**ML-POWERED OBJECTIFICATION AND SUBJECTIFICATION OF TEXT**

**By: Wang Hengyue, Mendoza Kieran Majel**

**Domain: Media, literature**

Often, the biases and the judgements of journalists get in the way of the facts and the truth. This project aims to first detect whether a piece of text is subjective or objective, then reframe it to eliminate the prejudices and be more factual and vice versa.

This project consists of two parts: the development of a model that detects the objectivity (and consequently, subjectivity) of the text using a supervised model, and the creation of a model that objectifies or subjectifies the text.

Results have shown that a supervised method of training the model is generally effective except in cases where the text is from news literature.

[**Objectives**](#_heading=h.i58rjzkdliv4) **1**

[**Development Processes and Tools Use**](#_heading=h.7ns3rdj3gokn) **2**

[**Methodology**](#_heading=h.jteplz43p3ei) **2**

[Part 0: Dataset preprocessing](#_heading=h.m8834hlcvwfe) 2

[Part 1: Objectivity detection](#_heading=h.d5mkoybdr7rd) 2

[Part 2: Objectification and Subjectification](#_heading=h.fm6r7ozgqfkd) 3

[Part 3: User Interface](#_heading=h.wjsg0l4o26gs) 3

[**Results and Discussion**](#_heading=h.77kejc1cut14) **3**

[Objectivity detection](#_heading=h.p7tkur61nnoc) 3

[Text Transformation](#_heading=h.5x8pzjiz8xcc) 4

[**Screenshots and Diagrams**](#_heading=h.g0mzyq2vw0yr) **4**

[**Limitations**](#_heading=h.jkgiljgoqcf9) **6**

[**Conclusion and Recommendations**](#_heading=h.eau8iq4swkit) **6**

[**Reflection**](#_heading=h.qhnugmjp6t5n) **6**

[**References**](#_heading=h.swv4hfnjf7ri) **7**

[**Acknowledgements**](#_heading=h.637uatlygyb) **7**

[**Work Distribution Matrix**](#_heading=h.xhz0crwgfxeh) **7**

# Objectives

1. Detection of author stance
2. Detection of subjective or objective text
3. Manipulation of text from being subjective to objective and vice-versa.

# Development Processes and Tools Use

* nltk - natural language processing library
* numpy - mathematical library and intuitive array functions
* pandas - large-scale data analysis and manipulation
* sklearn - machine learning library with useful algorithms
* tensorflow - neural-network centric machine learning library
* tkinter - simple user interface library, also base Python GUI
* pickle - serialisation and de-serialising objects

# Methodology

The dataset used for Part 1 of the project is 2004-2005 movie review data from Rotten Tomatoes, as well as iMDb plot summary data[1]. This is useful, since the former is well-known to produce more polarised, subjective sentiments and the latter generates factual, objective plot summaries. Both ends of the spectrum would prove useful for training our project.

## Part 0: Dataset preprocessing

The data above is indeed useful, but it has some formatting problems. For one, it uses custom file extensions, which means that they have to first be converted to the appropriate file format (.txt) before the data can be loaded.

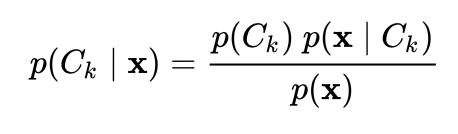
Additionally, as standard NLP practice, we also perform some operations on the text to ensure that the data is of consistent format. Our main method is to *lemmatize* the text, or to group together variants of the same root word. This means that we remove the pronouns (object’s becomes object) and reduce the word to its root form (hypnotizing becomes hypnotism).

## Part 1: Objectivity detection

This component of the project is a classification problem.

*Supervised learning* was used to train the model to detect the objectivity of the text. I used two approaches, the 1st is to detect text sentiment, and assume that if the model is unable to clearly distinguish whether the sentiment is positive or negative, that the text is positive. The other approach is to directly detect the objectivity of the text.

To train the model, the Naiïve Bayes algorithm was used[2]. To give some background, this classifier is one that can output a probability distribution of the possible outcomes. It is based on the Bayes’ theorem (equation below), and assumes that the features are independent of each other (which is true in this case since there are only two to begin with).



The equation above means that the posterior is equal to the product of the prior and likelihood, divided by the evidence. Evidence refers to the number of similar observations divided by the total number, the likelihood refers to the probability of finding that observation in the dataset, and the prior is the number of similar observations of the same class divided by the total number of that class type.

## Part 2: Objectification and Subjectification

We first attempt a naiïve way of making the text more objective or subjective. How it goes is that it simply finds the most objective/subjective synonym for the word (assuming it isn’t a stopword) and then replaces the original word. Obviously, this is prone to losing the gist of the content in the text and makes it grammatically incorrect. But this will serve as the baseline for our model for text transformation.

