets, we opted against implementing this due to time restraints. However, this should get further explored at a later date.

3.5 Implementation

The web application got implemented using the Python web library Flask. The web application used several industry-standard tools, for example, HTML, CSS, JavaScript, Bootstrap and dynamic content. The HTML, CSS, Bootstrap and JavaScipt was used to

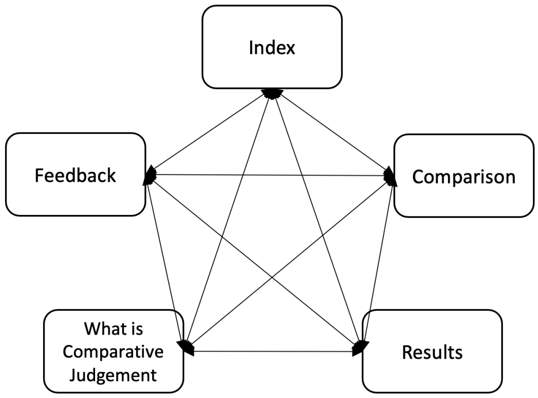


Figure 3.6: A visual representation of the web apps navigation. To see all web page designs, see appendix: A

handle the application's front end. The web application had a mesh style navigation system (see fig: 3.6). However, when the user was on the compare page, this would push to itself and update the users content based on what they had next in their comparison list.

Additional tools like Google's Firebase [77] was used to handle user authentication and store the web app's content in their real-time databases. The real-time databases are a NoSQL document notation database that updates in real-time.

A requirement of the app is for the user to be able to create an account. The account sign-up only requires an email and will generate all the additional requirements for the other parts of the app to work in the background. They are linking all the results for these comparisons to the user's ID. At the point of sign-up, a user position within the comparison cycle gets generated, a random selection of tweets to get compared against will be generated. The logic behind the sampling is that a user will only see a single tweet once. Therefore making sure that the user sees these tweets for the first time, every time, making it more of a fair comparison.

Heroku [80] handled the hosting of the web app. Heroku is a free-to-use web hosting provider. However, with it being a free-to-use service, it did bring about some undesirable aspects, mainly the website's slow loading time.

As previously mentioned, a user will have a random sample of the tweets, which will have a unique pairing. Therefore ensuring that a user will only see one tweet within the pairing once, to make the tweet's joke not lose its impact as the second or third time a user sees the same tweet, it naturally would lose its edge. Hence, each user will have their own predetermined set of comparisons at the point of sign-up but will one see, for example, tweet one once. As we mentioned, this was to keep the tweets fresh for the user and make them more likely to complete all the comparisons. Otherwise, if the user had to see all unique comparisons, they would have to see 45 different combinations in total just for ten different tweets. So if we put this into the context of a teacher, who would usually have 30 students in a class, several teachers will have to see 435 different combinations, which is just for one class. When this gets factored in, we are looking at around 11175 for 150 different students.

The app will query the database and look for the user's current position when presenting the tweets. Based on their position, the tweet combinations then get checked for that according to the round. The tweet IDs are then queried against the tweets' content and then presented to the web page. The user gets expected to select a tweet that they find funnier and then provide an opportunity to justify their choice, which is optional.

When the user presses the "Vote!" button, this saves the results to the database, updating the two result systems and the user's position. The process will save which tweet won and lost and update the Elo ranking and the standard CJ ranking. The standard ranking gets calculated by taking how many times a tweet has won minus the number it has lost. The implementation of the standard ranking system is to try to implement a more traditional comparative judgement ranking system. In contrast, the Elo system is using a more traditional approach (see figs: 3.4, 3.5) Which gets updated after every comparison. The implementation of the two systems allows us to see if the Elo or more standard version of CJ is the more effective one or if they naturally mirror each other. This process gets repeated until the user has completed all five comparisons.

To see the main Python scripts for the web app, please look at the appendix: E.

The NLP notebook is a more self-contained environment. The notebook has pre-written code and relies on all of the code getting executed to produce the required outputs and feedback. The notebook contains all of the information extraction techniques we explained in section: 3.1.3. To see the code, please look at the appendix: F.

Chapter 4

Results and Discussion

We will first look at the web application results based on the user's feedback, and then we will look into the insights and potential feedback the NLP process could provide the user. We will then also look to review the overall process.

We will compare the web application's results against the CJ, Elo ranking, and the score we created for the tweets on Twitter. With the insights of the NLP for feedback to the user, we will look at what insights got made. Additionally, we will look at if any of the knowledge extracted generated provides any meaningful feedback to the user.

4.1 Tweet Ranking Results

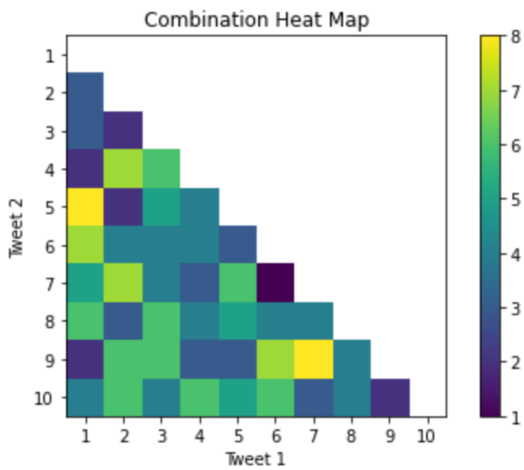


Figure 4.1: The web applicaitons generated results compared agaist each other.

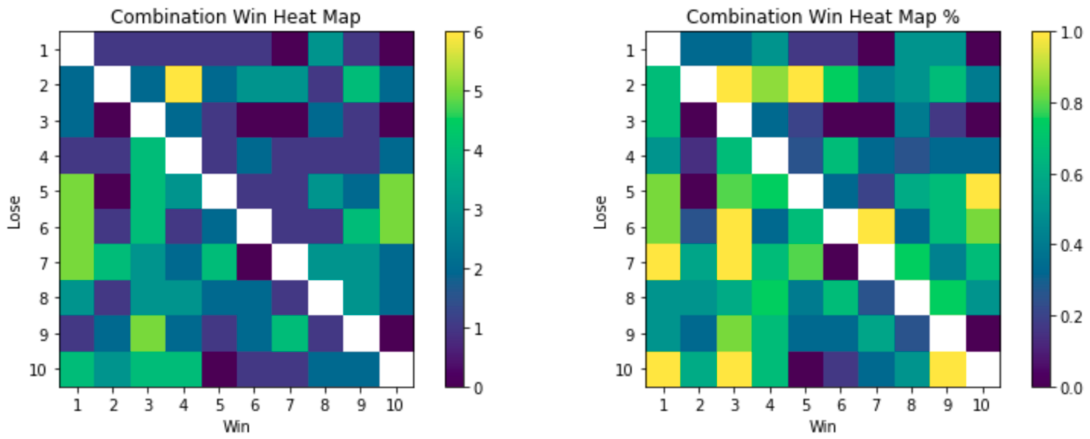


Figure 4.2: A heat map of the amount of times a tweet win or lost. Left - by total values. Right - By win percentages.

Forty different users take part in the comparison judgement within the web app. Through looking at fig: 4.1 we can see that all combinations got displayed to the users taking part in the comparisons. We can see that tweet 1 and tweet 5 appeared the most, while the combination appearing the lowest was tweet 6 and 7, with one comparison getting presented to the users.

When we look at winners and losers of the comparisons (see fig: 4.2), we can see that the tweet that won the most between a specific combination was tweet 4 and 2, with tweet 4 winning 6 times and tweet 2 winning only once. Additionally, when we look at the combination that appeared the most, 1 and 5, one came out on top 5 times, compared to 5 winning between the two once.

When we look at the winner heat map (see fig: 4.2), we can see that 2, 5, 6, 7 and ten had moments where they didn't win a head-to-head with another tweet. 2, 6, 7 and 10 didn't win against at least two different tweets, while the others were only against one tweet they failed to win. We can see that certain tweets never won against another tweet. For example, Tweet 10 never beat Tweet 9, which is also reflected in the ranking of the tweets, as Tweet 9 is ranked higher than Tweet 10 in both the Elo and CJ ranking table. The same can get said about Tweet 6 and 3, with Tweet 6 never beating Tweet 3, and Tweet 6 came 9th, and Tweet 3 came 1st in the rankings.

When we look at the two scores plotted against each other, Elo and CJ (see fig: 4.3), it shows that these values are linearly correlated. Additionally, the results returned as

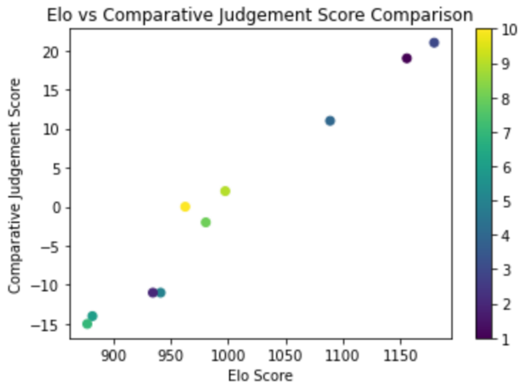


Figure 4.3: A scatter graph plotting each tweet againast their Elo and comparative judgement score.

0.98391595 when a Pearsons correlation test got conducted on these scores. Therefore, the two values are heavily linked, so when a tweet has a good Elo score, it also has a good CJ score. This correlation between the results shows that the Elo score is an excellent alternative to the CJ scoring system. Through using the Elo system, also provides the process with a lot more robustness. It allows the ranking to get done to a high degree of accuracy. Additionally, without every combination getting presented against each other. As a result, this would be a sound scoring system to implement at a national scaled-up scale.

While looking at fig 4.4, we can see that the Elo and CJ ranking generated very similar results. However, as we can see, the tweets coming in 6th, 7th, and 8th slightly vary in the results. These CJ results bring about some questions about whether further work is required to rank them more accurately. However, we need to ensure that the process does not end up having someone do multiple rounds and then expand the time required to complete the CJ, taking away any actual benefits. But it does bring to light how effective the Elo ranking system is and can handle these situations. It takes a score calculation based on the likelihood that the tweet will win, rather than a more dogmatic approach of the total wins minus the total losses.

While we look at the T-rating ranking compared to the Elo ranking (see fig: 4.5), we can see that the results ranking is very different. The tweet that came 1st in the T-rating came 4th in the Elo ranking. At the same time, the tweet that came 1st in the Elo ranking came 8th in the T-rating. Tweets that done worse in the Elo ranking compared to T-rating had an average difference in the ranked placing of 5 places, while the tweets that had a better Elo ranking compared to the T-ranking ranked an average of 4 places lower. Therefore, 4 of the top 5 tweets in the T-rating were actually in the bottom five of the Elo ranking.

Tweet ID	Content	ELO Ranking	ELO Score	CJ Ranking	CJ Score	+/-
3	Q: With Britain leaving the EU how much space was created? A: Exactly 1GB	1	1179.3849804860672	1	21	0
1	An Englishman, a Scotsman and an Irishman walk into a bar. The Englishman wanted to go so they all had to leave. #Brexitjokes	2	1155.592817447446	2	19	0
4	VOTERS: we want to give a boat a ridiculous name UK: no VOTERS: we want to break up the EU and trash the world economy UK: fine	3	1088.8199623047965	3	11	0
9	Hello, I am from Britain, you know, the one that got tricked by a bus	4	997.5535634725744	4	2	0
8	Say goodbye to croissants, people. Delicious croissants. We're stuck with crumpets FOREVER.	5	980.635912446213	6	-2	-1
10	How many Brexiteers does it take to change a light bulb? None, they are all walked out because they didn't like the way the electrician did it.	6	962.7368861475267	5	0	+1
5	#BrexitJokes How did the Brexit chicken cross the road? I never said there was a road. Or a chicken.	7	941.3060728832675	8	-11	-1
2	Why do we need any colour passport? We should just be able to shout, "British! Less of your nonsense!" and stroll straight through.	8	934.560236052883	7	-11	+1
6	After #brexit, when rapper 50 cent performs in GBR he'll appear as 10.000 pounds. #brexitjokes	9	881.9366306271611	9	-14	0
7	I long for the simpler days when #Brexit was just a term for leaving brunch early.	10	877.4729381320648	10	-15	0

Figure 4.4: The web applicaitons generated results compared agaist each other.

Only tweet ID 4 done one place better with the Elo ranking than it did in the T-ranking. However, two of the top three tweets in the T-rating were in the bottom three of the Elo ranking and vice versa. The T-rating score has a correlation score of -0.14360792 against Elo and -0.09776676 against CJ. They were showing us that there is a negative correlation between the scores. It does make it seem like how popular something is on Twitter doesn't mean it is necessarily a funnier tweet when carried out in a controlled environment.

However, even though these ended up with very different results (see fig: 4.6), due to the multiple variables at play regarding Twitter, in terms of likes, retweets, followers, how many followers retweeters have, a tweet might have, the random chance of someone seeing it. The T-rating system is a very ambiguous metric to use as an accurate ranking

Tweet ID	Content	T-Rating Ranking	T-Rating Score	CJ Ranking	+/-
9	Hello, I am from Britain, you know, the one that got tricked by a bus	1	0.57971014	4	-3
2	Why do we need any colour passport? We should just be able to shout, “British! Less of your nonsense!” and stroll straight through.	2	0.20507084	8	-6
6	After #brexit, when rapper 50 cent performs in GBR he'll appear as 10.00 pounds. #brexitjokes	3	0.14233577	9	-6
4	VOTERS: we want to give a boat a ridiculous name UK: no VOTERS: we want to break up the EU and trash the world economy UK: fine	4	0.13602305	3	+1
7	I long for the simpler days when #Brexit was just a term for leaving brunch early.	5	0.05430769	10	-5
8	Say goodbye to croissants, people. Delicious croissants. We're stuck with crumpets FOREVER.	6	0.03097458	5	+1
10	How many Brexiteers does it take to change a light bulb? None, they are all walked out because they didn’t like the way the electrician did it.	7	0.02849923	6	+1
3	Q: With Britain leaving the EU how much space was created? A: Exactly 1GB	8	0.01221757	1	+7
1	An Englishman, a Scotsman and an Irishman walk into a bar. The Englishman wanted to go so they all had to leave. #Brexitjokes	9	0.01165323	2	+7
5	#BrexitJokes How did the Brexit chicken cross the road? "I never said there was a road. Or a chicken".	10	0.00552061	7	+3

Figure 4.5: The Twitter tweet score ranking comparison against Elo ranking.

system. Additionally, with Twitter being a global app, the results on certain tweets could be affected by people’s views from outside the UK, drastically changing opinions. Another factor that is making this a difficult comparison to make is the sample size. The tweets on Twitter had many more people interacting with them than how many people took part in our study. Therefore, how the tweet did in the real world is not a valid comparison against the Elo rankings results. There is also room to suggest that this proves that the Elo system is better suited for this action, as it can handle random elements of its pairings.

