Can Deep Learning Approaches be Employed to Accurately Perform Text Readability Assessment?

Ashwathi Pillai
20029015

21/22 MSc Data Science and AI for the Creative Industries

University of the Arts London

Unit Title: Natural Language Processing for the Creative Industries

Unit Code: IU000131

Assignment : Element 1 (Critical Essay)

Course Leader: Dr. Louis McCallum

Unit Tutor: Dr. Rebecca Fiebrink

17/12/2021 Word Count: 2843

“Simplicity is the ultimate sophistication.”
— Leonardo da Vinci

INTRODUCTION:

Research into the area of text readability has become increasingly important over the past few years, given how much data is produced online on a daily basis. Writers are expected to produce original written thought, or “copy”, within a specified word count due to space constraints and low attention spans (Goldhaber, 1997), which makes it increasingly important to make sure that these compact texts still reaches a wide range of audience, and that they are easy to understand. Even teachers, while creating a reading list for students, might make use of a text readability software to ensure that they’re choosing books that are in tandem with their students’ reading level, as the student pool could comprise of children with learning disabilities, immigrant populations, and second or foreign language learners (Vajjala and Meurers, 2012). As a result, a consistent and reliable method for determining if a text is accessible to the target population is required.

BACKGROUND:

1) Traditional Formulas for Readability Assessment:

Many great strides have been made in the area of readability research, the earliest being manual formulas to compute how readable a text is. Most of them were designed in the first half of the 20th century, and only a small number of surface-level linguistic variables, such as word length, sentence length, and frequency of words, were taken into account while developing them (Klare, 1963). The most popular of them all is the Flesch-Kincaid Grade Level Formula, which outputs a score on a scale of an American student's reading comprehension level (Kincaid et al., 1975). A score of 9.3 indicates that the text is understandable to an American student in the ninth grade. The Flesch-Kincaid Reading ease is yet another popular formula that calculates how easy the text is to read on a scale from 1 to 100 - the higher the score, the more accessible the content is (Kincaid et al., 1975). Most states in America now require insurance forms to score between 40 to 50 on the Reading Ease test to be considered valid, along with various other government documents (Schriver, 2017). Other formulas such as the SMOG Readability Test, The Coleman–Liau Index, The Dale–Chall Readability Formula, the Gunning Fog

Index, etc. are used in various industries as well, and they operate on similar principles (Kincaid et al., 1975).

However, these formulas have been challenged by several critics as being "reductionist and having a weak statistical foundation" (Martinc, Pollak and Robnik-Šikonja, 2019). The Flesch-Kincaid tests were deduced to be quite unadaptable, as a lot of variation between scores in American English and English spoken in other regions of the world were found - since readability formulas were highly dependent on the syntactic constructions of the language (Klare, 1963). Most importantly, factors such as sentence length couldn't accurately compute complexity, as an unnecessarily long sentence with very simple vocabulary would score lower on Reading Ease than a short sentence with very complex words. Therefore, other approaches were required to be devised.

2) Cloze Procedure

Cloze Procedure is a technique for "omitting words from a passage so that the reader is forced to use background experience, knowledge of syntax, vocabulary, interest, and, generally, higher-order thinking skills to fill in the blank and complete the thought." (Student Compass, 2021). Researchers have used this method to evaluate the readability score by letting readers fill in missing content after every five words in sentences (Horton, 1974). The level of the education of the reader is then used to define the readability level of the text. Evidently, this isn't the most reliable way of measuring text readability as experience with the language could be very subjective despite their education level on paper, and the choice of participants for this activity really defines the whole scale which is not ideal as it's not always a representative sample.

3) Deep Learning Approaches:

Data-driven language models are the newest area of research in the field of text readability. Some of these efforts involve using high-level textual features, such as “semantic and discursive properties of texts” (Martinc, Pollak and Robnik-Šikonja, 2019), for readability modelling. Modern readability checkers such as Grammarly work using this technique. This project also revolves around exploring deep learning approaches and implementing them on manually crafted datasets.