Afterwards, we attempt to modify *Structured content preservation for unsupervised text style transfer*[6] to make the text more subjective or objective. The outcome is slightly better than the naiïve implementation but still requires more work. We think that with more data the model would be more robust and able to transform sentences better.

## Part 3: User Interface

(note that in some parts of the report, the UI is not the current version, as we were making regular UI improvements then)

The theme of the user interface is one that comes across as robotic and minimalist and coming from the 1980s-90s, while also being functional. This emphasizes how, ironically, a logic-based and highly mathematical machine is attempting to figure out the dynamics of linguistics.

The tkinter -based UI was used because of easy integration with pickle which enables for a pre-trained model to be saved and then loaded on compilation, saving computational time and resources. It is also simple to learn and is highly intuitive to use. It is somewhat lacking in aesthetic customization options, though.

The simple Python application has the two main functionalities inside it, the objectivity detection and the text transformation ability. To attempt to make it easier to import large amounts of text, importing of .txt files is also enabled and fed into the text field (Figure 5).

The objectivity detection button outputs a probability of the text being subjective and objective, and the two transformation buttons (make text more subjective and make text more objective) attempts to make the text more of a certain voice, as the name suggests. The output is stored in a text file in the same file directory.

# Results and Discussion

## Objectivity detection

Using the first approach, we see that the accuracy of the model is acceptable, with an accuracy score of 75%. This means that the classifier is able to correctly predict the sentiment in the sentence ¾ of the time. However, the second approach (the direct objectivity detection) provides more accurate results at 90%. The confusion matrix can be found in Figure 1a and 1b.

The distinction between objective and subjective statements is clearly depicted in Figure 2a and 2b, which shows a subjective statement and objective statement on the same situation, respectively. Note how the two examples are highly polarised, and likely unrealistic, to show this difference.

Referring to Figure 3a and 3b, we can see that certain words have a large influence on the overall subjectivity or objectivity of the text. In this case, the addition of the word ‘absolutely’ adds a large amount of subjectivity in the sentence, with an increase of probability of 7%.

## Text Transformation

The Naiïve approach, as expected, produces sub-par results, as shown by the poor substitution of words and the incoherence of the sentence in Figure 4.

# Screenshots and Diagrams

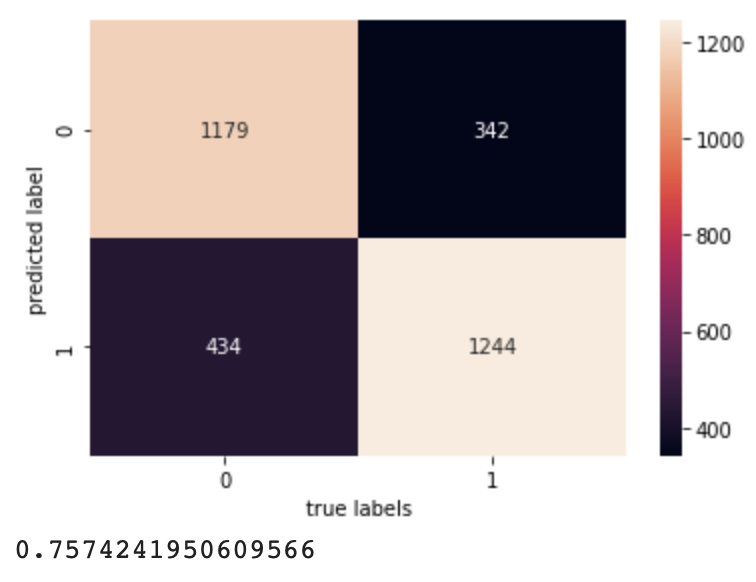
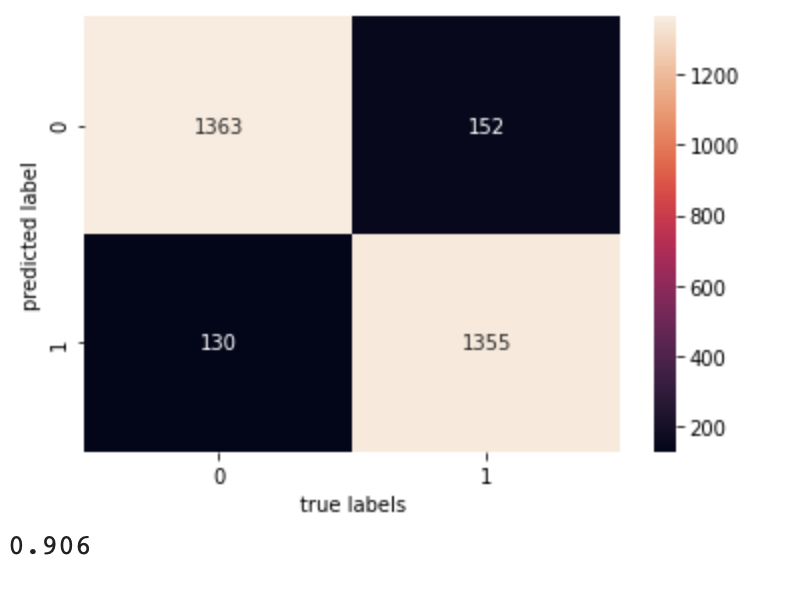
 

