Wikipedia Sentences Complexity Classification

Team 24, Tony Lan (tonylan@umich.edu) and Thanuja Stewart (tpstewar@umich.edu)
February 2023

Introduction

This project analyzes sentences labeled with complexity labels. Understanding English sentence complexity would allow public websites such as Wikipedia to automatically flag text for review and revision. This would enhance readability for all readers, especially for those whom English is a second language. The motivation for this project is to explore the effectiveness of different feature representations and machine learning models on natural language. Findings here could be applied to other text classification tasks.

For supervised classification, we engineered features and trained various models against these labels. The supervised models explored were LinearSVC (Support Vector Classifier) [9], MLP (Multilayer Perceptron) [10], RF (Random Forest) [11], GRU (Gated Recurrent Unit) [12], and LLM (Large Language Model, specifically DistilBERT [1] through Huggingface [13]). LinearSVC, MLP, and RF were trained on both a bag of words approach and engineered features. GRU was trained on parts of speech sequences, which we believe is a novel contribution to this classification task. And finally, the LLM was trained on raw text that transformed using DistilBERT's default tokenization and word embeddings. This was a challenging task for supervised learning, and the best accuracy of about 75% was achieved with large language models. Looking deeper into the supervised results, we'll show that complexity depends on both *structure* and *content*.

For unsupervised learning, vectorized and non-vectorized features were used in the K-means [14] clustering algorithm. The impact of PCA [15] and scaling of data on accuracy was also explored. Hyperparameter tuning was conducted to optimize outcomes. Confusion matrix and Error Analysis were examined to gain a deeper understanding into the nature and characteristics of classification errors by the algorithm.

Related Work

In *Text Complexity Classification Based on Linguistic Information: Application to Intelligent Tutoring of ESL* (M. Zakaria Kurdi) [2], large ESL (English as a Second Language) passages are classified by text complexity. M. Zakaria Kurdi's paper is focused on passage, rather than sentence complexity. This is similar to our own project but our target task is binary rather than multi-class. Additionally, Kurdi's paper is focused on "scalability", which in this context means being able to apply training from one domain to another. Our experiments will stay within one domain (wiki text).

In *A Language Modeling Approach to Predicting Reading Difficulty* (K. Collins-Thomson and J. Callan) [3], web passages are classified into 12 grade reading levels. This text classification is similar to our own in that the passages can be short, but the modeling approach is different. K. Collins-Thomson and J. Callan form a probability distribution for each word across grade levels, and use a multinomial naive Bayes model to predict the reading level. We focus on training pre-existing supervised models.

In *Readability Assessment for Text Simplification: From Analyzing Documents to Identifying Sentential Simplifications* (S. Vajjala and D. Meurers) [4], readability classification is explored in a regression model among a wide range of engineered features. Our exploration does not explore so many features, and we do not use regression analysis. Regression analysis makes it easier to determine the weight/importance of each feature, but we use more sophisticated models to focus on predictive accuracy.

Unlike any of these related works, we explore deep learning (GRU, LLM) and unsupervised learning.

Data Source

This project is based on the default Kaggle dataset. This comma-separated plaintext dataset (**WikiLarge_Train.csv**) consists of 416k sentences from English Wikipedia articles, along with labels indicating the complexity of each sentence. The sentences are preparsed into whitespace separated tokens. So, for example, "`Poanes benito Freeman, 1979--Benito's Skipper`" becomes "`Poanes benito Freeman , 1979 -- Benito 's Skipper`", with extra whitespace added between tokens.

In addition to WikiLarge_Train, there are auxiliary datasets in the Kaggle project that provide useful information about English words:

- **Dale_chall.text** - Dale-Chall list of 3000 elementary words [5].
- **Concreteness_ratings_Brysbaert_et_al_BRM.txt** - Concreteness ratings of 40,000 words [6].
- **AoA_51715_words.csv** - Age of acquisition of 51,000 words, dominant part of speech (POS), and word frequency in general usage [7].

The auxiliary datasets were joined to WikiLarge_Train by splitting each sentence on whitespace, and then looking up each word's simplicity (Dale_chall), concreteness (Brysbaert et. al.), and age of acquisition (AoA_51715_words). Lemmatization was used for word matching only when the raw word could not be matched.

Feature Engineering

Feature Engineering for Supervised Evaluation

- ***Bag of Words:*** TFIDF [20] (term frequency/inverse document frequency) vector of each sentence.
- ***Engineered Features:*** Computed from the words in the sentence without considering order or structure. These include (1) average/max age of acquisition, (2) average/ max/ standard deviation of concreteness scores, (3) average word frequency in common usage, (4) percentage of simple words, (5) count of each type of part of speech (i.e. Nouns, Verbs), (6) unique 1, 2, and 3-grams for parts of speech, (7) word count, (8) average syllables per word, and (9) count of numbers and punctuation.
- ***Parts of Speech Sequences:*** Each sentence was reduced to the dominant parts of speech for each word. For example, "Toshihide Saito is a Japanese football -LRB- soccer -RRB- defender ." becomes "ProperNoun ProperNoun Verb Article ProperNoun Noun Punct Noun Punct Noun Punct."
- ***Raw Text:*** For DistilBERT, tokenization and word embedding was left to the defaults trained for this large language model.

Feature Engineering for Unsupervised Evaluation

For the unsupervised portion, the following two groups of features were explored, vectorized (TFIDF and Count vectorization), and non-vectorized (Dale-Chall simple words, Age of Acquisition, and Concreteness ratings).

Vectorized features: TFIDF and count vectorizers [19] were used to vectorize each sentence in the WikiLarge_Train dataset.

Non-vectorized features: In order to prepare the data for non-vectorized feature extraction, each sentence in the WikiLarge_train dataset was tokenized and lemmatized using the NLTK library [8]. Additionally, each of the words contained in the supplementary datasets (Dale-Chall, AoA, Concreteness), were also lemmatized (if not already included in the dataset). The lemmatization was intended to increase the chance of matching between the main dataset and supplementary datasets. The lemmatized words from each sentence were matched to words from the supplemental datasets. Features were extracted from these matched words. Examples of features include mean AoA of all matched words in a sentence, percent of words with older AoA, percent of words with younger AoA…etc.

Supervised Learning

Methods

The working hypothesis was that sentence complexity is determined by two key dimensions, (1) **structural complexity** and (2) **content obscurity**. Structural complexity is how much structure a sentence has. So “John walked the dog” is more simple than “John walked the dog to the store”, which is more simple than “Jane said that John walked the dog to the store.” Content obscurity would be how obscure or rare each word is. So “The groveling dog threw the shoemaker’s tools out the window” would be simpler than “The obsequious canine defenestrated the cobbler’s instruments”.

Without a sophisticated parsing of sentences, the structural complexity is difficult to directly measure. However, many easy to compute metrics are reasonable proxies. These include sentence length and 2-grams on parts of speech. The parts of speech are provided by the age of acquisition and concreteness datasets. The parts of speech for each word are also used to form a new sequence, which maintains sentence structure without any content.