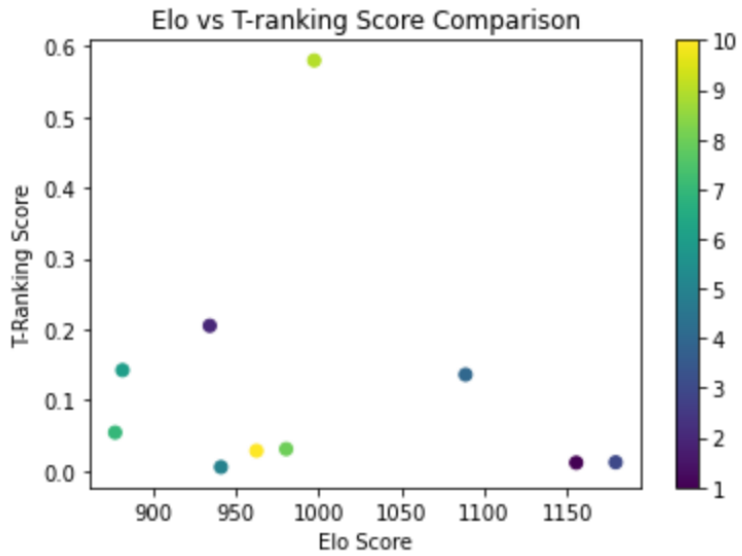


Figure 4.6: The Twitter tweet score ranking plotted against Elo ranking.

However, this comparison has brought to light a valid point: do we want the results to be decided upon by a local group of specialised people? Or do we want the results to get agreed upon as a global element? For example, teachers within a school in the UK would possibly be looking for different work factors compared to a teacher in Finland. Therefore, creating contrast in views. Additionally, GCSE awards bodies might also have different focuses within their assessments, even when it comes to subjects like English. So this could have a huge impact on views getting generated around the ranking of students work which would need further investigating.

Within the forty participants, twenty-two of them left a justification for why they select one tweet over the other. However, the participants' responses were varied in the amount of provided feedback. Some proved a justification for all five combinations. On the other hand, some only left them for a few and not all. The users gave a total of sixty-three explanations to their decisions on which tweet they had chosen.

One user stated, "I just think it is a clever way to put our departure from EU, plus it did make me giggle." The comment was in regards to tweet 3 beating tweet 8. Tweet 3 did provide several justifications, a lot of them to do around its tech theme on Brexit. Some of the rationales are "Comp sci wordplay", "everyone loves a tech joke", "Because it's the nerdier option", the "First tweet just lol", and "Actually laughed out loud".

Another tweet, tweet 10 beating tweet 8, had the justification for winning as 'because of the wordplay'. So we can see that several tweets had some form of explanation around the lines of good wordplay. Therefore, creating user feedback has not made an excellent source of information to help build feedback. However, it has given some context to why they had made their decisions.

4.2 NLP Feedback and Insights

The Jupyter notebook was able to conduct the NLP tasks that we required successfully. We presented the user the POS tagging insights of how many POS tags were present in each tweet. We were also able to visualise the POS tagging to reflect the user how the tweet got broken down structure-wise (see fig: 4.7).

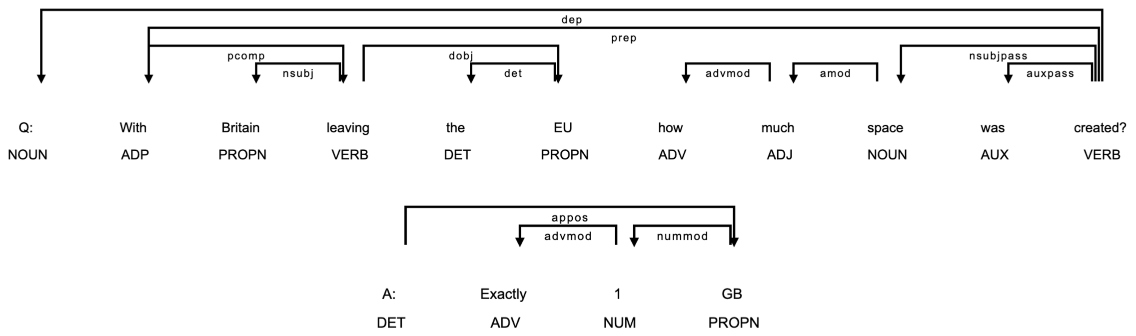


Figure 4.7: An example of a POS tagging visulaisation. To see all the outputs, please look at appendix: G

We were also able to present to the user the NER that the pre-trained model supplied by spaCy was able to identify. These were presented to the user in text format as well within a visualisation, identifying the NERs within the sentence (see fig: 4.8).

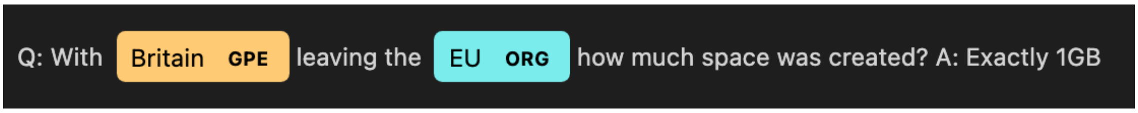


Figure 4.8: An example of a NER tagging visulaisation. To see all the outputs, please look at appendix: H

The notebook was also able to present back to the user the top ten tweets on how similar they were by the whole tweet (see fig: 4.10) and by NERs (see fig: 4.9). When looking at the results for the similarity scoring between the NERs, we can see that the most similar tweets are tweet three and tweet 4. These tweets have a similarity score of 0.857896 based on the NER values Britain, EU (Tweet 1) and UK, EU (Tweet 4). The tweets with the least similarity are Tweet 2, British, and Tweet 6, 50, cent, 10.00, pounds, with a similarity score of -0.025753.

	similarity	tweet1	tweet1 NE Span	tweet2	tweet2 NE Span
17	0.857896	3	(Britain, EU)	4	(UK, EU)
1	0.788178	1	(Scotsman, Irishman, Englishman)	3	(Britain, EU)
33	0.771924	5	(Brexit)	9	(Britain)
2	0.720223	1	(Scotsman, Irishman, Englishman)	4	(UK, EU)
18	0.688950	3	(Britain, EU)	5	(Brexit)
22	0.646520	3	(Britain, EU)	9	(Britain)
24	0.598866	4	(UK, EU)	5	(Brexit)
16	0.549264	2	(British)	10	(Brexiteers)
7	0.510660	1	(Scotsman, Irishman, Englishman)	9	(Britain)
3	0.510251	1	(Scotsman, Irishman, Englishman)	5	(Brexit)

Figure 4.9: A table displaying the top ten similar tweets based on the tweet's NERs.

The results show us, in regards to the whole tweets, that Tweet 5 and Tweet 10 were the most similar with a similarity score of 0.576191. The tweet's contents were '#BrexitJokes How did the Brexit chicken cross the road? "I never said there was a road. Or a chicken".' (Tweet 5) and 'How many Brexiteers does it take to change a light bulb? None, they are all walked out because they didn't like the way the electrician did it.' (Tweet 10). The tweets with the least similarity are Tweet 4, 'VOTERS: we want to give a boat a ridiculous name UK: no VOTERS: we want to break up the EU and trash the world economy UK: fine', and Tweet 6, 'After #brexit, when rapper 50 cent performs in GBR he'll appear as 10.00 pounds. #brexitjokes', with a similarity score of -0.041637.

	similarity	text 1	text 2	tweet1	tweet2
34	0.576191	(#, BrexitJokes, How, did, the, Brexit, chicke...	(How, many, Brexiteers, does, it, take, to, ch...	5	10
3	0.575611	(An, Englishman, ,, a, Scotsman, and, an, Iris...	(#, BrexitJokes, How, did, the, Brexit, chicke...	1	5
16	0.490846	(Why, do, we, need, any, colour, passport, ?, ...	(How, many, Brexiteers, does, it, take, to, ch...	2	10
2	0.490278	(An, Englishman, ,, a, Scotsman, and, an, Iris...	(VOTERS, :, we, want, to, give, a, boat, a, ri...	1	4
14	0.489872	(Why, do, we, need, any, colour, passport, ?, ...	(Say, goodbye, to, croissants, ,, people, ., D...	2	8
8	0.462386	(An, Englishman, ,, a, Scotsman, and, an, Iris...	(How, many, Brexiteers, does, it, take, to, ch...	1	10
11	0.458674	(Why, do, we, need, any, colour, passport, ?, ...	(#, BrexitJokes, How, did, the, Brexit, chicke...	2	5
18	0.456565	(Q, :, With, Britain, leaving, the, EU, how, m...	(#, BrexitJokes, How, did, the, Brexit, chicke...	3	5
20	0.439537	(Q, :, With, Britain, leaving, the, EU, how, m...	(I, long, for, the, simpler, days, when, #, Br...	3	7
7	0.406573	(An, Englishman, ,, a, Scotsman, and, an, Iris...	(Hello, ,, I, am, from, Britain, ,, you, know,...	1	9

Figure 4.10: A table displaying the top ten similar tweets based on the whole tweet.

The information extraction process identified several interesting aspects from the tweets (see fig: 4.11). The results show that six of the tweet's sentiments scoring got classified as positive, and out of those six, five were in the top 5 results. We can't say for certain that having a positive tweet will likely score higher, as the dataset is not big enough to make that kind of claim. However, it does provide some good feedback and insights to the user. The NLP process also provided some excellent extraction of key phrases from the tweets. The only tweet's key phrase that didn't prove any meaningful information was Tweet 7's 'Brexit was'. Considering that these information extraction techniques, NER and key phrases, have not had any additional training, other than what comes out of the box, they have performed well in providing insights and feedback to the user.

Tweet ID	Named Entity Recognition	Sentiment Analysis	Key Phrases
1	Scotsman - PERSON - People, including fictional Irishman - NORP - Nationalities or religious or political groups Englishman - PERSON - People, including fictional	Positive	Englishman wanted
2	British - NORP - Nationalities or religious or political groups	Positive	need ing passport
3	Britain - GPE - Countries, cities, states EU - ORG - Companies, agencies, institutions, etc.	Positive	Britain leaving
4	UK - GPE - Countries, cities, states EU - ORG - Companies, agencies, institutions, etc.	Positive	trash ing UK
5	Brexit - PERSON - People, including fictional	Negative	chicken cross
6	50 cent - MONEY - Monetary values, including unit 10.00 pounds - MONEY - Monetary values, including unit	Negative	appear ing performs
7	the simpler days - DATE - Absolute or relative dates or periods Brexit - PERSON - People, including fictional	Negative	Brexit was
8	FOREVER - WORK_OF_ART - Titles of books, songs, etc.	Positive	Say ing goodbye
9	Britain - GPE - Countries, cities, states	Positive	False
10	Brexiteers - WORK_OF_ART - Titles of books, songs, etc.	Negative	electrician did

Figure 4.11: A table displaying the key information extracted from the NER, Sentiment analysis and Key Phrases NLP processes.

Using the TF-IDF, we extracted the key token features from all of the tweets. The higher the value, the more important that feature is for that tweet (see fig: 4.12). However, this information does not provide much feedback for a user, but it would highly likely be adequate for training some form of ML models.

	all	and	brexit	brexitjokes	britain	did	eu	how	just	leaving	the eu	they	to	was	we	when	with
0	0.427075	0.373640	0.000000	0.373640	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.427075	0.596656	0.000000	0.000000	0.000000	0.000000
1	0.000000	0.379486	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.433757	0.000000	0.000000	0.000000	0.302996	0.000000	0.758972	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.391305	0.000000	0.391305	0.342346	0.000000	0.391305	0.391305	0.000000	0.000000	0.342346	0.000000	0.000000	0.391305
3	0.000000	0.313682	0.000000	0.000000	0.000000	0.000000	0.358542	0.000000	0.000000	0.000000	0.358542	0.000000	0.500911	0.000000	0.627365	0.000000	0.000000
4	0.000000	0.000000	0.434107	0.434107	0.000000	0.496189	0.000000	0.434107	0.000000	0.000000	0.000000	0.000000	0.000000	0.434107	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.549943	0.549943	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.628591	0.000000
6	0.000000	0.000000	0.411017	0.000000	0.000000	0.000000	0.000000	0.000000	0.469798	0.469798	0.000000	0.000000	0.000000	0.411017	0.000000	0.469798	0.000000
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.465343	0.000000	0.582818	0.000000	0.666168
8	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	0.371304	0.000000	0.000000	0.000000	0.000000	0.371304	0.000000	0.324847	0.000000	0.000000	0.000000	0.742609	0.259370	0.000000	0.000000	0.000000	0.000000

Figure 4.12: A table showing the key tokens within each tweet and thir importance to that tweet.

In contrast, the information extraction techniques of finding word sequence patterns and utterance pattern matching did not provide any meaningful information. The finding word sequence pattern presented only "he'll appear" about Tweet 6, and the utterance pattern matching showed that a pattern was found in Tweet 6 too. These techniques have not provided much use currently but could be helpful when scaling up and using a much bigger dataset, like exam papers.

4.3 Overall Results

Overall we can suggest that the Elo ranking is a great alternative ranking system to the ACJ. It provides a robust scoring system because the combination process is random, removing any opportunity for Elo's flaws to be taken advantage of and does remove any ACJ bias from marking. It also provides the ability to try 'what if' calculations with potential comparison outcomes.

On the other hand, the NLP information extraction provided some good information but was too basic to offer any real insights to the user to digest easily. While there is a lot of promise regarding the NLP, more fine-tuning is required to make this feature to provide feedback worthwhile. However, we believe this is a step worth taking with appropriate building blocks that have been put in place to expand upon.