Figure 1a (left) and Figure 1b (right) - confusion matrix from Approach 1 and Approach 2, respectively.

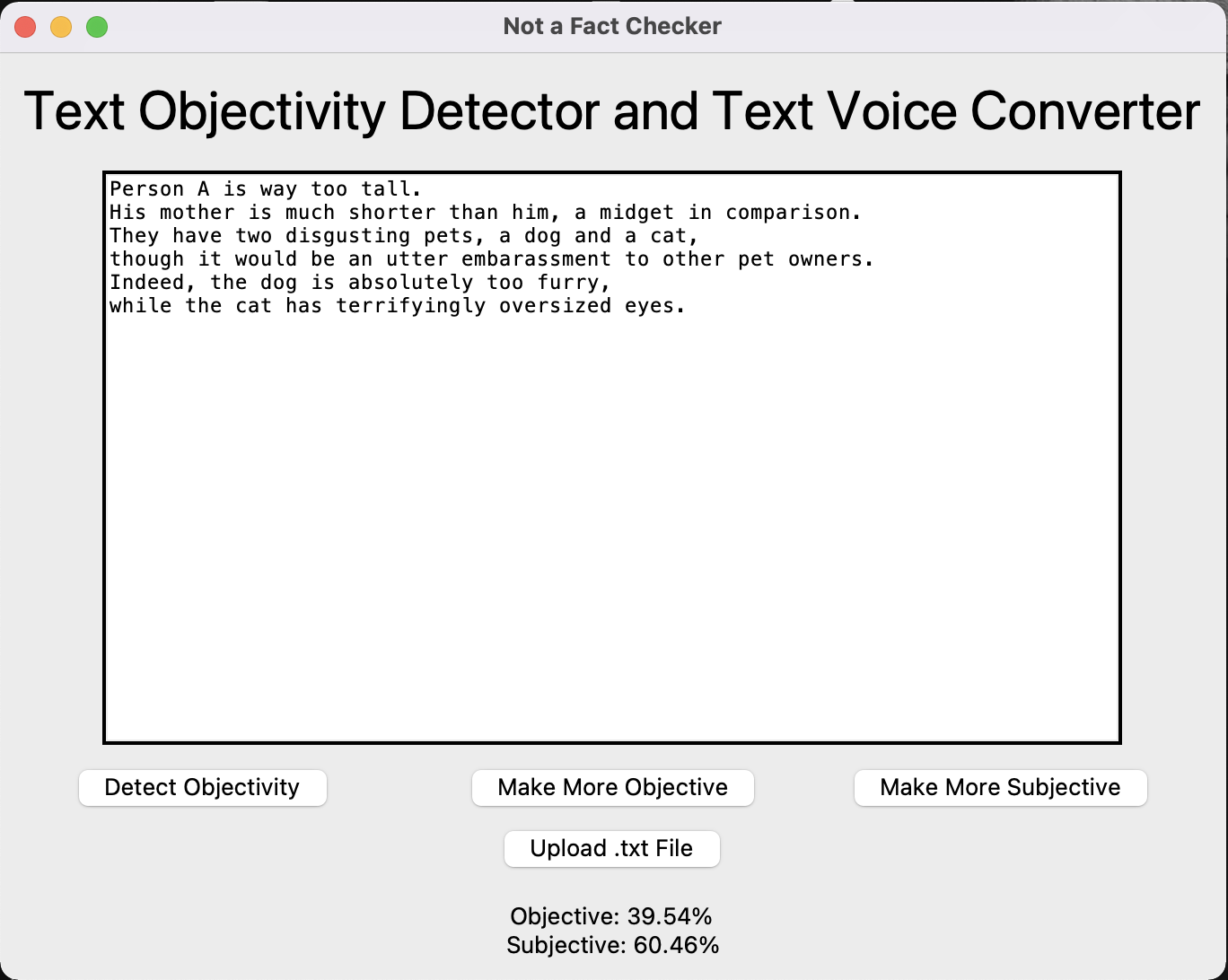
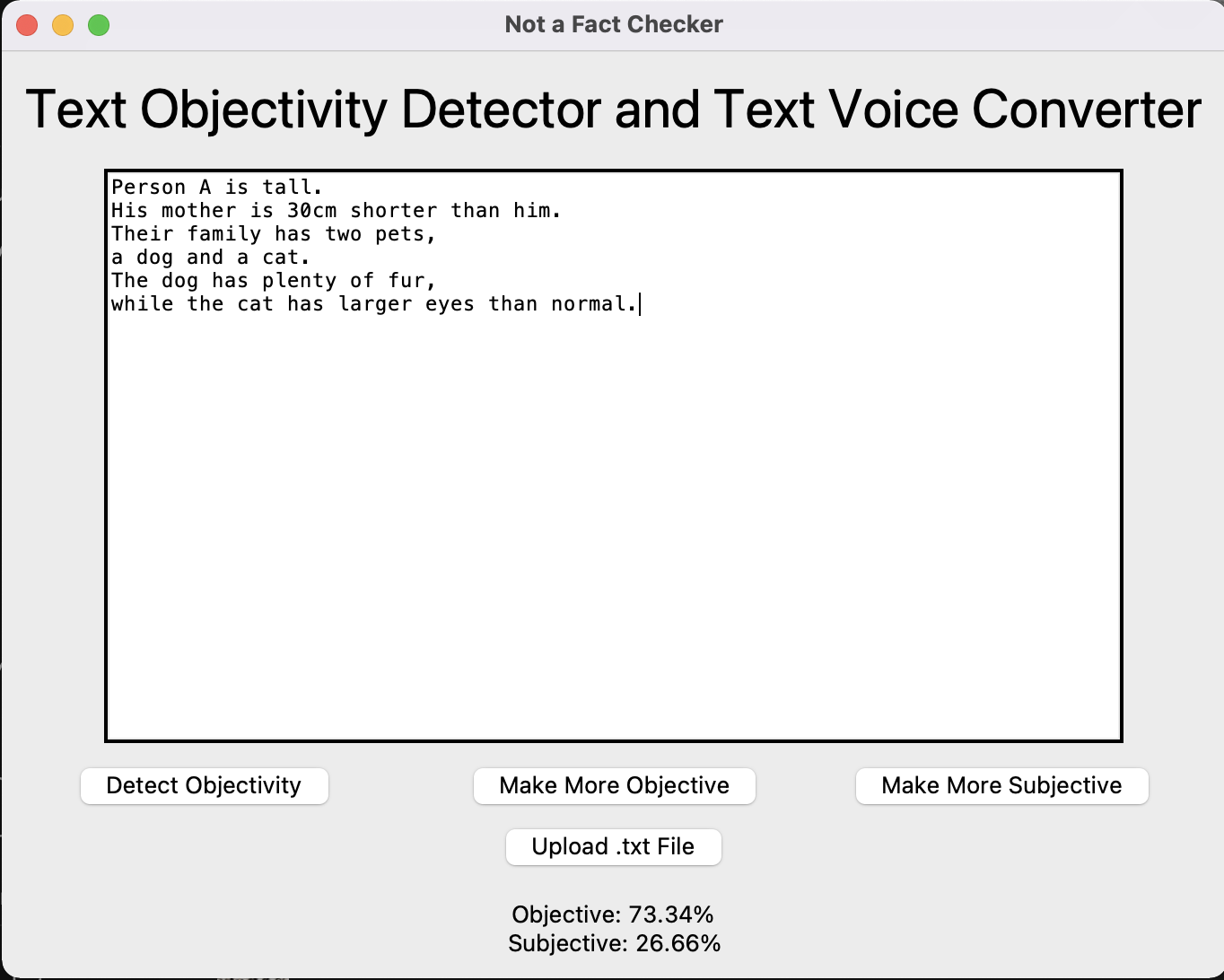