Content obscurity is easier to measure. The external datasets allow us to determine the age of acquisition, concreteness, usage frequency, and simplicity of each word. Bag of Words removes structure allowing models to train on content only. And the raw text representation maintains both structure and content.

While structure and content both contribute to sentence complexity, we expect these to not be independent. For example, 1st grade texts would have both simple structure and vocabulary, while graduate texts would have both more complex structure and complex words. The greatest predictive performance was obtained by training models on both structure and content.

Models and Hyperparameters:

- ***MLP*** was chosen as a sort of benchmark because of how commonly used neural networks are in machine learning.
  - hidden_layer_sizes=(20,), max_iter=50
- ***LinearSVC*** was chosen for its speed on high dimensional datasets, and to see how well a linear decision boundary would model the data. If some of the features are highly correlated to the label, then this would come out in the performance of simpler models.
  - max_iter=1000, dual=False, penalty=’l1’
- ***Random Forest*** was chosen for its speed (it can be parallelized easily), and for its ability to model highly complex non-linear decision boundaries.
  - max_leaf_nodes=25600, n_estimators=100, max_features=’sqrt’
- ***GRU*** - Both Bag of Words and the engineered features do not consider word order. That is, both transformations remain the same even if the word order of each sentence is shuffled. So GRU was used on parts of speech sequences to see how important word order is for prediction accuracy.
  - Layers are, in order, an identity embedding, GRU with 50 units, BatchNormalization, Dense(100), Dense(10), Dense(1) (output)
  - Loss is binary crossentropy, Adam optimizer, default learning rate, 7 epochs.
- ***LLM*** - Finally, a large language model (DistilBERT) was used. This LLM was pretrained on English Wikipedia text along with other data sources, making it an ideal fit for the complexity classification of English Wikipedia sentences. DistilBERT was chosen as it has similar performance but 40% fewer parameters to train than BERT.
  - Model defaults, Adam optimizer, learning rate of 2e-5, 1 epoch.

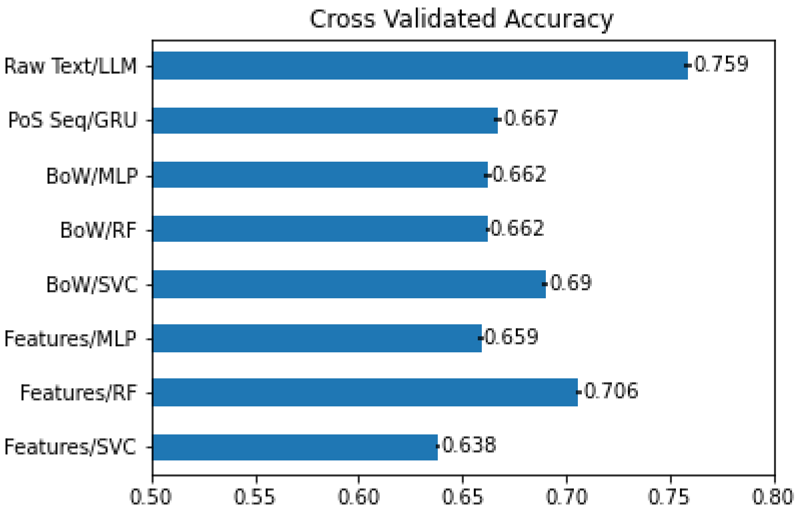
Hyperparameters for MLP, SVC, and Random Forest were chosen more for computational feasibility than predictive accuracy. MLP layer sizes and max iterations were capped to avoid excessive training time. But ad hoc experiments on Tensorflow with more and larger layers did not yield significant accuracy increases, so the limited MLP was used for overall analysis.

Scikit Learn’s SVC model can be slow, especially on high dimensions. The L1 regularization was used due to the high dimensionality, as this reduced training time and increased accuracy. L1 regularization tends to zero out features, effectively performing feature selection [21]. The dual=False was set as this is a requirement for LinearSVC when using L1 regularization. And the iterations were capped to avoid overly extended training times, but the iteration cap was not hit on standard training.

Random Forest was limited by max_leaf_nodes to avoid overfitting and decrease training time. Without a parameter limit, RF was not computationally tractable. Estimators and max features were left at their defaults. But these are explicitly called out here as they are tuned later.

No systematic hypertuning was done for GRU or LLM due to the computational intensity of these models. Reasonable defaults were chosen and the learning rate slightly tweaked for LLM.

Supervised Evaluation



Evaluation Metric

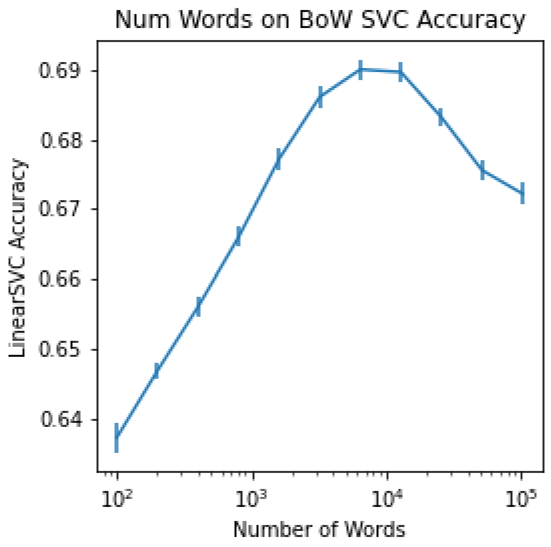
The dataset is balanced between complex and simple sentences, and the target task is to properly classify the greatest percentage of samples. So the chosen metric for evaluation was *accuracy*. Each model's accuracy was evaluated with a 5-fold cross validation. Given the amount of training data, the standard deviation of accuracy was less than 0.22% for all models.

Overall Summary

LLM performed the best overall with an accuracy of 75.9%. RF on engineered features was next, achieving 70.6% accuracy. LinearSVC on Bag of Words had 69.0% accuracy.

Feature Sensitivity

In terms of decreasing performance, the best feature representations were (1) raw text (word embeddings), (2) engineered features, (3) bag of words, and (4) parts of speech sequences. This is expected as the LLM is able to infer based on content and structure. The engineered features also give some information about content and structure. But structural information is limited since engineered features are invariant under word order permutations. Bag of words gives only content based information, which is more useful than only structural information given by the part of speech sequences.



For the **bag of words** features, LinearSVC performed the best. This is expected due to the high dimensionality of this data [17]. The higher number of dimensions means that there are more potential hyperplanes through the data space. And so there is more potential for a linear decision boundary to achieve higher accuracy. MLP and RF were impacted by the curse of dimensionality [16], but even LinearSVC was affected at higher vocabulary sizes. We see that the vocabulary size has an optimal value past which adding more words from the dataset decreases LinearSVC performance.