Chapter 5

Conclusions and Future Work

The process of CJ is undeniable in reducing cognitive load, as our brains are much more adapt to comparing one thing to another and saying one is better. The literature around CJ firmly claims that ACJ is a better alternative to more traditional marking methods, for example, using a rubric. CJ does have several flaws. One of the flaws is that the whole process can take longer than traditional marking in the first place. Additionally, the adaptive nature of ACJ can generate bias within its results by getting the markers to mark more often, especially when the results that get closely ranked to each other. It gets claimed that a random pairing is better than the adaptive approach. A considerable flaw within the CJ/ACJ process is that it does not provide personalised feedback to the learners. Giving feedback is a vital part of education today, ensuring that students know where they are and where they need to improve. Instead, CJ's feedback approach is to allow students to peer-assess each other and then gain their insights from their understanding. However, this relies on the students understanding the marking criteria in the first place and extracting what they need to improve on.

While CJ generates results to create a ranking of the students' work, CJ is not the only ranking method available. There are multiple ranking systems that can get used within competitive chess and e-Sports. Two such methods are the Elo and Glicko ranking. While the Glicko system is a proposed improved system over Elo, the Glicko system introduces features that we did not need, and the flaws within the Elo system would not get abused within our proposed solution. Therefore we decided to use the Elo ranking system.

Therefore, we created a web app that allowed users to compare two tweets and declare what tweet they prefered. The results then got used to calculate a simplified

CJ score and an Elo score, allowing us to compare the final results of the two ranking systems. Additionally, a Jupyter notebook got created to carry out information extraction techniques. These techniques include POS tagging, NER, feature extraction, sentiment analysis, text similarity scoring, utterance pattern matching, finding word sequence patterns and finally extracting key phrases.

The results from the web app presented that the final Elo ranking and the CJ score a strongly correlated, with a score of 0.98391595. The web app allowed the users to complete the comparisons very quickly and only have to do one round of judgements. Therefore, reducing cognitive load and reducing the time required for marking. However, the scores only became truly useful after a number of users had completed the comparison. Still, the more users took part, the more sure the final results became, with the results showing that the Elo system is a suitable method for ranking the results.

In contrast, when we compared Elo's scores ranking against the T-rating, these did not correlate with each other. But we believe that this is not a very straightforward comparison, but it does bring up questions to think about. For example, do we want a selection of specialised local markers to conduct the CJ in the future or is using a global approach ok? Also, how would the outcome be with a larger sample size getting used, rather than the 40 users who took part?

While the web app generated a strong argument for using the Elo ranking system, the NLP notebook for information extraction did not provide the exact outcome we expected. While the notebook did complete all the NLP tasks we required, it did produce some good insights into the tweets. It did not manage to provide any real insights that an end-user could use to provide personalised feedback. However, it did create great building blocks to build upon.

Overall, we conclude that the research ended up with many positives, but some areas need development, especially when providing feedback using NLP techniques. However, the study has shown that the Elo system has a solid case for getting used for ranking work. As it massively reduces the time required to complete compared to ACJ methods. Additionally, the process also being based around CJ reduces the cognitive load for anyone taking part in the judging. Therefore, we believe there is a lot of potential within combining these techniques.

5.1 Contributions

The main contributions of this work can be seen as follows:

- **A web applicaiton to conduct the comparative judgement**

  We created a web application and hosted it to crowdsource users views on ten tweets based on Brexit. The app provided at random five unique pair comparisons while updating the CJ score and Elo score.

- **A comparion of two different ranking systems**

  Metrics are being stored and calculated based on the two ranking systems, a CJ style and an Elo ranking system. Therefore, the results provide us with a way to compare the effectiveness of the two ranking systems. As a result, they are allowing us to see which one works better in our required situation.

- **An exploration into NLP techniques to provide feedback to the user**

  We created a Jupyter notebook exploring NLP information extraction techniques to provide feedback to the user from information extracted from the ten tweets.

5.2 Future Work

While the research found some good insights, we believe much future work can get done. We believe that a bigger pool of samples needs to occur for the Elo system to be assured as an alternative to the ACJ method. Additionally, introducing the markers and seeing how long it takes for the sample pool to be marked and how well it ranks against a more traditional rubric marking method.

More work can be done with the Elo score and converting the results into grades from A* to F. We believe that a process can convert the results created by the Elo score into standardised GCSE grades. For example, an Elo score greater than 1800 is equivalent to an A*, or a score greater than 1700 resulting in an A grade.

However, where we feel a lot more research can get done is within the NLP capabilities. We believe that the ability to extract the information from a student's work and then provide personised feedback would be a fantastic addition to the CJ process. Therefore, allowing teachers to reduce their cognitive load and workload, as giving feedback would taking a time consuming and draining task away from them. Having the NLP processes automated, but allowing the teacher to have overall control, would be a massive addition to any teacher's toolbox. Ultimately reducing their workload and allowing the teacher to do what they are best at, creating engaging lessons for their students.

Bibliography

[1] L. L. Thurstone, "A law of comparative judgment." *Psychological review*, vol. 34, no. 4, p. 273, 1927.

[2] UK Public General Acts, "Education act 1988," 1988.

[3] D. Hutchison and I. Schagen, *How reliable is National Curriculum assessment?* NFER, 1994.

[4] J. Dillon and M. Maguire, *Becoming a teacher: Issues in secondary education*. McGraw-Hill Education (UK), 2011.

[5] J. Wellington, *Secondary education: The key concepts*. Routledge, 2007.

[6] P. Black and D. William, "Inside the black box: Raising standards through classroom assessment. phi delta kappam," 1998.

[7] A. Pollitt, "The method of adaptive comparative judgement," *Assessment in Education: principles, policy & practice*, vol. 19, no. 3, pp. 281–300, 2012.

[8] L. L. Thurstone, "Psychophysical analysis," *The American journal of psychology*, vol. 38, no. 3, pp. 368–389, 1927.

[9] RM Compare. (2021) Assessment for learning with rm compare. [Online]. Available: https://www.rm.com/products/rm-compare

[10] T. Bramley, "Investigating the reliability of adaptive comparative judgment," *Cambridge Assessment, Cambridge*, vol. 36, 2015.

[11] S. McMahon and I. Jones, "A comparative judgement approach to teacher assessment," *Assessment in Education: Principles, Policy & Practice*, vol. 22, no. 3, pp. 368–389, 2015.

[12] J. T. Steedle and S. Ferrara, "Evaluating comparative judgment as an approach to essay scoring," *Applied Measurement in Education*, vol. 29, no. 3, pp. 211–223, 2016.

[13] A. E. Elo, *The Rating of Chessplayers, Past and Present*. New York: Arco Pub., 1978. [Online]. Available: http://www.amazon.com/Rating-Chess-Players-Past-Present/dp/0668047216

[14] M. E. Glickman, "The glicko system," *Boston University*, vol. 16, pp. 16–17, 1995.

[15] M. Glickman, "Example of the glicko-2 system," *Boston University*, pp. 1–6, 2012.

[16] UK Public General Acts, "Education act 1918," 1918.

[17] BBC News. (2004) Primary school tests toned down. [Online]. Available: http://news.bbc.co.uk/1/hi/education/3656244.stm

[18] BBC. (2008) Tests scrapped for 14-year-olds. [Online]. Available: http://news.bbc.co.uk/1/hi/education/7669254.stm

[19] Department for Education. (2013) Assessing without levels. [Online]. Available: https://webarchive.nationalarchives.gov.uk/ukgwa/20130802141012/https://www.education.gov.uk/schools/teachingandlearning/curriculum/nationalcurriculum2014/a00225864/assessing-without-levels

[20] H. Torrance and J. Pryor, *Investigating formative assessment: Teaching, learning and assessment in the classroom*. McGraw-Hill Education (UK), 1998.

[21] P. Black and C. Harrison, "Feedback in questioning and marking: The science teacher's role in formative assessment," *School science review*, vol. 82, no. 301, pp. 55–61, 2001.

[22] OECD. (2005) Formative assessment: Improving learning in secondary classrooms. [Online]. Available: https://www.oecd.org/education/ceri/35661078.pdf

[23] D. William, "National curriculum assessment arrangements," *British Journal for Curriculum and Assessment*, vol. 1, pp. 8–12, 1990.

[24] Reasearch ED. (2018) Comparative judgement: the next big revolution in assessment? [Online]. Available: https://researched.org.uk/2018/07/06/comparative-judgement-the-next-big-revolution-in-assessment-2/

[25] J. Arbuckle and J. H. Nugent, "A general procedure for parameter estimation for the law of comparative judgement," *British Journal of Mathematical and Statistical Psychology*, vol. 26, no. 2, pp. 240–260, 1973.

[26] R. M. Furr, *Psychometrics: an introduction*. SAGE publications, 2021.

[27] G. A. Gescheider, *Psychophysics: the fundamentals*. Psychology Press, 2013.

[28] S. E. Embretson and S. P. Reise, *Item response theory*. Psychology Press, 2013.

[29] B. D. Wright and M. Mok, "Understanding rasch measurement: Rasch models overview." *Journal of applied measurement*, 2000.

[30] A. Pollitt and N. L. Murray, "What raters really pay attention to," *Studies in language testing*, vol. 3, pp. 74–91, 1996.

[31] D. Andrich, "A rating formulation for ordered response categories," *Psychometrika*, vol. 43, no. 4, pp. 561–573, 1978.

[32] P. Newton, J.-A. Baird, H. P. Harvey Goldstein, and P. Tymms, "Paired comparison methods," 2007.

[33] A. Pollitt, "Let's stop marking exams," 01 2004.

[34] ——, "Abolishing marksism and rescuing validity," *International Association for Educational Assessment, Brisbane, Australia. http://www. iaea. info/documents/paper_4d527d4e. pdf*, 2009.

[35] S. R. Bartholomew, L. Zhang, E. Garcia Bravo, and G. J. Strimel, "A tool for formative assessment and learning in a graphics design course: Adaptive comparative judgement," *The Design Journal*, vol. 22, no. 1, pp. 73–95, 2019.

[36] N. Seery, D. Canty, and P. Phelan, "The validity and value of peer assessment using adaptive comparative judgement in design driven practical education," *International Journal of Technology and Design Education*, vol. 22, no. 2, pp. 205–226, 2012.

[37] T. Potter, L. Englund, J. Charbonneau, M. T. MacLean, J. Newell, I. Roll *et al.*, "Compair: A new online tool using adaptive comparative judgement to support learning with peer feedback," *Teaching & Learning Inquiry*, vol. 5, no. 2, pp. 89–113, 2017.

[38] N. Seery, J. Buckley, T. Delahunty, and D. Canty, "Integrating learners into the assessment process using adaptive comparative judgement with an ipsative approach to identifying competence based gains relative to student ability levels," *International Journal of Technology and Design Education*, vol. 29, no. 4, pp. 701–715, 2019.

[39] R. C. Weng and C.-J. Lin, "A bayesian approximation method for online ranking." *Journal of Machine Learning Research*, vol. 12, no. 1, 2011.

[40] N. Silver and R. Fischer-Baum, "How we calculate nba elo ratings," *Dostopno na: http://fivethirtyeight. com/features/how-we-calculate-nba-elo-ratings*, 2015.

[41] S. Pradhan and Y. Abdourazakou, ""power ranking" professional circuit esports teams using multi-criteria decision-making (mcdm)," *Journal of Sports Analytics*, vol. 6, no. 1, pp. 61–73, 2020.

[42] C. Sullivan and C. Cronin, "Improving elo rankings for sports experimenting on the english premier league," *Virginia Tech CSx824/ECEx424 technical report, VA, USA*, 2016.

[43] H. L. Friedman, *Playing to win*. University of California Press, 2013.

[44] S. Edelkamp, "Elo system for skat and other games of chance," *arXiv preprint arXiv:2104.05422*, 2021.

[45] G. J. Williams, *Abstracting Glicko-2 for team games*. University of Cincinnati, 2013.

[46] M. Glickman. (2021) Mark glickman's world. [Online]. Available: http://www.glicko.net/glicko.html

[47] Y. Vasiliev, *Natural Language Processing with Python and SpaCy: A Practical Introduction*. No Starch Press, 2020.

[48] S. Vajjala, B. Majumder, A. Gupta, and H. Surana, *Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems*. O'Reilly Media, 2020.

[49] N. M. Marking.

[50] C. Wheadon. (2014) Open sourcing our comparative judgement algorithms. [Online]. Available: https://blog.nomoremarking.com/open-sourcing-our-comparative-judgement-algorithms-84d12d92f9c4

[51] npm. (2021) comparative-judgement. [Online]. Available: https://www.npmjs.com/package/comparative-judgement

[52] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The annals of mathematical statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[53] H. Hapke, C. Howard, and H. Lane, *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python*. Simon and Schuster, 2019.

[54] D. Sarkar, *Text Analytics with python*. Springer, 2016.

[55] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[56] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[57] spaCy. (2021) Linguistic features. [Online]. Available: https://spacy.io/usage/linguistic-features#vectors-similarity

[58] N. Béchet, P. Cellier, T. Charnois, and B. Crémilleux, "Discovering linguistic patterns using sequence mining," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2012, pp. 154–165.

[59] C. Nédellec, "Machine learning for information extraction in genomics—state of the art and perspectives," in *Text Mining and its Applications*. Springer, 2004, pp. 99–118.

[60] E. Riloff, "Automatically generating extraction patterns from untagged text," in *Proceedings of the national conference on artificial intelligence*, 1996, pp. 1044–1049.

[61] Trello. (2021) 6 essential trello templates you need to run a business. [Online]. Available: https://blog.trello.com/essential-trello-boards-for-every-business

[62] M. Rehkopf, "What is a kanban board?" Retrieved April 29, 2020 from: https://www.atlassian.com/agile/kanban/boards.

[63] K. Arnold, J. Gosling, and D. Holmes, *The Java programming language*. Addison Wesley Professional, 2005.

[64] S. S. Bakken, Z. Suraski, and E. Schmid, *PHP Manual: Volume 1*. iUniverse, Incorporated, 2000.

[65] D. Flanagan, *JavaScript: the definitive guide*. " O'Reilly Media, Inc.", 2006.