PURPOSE OF THIS PROJECT:

Aim: To construct a model that allows writers to have more control over their writing, so that they could structure their work according to their intended audience.

The purpose of this project was to build and analyse various deep learning models for text readability classification in order to determine whether a piece of text is easy or hard to read. Naive Bayes, Convolutional Neural Network, and distilBERT models were trained on a self-curated corpus which was divided into three distinct labelled datasets. Each of these datasets were also analysed using word embeddings with t-SNE to get a better understanding of it. Lastly, upon all the classifiers being ready after manual parameter-tweaking, they were tested using Wiki-Simple and Wiki-Normal pages.

METHODOLOGY:

1) Corpus Development:

English textbooks from India, Afghanistan, Indonesia, and the Islamic State were interchangeably used to form three separate datasets. This was done for several different aims: to see if variances in grade ranges affect the performance, to study how the same language is taught in different regions, to examine if changes in region of the textbook utilized for the same classifier affected performance, and to verify if the model could accurately classify English phrases from across the world using

non-western data. Due to this data diversity, one can evaluate what effects the specifics of each dataset has on the general robustness of the classifier. The nature of the content was common in all the textbooks, consisting of stories, poems, plays, and English language acquisition exercises. All of these texts were sourced from Library Genesis, and extracted from PDF files using an online file converter. The three finalised datasets with labels were as follows:

a) Dataset 1: First and Tenth Grade English Textbooks from India by the National Council of Educational Research and Training Publication (2009 Edition)

b) Dataset 2: Fourth and Twelfth Grade English Textbooks from Afghanistan by the country's Ministry of Education (2011 Edition)

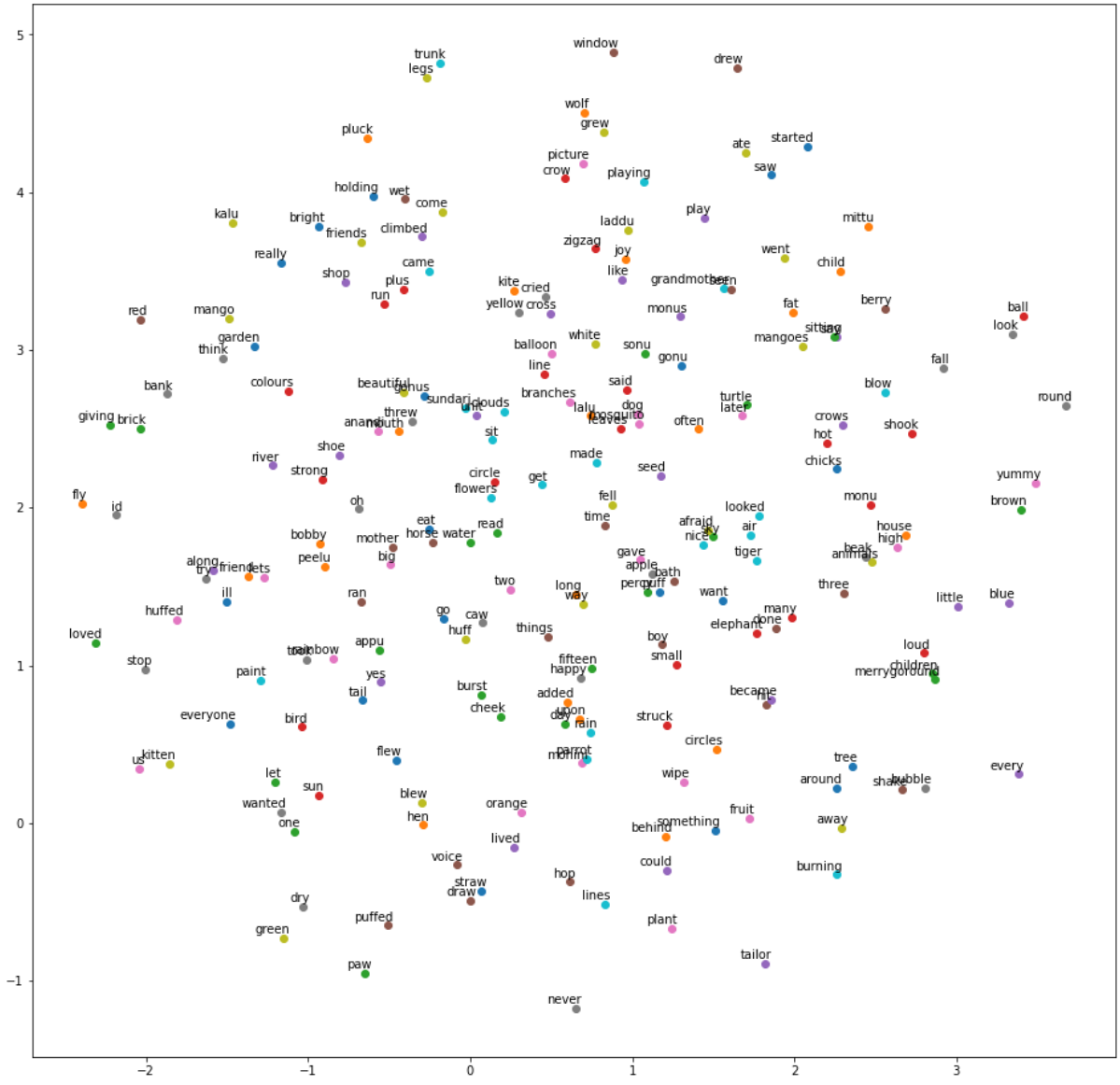
c) Dataset 3: A Second Grade English Textbook from Indonesia by the country's Ministry of Education and Culture (2008 Edition), and a Sixth Grade Textbook from the Islamic State (Publisher Unknown).