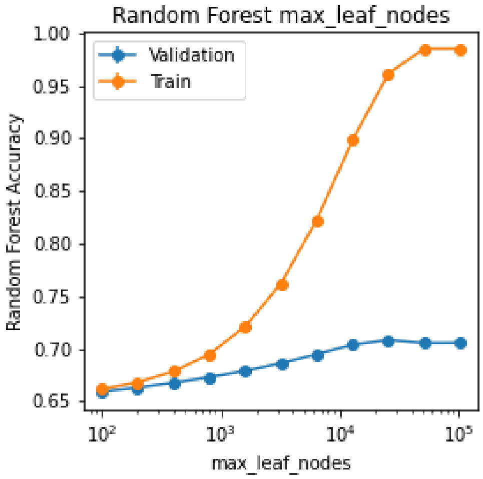
For the **engineered features**, Random Forest performed the best. This shows that the limited number of features had a highly complex decision boundary. MLP also can handle highly complex decision boundaries, but did not perform as well. MLP parameters were limited due to Scikit-Learn's software implementation inefficiencies with large models [18] and the high dimensionality of the bag of words. As RF is easily parallelizable, and somewhat resilient to high dimensionality, the number of parameters could be increased more. But ad-hoc experiments with larger MLP models on Tensorflow did not increase accuracy. The exact reason MLP did not perform as well as RF on this task would take significant effort to determine as neural networks generally lack interpretability [23], but we should not be surprised that MLP sometimes performs poorly due to the No Free Lunch Theorem [22].

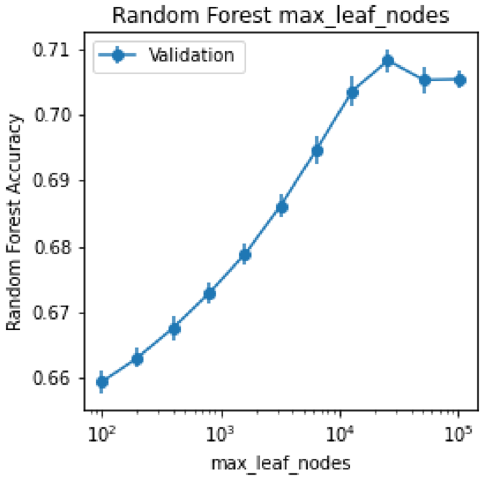
Hypertuning Sensitivity

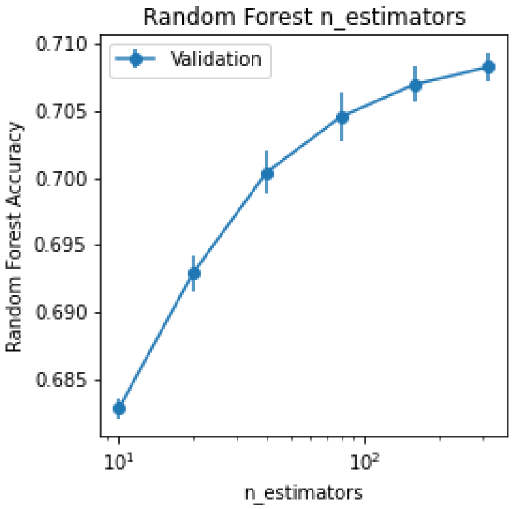
Only Random Forest was hypertuned as it was the best performing model that did not have the computational cost of deep learning. For each parameter graphed below, all parameters were held at their optimal values, except for the parameter being tuned.

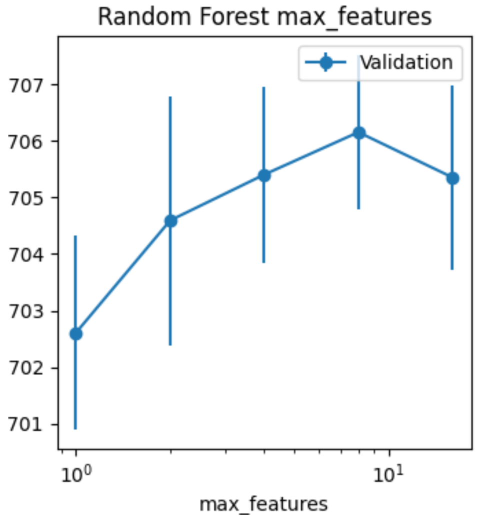
During hypertuning we see that validation accuracy only modestly increases while training accuracy rapidly approaches 1. This is only shown for max_leaf_nodes to make the minor validation accuracy changes for other parameters apparent, but all parameters exhibited similar behavior.

The most important hyperparameter was max_leaf_nodes. After 25600 leaf nodes, it starts overfitting. But the overfitting penalty is quite minor. Notably, we see that exponential increases in estimators yields a less than linear increase in validation accuracy. Changes to max features (per tree) had a very minor effect.



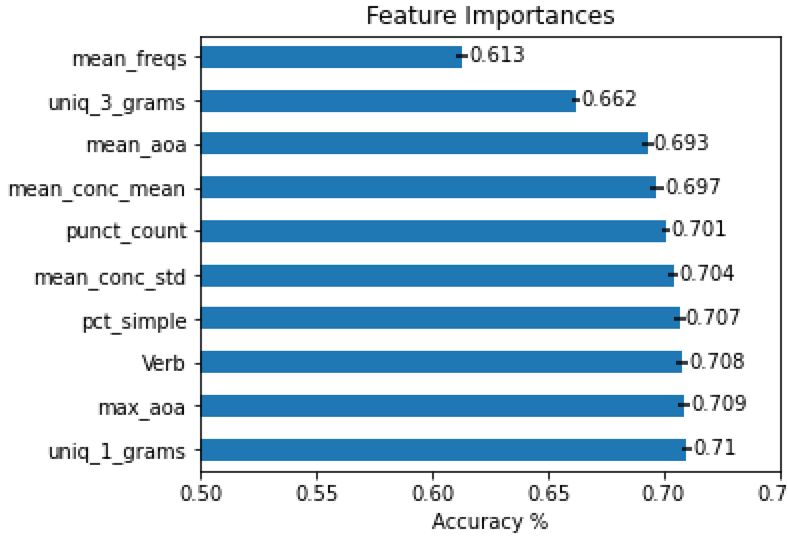




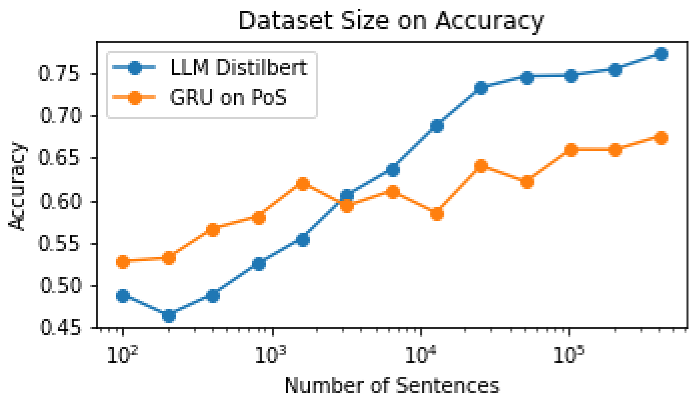


Feature Importances

Feature importance was determined additively. Starting with an empty set, the single feature with the best increase in accuracy was chosen to add. This process was repeated 10 times. Based on this procedure, we see that the most important features are frequency in common usage (mean_freq), unique parts of speech 3-grams (uniq_3_grams), and age of acquisition (mean_aoa). Other features only contributed a fraction of a percentage in accuracy each. Of the top ten engineered features, four are structural (uniq_3_grams, punct_count, Verb (count), and uniq_1_grams), and the remaining six are content based. This shows that Random Forest is relying heavily on both structural and content based data to make its predictions.