[66] Python Core Team, *Python: A dynamic, open source programming language*, Python Software Foundation, Vienna, Austria, 2020. [Online]. Available: https://www.python.org/

[67] E. Loper and S. Bird, "Nltk: The natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[68] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017, to appear.

[69] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[70] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[71] Django. (2021) Meet django. [Online]. Available: https://www.djangoproject.com/

[72] M. Grinberg, *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.", 2018.

[73] Gensim. (2021) Topic modelling for humans. [Online]. Available: https://radimrehurek.com/gensim/

[74] CoreNLP. (2021) Corenlp. [Online]. Available: https://stanfordnlp.github.io/CoreNLP/

[75] S. Loria, "textblob documentation," *Release 0.15*, vol. 2, 2018.

[76] J. Roesslein, "Tweepy: Twitter for python!" *URL: https://github.com/tweepy/tweepy*, 2020.

[77] L. Moroney, "The firebase realtime database," in *The Definitive Guide to Firebase*. Springer, 2017, pp. 51–71.

[78] C. W. Churchman and R. L. Ackoff, "An approximate measure of value," *Journal of the Operations Research Society of America*, vol. 2, no. 2, pp. 172–187, 1954.

[79] S. Aleidi, D. Alsuhaibani, N. Alrajebah, and H. Kurdi, "A tweet-ranking system using sentiment scores and popularity measures," in *International Conference on Computing*. Springer, 2019, pp. 162–169.

[80] N. Middleton and R. Schneeman, *Heroku: up and running: effortless application deployment and scaling*. " O'Reilly Media, Inc.", 2013.

Appendix A

Web App Pages

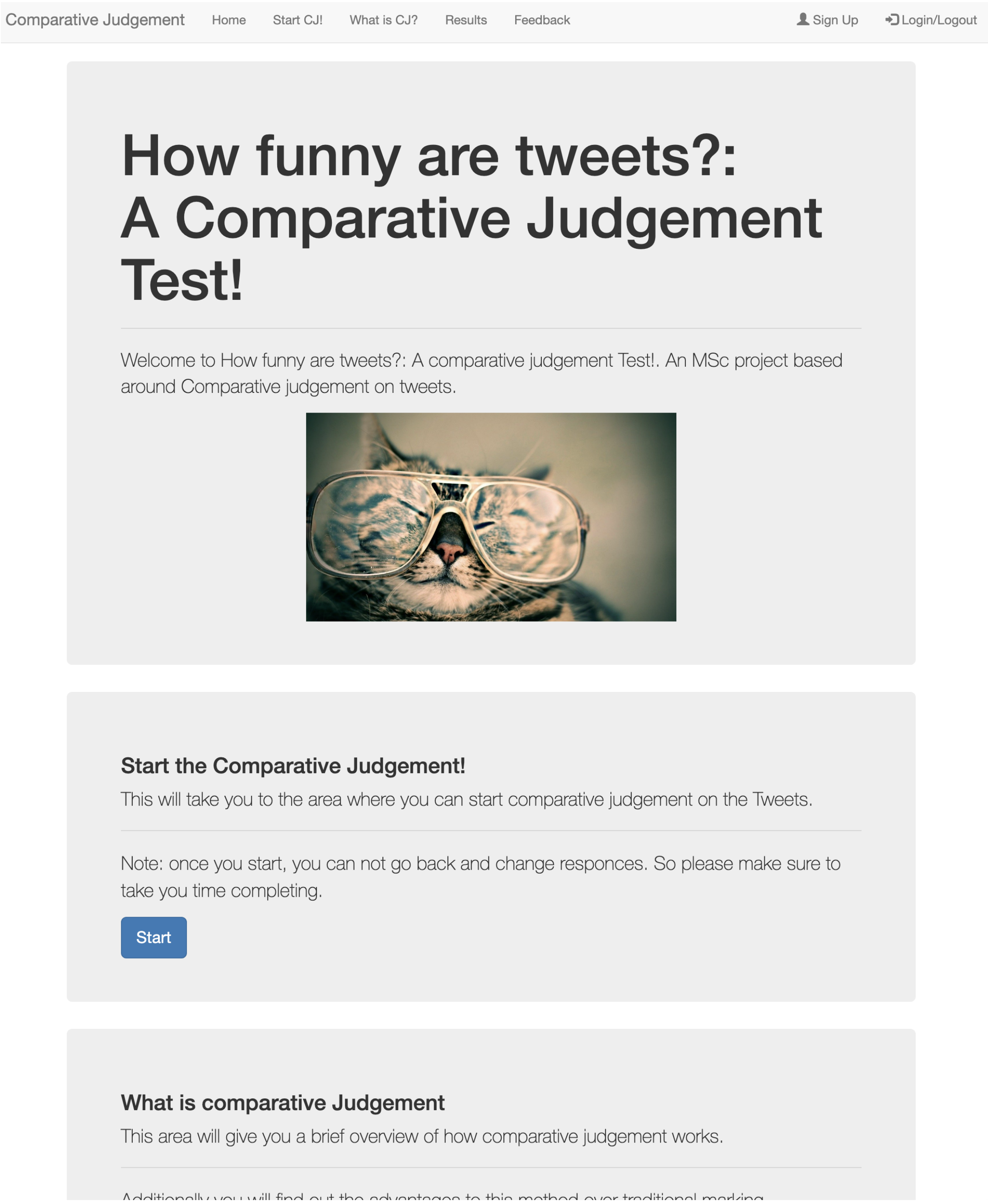

Comparative Judgement
Home
Start CJ!
What is CJ?
Results
Feedback
Sign Up
Login/Logout
How funny are tweets?:
A Comparative Judgement
Test!
Welcome to How funny are tweets?: A comparative judgement Test!. An MSc project based around Comparative judgement on tweets.
Start the Comparative Judgement!
This will take you to the area where you can start comparative judgement on the Tweets.
Note: once you start, you can not go back and change responces. So please make sure to take you time completing.
Start
What is comparative Judgement
This area will give you a brief overview of how comparative judgement works.

Figure A.1

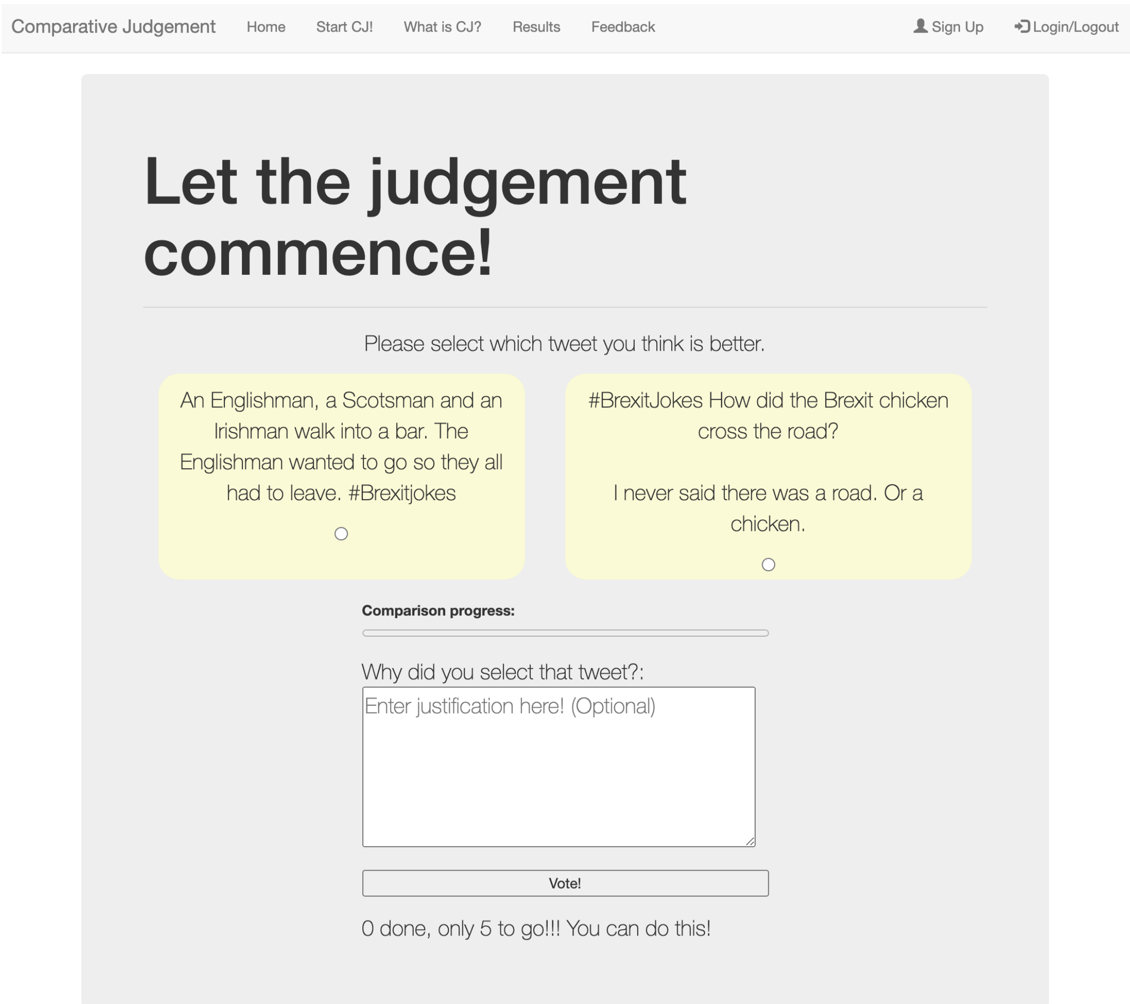

Comparative Judgement
Home
Start CJ!
What is CJ?
Results
Feedback
Sign Up
Login/Logout
Let the judgement commence!
Please select which tweet you think is better.
An Englishman, a Scotsman and an Irishman walk into a bar. The Englishman wanted to go so they all had to leave. #Brexitjokes
#BrexitJokes How did the Brexit chicken cross the road?
I never said there was a road. Or a chicken.
Comparison progress:
Why did you select that tweet?:
Enter justification here! (Optional)
Vote!
0 done, only 5 to go!!! You can do this!

Figure A.2

Comparative Judgement Home Start CJ! What is CJ? Results Feedback Sign Up Login/Logout

Comparative Judgement Results

Welcome to How funny are tweets?: A comparative judgement Test!. An MSc project based around Comparative judgement on tweets.

The ELO Results are in........!

1. Tweet: 3
   Score: 1175.1014325267206
   Content:
   Q: With Britain leaving the EU how much space was created?
   A: Exactly 1GB

2. Tweet: 1
   Score: 1160.4817903284847
   Content:
   An Englishman, a Scotsman and an Irishman walk into a bar. The Englishman wanted to go so they all had to leave. #Brexitjokes

3. Tweet: 4
   Score: 1105.4998341784085
   Content:
   VOTERS: we want to give a boat a ridiculous name
   UK: no
   VOTERS: we want to break up the EU and trash the world economy
   UK: fine

4. Tweet: 10
   Score: 1011.5592350105783
   Content:
   How many Brexiteers does it take to change a light bulb?
   None, they are all walked out because they didn't like the way the electrician did it.

5. Tweet: 8
   Score: 971.2347709546959
   Content:
   Say goodbye to croissants, people. Delicious croissants. We're stuck with crumpets FOREVER.

Figure A.3

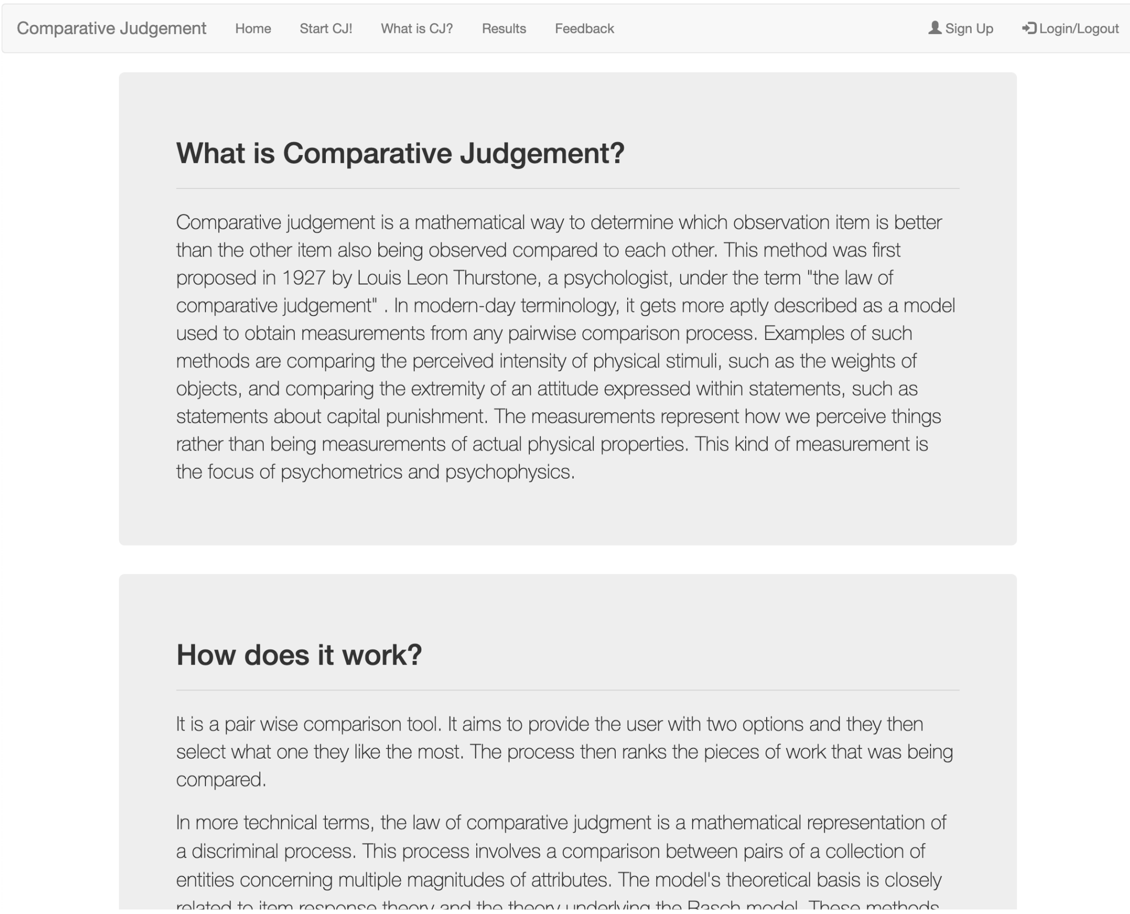

Comparative Judgement
Home
Start CJ!
What is CJ?
Results
Feedback
Sign Up
Login/Logout
What is Comparative Judgement?
Comparative judgement is a mathematical way to determine which observation item is better than the other item also being observed compared to each other. This method was first proposed in 1927 by Louis Leon Thurstone, a psychologist, under the term "the law of comparative judgement" . In modern-day terminology, it gets more aptly described as a model used to obtain measurements from any pairwise comparison process. Examples of such methods are comparing the perceived intensity of physical stimuli, such as the weights of objects, and comparing the extremity of an attitude expressed within statements, such as statements about capital punishment. The measurements represent how we perceive things rather than being measurements of actual physical properties. This kind of measurement is the focus of psychometrics and psychophysics.
How does it work?
It is a pair wise comparison tool. It aims to provide the user with two options and they then select what one they like the most. The process then ranks the pieces of work that was being compared.
In more technical terms, the law of comparative judgment is a mathematical representation of a discriminal process. This process involves a comparison between pairs of a collection of entities concerning multiple magnitudes of attributes. The model's theoretical basis is closely