2) Data pre-processing:

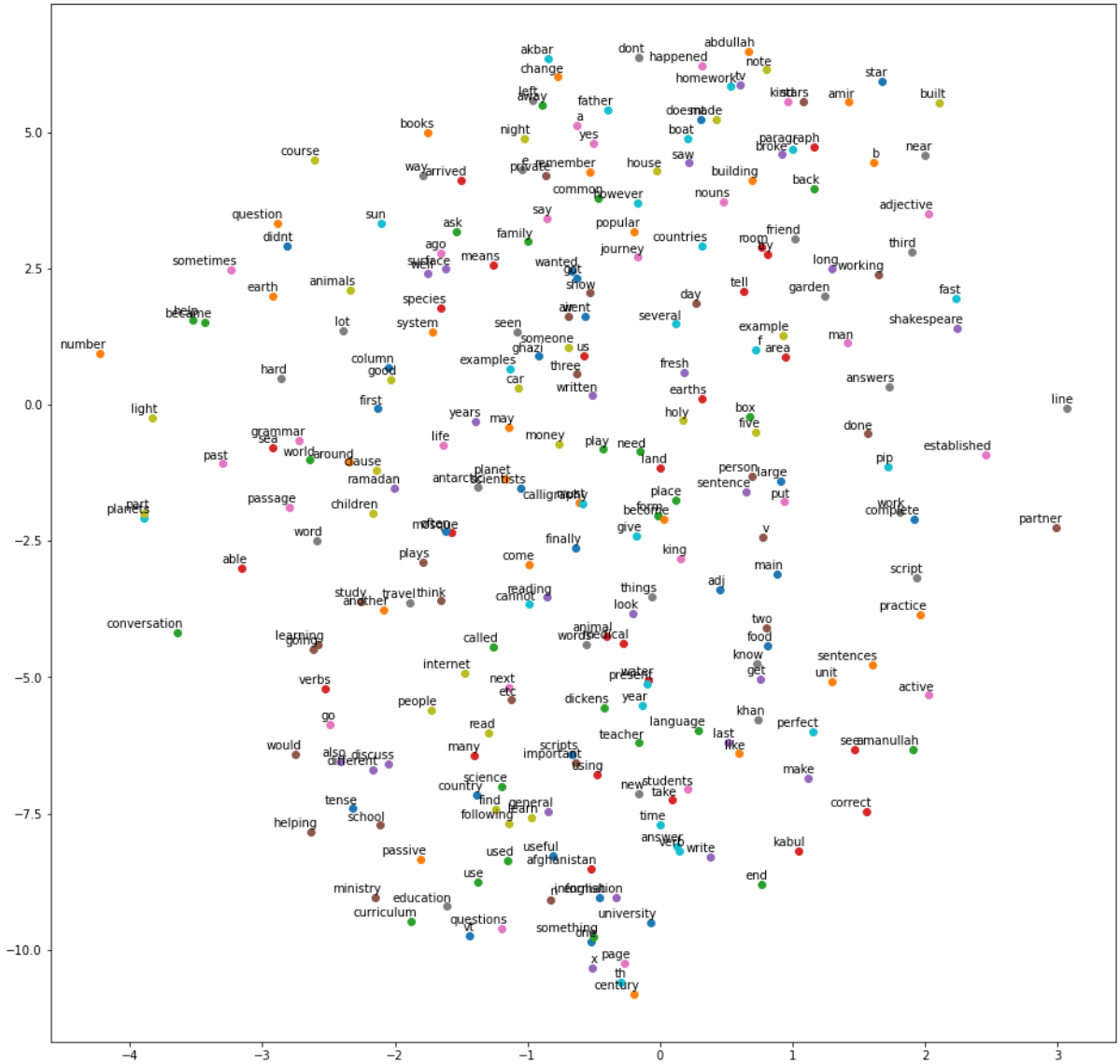
Data Cleaning was performed on the extracted texts using Regex, the CleanText Library (clean-text, 2021), and Data Slicing to remove list of contents and unrequired noise such as special characters, numbers, URLs, etc. Next, the question “What constitutes a sentence in this dataset?” was to be answered. A 'read_and_clean' function was defined to read any given text file and clean the data in it (so that the code doesn’t repeat in the notebook), whilst replacing all line-breaks to full stops if two or more line breaks are together and if two or more spaces are together, and replacing all line-breaks to spaces if it is followed by a small letter. This was done as the text files for this project weren’t following a particular pattern with grammar since it was extracted from an image-heavy PDF, resulting in random line-breaks. After that, a for-loop for sentence-splitting after every full stop was utilised, and a conditional was added to discard any sentences with less than two words.

3) Dataset Analysis:

Given the diversity of the textbooks in the three final datasets, the textbooks were then analysed to better understand the corpus. For this purpose, word embeddings from the datasets were visualized onto a 2-dimensional space using t-distributed Stochastic Neighbor Embedding (t-SNE), which would provide us with some further insight into any syntactic relationships that stand out (Chah, 2021). The basic idea behind this is “to keep similar words close together on the plane, while maximizing the distance between dissimilar words” (Delaney, 2021). The visualisations generated were found to be slightly messy in certain areas, despite playing around with the min_count. In the ‘Grade One Textbook from India’, the words ‘draw’ and ‘straw’ were closely placed possibly because they both rhymed with each other. Also, the words ‘sundari’ and ‘beautiful’ were closely placed which was quite interesting since ‘sundari’ means beautiful in Hindi, which concludes that these techniques can also compute similarities in different languages along with English. Even ‘loud’ and ‘children’ were closely placed which was a funny correlation.



Data from the ‘Grade Twelve Textbook from Afghanistan’ was the only textbook that had country-specific words like ‘Kabul’ and ‘Afghanistan’ that showed up, possibly suggesting that the curriculum must have emphasized on inculcating national pride amongst students.