Figure 2a (left) and figure 2b (right). Both depict the same situation, but have different voices about it.

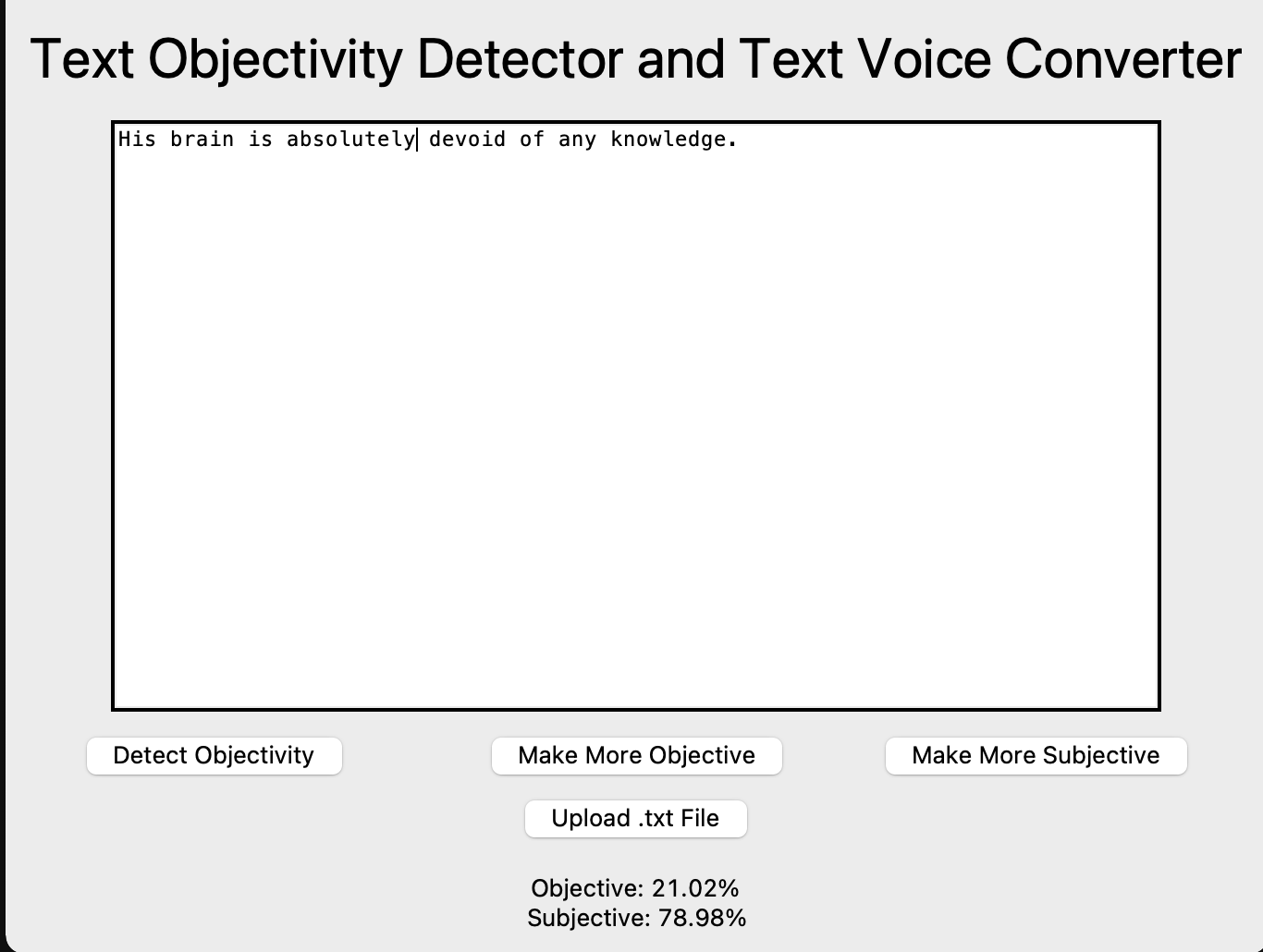
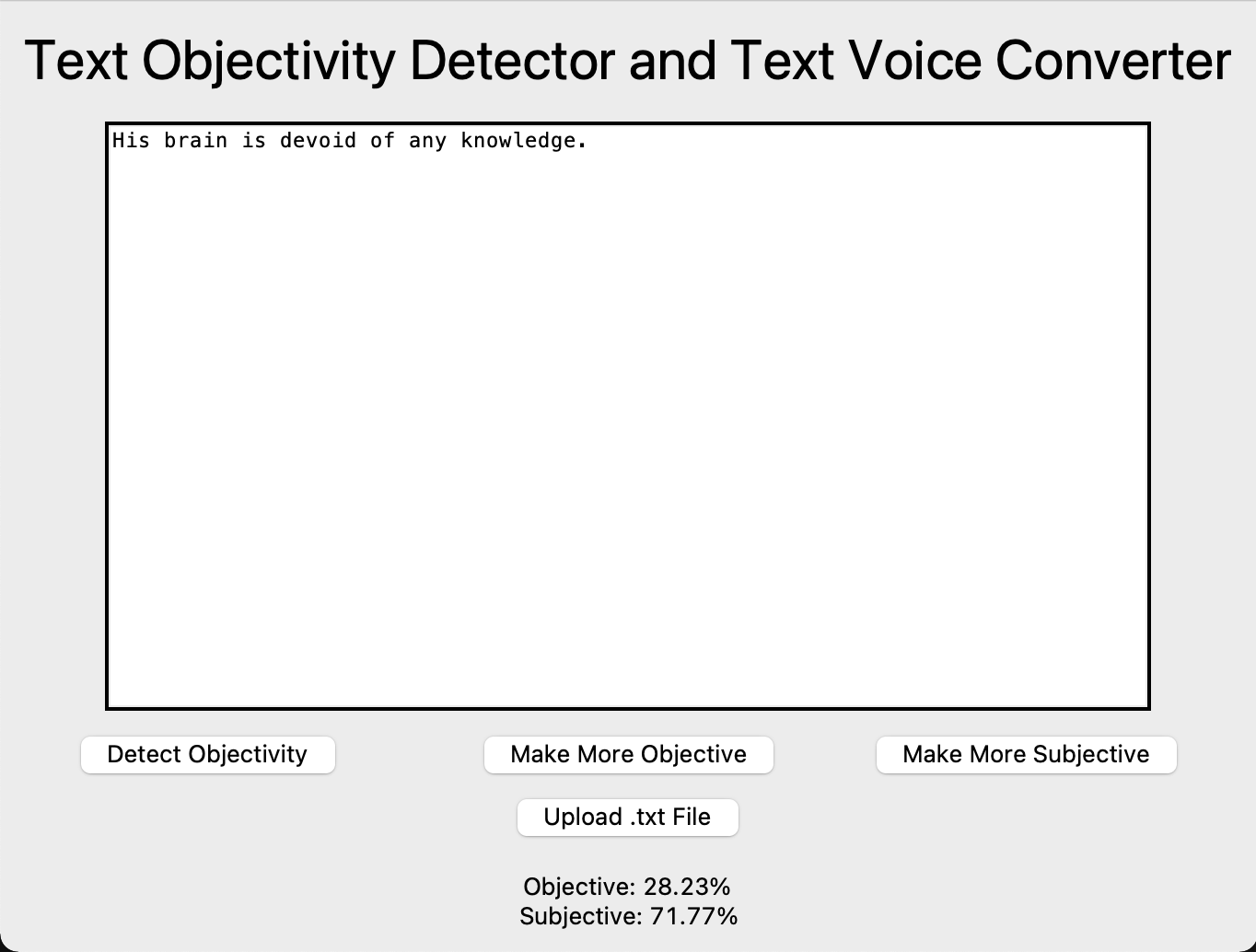