Figure A.4

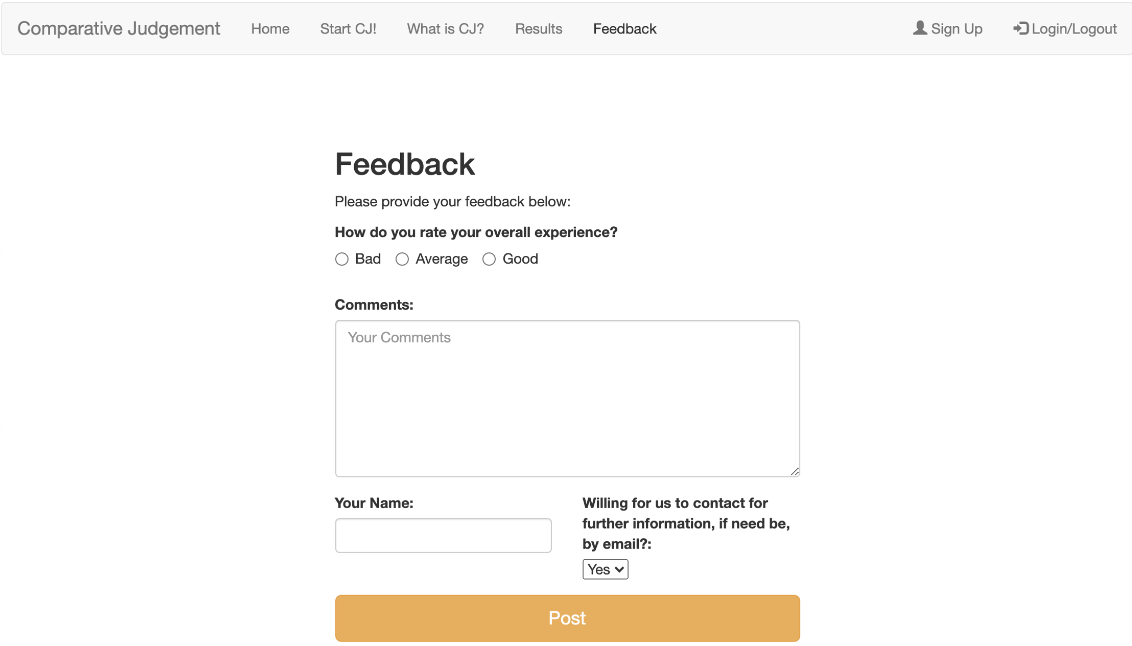


Figure A.5

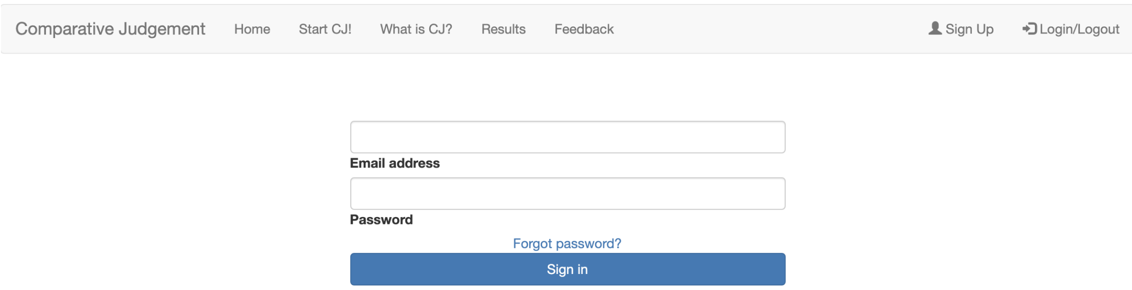


Figure A.6

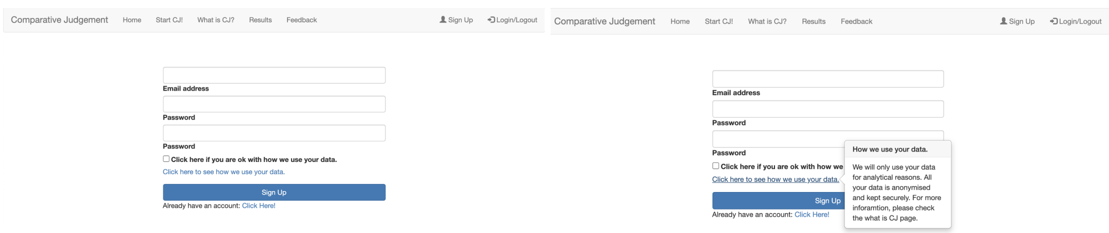


Figure A.7

Appendix B

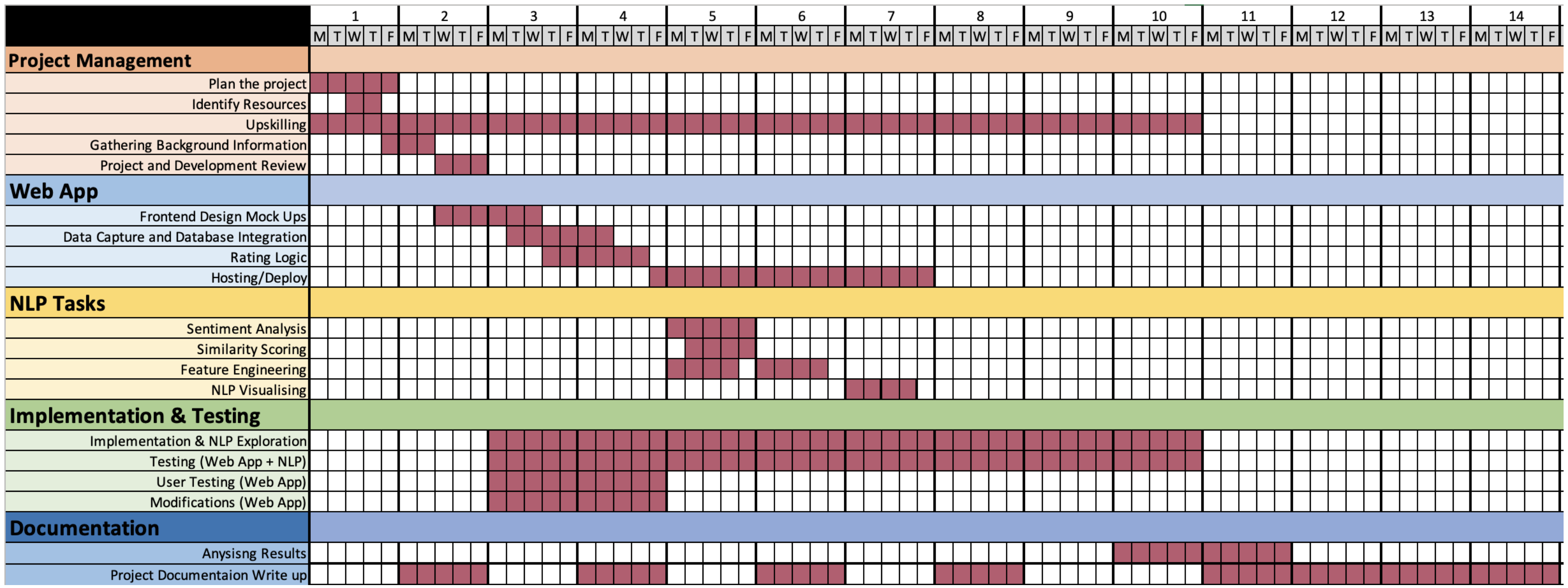
Risks

*S= Severity, L = Likelihood, D= Detection

Risk	S	L	D	RPN	Mitigation
The application is not user friendly.	6	3	2	36	Through user testing, to gain feedback and review.
Application does not meet expectation of the user.	6	3	3	54	User testing must be carried out and feedback taken to adapt the app.
Application has foundation bugs which effects performance.	9	3	6	162	Making sure app that the app is carrying out the core requirements correctly is essential.
Loss of Data/ Application.	8	3	7	168	To make sure that solution is back up by using services like GitHub and other back-up solutions.
More time needed to complete required tasks.	7	4	6	168	Any additional tasks that are not essentially required will have to get discarded.
Not enough time to learn required libraries to highest level.	4	4	6	96	Make sure that NLP and Flask is learnt well enough to be able to put the main concept together.
Inability to incorporate NLP into the research.	6	6	7	252	Make sure that the ELO and Comparative Judgement rankings are carried out correctly.
Under estimation of the project's complexity.	7	5	3	105	Define the projects scope clearly and learn required skills needed to complete the task.
Unrealistic time estimations.	7	4	1	28	Essential that all times requirements are followed. If falling behind, then escalation to project supervisor is required and time management redone.
Failure to follow the project's planed methodology.	6	3	1	18	Ensure requirements to methodology are clear.

Appendix C

Schedule



Appendix D

Testing

The web application was the part of the implementation that required rigorous testing. The testing was because the web app was the bit that users would be interacting with the study. Therefore, we needed to ensure the app was to a high standard not to detract away from the users' experience and solely focus on the application purpose, which is to select which tweet they think is funnier.

We conducted multiple in-house testing using an internal server's localhost to ensure that the app was suitable. Additionally, we allowed a small number of users to test out the application. Once we were happy with the feedback, the application's data got reset and published to potential users.

Appendix E

Implementation of a Web App

```python
from flask import Flask, render_template, request, url_for, session, redirect, flash,
     Markup
from flask_cors import CORS
from models import *
from logic import *

import pyrebase
import os
import sys
import logging


app = Flask(__name__)

app.logger.addHandler(logging.StreamHandler(sys.stdout))
app.logger.setLevel(logging.ERROR)

CORS(app)
app.secret_key = "lets_judge"

# Home form load
@app.route('/', methods=['GET','POST'])
def index():
    return render_template('index.html')


# CJ compare form load
@app.route('/compare/', methods=['GET','POST'])
def compare():
    if request.method == 'GET':
        try:
```

```
            if "user" in session:
                round_number       = get_round_num(session['user'])
                percent            = int(round(((round_number - 1) / 5) * 100, 0))
                total_combinations = get_total_combinations(session['user'])
                if round_number != total_combinations:
                    combo_id       = get_combinations(round_number,session['user'])
                    tweet1_content = get_tweet_content(combo_id['tweet_1'])
                    tweet2_content = get_tweet_content(combo_id['tweet_2'])

                    tweet1_content = Markup(tweet1_content.replace('_b', '<br><br>'))
                    tweet2_content = Markup(tweet2_content.replace('_b', '<br><br>'))

                    tweet1, tweet2, tweet1_id, tweet2_id = tweet1_content, tweet2_content, combo_id['tweet_1'], combo_id['tweet_2']
                else:
                    msg = "You have complaed all the comparisons, please provide feedback on your experience."
                    flash(msg, 'info')
                    return redirect(url_for('feedback'))
            else:
                return redirect(url_for('signup'))
        except:
            return redirect(url_for('logout'))


    if request.method == 'POST':
        radio_1       = request.form.get('radio')
        justification = request.form.get('content')

        if radio_1 == None:
            message = "You have missed some required information. Please try again"
            flash(message, "info")
            return redirect(url_for('compare'))
        else:
            round_number = get_round_num(session['user'])
            percent = round_number / 5
            update_result(round_number,radio_1,session['user'])
            record_justification(round_number,session['user'],justification)
            update_round_number(session['user'])
            update_cj_score()

            return redirect(url_for('compare'))

    return render_template('compare.html', tweet1 = tweet1, tweet2 = tweet2,
                            tweet1_id = tweet1_id, tweet2_id = tweet2_id,
                            percent = int(percent),
                            tweet_count = round_number)
```

```


# CJ Explination form load.
@app.route('/explination/')
def explination():
    return render_template('explination.html')


# CJ Results form load.
@app.route('/results/', methods=['GET','POST'])
def results():
    if request.method == 'GET':
        rank, content = display_ranking()

        elo_rank, elo_content = display_elo_ranking()



    if request.method == 'POST':
        pass

    return render_template('results.html', rank=rank, content=content,
                            elo_rank=elo_rank, elo_content=elo_content)


# Feedback form load
@app.route('/feedback/', methods=['GET','POST'])
def feedback():
    if request.method == 'GET':
        if "user" in session:
            return render_template('feedback.html')
        else:
            return redirect(url_for('login'))

    if request.method == 'POST':
        name     = request.form.get('name')
        contact  = request.form.get('contact')
        feedback = request.form.get('comments')
        rating   = request.form.get('experience')

        create_feedback(name, feedback, rating, session, contact)
        msg = "thank you for the feedback!"
        flash(msg, 'info')
        return redirect(url_for('index'))


# Loging form load
```

```
@app.route('/login/', methods=['GET','POST'])
def login():
    if request.method == 'GET':
        try:
            if "user" in session:
                return redirect(url_for('logout'))
            else:
                return render_template('login.html')
        except:
            msg = "An issue happened. Please try again."
            flash("You have been signed up successfully.", "info")
            return redirect('index')

    if request.method == 'POST':
        try:
            email    = request.form.get('email')
            password = request.form.get('password')
            user     = login_user(email,password)

            if user == None:
                msg = "This email address or password mightbe wrong, please try again
                    . Additionally, You might need to sign up instead."
                flash(msg, 'info')
                return redirect(url_for('login'))
            else:
                session['user']  = user
                session['email'] = email
                flash("You have been logged in successfully.", "info")
                return redirect(url_for('index'))
        except:
            flash("Email address does not exist, please sign up.", "info")
            return redirect(url_for('signup'))


# Signup form load
@app.route('/signup/', methods=['GET','POST'])
def signup():
    if request.method == 'GET':
        return render_template('signup.html')

    if request.method == 'POST':
        email    = request.form.get('email')
        password = request.form.get('password')
        password_check = request.form.get('password_check')
        data_confirm   = request.form.get('confirm')

        print(data_confirm)
```

```

        if data_confirm == 'on':
            if password == password_check:
                success, user_id = signup_user(email,password)
                session['user']  = user_id
                session['email'] = email

                if success == True:
                    flash("You have been signed up successfully.", "info")
                    return redirect(url_for('index'))
                else:
                    flash("Email address already exists, please try logging in
                        instead.", "info")
                    return redirect(url_for('signup'))
            else:
                flash("Invalid email and/or passwords do not match.", "info")
                return redirect(url_for('signup'))
        else:
            flash("Please confirm you are happy with how we use your data.", "info")
            return redirect(url_for('signup'))


# Password reset form load
@app.route('/reset_password/', methods=['GET','POST'])
def reset_password():
    if request.method == 'GET':
        return render_template('forgotten_password.html')

    if request.method == 'POST':
        auth  = init_auth()
        email = request.form.get('email')

        print(email)
        auth.send_password_reset_email(email)

        return redirect(url_for('login'))


# Log out form load
@app.route('/logout/')
def logout():
    if "user" in session:
        user    = session["user"]
        message = "You have been logged out succesfully"
        flash(message, "info")

    session.pop("user", None)
```

```

    return redirect(url_for("index"))


if __name__ == '__main__':
    app.run(debug=True)
```