Note that when limited to 10 features, RF accuracy actually increases slightly to 71% (from 70.6%). This shows that too many features can actually decrease accuracy.

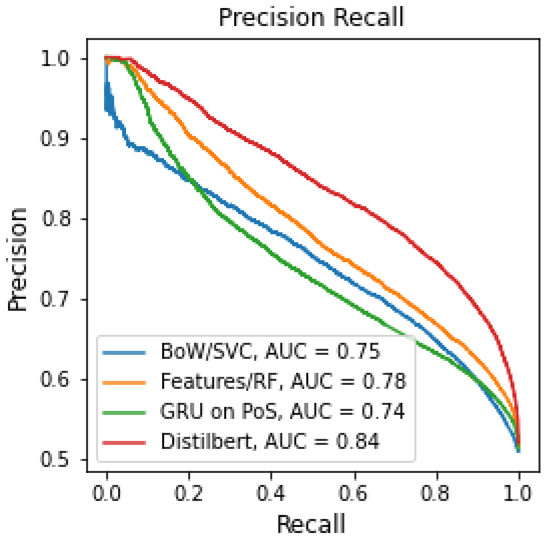


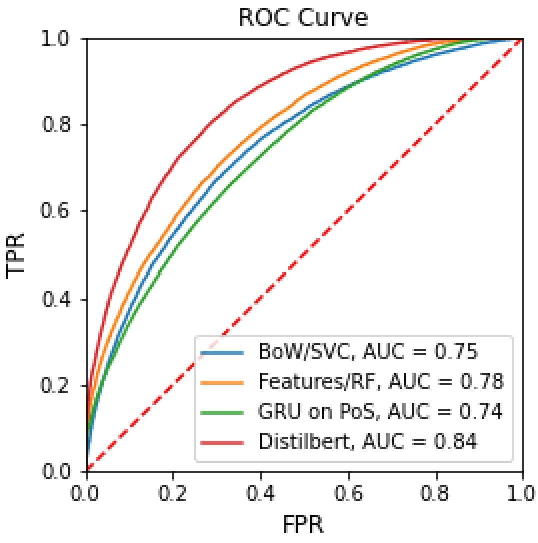
Training Data Size vs. Accuracy Tradeoff

The LLM has 67 million trainable parameters, and requires significant data to converge. The GRU model only has 36 thousand trainable parameters, and the reduction to parts of speech limits the data space. As a result, the GRU model accuracy was less responsive to additional data.

TPR/FPR and Precision/Recall Tradeoffs

Only the best models for each feature representation were graphed for clarity. Distilbert and RF on features were clearly first and second best at all decision thresholds, respectively. Overall LinearSVC on BoW outperformed GRU on PoS, but SVC could not match the precision of GRU at very high and low recall levels. Similarly, SVC had a lower TPR (true positive rate) than GRU at very low and high FPR (false positive rate). So we would prefer GRU over SVC here if high recall, high precision, high TPR, or low FPR is required.





Failure Analysis

Prediction probabilities on a hold-out evaluation set were obtained for all models, and averaged together. A prediction probability of 0 means the model has 100% confidence that the sample is simple, and a prediction probability of 1 means the model has 100% confidence that the sample is complex. Probabilities between 0 and 1 represent fractional confidence, with the decision threshold being 0.5. Error was determined by the absolute difference between the probability and the label.

The predictions with the greatest error were analyzed. It becomes immediately clear that the dataset has noise. There is no reasonable interpretation in which the first two sentences below are complex, and the next two are simple. But that's what's in the dataset.

Dataset Label	Mean Model Probability	Sentence
1	0.0465	he plays for machida zelvia .
1	0.0641	it was created in 1858 .
0	0.9091	the mongolian language -lrb- , mongol kele , cyrillic : , mongol khel -rrb- is the best-known member of the mongolic language family and the language of most of the residents of

		mongolia , where it is officially written with the cyrillic alphabet and of around three million mongolian speakers in the inner mongolia autonomous region of china , where it is officially written with the traditional mongolian script .
0	0.8945	during the war , the white ã migrã s came into contact with former soviet citizens from german-occupied territories who used the german retreat as an opportunity to flee from the soviet union or were in germany and austria as pows and forced labourers and preferred to stay in the west , often referred to as the second wave of ã migrã s -lrb- often also called dps - displaced persons , see displaced persons camp -rrb- .

Without relabeling the entire dataset, it would be difficult to measure how pervasive the noise is. But it would be worth further investigation.

Next are predictions with the greatest variance between models, as measured by the standard deviation of their prediction probabilities.

Ref	label	Features on RF	SVC on BoW	LLM	GRU on PoS	Sentence
(a)	1	0.573	0.860	0.958	0.003	paolo conti -lrb- born 1 april 1950 -rrb- is a former italian football goalkeeper .
(b)	1	0.817	0.786	0.949	0.015	link
(c)	0	0.657	0.234	0.003	0.945	is a 1982 platforming video game made by activision for atari 2600 , atari 5200 , atari 8-bit , colecovision , commodore 64 , intellivision and the sega sg-1000 .
(d)	0	0.034	0.863	0.018	0.014	1972 - pavel nedved , czech footballer
(e)	1	0.508	0.188	0.991	0.046	c.
(f)	0	0.508	0.188	0.974	0.046	s.
(g)	1	0.508	0.188	0.970	0.046	d.
(h)	0	0.752	0.309	0.027	0.918	james francis cagney , jr. -lrb- july 17 , 1899 -- march 30 , 1986 -rrb- was an american movie actor who became very famous for many roles in his long career , and won the oscar for best actor in 1942 for his role in yankee doodle dandy .

We see some interesting divergence between GRU and LLM. In (a), there isn't much structure to this sentence, which explains GRU confidence in its simplicity. But LLM and SVC are confident in its complexity. Perhaps the date of birth put in brackets is a signal of complexity from the training set. For (b), the word "link" is clearly a marker for complexity in the training set, as the BoW (SVC) model gets this correct, while GRU sees only a single noun and confidently predicts simplicity. Sentence (c) is an interesting failure mode for GRU. A long list of nouns is not necessarily complex, which LLM detects. Sentence (d) shows a significant misprediction for BoW (SVC) as either "czech" or "footballer" must be significantly correlated with complexity in the training set. Sentences (e), (f), and (g) are single letter noise. It's unclear why "c." would be complex and "s." simple. Sentence (h) is interesting in that LLM matches the dataset label so confidently, but it seems like this sentence should have been classified as complex. Data leakage or memorization seems unlikely as LLM got (f) incorrect. The words in the sentence are relatively simple, but the structure is not. More investigation would be needed here.

Future Improvements

LLM was correct on all the GRU failures. Reducing dataset noise, such as single letter or single word sentences would help. But improved part of speech tagging would also help GRU. See *Solution Extensions* in the discussion section below.