Figure 3a (left) and figure 3b (right). The only difference between the two is the addition of a single word.

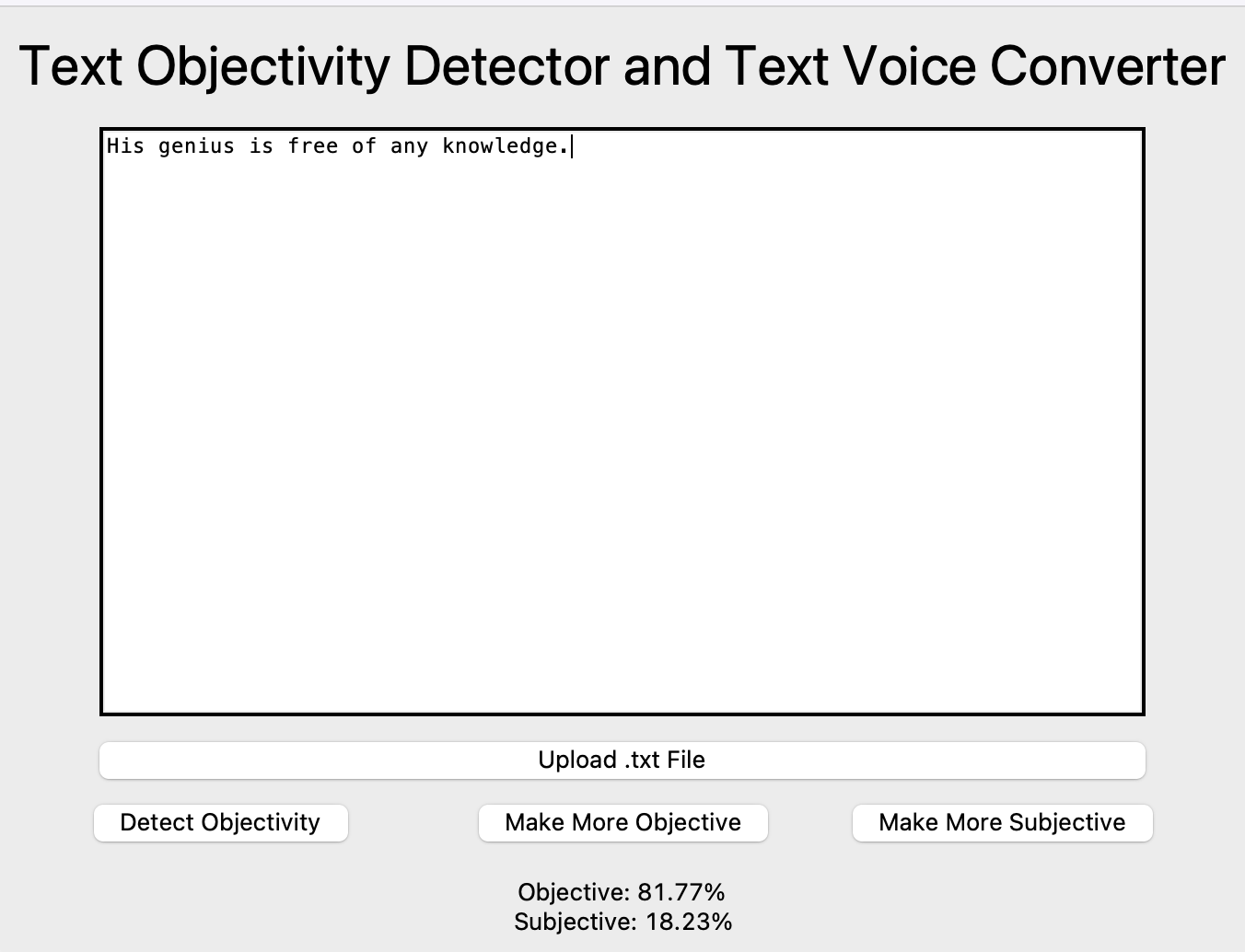


Figure 4 (above), an attempted objectification using the Naiïve method. The original text is the same as Figure 3, but notice the poor substitution of words despite the higher objectivity rate.