Listing E.1: The implemented code for handling the main control of the web app.

```
#import sqlite3 as sql
from os        import path, remove
from itertools import combinations as combs

from sklearn.utils import shuffle
from flask         import sessions, Markup

import operator
import random
import pytz
from datetime import datetime, date

import pandas as pd
import numpy  as np

from models import *
import pyrebase


def create_feedback(name,feedback, user_rating, session, contact):
    db = init_db()

    info = {
        'email': session['email'],
        'name': name,
        'user_rating': user_rating,
        'feedback': feedback,
        'contact': contact,
        'user_id': session['user']
    }

    db.child("user_feedback").child(session['user']).update(info)


############################ Firebase Connections
    ######################################
def store_feedback_cloud(textfile_name, session):
    storage        = init_storage()
    filename       = textfile_name
```

```python
    cloud_filename = "feedback/user_"+str(session["user"])

    storage.child(cloud_filename).put(filename)


def store_user_docs(textfile_name, session):
    storage        = init_storage()

    filename       = textfile_name
    cloud_filename = "feedback/user_"+str(session["user"])

    storage.child(cloud_filename).put(filename)


def get_user_storage_docs():
    storage        = init_storage()

    stored_doc = storage.child("doc name.txt").download("","server name.txt")

    return stored_doc

def login_user(id, password):
    """
    Connecting the web app to the firebase authenication to return a user ID.

    Args:
        id ([str]): the users email address to be checked for auth.
        password ([str]): the users password to conform the auth.

    Returns:
        token [str]: this contains the returned local id for the auth.
    """
    auth = init_auth()

    try:
        user  = auth.sign_in_with_email_and_password(id,password)
        token = user['localId']

        return token
    except:
        print("invalid user or password. Please try again")


def signup_user(id,password):
    auth = init_auth()
    db = init_db()

```

```
    try:
        user = auth.create_user_with_email_and_password(id,password)
        init_cj_round_number(user['localId'])

        auth.send_email_verification(user['idToken'])

        tweet_id = [i for i in range(1,11)]
        id_combs = list(combs(tweet_id, 2))
        random.shuffle(id_combs)

        used_nums = []
        new_pairs = []

        for each_pair in id_combs:
            if each_pair[0] not in used_nums:
                if each_pair[1] not in used_nums:
                    used_nums.append(each_pair[0])
                    used_nums.append(each_pair[1])
                    new_pairs.append(each_pair)

        combs_df = pd.DataFrame()

        r = 1
        for each_combination in new_pairs:
            #split = each_combination.split(' , ')
            combs_df = combs_df.append({
                "combination_id": str(r),
                "tweet_1": str(each_combination[0]),
                "tweet_2": str(each_combination[1])
            }, ignore_index=True)

            r += 1

        combination_df = combs_df.reset_index(drop=True)

        for i in combination_df.index:
            dict_data = combination_df.loc[i].to_dict()
            tweet_id = i+1
            db.child("combinations").child(user['localId']).child(tweet_id).set(
                dict_data)

        return True, user['localId']
    except:
        return False, None


def init_cj_round_number(user_id):
```

```
    db = init_db()
    db.child("cj_position").child(user_id).update({'comparison_no': 1})


########## Firebase Content Handling ########
def update_round_number(user_id):
    db = init_db()
    current_round = get_round_num(user_id)

    db.child("cj_position").child(user_id).update({'comparison_no': current_round +
        1})


def get_round_num(user_id):
    db = init_db()
    round_info = db.child("cj_position").child(user_id).get()

    for cj_position in round_info.each():
        current_num = cj_position.val()

    return current_num


def record_justification(round_number,user_id,justification):
    db = init_db()
    db.child("combinations").child(user_id).child(round_number).update({'
        justification': justification})


def get_time_stamp():
    today = date.today()
    d1 = today.strftime("%d/%m/%Y")

    london_tz = pytz.timezone('Europe/London')
    now = datetime.now(london_tz)
    time = now.strftime("%H:%M:%S")
    time_stamp = f"{time} {d1}"

    return time_stamp


def update_result(round_number,winner_id,user_id):
    db = init_db()
    combination = get_combinations(round_number,user_id)

    time_stamp = get_time_stamp()

```

```
    if winner_id == combination['tweet_1']:
        loser_id = int(combination['tweet_2'])
    else:
        loser_id = int(combination['tweet_1'])

    tweets = db.child("results").child(int(winner_id)).get()
    tweet_dict = {}
    for tweet in tweets.each():
        tweet_dict[tweet.key()] = tweet.val()
    tweet_dict['win'] += 1

    other_tweet = db.child("results").child(loser_id).get()
    other_tweet_dict = {}
    for tweet in other_tweet.each():
        other_tweet_dict[tweet.key()] = tweet.val()
    other_tweet_dict['lose'] += 1

    winner_new_score = elo_rating(tweet_dict['elo_score'],other_tweet_dict['elo_score'],1)
    loser_new_score = elo_rating(other_tweet_dict['elo_score'],tweet_dict['elo_score'],0)

    db.child("results").child(winner_id).update({"win": tweet_dict['win'], "elo_score": winner_new_score})
    db.child("results").child(loser_id).update({"lose": other_tweet_dict['lose'], "elo_score": loser_new_score})
    db.child("combinations").child(user_id).child(round_number).update({"winner": winner_id, "loser": loser_id, 'time_stamp': str(time_stamp)})


def predict_elo_result(A, B):
    p_a_wins = 1 / (1 + (10**((B-A)/400)))

    return p_a_wins


def elo_rating(A, B, score):
    expected_score = predict_elo_result(A, B)
    rating = A

    new_score = rating + (32 * (score - expected_score))

    return new_score


def get_combinations(round_number,user_id):
    db = init_db()
```

```
    combination = db.child("combinations").child(user_id).child(round_number).get()
    combo_dict = {}
    for combo in combination.each():
        combo_dict[combo.key()] = combo.val()

    return combo_dict


def get_tweet_content(id):
    db = init_db()
    tweets = db.child("results").child(id).get()
    dict = {}
    for tweet in tweets.each():
        dict[tweet.key()] = tweet.val()

    return dict['content']


def get_total_combinations(user_id):
    db = init_db()
    rounds_no = db.child('combinations').child(user_id).get()

    count = 0
    for each_combo in rounds_no.each():
        count += 1

    return count


def calculate_score(id):
    db = init_db()
    tweets_scores = db.child("results").child(id).get()
    dict = {}
    for tweet in tweets_scores.each():
        dict[tweet.key()] = tweet.val()

    result = dict['win'] - dict['lose']

    return result


def display_ranking():
    db = init_db()

    order_dict = {}
    for i in range(1,11):
        tweet_details = db.child("results").child(i).get()
```

```
        dict = {}
        for tweet in tweet_details.each():
            dict[tweet.key()] = tweet.val()

        order_dict[i] = dict

    new_order = {}
    for i in range(1,11):
        new_order[i] = order_dict[i]['score']

    new_order = sorted(new_order.items(), key=lambda kv: kv[1], reverse=True)

    final_order = {}
    for i in range(len(new_order)):
        final_order[new_order[i][0]] = new_order[i][1]

    final_order_content = {}
    for key in final_order:
        text = get_tweet_content(key)
        text = Markup(text.replace('_b', '<br>'))
        final_order_content[key] = text

    return final_order, final_order_content


def display_elo_ranking():
    db = init_db()

    order_dict = {}
    for i in range(1,11):
        tweet_details = db.child("results").child(i).get()
        dict = {}
        for tweet in tweet_details.each():
            dict[tweet.key()] = tweet.val()

        order_dict[i] = dict

    new_order = {}
    for i in range(1,11):
        new_order[i] = order_dict[i]['elo_score']

    new_order = sorted(new_order.items(), key=lambda kv: kv[1], reverse=True)

    final_order = {}
    for i in range(len(new_order)):
        final_order[new_order[i][0]] = new_order[i][1]

```

```
    final_order_content = {}
    for key in final_order:
        text = get_tweet_content(key)
        text = Markup(text.replace('_b', '<br>'))
        final_order_content[key] = text

    return final_order, final_order_content


def update_cj_score():
    db = init_db()
    for i in range(1,11):
        score = calculate_score(i)
        db.child("results").child(i).update({'score': score})
```

Listing E.2: The implemented code for handling the main web app logic.

```
import pyrebase


def connect_to_firebase():
    firebase_config = {
        "apiKey": "Removed",
        "authDomain": "Removed",
        "databaseURL": "Removed",
        "projectId": "Removed",
        "storageBucket": "Removed",
        "messagingSenderId": "Removed",
        "appId": "Removed",
        "measurementId": "Removed"
    }

    firebase = pyrebase.initialize_app(firebase_config)

    return firebase


def init_db():
    firebase = connect_to_firebase()
    firebase_db = firebase.database()

    return firebase_db


def init_auth():
    firebase = connect_to_firebase()
    firebase_auth = firebase.auth()

```

```
    return firebase_auth


def init_storage():
    firebase = connect_to_firebase()
    firebase_storage = firebase.storage()

    return firebase_storage
```

Listing E.3: The implemented code for handling Firebase Connections.