Unsupervised Learning

Methods

Algorithms
The main unsupervised algorithms explored were PCA and K-means clustering. For the classification of sentence complexity, there would be a high dimensional set of features. A common method for processing high dimensional features is dimensional reduction with PCA. K-means clustering is a common unsupervised clustering algorithm used to process unlabeled data. Both PCA and K-means were reasonable approaches to analyzing the Wiki classification data. Each K-means analysis consisted of ten iterations, the mean and standard deviation of the results were reported. Initial K-means parameters were set to **n_clusters = 2, max_iter = 100, n_init = 5**. Further hyperparameter tuning was performed later in the process.

Evaluation Metric
Accuracy score was chosen to evaluate the performance of a model or feature set. In a typical unsupervised learning situation, the dataset does not contain labels. Therefore it is not possible to calculate an accuracy score since there are no true labels to compare with the model predicted labels. However, this dataset does contain labels that need to be removed when fitting the unsupervised model. Once the unsupervised model has produced predicted labels, these labels can then be compared with the original labeled dataset to calculate an accuracy score. The method used for accuracy score was **accuracy_score(***y_true, y_pred***)** from SKLearn library. Each accuracy score reported is the mean and standard deviation of the results of ten iterations of the K-means clustering algorithm. Once the highest accuracy scoring algorithm is determined, a confusion matrix will be generated to further understand the results.

Results

Initial feature analysis
Having established the algorithm and evaluation metric, the performance of various features was explored.
The initial comparison was between two broad categories of features, vectorized versus non-vectorized (refer to Feature Engineering Section for more details of features). Results of K-means clustering accuracy scores are shown in the table below.

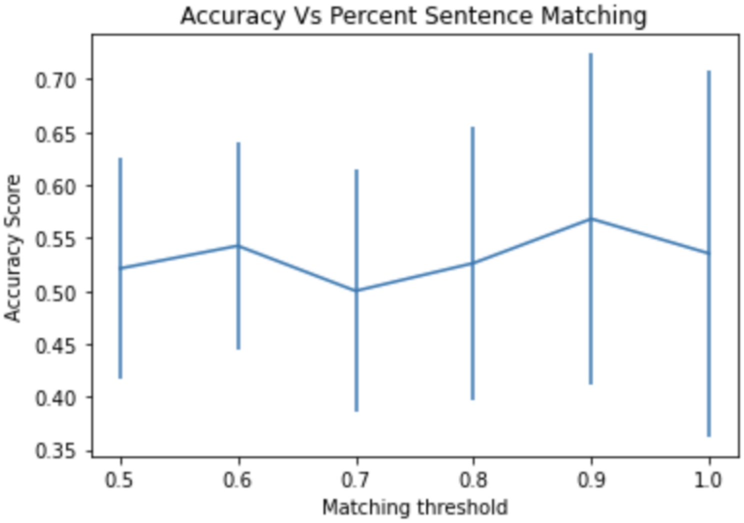
Feature	TFIDF Vectorizer	Count Vectorizer	Non-vectorized features
Accuracy Score	0.511 ± 0.011	0.498 ± 0.028	0.539 ± 0.088

The vectorized accuracy scores were low, around 50%, meaning that they are no better than random chance at correctly labeling a sentence. The non-vectorized features had a slightly better accuracy score. Non-vectorized features were further examined to see if improvements could be made to the accuracy score.

Accuracy Score by Matching Threshold
Since the non-vectorized features were largely derived from supplemental datasets such as AoA, Concreteness rating, and Dale-Chall, the amount of information available for each sentence would depend on the overlap between the words in a sentence and the words in the supplemental datasets. For example, if a sentence had 10 words and only five of them were contained in the supplemental datasets, then only 50% of the sentence information would be available for analysis. This could certainly impact the accuracy of the algorithm. An analysis of the Wiki sentences showed the following breakdown at various matching thresholds.

% Match	>= 50%	>= 60%	>= 70%	>= 80%	>= 90%	>= 100%
# of Sentences	334,937	263,867	193,787	130,348	68,551	51,210

Intuitively, it can be assumed that the higher the percentage, the better chance of being classified accurately. This hypothesis was tested by evaluating the K-means clustering accuracy of the Wiki sentence dataset at each threshold.



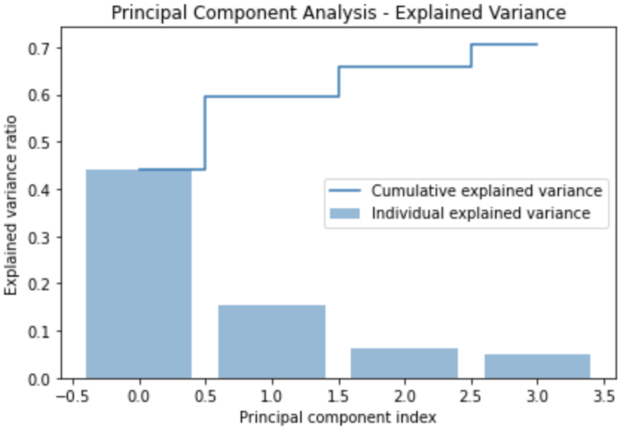
It was surprising to see that there was no clear trend of improved accuracy as the matching threshold increased. Particularly unexpected was the 100% matched sentences had a lower accuracy score compared to the 90% matching threshold. Even though the graph did not show a conclusive trend, the decision was made to follow the intuitive assumption that those sentences that were 100% matched to the supplemental datasets would give the most accurate results. This is the dataset that is used for all upcoming analyses.

Effect of Scaling Data

Up until this point, raw data features have been used in the analyses. However, scaling is important when there are different units that can lead to large numeric differences between features. In this dataset, the units were counts and percentages. By definition, percentages are numerically smaller than counts. Scaling data may help improve accuracy scores. Two common scalers, StandardScaler and MinMaxScaler, were applied to the data. The following table shows the accuracy scores of the 100% Matched dataset.

Data Type	Raw data	Standard scaled	Min Max scaled
Accuracy Score	0.535 ± 0.173	0.607 ± 0.143	0.511 ± 0.054

The StandardScalar led to a noticeable improvement in the accuracy score. Conversely, the MinMaxScaler led to a decrease in accuracy score. StandardScaler scales the mean of the data to 0 and each data point to corresponding unit variance. MinMaxScaler scales all data between a range [0:1], this is useful in situations where the boundaries of data are well established. For this dataset, there are no well established boundaries for the features.