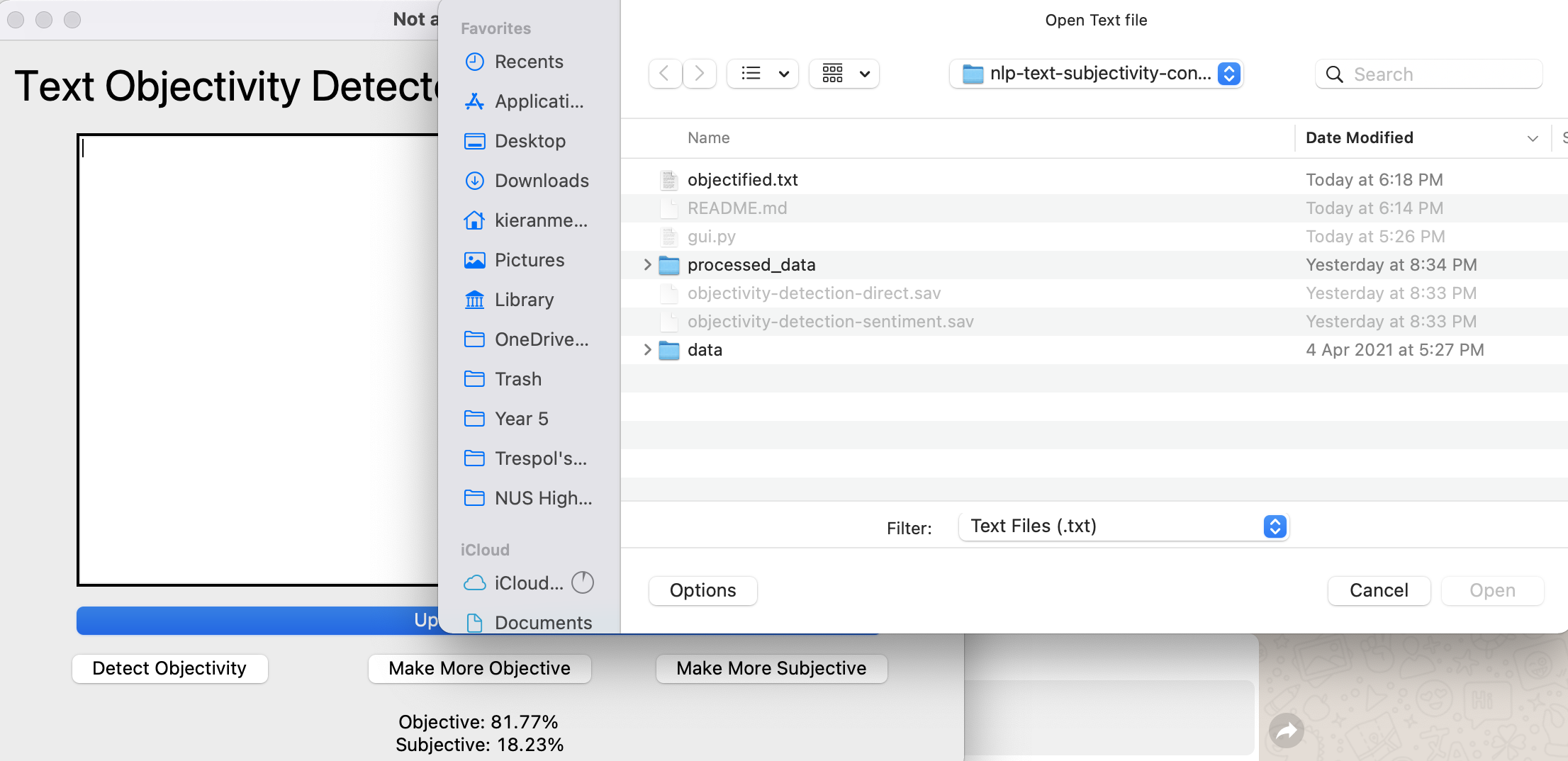


Figure 5 (above). File dialog to upload .txt files only.

# Limitations

A large limitation of our project is that it is trained on movie review data, and while subjective indeed (it is from Rotten Tomatoes), it may not be well-suited for other real-life cases. Testing has shown that the model struggles more when it comes to more opinionated and lengthy pieces, like op-eds and commentaries from news websites. It may also be unfamiliar with slang and slur words and cannot therefore be fully accurate in the modern context.

A justification for this phenomena is that for op-eds, the subjectivity of the stance is diluted by the evidence and facts that back the author’s opinion, therefore the net result is that the overall text turns out to be objective. This leads to a different ML problem to be solved, which is determining the ‘crux’ of the passage and determining the literature type - commentary (subjective) or simply news (objective). Currently, our model is unable to discern the two.

# Conclusion and Recommendations

The implementation of objectivity detection was done acceptably, with the model working especially well for most text, with the exception of length op-eds. It is able to capture the subjectivity of the model quite well, with the addition of modifiers like ‘absolutely’, ‘indeed’ and ‘very’ having noticeable impact on the text objectivity.

# Reflection

Thinking about the flow of the entire project, I feel that the work distribution was done in an equal and efficient manner, and there was not much overlapping in the work that needed to be done. The delegation of tasks was relatively well executed, and there was no major conflict on where the project direction should be. In short, the project flow was relatively seamless and well-organised.

The machine learning part of the project turned out to be quite enjoyable, and we didn’t need to reinvent the wheel most of the time, giving us greater space to explore and build on existing frameworks to generate something greater. Seeing how our application has potential use cases in the real world, this is quite a satisfying experience.

One main challenge in the project was finding a sufficiently comprehensive dataset to train our models on. There weren’t many out there, so we settled with movie review data and plot summaries as the basis of our objective-subjective dataset. I felt that it was a good call, since they fit into the extremes of the spectrum, and turned out to be useful in the training of our model.

Another was the difficult nature of the text transformation. It is much more complicated than the first part which is simply a classification problem, since it requires the voice to be changed while ensuring that the content remains relatively similar. The inherent difficulty of the text transformation can be seen by the greater amount of time required to train the model.

# References

[1] <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

[2] <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>

[3] <https://ojs.aaai.org/index.php/AAAI/article/view/6433>

[4] <https://arxiv.org/abs/1809.04556>

[5]<https://papers.nips.cc/paper/2017/file/2d2c8394e31101a261abf1784302bf75-Paper.pdf> Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. NIPS 2017\

[6]Youzhi Tian, Zhiting Hu, and Zhou Yu. 2018. Structured content preservation for unsupervised text style transfer. arXiv preprint arXiv:1810.06526.

# Acknowledgements

We’d like to take this opportunity to thank Mr Claude Chua for his guidance in the ideation process, without which we would be unable to come up with a concrete idea to work on. Additionally, we’d like to thank Cornell University for compiling and publishing the dataset, which serves as the basis for our project. Last but not least, we’d like to recognise the open-source community for providing useful and functional frameworks for us to utilise and accelerate our project development, without needing to reinvent the wheel.

**Note:** You may modify or include additional sections if there is a need. The above is just a suggested template.

# Work Distribution Matrix

|  |  |  |
| --- | --- | --- |
| **Work Description** | **Kieran** | **Hengyue** |
| Ideation & Proposal | ✔ | ✔ |
| Data Fetching |  | ✔ |
| Data Cleaning | ✔ |  |
| Classifier Design | ✔ |  |
| Classifier Evaluation | ✔ |  |
| Text Transformer Design |  | ✔ |
| Text Transformer Evaluation |  | ✔ |
| UI | ✔ |  |
| Report | ✔ | ✔ |
| Presentation Slides | ✔ |  |