Appendix F

NLP Jupyter Notebook

```
import spacy
import pandas as pd
from itertools import combinations as combs
from spacy.matcher import Matcher
from spacy import displacy

import nltk

import numpy as np

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, Dropout, SpatialDropout1D
from tensorflow.keras.layers import LSTM
from tensorflow.keras.models import load_model

from collections import Counter
import text_normalizer as tn
import model_evaluation_utils as meu

from keras.preprocessing import sequence
from sklearn.preprocessing import LabelEncoder

# %% [markdown]
# ## Data Pipeline

# %%
nlp = spacy.load('en_core_web_sm')

doc1  = nlp(u'An Englishman, a Scotsman and an Irishman walk into a bar. The
    Englishman wanted to go so they all had to leave. #Brexitjokes')
```

```
doc2  = nlp(u'Why do we need any colour passport? We should just be able to shout, British! Less of your nonsense! and stroll straight through.')
doc3  = nlp(u'Q: With Britain leaving the EU how much space was created? A: Exactly 1 GB')
doc4  = nlp(u'VOTERS: we want to give a boat a ridiculous name UK: no VOTERS: we want to break up the EU and trash the world economy UK: fine')
doc5  = nlp(u'#BrexitJokes How did the Brexit chicken cross the road? I never said there was a road. Or a chicken.')
doc6  = nlp(u'After #brexit, when rapper 50 cent performs in GBR he\'ll appear as 10.00 pounds. #brexitjokes')
doc7  = nlp(u'I long for the simpler days when #Brexit was just a term for leaving brunch early.')
doc8  = nlp(u'Say goodbye to croissants, people. Delicious croissants. We\'re stuck with crumpets FOREVER.')
doc9  = nlp(u'Hello, I am from Britain, you know, the one that got tricked by a bus')
doc10 = nlp(u'How many Brexiteers does it take to change a light bulb? None, they are all walked out because they didn\'t like the way the electrician did it.')

docs = [
    doc1,
    doc2,
    doc3,
    doc4,
    doc5,
    doc6,
    doc7,
    doc8,
    doc9,
    doc10]


# %%
#Creating DF for LSTM
tweets = np.array([
    ["An Englishman, a Scotsman and an Irishman walk into a bar. The Englishman wanted to go so they all had to leave. #Brexitjokes"],
    ["Why do we need any colour passport? We should just be able to shout, British! Less of your nonsense! and stroll straight through."],
    ["Q: With Britain leaving the EU how much space was created? A: Exactly 1GB"],
    ["VOTERS: we want to give a boat a ridiculous name UK: no VOTERS: we want to break up the EU and trash the world economy UK: fine"],
    ["#BrexitJokes How did the Brexit chicken cross the road? I never said there was a road. Or a chicken."],
    ["After #brexit, when rapper 50 cent performs in GBR he'll appear as 10.00 pounds. #brexitjokes"],
    ["I long for the simpler days when #Brexit was just a term for leaving brunch early."],
```

```
    ["Say goodbye to croissants, people. Delicious croissants. We're stuck with
        crumpets FOREVER."],
    ["Hello, I am from Britain, you know, the one that got tricked by a bus"],
    ["How many Brexiteers does it take to change a light bulb? None, they are all
        walked out because they didn\'t like the way the electrician did it."]])

tweet_df = pd.DataFrame(tweets, columns=['tweet_content'])
tweet_df.head()

# Removing Stop words
stop_words = nltk.corpus.stopwords.words('english')
stop_words.remove('no')
stop_words.remove('but')
stop_words.remove('not')

# %% [markdown]
# ___
# %% [markdown]
# ## Part of Speach Tagging

# %%
tweet_no = 1
for doc in docs:
    print(f'Tweet: {tweet_no}')
    for token in doc:
        print(f'{token.text:{10}} - {token.pos_:{10}} - {token.tag_:{10}} - {spacy.
            explain(token.tag_)}')
    tweet_no += 1



# %%
# POS Counts
tweet_no = 1
for doc in docs:
    print(f'Tweet: {tweet_no}')
    POS_counts = doc.count_by(spacy.attrs.POS)
    for k,v in sorted(POS_counts.items()):
        print(f'{k}: {doc.vocab[k].text:{5}} {v}')

    print('\n')
    tweet_no += 1


# %%
# Visualising POS
options = {
```

```
    'distance':95,
    'compact':'True'
}

for doc in docs:
    spans = list(doc.sents)
    displacy.render(spans,style='dep',jupyter=True, options = options)

# %% [markdown]
# ___
# %% [markdown]
# ## Named Entity Recognition

# %%
def show_ents(doc):
    no_ents = 0
    if doc.ents:
        for ent in doc.ents:
            print(f'{ent.text} - {ent.label_} - {spacy.explain(ent.label_)}')
            no_ents += 1
        print(f'Total number of entities: {no_ents}')
    else:
        print('No entites found')


# %%
tweet_no = 1
for doc in docs:
    print(f'Tweet: {tweet_no}')
    show_ents(doc)
    print('\n')
    tweet_no += 1


# %%
tweet_no = 1
for doc in docs:
    print(f'Tweet: {tweet_no}')
    displacy.render(doc, style="ent")
    tweet_no += 1

# %% [markdown]
# ___
# %% [markdown]
# ## Feature Extraction

# %%
```

```
tweet_df.isnull().sum() #delete at a later date


# %%
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer,
    TfidfVectorizer


# %%
tfidf = TfidfVectorizer(min_df=2, max_df=0.5, ngram_range=(1,2))


# %%
doc1  = ('An Englishman, a Scotsman and an Irishman walk into a bar. The Englishman
    wanted to go so they all had to leave. #Brexitjokes')
doc2  = ('Why do we need any colour passport? We should just be able to shout, \"
    British! Less of your nonsense!\" and stroll straight through.')
doc3  = ('Q: With Britain leaving the EU how much space was created? A: Exactly 1GB')
doc4  = ('VOTERS: we want to give a boat a ridiculous name UK: no VOTERS: we want to
    break up the EU and trash the world economy UK: fine')
doc5  = ('#BrexitJokes How did the Brexit chicken cross the road? \"I never said
    there was a road. Or a chicken\".')
doc6  = ('After #brexit, when rapper 50 cent performs in GBR he\'ll appear as 10.00
    pounds. #brexitjokes')
doc7  = ('I long for the simpler days when #Brexit was just a term for leaving brunch
     early.')
doc8  = ('Say goodbye to croissants, people. Delicious croissants. We\'re stuck with
    crumpets FOREVER.')
doc9  = ('Hello, I am from Britain, you know, the one that got tricked by a bus')
doc10 = ('How many Brexiteers does it take to change a light bulb? None, they are all
     walked out because they didn\'t like the way the electrician did it.')

fe_docs = [
    doc1,
    doc2,
    doc3,
    doc4,
    doc5,
    doc6,
    doc7,
    doc8,
    doc9,
    doc10]


# %%
features = tfidf.fit_transform(fe_docs)
```

```python


# %%
fe_df = pd.DataFrame(features.todense(),columns=tfidf.get_feature_names())


# %%
fe_df

# %% [markdown]
# ___
# %% [markdown]
# ## Sentiment Analysis

# %%
# Load pre-trained model
model = load_model('LSTM_model.h5')


# %%
norm_tweets = tn.normalize_corpus(tweet_df['tweet_content'], stopwords=stop_words)
tokenized_tweets  = [tn.tokenizer.tokenize(text) for text in norm_tweets]

# build word to index vocabulary
token_counter = Counter([token for review in tokenized_tweets for token in review])
vocab_map     = {item[0]: index+1 for index, item in enumerate(dict(token_counter).
    items())}
max_index     = np.max(list(vocab_map.values()))

vocab_map['PAD_INDEX']       = 0
vocab_map['NOT_FOUND_INDEX'] = max_index+1

vocab_size    = len(vocab_map)

# view vocabulary size and part of the vocabulary map
print('Vocabulary Size:', vocab_size)
print('Sample slice of vocabulary map:', dict(list(vocab_map.items())))

#get max length of train corpus and initialize label encoder
le          = LabelEncoder()
num_classes = 2 # positive -> 1, negative -> 0
max_len     = np.max([len(review) for review in tokenized_tweets])


## Test reviews data corpus
# Convert tokenized text reviews to numeric vectors
```

```
tweet_ready = [[vocab_map[token] for token in tokenized_review] for tokenized_review in tokenized_tweets]
tweet_ready = sequence.pad_sequences(tweet_ready, maxlen=max_len) # pad


# view vector shapes
print('Max length of tweet review vectors:', max_len)
print('Tweet vectors shape:', tweet_ready.shape)


# %%
my_pred_test = model.predict(tweet_ready)


# %%
pred_score = [1 if p > 0.5 else 0 for p in my_pred_test]
pred_sent = ['Positive' if p > 0.5 else 'Negative' for p in my_pred_test]


# %%
for i in range(len(pred_score)):
    print(f'Tweet {i+1}:\nActual Score: {my_pred_test[i]} - Score: {pred_score[i]} - Sentiment: {pred_sent[i]}')

# %% [markdown]
# ___
# %% [markdown]
# ## Tweet Similarity Scoring
# %% [markdown]
# ### Document Similarity

# %%
tweet_id = [i for i in range(1,11)]
id_combs = list(combs(tweet_id, 2))


# %%
doc_df = pd.DataFrame()

for each_pair in id_combs:
    doc_similarity = docs[each_pair[0]-1].similarity(docs[each_pair[1]-1])
    doc_results = {
        'tweet1': int(each_pair[0]),
        'tweet2': int(each_pair[1]),
        'similarity': doc_similarity,
        'text 1': docs[each_pair[0]-1],
        'text 2': docs[each_pair[1]-1]
```

```
    }

    doc_df = doc_df.append(doc_results, ignore_index=True)


# %%
doc_df['tweet1'] = doc_df['tweet1'].astype(int)
doc_df['tweet2'] = doc_df['tweet2'].astype(int)
doc_df.head()


# %%
doc_df_ordered = doc_df.sort_values(by=['similarity'], ascending=False)
doc_df_ordered.head(10)


# %%
doc_df_ordered.tail(10)

# %% [markdown]
# ### Term Similarity

# %%
spans = {}


# %%
for j,doc in enumerate(docs):
    named_entity_span = [doc[i].text for i in range(len(doc)) if doc[i].ent_type !=
        0]
    print(named_entity_span)
    named_entity_span = ' '.join(named_entity_span)
    named_entity_span = nlp(named_entity_span)
    spans.update({j:named_entity_span})


# %%
df = pd.DataFrame()

for each_pair in id_combs:
    similarity = spans[each_pair[0]-1].similarity(spans[each_pair[1]-1])
    #print(f'doc{each_pair[0]} is similar to doc{each_pair[1]} by: {similarity}') #Un
        -comment if you want to see individual scores printed.
    results = {
        'tweet1': int(each_pair[0]),
        'tweet2': int(each_pair[1]),
        'similarity': similarity,
```

```
        'tweet1 NE Span': spans[each_pair[0]-1],
        'tweet2 NE Span': spans[each_pair[1]-1]
    }

    df = df.append(results, ignore_index=True)


# %%
# Chaning Data Types
df['tweet1'] = df['tweet1'].astype(int)
df['tweet2'] = df['tweet2'].astype(int)


# %%
# Saving to/loading from CSV
#df = pd.read_csv('similarity_scores_v2.csv') #Uncomment to load.
#df.to_csv('similarity_scores_v2.csv') #Uncomment to resave.


# %%
df_ordered = df.sort_values(by=['similarity'], ascending=False)


# %%
# Display the Top 10 Simialr Combinations
df_ordered.head(10)


# %%
# Display the Bottom 10 Simialr Combinations
df_ordered.tail(10)

# %% [markdown]
# ___
# %% [markdown]
# ## Utterence Pattern Matching

# %%
def dep_pattern(doc):
    for i in range(len(doc)-1):
        if doc[i].dep_ == 'nsubj' and doc[i+1].dep_ == 'aux' and doc[i+2].dep_ == '
            ROOT':
            for tok in doc[i+2].children:
                if tok.dep_ == 'dobj':
                    return True
    else:
        return False
```

```


# %%
for i in docs:
    if dep_pattern(i):
        print(f'Found in: {i}')
    else:
        print('Not Found')

# %% [markdown]
# ___
# %% [markdown]
# ## Finding Word Sequence Patterns

# %%
matcher = Matcher(nlp.vocab)
pattern = [{
    'DEP':"nsubj"},
    {"DEP":"aux"},
    {"DEP":"ROOT"}
    ]

matcher.add("NsubjAuxRoot", [pattern])

tweet_no = 1

for doc in docs:
    matches = matcher(doc)
    print(f'Tweet: {tweet_no}')
    for match_id, start, end in matches:
        span = doc[start:end]
        print(f"Span: {span.text}")
        print(f"The position in the doc are: {start} - {end}\n")
    else:
        print("None found.\n")
    tweet_no += 1

# %% [markdown]
# ___
# %% [markdown]
# ## Key Phrases

# %%
def keyphrase(doc):
    for t in doc:
        if t.dep_ == 'probj' and (t.pos_ == 'NOUN' or t.pos_ == "PROPN"):
```

```
            return (' '.join([child.text for child in t.lefts]) + ' ' + t.text).lstrip()
    for t in reversed(doc):
        if t.dep_ == 'nsubj' and (t.pos_ == 'NOUN' or t.pos_ == 'PROPN'):
            return t.text + ' ' + t.head.text
    for t in reversed(doc):
        if t.dep_ == 'dobj' and (t.pos_ == 'NOUN' or t.pos_ == 'PROPN'):
            return t.head.text + ' ' + 'ing' + ' ' + t.text
    return False


# %%
tweet_no = 1
for doc in docs:
    print(keyphrase(doc))
    tweet_no += 1

# %% [markdown]
# ___
```

Listing F.1: The implemented code for the NLP Information Extraction.