4) Passing the Data to Various Machine Learning Algorithms:

The pre-processed datasets were then split into training and testing sets after extracting features, and then passed onto three different algorithms for training. The details of the same are described below:

a. Naive Bayes (Machine Learning):

The three datasets were passed onto the Naive Bayes model, which is a "simple classifier that assumes that the presence of a particular feature in a

class is unrelated to the presence of any other feature." (Ray, 2021). TF-IDF features were used for the same.

Dataset 1 had a training set size of 90% and test set size of 10%, resulting in a model accuracy of 91.40%

Dataset 2 had a training set size of 70% and test set size of 30%, resulting in a model accuracy of 96.31%

Dataset 3 had a training set size of 80% and test set size of 20%, resulting in a model accuracy of 74.18%

Out of all the three, Dataset 1 performed the best whilst other predictive models were possibly facing an issue called 'overfitting', meaning "a model learning the details and noise in the training data to such an extent that it negatively impacts the performance of the model on new data" (Brownlee, 2021). Various test-size combinations were tried out, and cross-checking was done between both the labelled corpus to ensure that data wasn't imbalanced - but still poor results were noticed, possibly suggesting an issue with the dataset. Hence, the Naive Bayes classifier for Dataset 1 was selected for final testing as it performed the best out of the three.

b. Convolutional Neural Network (Deep Learning):

"CNN is a class of deep, feed-forward artificial neural networks which use a variation of multilayer perceptrons designed to require minimal preprocessing" (Maheshwari, 2021). Despite being typically used for image classification tasks, these algorithms can also be utilized for text classification problems. Pre-trained word vectors from Google trained on a News Dataset of about 100 billion words were used (Code.google.com, 2013).

Dataset 1 had a training set size of 80% and test set size of 20%. Various kernel sizes from 2 to 10 were inputted (10 being the most ideal in this case), with the dropout layer being increased to 0.4 to avoid overfitting. Lastly, the model was built and trained on 7 epochs, resulting in a model accuracy of 27.31%. This was really surprising since the same dataset performed really well when passed onto the Naive Bayes algorithm. A lot of manual experimentation with the aforementioned parameters were done, but unfortunately the model accuracy did not increase.

Dataset 2 had a training set size of 80% and test set size of 20%. Various kernel sizes from 2 to 10 were inputted, 5 being the most optimum in this case. Dropout layer was also increased to 0.3 to avoid overfitting. Lastly, the model was built and trained on 8 epochs, resulting in a model accuracy of 94%. Given that Dataset 2 also had a high accuracy with Naive Bayes, this wasn't surprising.

Dataset 3 had a training set size of 70% and test set size of 30%. Various kernel sizes from 2 to 10 were inputted, 5 being the most optimum in this case. Dropout layer was kept at 0.1 to avoid overfitting. Lastly, the model was built and trained on 9 epochs, resulting in a model accuracy of 85.14%. It was also noticed that when data was balanced, the model wouldn't perform well - which was an interesting finding.

To explore why Dataset 1 performed badly, a new dataset was created combining the First Grade Indian textbook and the Twelfth Grade Afghan Textbook. The exact same parameters were used as Dataset 1, but the model accuracy turned out to be 90.88%. It is unclear as to why this happened, as this meant that the model wasn't learning at all with Dataset 1. A possible explanation for this could be the variation in the dataset, since textbooks from the same country and publisher might have repetitive vocabulary. But the fact that the Naive Bayes classifier performed well with the same dataset negates this fact.

To properly confirm whether variation in dataset increases model accuracy, other textbook combinations were used to create new datasets to pass it on to the CNN algorithm, but they were unable to yield high model accuracy scores too. Hence, the hypothesis about data variation was proved to be false. In the end, the CNN model with Dataset 2 was selected for final testing.

c. DistilBERT (Transfer Learning):

"DistilBERT is a small, fast, cheap and light Transformer model based on the BERT architecture, wherein knowledge distillation is performed during the pre-training phase to reduce the size of a BERT model by 40%" (Sanh, Debut, Chaumond and Wolf, 2019). It was pre-trained on the same raw texts as BERT, with zero manual labelling. Since computers with graphics cards weren't used, fine tuning DistilBERT proved to be the ideal option for faster computational speed.