# of Principal Components	**Accuracy Score**
1	0.571 ± 0.162
2	0.571 ± 0.164
3	0.643 ± 0.107
4	0.464 ± 0.175

Principal Component Analysis

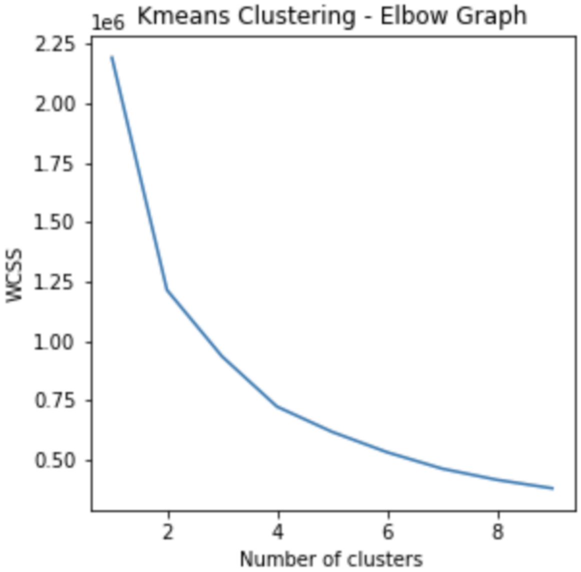
One objective for this analysis was to understand the impact of PCA on the accuracy of K-means clustering. PCA was performed on the standard scaled features from the 100% matched dataset. Four components account for approximately 70% of the variance. K-means clustering was fitted on 1 to 4 components to evaluate accuracy scores.

A large improvement in accuracy score was seen when K-means was fitted with three principal components (accuracy = 0.643 ± 0.107). Increasing the number of principal components beyond three did not lead to an increase in accuracy.

Hyperparameter Tuning

The model that gave us the highest accuracy score of 0.643 ± 0.107 was K-means clustering on three principal components of StandardScaled features. The K-means clustering algorithm has hyperparameters that can be tuned. The following section will evaluate three hyperparameters **n_clusters, max_iter,** and **n_init**.

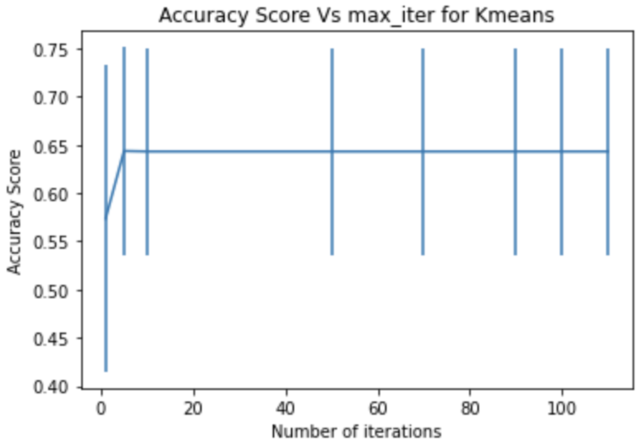
N_clusters: Up to this point, the K-means algorithm had been set to k=2 clusters. This was because the original dataset was binary classified. In order to determine if k=2 clusters was indeed the optimal number of clusters, the Elbow Method and Silhouette Coefficients were used.

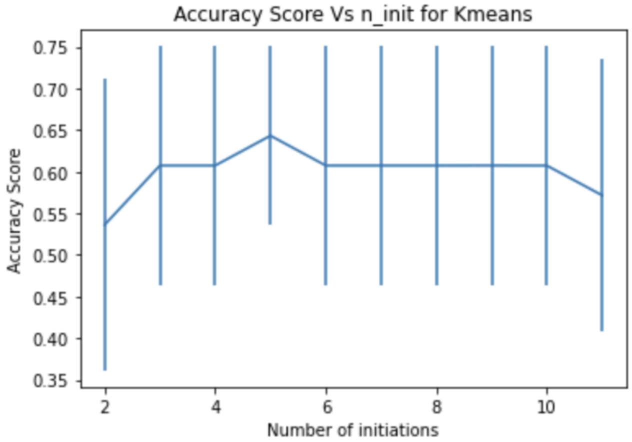


K-means Clusters	Silhouette Coefficient
2	0.394 ± 0.004
3	0.336 ± 0.005
4	0.322 ± 0.004

The Elbow Graph showed an inflection point between 2 to 4 clusters. The Silhouette Coefficient for 2 clusters was the highest and confirmed the use of k=2 for the K-means clustering algorithm.

Max_iter and N_init: The results of hypertuning for max_iter and n_init also confirmed the initial values set for these parameters. From the graphs, it can be seen that the accuracy score plateaus when max_iter reaches 5. Therefore the initial value of 100 is well above threshold. Further increase in max_iter would not have led to increases in accuracy. For n_init, the accuracy peaks at n_init = 5. This also confirms the initial parameter that was set.

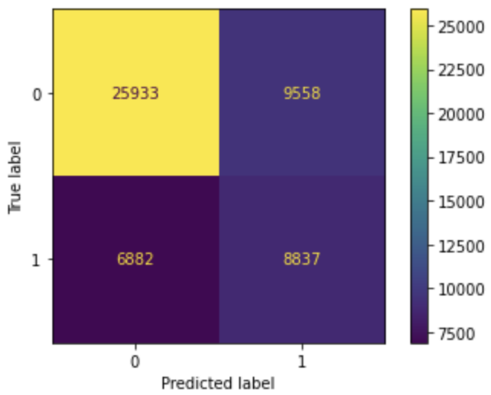




Based on the results of the hyperparameter analyses, no adjustments were needed for the original K-means parameters. The result of the unsupervised analysis was K-means clustering (k=2, max_iter=100, n_init = 5) fitted to three PCA components of standard scaled non-vectorized features, producing the highest accuracy score of 0.643 ± 0.107.

Confusion Matrix

To gain further insights into how the clustering is classifying the sentences, a confusion matrix was generated. The algorithm has moderate success in correctly predicting sentences that do not need to be simplified (0). However, the algorithm has difficulty predicting when a sentence does need to be simplified (1).



Error Analysis

A detailed inspection of specific errors revealed certain patterns. The two sentences that were misclassified as not needing to be simplified, were significantly shorter in length than the other complex sentences.

Sentence	True Label	Pred Label
This marked the first motorcycle racing event at the facility since its first month of operation , in August 1909	1	1
A very wide covered footbridge joins all platforms at their western ends but does not provide entry to or egress from the station	1	1
He has subsequently written a further nine plays	1	0
a family of	1	0
Constitutional monarchy -- A government that has a monarch , but one whose powers are limited by law or by a formal constitution	1	1

The two sentences misclassified as needing to be simplified, were significantly longer than the other simple sentences.