Appendix G

NLP POS Tagging Visulaisations

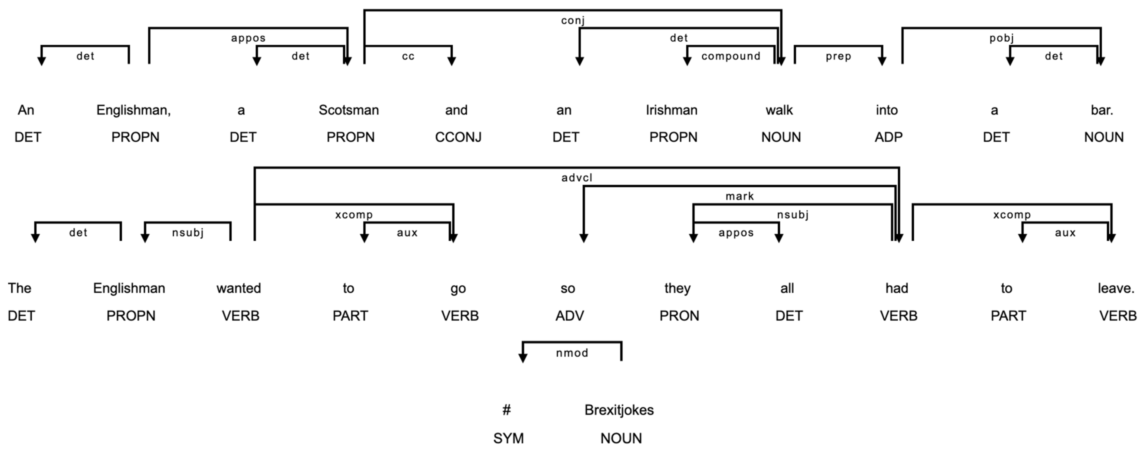


Figure G.1

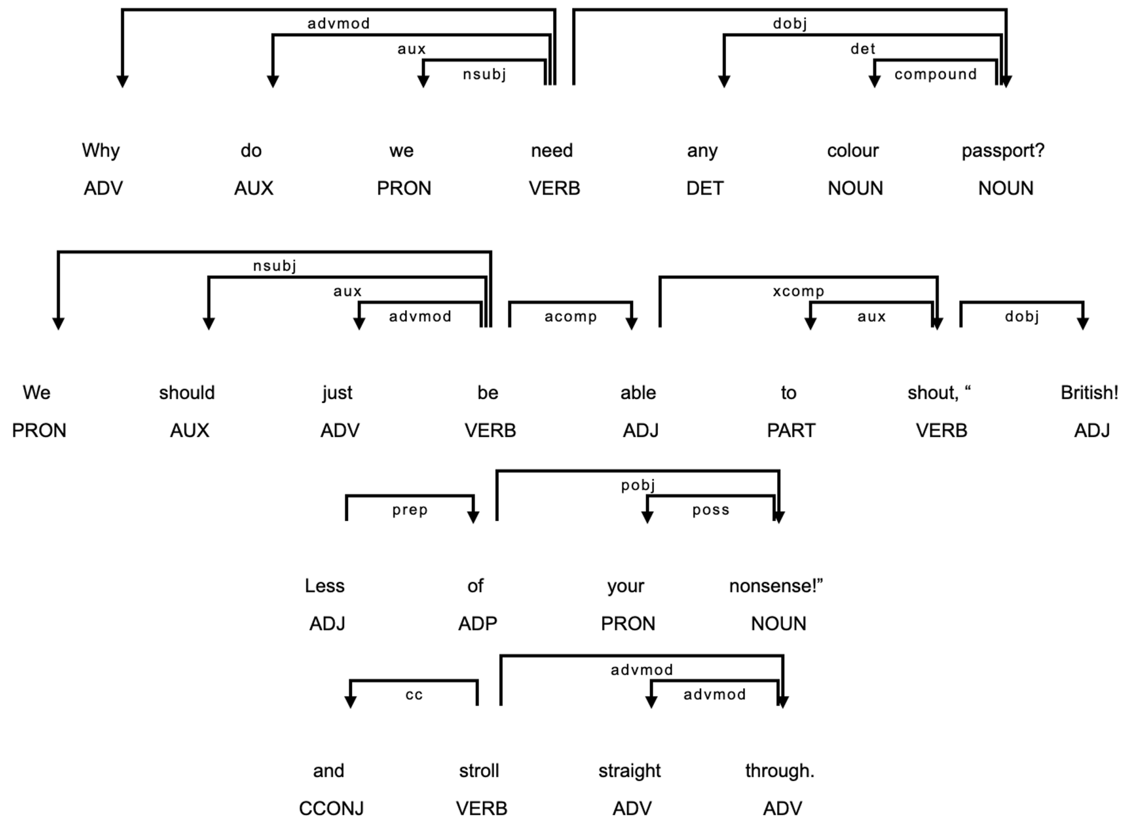


Figure G.2

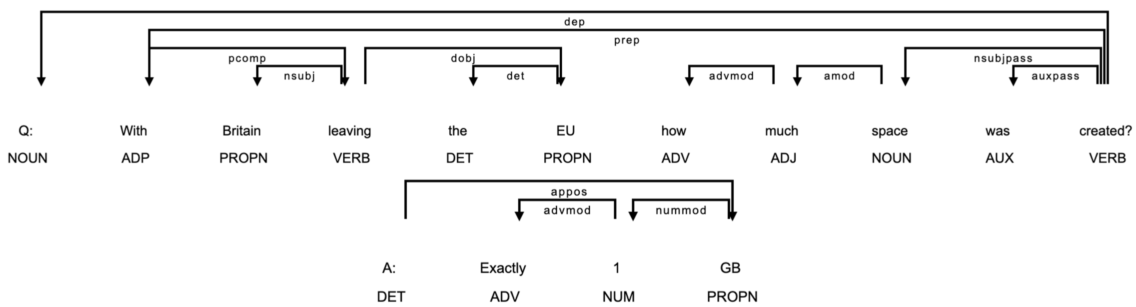


Figure G.3

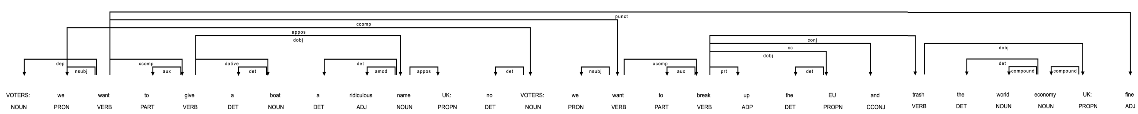


Figure G.4

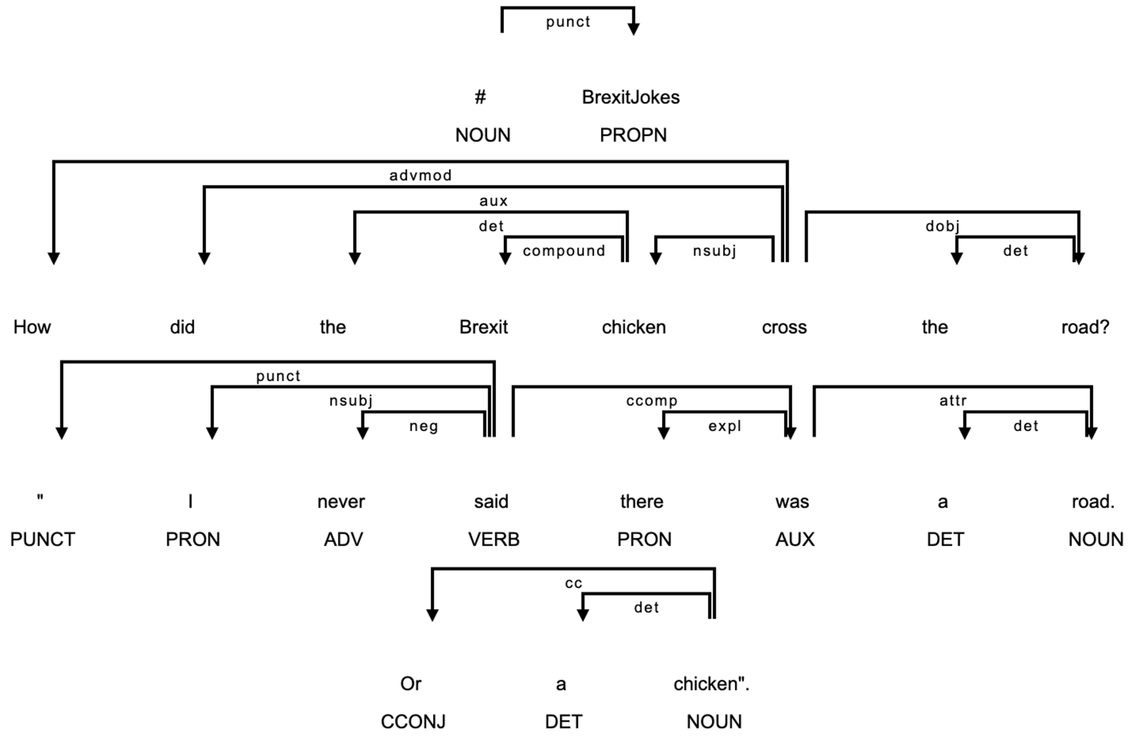


Figure G.5

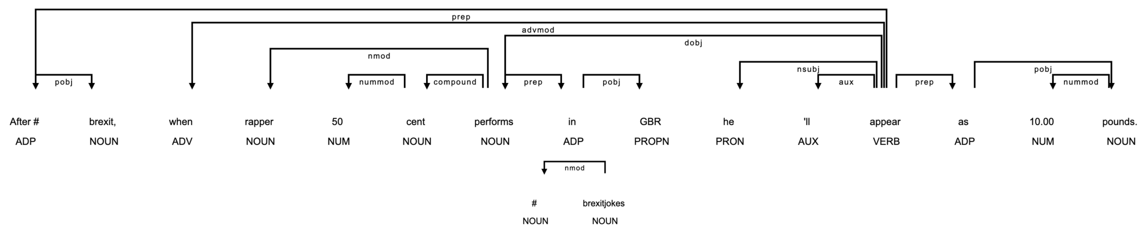


Figure G.6

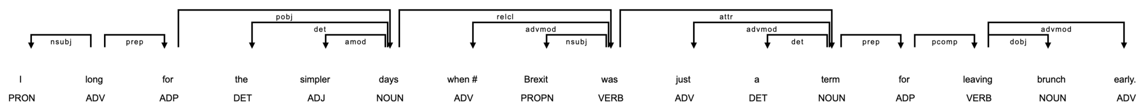


Figure G.7

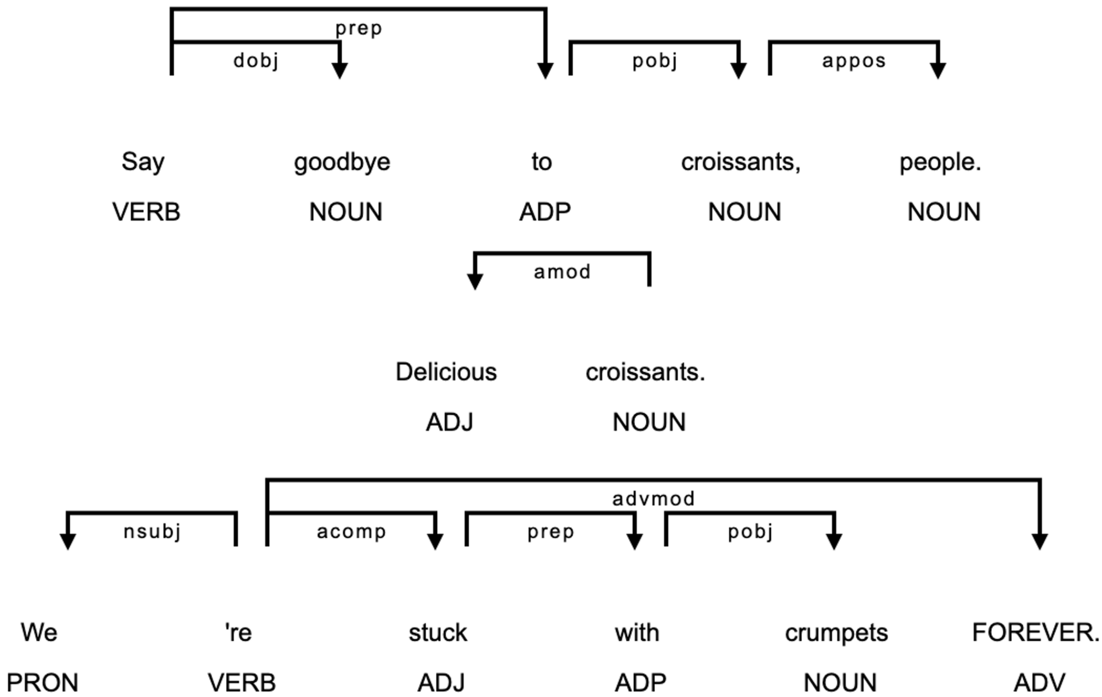


Figure G.8

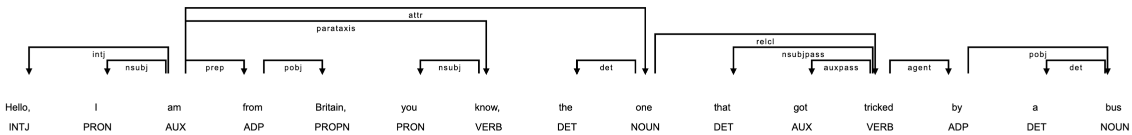


Figure G.9

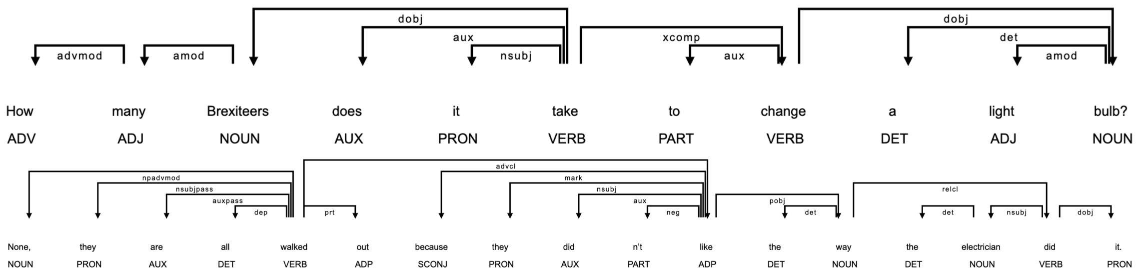


Figure G.10

Appendix H

NLP NER Visualisations

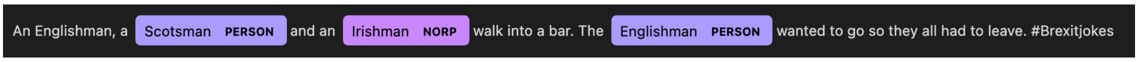


Figure H.1

Figure H.2

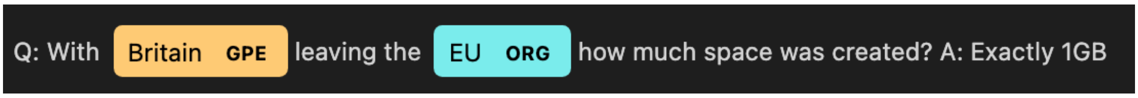


Figure H.3

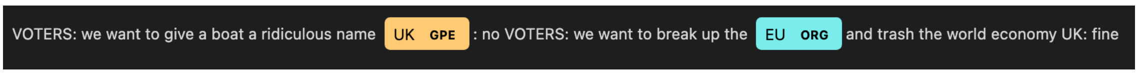


Figure H.4

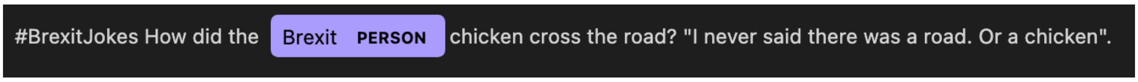


Figure H.5

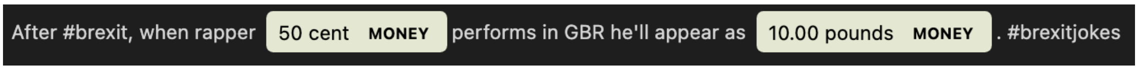


Figure H.6

Figure H.7

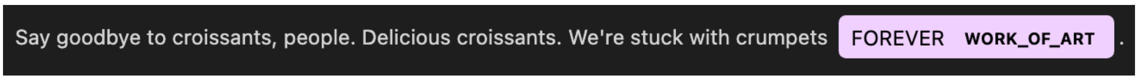


Figure H.8

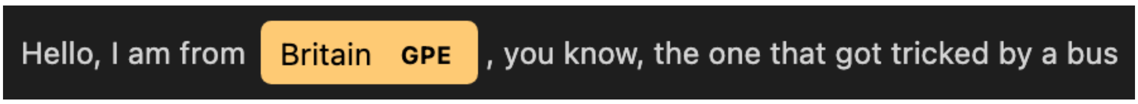


Figure H.9

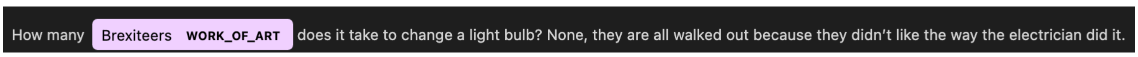


Figure H.10