Dataset 1 had a training set size of 80%, and test set size of 20%. The batch size for evaluation was increased from 16 to 64, as better results were noticed this way. Various epoch values were inputted, and it was noticed that the model accuracy kept increasing (upto 95%) when epochs were increased. However, when the dataset was balanced, model accuracy dropped to 90%. Test set size was then changed to 10%.

Dataset 2 had a training set size of 90%, and a test size of 10%. The batch size for evaluation was increased from 16 to 64, as better results were noticed this way. Various epoch values were inputted, and it was noticed that the model accuracy was constant throughout the process (at 98%). However, when the dataset was balanced, model accuracy dropped to 97%. Test set size was then changed to 20% but the same model accuracy was observed.

Dataset 3 took the longest amount of time to run, due to the data size of the Indonesian textbook being very huge. The training set size was 70%,

and the test set size was 30%. The batch size for evaluation was increased from 16 to 64, as better results were noticed this way. The model accuracy was observed to be 89% for 5, 6 and 9 epochs.

The DistilBERT model with Dataset 1 was selected for final testing.

RESULTS:

Testing the Final Classifiers:

Articles from Wiki-simple (Wikipedia, 2021) and Wiki-normal (Wikipedia, 2021) were used to test the final classifiers. Lines from the Saturn article (Saturn - Simple English Wikipedia, 2021) were selected to test if the classifier could label them as 0 or ‘Easy’. The following are the results:

Naive Bayes:

```
In [23]: # https://git.arts.ac.uk/lmccallum/nlp-21-22/blob/master/NLP%20Week%204.1%20-%20Classification%20Task.ipynb
         # trying new data
         your_new_data = new_tfidf
         y_pred = model.predict(your_new_data)
         # looking at predictions
         for i, t in enumerate(my_new_text):
             print(t," -> class:", y_pred[i])

         Saturn is the sixth planet from the Sun located in the Solar System  -> class: 0
```

CNN:

```
In [36]: text = np.array(["Saturn is the sixth planet from the Sun located in the Solar System"])
         feat = tokenize_and_vectorize(text)
         feat = pad_trunc(feat, maxlen)
         feat = np.reshape(feat, (len(feat), maxlen, embedding_dims))

In [37]: predict = model.predict(feat)[0][0]
         if predict >0.5:
           print(1)
         else:
           print(0)

         1
```

DistilBERT:

```
In [30]: inputs = tokenizer("Saturn is the sixth planet from the Sun located in the Solar System", return_tensors="pt")
         labels = torch.tensor([1]).unsqueeze(0)
         outputs = model(**inputs, labels=labels)
         loss = outputs.loss
         logits = outputs.logits

In [31]: logits

Out[31]: tensor([[-1.4160,  1.4341]], grad_fn=<AddmmBackward0>)

In [32]: if logits[0][0] > logits[0][1]:
           print(0)
         else:
           print(1)

         1
```

Lines from the Civil Rights Congress (Civil Rights Congress - Wikipedia, 2021) were selected to test if the classifier could label them as 1 or ‘Hard’. The following are the results:

Naive Bayes:

```
In [21]: my_new_text = ["The CRC used a two-pronged strategy of litigation and demonstrations, with extensive public communicati

In [22]: # https://git.arts.ac.uk/lmccallum/nlp-21-22/blob/master/NLP%20Week%204.1%20-%20Classification%20Task.ipynb
         # turning new text into vectors
         new_tfidf = tfidf_model.transform(my_new_text).todense()

In [23]: # https://git.arts.ac.uk/lmccallum/nlp-21-22/blob/master/NLP%20Week%204.1%20-%20Classification%20Task.ipynb
         # trying new data
         your_new_data = new_tfidf
         y_pred = model.predict(your_new_data)
         # looking at predictions
         for i, t in enumerate(my_new_text):
             print(t," -> class:", y_pred[i])

         The CRC used a two-pronged strategy of litigation and demonstrations, with extensive public communications, to call a
         ttention to racial injustice in the United States.   -> class: 1
```