Sentence	True Label	Pred Label
He is one of the most successful drivers in the history of the Champ Car World Series . He won four championships in a row , from 2004 to 2007	0	1
Their hair may be curly , wavy , or almost straight	0	0
Kimonos are very expensive	0	0
Cell theory says that the cell is the basic unit of life	0	0
The words were written and sung by The Edge	0	0
Battleship	0	0
Major League Soccer : 2	0	0
He also worked with Timothy Leary to promote futurist ideas of space migration , life extension , and intelligence enhancement	0	1

Discussion of Supervised Learning

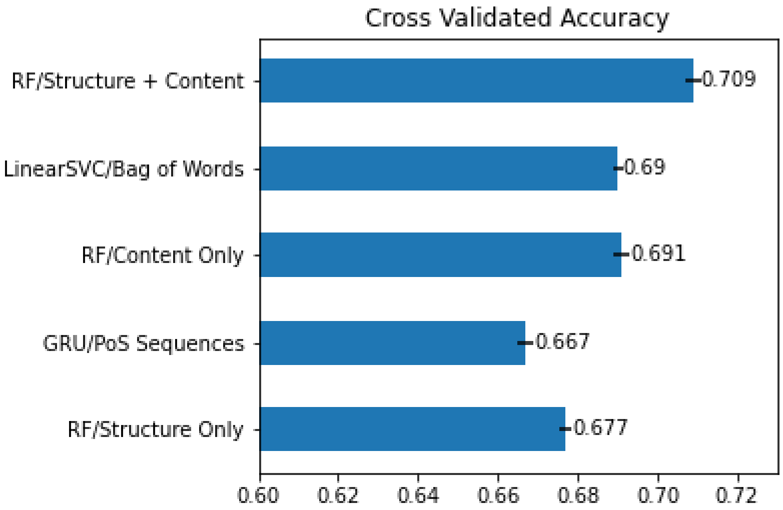
In the supervised learning experiments, we learned that there are two important feature dimensions: **content** and **structure**. These are highly correlated, but feature importances show that both are necessary for optimal prediction. The best feature representations were, respectively:

1) Content and Full Structure (Distilbert)
2) Content and Partial Structure (Random Forest on Engineered Features)
3) Content only (LinearSVC on Bag of Words)
4) Structure only (GRU on Parts of Speech Sequences)

The engineered features only partially capture structure because they are invariant under word order changes. This shows that there is a tradeoff between the amount of information available in the engineered features and model performance.

Looking specifically at the engineered features, we can divide these into structure and content. The structural features include the counts for each part of speech or token type (numbers, punctuation), the total word count, and parts of speech n-grams. The content features include information about words: age of acquisition, concreteness, simplicity, number of syllables, and frequency in common usage.

When limiting Random Forest to content or structural features only, we see a similar trend. RF trained on content features only has a predictive accuracy that almost exactly matches LinearSVC trained on bags of words. These models are very different, but both are trained on content features only. Similarly, RF trained on structural features has a predictive accuracy that is only 1% better than GRU trained on parts of speech sequences. Again, the commonality of these very different models is that they are both trained on structural features only.



While models trained on structural features can achieve accuracy 17% above random guessing, and models trained on content features can do 19%, content and structure are highly correlated. Otherwise combining these features would achieve a much higher accuracy. But even the best model, LLM trained on word embeddings, can only achieve an accuracy of 26% above random guessing.

Surprising Results

There were several surprises encountered while training supervised models. The first is how difficult the training task was. We expected simple features like sentence length to be more highly correlated with sentence complexity. The second is training data noise. There were many unmeaningful sentences consisting of a single word, letter, or punctuation mark. And when inspecting the worst mispredictions, it's clear that some of the dataset is mislabeled. The third surprising result is that content was more predictive of sentence complexity than structure.

Challenges

The biggest challenge was simply setting up Tensorflow to train GRU and Distilbert. This was surprisingly difficult. It proved to be easier to download and run a docker image for Tensorflow, despite no previous familiarity with docker. Another challenge was what to do with unknown words, since part of speech information is not available for every person, place, or thing referenced in the dataset. We decided to treat title-case words as proper nouns, and otherwise unmatched words would be classified as unknown.

Solution Extensions

The dominant parts of speech used for GRU training are static for each word. But words can participate in multiple parts of speech. For example, “link” can refer to the act of linking things together or the product of this act, such as a link in a chain. LLMs can be trained to tag words in sentences. Here is an example for named entity recognition tags, but similar training can tag parts of speech. This would also allow properly tagging more obscure words such as lesser known species names. With these more correctly tagged parts of speech, GRU could be trained again. This would tell us how much predictive accuracy full structure (order and length) would provide.

Discussion of Unsupervised Learning

Overall an accuracy score of 0.643 ± 0.107 was achieved utilizing PCA and K-means clustering with standard scaled non-vectorized features. This indicates a probability of correct classification better than random chance, however, it may not be useful in a real world scenario.

In the analysis of accuracy by matched thresholds, it was surprising to see that there was a drop in accuracy from 90% matched to 100% matched. Intuitively, we would assume that 100% matched sentences would have more information per sentence and therefore be able to be more accurately labeled.

The confusion matrix and error analysis were particularly insightful in understanding the nature of misclassifications. Based on the confusion matrix, it appears that the features extracted were more oriented towards determining whether a sentence was a simple sentence, and were not useful in determining if a sentence was a complicated sentence.

Error analysis revealed that an overarching rule followed by the algorithm could be summarized as long sentence = complex sentence and short sentence = simple sentence. The challenge is correctly classifying short complex sentences and long simple sentences.

The error analysis also uncovered potential data quality issues. For example, one error was the misclassification of the sentence “*a family of*”. This was given a true label of 1 in the dataset, meaning that it was a complex sentence needing to be simplified. In this instance, it could be argued that the predicted label of 0, meaning it is a simple sentence, is actually a more accurate classification. Conversely, another error was the misclassification of the sentence “*He also worked with Timothy Leary to promote futurist ideas of space migration , life extension , and intelligence enhancement*”. This was given a true label of 0 in the dataset, meaning that it was a simple sentence. In this instance, it could also be argued that the predicted label of 1, is a more accurate label for this sentence.

Further improvements

Future analyses could include the exploration of combining vectorized and non-vectorized features into a single dataset. It would be interesting to see if the combination of these two types of features would result in any improvements in accuracy. Further work on features to address the area of challenge (short complex sentences and long simple sentences) would also lead to improvement. Another area that could improve accuracy would be to find additional supplemental data that could be used to match more words in the Wiki dataset. Only 30% of the sentences had an 80% or more match to the supplemental datasets. Finally, a detailed manual review of the dataset could be warranted as there were some instances where the true labels did not seem to be accurate.

Ethical Considerations

The ethical considerations for supervised and unsupervised learning are similar, as the target objective in both cases was to improve the classification of text complexity.

Text classification has ethical considerations that relate to literacy and education equity. With the current level of accuracy achieved in this analysis, the deployment of this algorithm into real world settings could lead to frustration on the part of learners. This is not inconsequential considering the empowerment that literacy provides an individual in society. A learner of language seeking to improve their literacy could miss materials that would have benefitted him or her, but was misclassified as too complex. The reverse scenario would be just as detrimental, to have materials provided that were classified as simple but in reality were complex. This could lead to frustration, discouragement, and other negative consequences for the learner.

Algorithms trained on one corpus could be especially problematic when applied to other corpora. In this specific case, Wikipedia text is different in kind from most grade-school materials, such as elementary school reading books and middle school geometry. This may lead to misclassification of these texts, further complicating the ethical applications of these algorithms in regard to literacy and education equity.

Statement of Work

Tony Lan
Unsupervised analyses of Wikipedia dataset and report writing relevant to these analyses.