CNN:

```
In [36]: text = np.array(["The CRC used a two-pronged strategy of litigation and demonstrations, with extensive public communica
         feat = tokenize_and_vectorize(text)
         feat = pad_trunc(feat, maxlen)
         feat = np.reshape(feat, (len(feat), maxlen, embedding_dims))

In [37]: predict = model.predict(feat)[0][0]
         if predict >0.5:
           print(1)
         else:
           print(0)

         1
```

DistilBERT:

```
In [33]: inputs = tokenizer("The CRC used a two-pronged strategy of litigation and demonstrations, with extensiv
         labels = torch.tensor([1]).unsqueeze(0)
         outputs = model(**inputs, labels=labels)
         loss = outputs.loss
         logits = outputs.logits

In [34]: logits

Out[34]: tensor([[-1.4160,  1.4341]], grad_fn=<AddmmBackward0>)

In [35]: if logits[0][0] > logits[0][1]:
           print(0)
         else:
           print(1)

         1
```

The Naive Bayes model performed the best out of the three, getting all five lines from Simple and Normal Wiki Pages accurately. Interestingly, CNN and the DistilBERT model didn't perform well as they kept predicting class 1 for every text – despite being known for their state of the art performance. This reason for this is unclear, but could perhaps be due to insufficient data or overfitting.

DISCUSSION / FUTURE ITERATIONS:

Having explored how impressive these models are, it is also important to realise the environmental and ethical implications behind running such powerful software. It is estimated that "training a BERT model…consumes as much energy as a trans-American flight." (McAuliffe, 2021).

For future work, given that the Naive Bayes classifier in this project proved to perform sufficiently well - the model could perhaps be used to test out a use case

wherein speech styles of various politicians are analysed in order to examine if ease of comprehension has any corelation to better outreach.

The next logical step to build upon this project would be to include more classes, add more features, and to try out new algorithms such as Linear SVM and the Hierarchical Attention Network (Maheshwari, 2021) to see if performance is improved. Also, making use of a classification matrix to visualise the results could be done.

Lastly, making use of a more structured database like the WeeBit Corpus (Martinc, Pollak and Robnik-Šikonja, 2019) would be ideal before concluding that the Naive Bayes is the perfect solution for this task. The beauty of Natural Language Processing is how fast-growing the field is, with new advancements almost every month (McAuliffe, 2021). Hence, more robust solutions could definitely be employed in the near future.

REFERENCES:

1) Bormuth, J., 1966. Readability: A New Approach. Reading Research Quarterly, [online] 1(3), pp.3-8. Available at: <https://www.jstor.org/stable/747021?seq=1#metadata_info_tab_contents> [Accessed 14 December 2021].

2) Brownlee, J., 2021. Overfitting and Underfitting With Machine Learning Algorithms. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/> [Accessed 17 December 2021].

3) Chah, N., 2021. Apply word2vec to all sorts of text documents.. [online] GitHub. Available at: <https://github.com/nchah/word2vec4everything> [Accessed 14 December 2021].

4) Code.Google.com, 2021. Word2Vec - Google Code. [online] Code.Google.com Available at: <https://code.google.com/archive/p/word2vec/> [Accessed 17 December 2021].

5) Olms.cte.jhu.edu, 2021. Cloze Procedure : Student Compass: Instructional Strategies Bank. [online] Available at: <http://olms.cte.jhu.edu/olms2/7243> [Accessed 17 December 2021].

6) Delaney, J., 2021. Visualizing Word Vectors with t-SNE. [online] Kaggle.com. Available at: <https://www.kaggle.com/jeffd23/visualizing-word-vectors-with-t-sne> [Accessed 16 December 2021].

7) En.wikipedia.org. 2021. Civil Rights Congress - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Civil_Rights_Congress> [Accessed 17 December 2021].

8) En.wikipedia, 2021. Normal - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Main_Page> [Accessed 17 December 2021].

9) Goldhaber, M., 1997. The attention economy and the Net. First Monday, [online] Available at: <https://firstmonday.org/article/view/519/440> [Accessed 13 December 2021].

10) Horton, R., 1974. The Construct Validity of Cloze Procedure: An Exploratory Factor Analysis of Cloze, Paragraph Reading, and Structure-of-Intellect Tests. Reading Research Quarterly, [online] 10(2), p.248. Available at: <https://www.jstor.org/stable/pdf/747185.pdf?refreqid=excelsior%3Acf937d464f35e0adfc10d11b7b1496f3> [Accessed 6 December 2021].

11) Kincaid, J., Fishburne, R., Rogers, R. and Chissom, B., 1975. Derivation Of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel. [online] Available at: <https://stars.library.ucf.edu/cgi/viewcontent.cgi?article=1055&context=istlibrary> [Accessed 15 December 2021].

12) Klare, G., 2000. The measurement of readability. ACM Journal of Computer Documentation, [online] 24(3), pp.107-121. Available at: <https://dl.acm.org/doi/pdf/10.1145/344599.344630> [Accessed 5 December 2021].

13) Libgen.is. 2021. Library Genesis. [online] Available at: <http://libgen.is/> [Accessed 17 December 2021].

14) Maheshwari, A., 2021. Report on Text Classification using CNN, RNN & HAN. [online] Medium. Available at: <https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f> [Accessed 17 December 2021].

15) Martinc, M., Pollak, S. and Robnik-Šikonja, M., 2019. Supervised and Unsupervised Neural Approaches to Text Readability. Computational Linguistics, [online] 47(1), pp.141-179. Available at: <https://direct.mit.edu/coli/article/47/1/141/97334/Supervised-and-Unsupervised-Neural-Approaches-to> [Accessed 16 December 2021].

16) McAuliffe, D., 2021. Developments in AI: Language models - Arabesque. [online] Arabesque. Available at: <https://www.arabesque.com/2021/07/06/developments-in-ai-language-models/> [Accessed 17 December 2021].

17) Pogiatzis, A., 2021. NLP: Contextualized word embeddings from BERT. [online] Medium. Available at: <https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b> [Accessed 17 December 2021].

18) PyPI. 2021. clean-text. [online] Available at: <https://pypi.org/project/clean-text/> [Accessed 17 December 2021].

19) Ray, S., 2021. Learn Naive Bayes Algorithm | Naive Bayes Classifier Examples. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/> [Accessed 16 December 2021].

20) Sanh, V., Debut, L., Chaumond, J. and Wolf, T., 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. [online] Available at: <https://arxiv.org/pdf/1910.01108v4.pdf> [Accessed 17 December 2021].

21) Schriver, K., 2017. Plain Language in the US Gains Momentum: 1940–2015. IEEE Transactions on Professional Communication, [online] 60(4), pp.343-383. Available at: <https://www.plainlanguage.gov/media/Schriver%20Plain%20Language%20in%20US%20Gains%20Momentum%201940_2015%20Draft.pdf> [Accessed 12 December 2021].

22) Simple.wikipedia.org. 2021. Wikipedia. [online] Available at: <https://simple.wikipedia.org/wiki/Main_Page> [Accessed 17 December 2021].

23) Vajjala, S. and Meurers, D., 2012. On Improving the Accuracy of Readability Classification using Insights from Second Language Acquisition. [online] pp.163-165. Available at: <https://dl.acm.org/doi/pdf/10.5555/2390384.2390404> [Accessed 13 December 2021].