Feature engineering of the following features:
'Nletters_abv_mid_count', 'Nletters_blw_mid_count', 'Nletters_hi_count', 'Nletters_lo_count', 'Nphon_abv_mid_count', 'Nphon_blw_mid_count', 'Nphon_hi_count', 'Nphon_lo_count', 'AoA_abv_mid_count', 'AoA_blw_mid_count', 'AoA_hi_count', 'AoA_lo_count', 'Conc_abv_mid_count', 'Conc_blw_mid_count', 'Conc_hi_count', 'Conc_lo_count', 'Simple_word_count', 'Total_matched', 'Article', 'Noun', 'Verb', 'Name', 'Adjective', 'Adverb', 'Preposition', 'Abbreviation', 'Interjection', 'Determiner', 'Conjunction', 'Unclassified', 'Pronoun', '%_Nletters_abv_mid_count', '%_Nletters_blw_mid_count', '%_Nletters_hi_count', '%_Nletters_lo_count', '%_Nphon_abv_mid_count', '%_Nphon_blw_mid_count', '%_Nphon_hi_count', '%_Nphon_lo_count', '%_AoA_abv_mid_count', '%_AoA_blw_mid_count', '%_AoA_hi', '%_AoA_lo', '%_Conc_hi_count', '%_Conc_lo_count', '%_Conc_abv_mid_count', '%_Conc_blw_mid_count', '%_Simple_word_count', '%_Total_matched', 'Freq_abv_mid_count', 'Freq_blw_mid_count', 'Freq_hi_count', 'Freq_lo_count', '%_Freq_abv_mid_count', '%_Freq_blw_mid_count', '%_Freq_hi_count', '%_Freq_lo_count', '%_Article', '%_Noun', '%_Verb', '%_Name', '%_Adjective', '%_Adverb', '%_Preposition', '%_Abbreviation', '%_Interjection', '%_Determiner', '%_Conjunction', '%_Unclassified', '%_Pronoun', ‘Total_matched’, ‘%_Total_matched’

Thanuja Stewart
Supervised analyses of Wikipedia dataset and report writing relevant to these analyses. This included Tensorflow/Keras modeling of GRU and training a pre-built LLM through Huggingface. I also contributed to feature engineering, reimplementing a subset of the same features in a parallel, but duplicative effort.

References

1. Sanh, Victor & Debut, Lysandre & Chaumond, Julien & Wolf, Thomas. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.
2. Kurdi, Mohamed. (2020). Text Complexity Classification Based on Linguistic Information: Application to Intelligent Tutoring of ESL.
3. Collins-Thompson, Kevyn & Callan, James. (2004). A Language Modeling Approach to Predicting Reading Difficulty.. HLT-NAACL.. 193-200.
4. V., Sowmya & Meurers, Detmar. (2014). Readability Assessment for Text Simplification: From Analyzing Documents to Identifying Sentential Simplifications. International Journal of Applied Linguistics, Special Issue on Current Research in Readability and Text Simplification. 165. 10.1075/itl.165.2.04vaj.
5. Dale E; Chall J (1948). "A Formula for Predicting Readability". Educational Research Bulletin. 27: 11–20+28.
6. Brysbaert, M., Warriner, A.B., & Kuperman, V. (2014). Concreteness ratings for 40 thousand generally known English word lemmas. Behavior Research Methods, 46, 904-911.
7. Crr » Age-of-Acquisition (AoA) Norms for over 50 Thousand English Words. http://crr.ugent.be/archives/806. Accessed 27 Feb. 2023.
8. NLTK :: Natural Language Toolkit. https://www.nltk.org/. Accessed 27 Feb. 2023.
9. “Sklearn.Svm.LinearSVC.” *Scikit-Learn*, https://scikit-learn/stable/modules/generated/sklearn.svm.LinearSVC.html. Accessed 27 Feb. 2023.
10. “Sklearn.Neural_network.MLPClassifier.” *Scikit-Learn*, https://scikit-learn/stable/modules/generated/sklearn.neural_network.MLPClassifier.html. Accessed 27 Feb. 2023.

11. “Sklearn.Ensemble.RandomForestClassifier.” *Scikit-Learn*, https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Accessed 27 Feb. 2023.
12. Team, Keras. *Keras Documentation: GRU Layer*. https://keras.io/api/layers/recurrent_layers/gru/. Accessed 27 Feb. 2023.
13. *Text Classification*. https://huggingface.co/docs/transformers/tasks/sequence_classification. Accessed 27 Feb. 2023.
14. “Sklearn.Cluster.KMeans.” *Scikit-Learn*, https://scikit-learn/stable/modules/generated/sklearn.cluster.KMeans.html. Accessed 27 Feb. 2023.
15. “Sklearn.Decomposition.PCA.” *Scikit-Learn*, https://scikit-learn/stable/modules/generated/sklearn.decomposition.PCA.html. Accessed 27 Feb. 2023.
16. “Curse of Dimensionality.” *Wikipedia*, 29 Jan. 2023. *Wikipedia*, https://en.wikipedia.org/wiki/Curse_of_dimensionality. Accessed 27 Feb. 2023.
17. “1.4. Support Vector Machines.” *Scikit-Learn*, https://scikit-learn/stable/modules/svm.html. Accessed 27 Feb. 2023.
18. “1.17. Neural Network Models (Supervised).” *Scikit-Learn*, https://scikit-learn/stable/modules/neural_networks_supervised.html. Accessed 27 Feb. 2023.
19. “Sklearn.Feature_extraction.Text.CountVectorizer.” *Scikit-Learn*, https://scikit-learn/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html. Accessed 27 Feb. 2023.
20. “Sklearn.Feature_extraction.Text.TfidfVectorizer.” *Scikit-Learn*, https://scikit-learn/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. Accessed 27 Feb. 2023.
21. “1.1. Linear Models.” *Scikit-Learn*, https://scikit-learn/stable/modules/linear_model.html#lasso. Accessed 27 Feb. 2023.
22. Wolpert, D.H., Macready, W.G. (1997), "No Free Lunch Theorems for Optimization", IEEE Transactions on Evolutionary Computation 1, 67.
23. Zhang, Yu & Tino, Peter & Leonardis, Ales. (2020). A Survey on Neural Network Interpretability.

Appendix: Engineered Features

Feature	Unsupervised?	Supervised?	Description
word_count_x	✓	✓	Count of words in sentence
num_count	✓	✓	Count of numbers in sentence
punct_count	✓	✓	Number of punctuations in a sentence
uniq_1_grams		✓	Number of unique parts of speech
uniq_2_grams		✓	Number of unique 2 grams in a sentence
uniq_3_grams	✓	✓	Number of unique 3 grams in a sentence
mean_syll	✓	✓	The mean number of syllables of words in a sentence
mean_aoa	✓	✓	The mean Age of Acquisition of words in a sentence that matched to AoA dataset.
max_aoa		✓	The max Age of Acquisition of words in a sentence that matched to AoA dataset.
mean_conc_mean	✓	✓	The mean concreteness score of words in a sentence that matched to Concreteness dataset
max_con_mean		✓	The max concreteness score of words in a sentence that matched to Concreteness dataset
mean_conc_std	✓	✓	The mean standard deviation of concreteness scores of words in a sentence that matched to Concetness dataset
word_count_y	✓		Count of words in sentence after removing stopwords, tokenizing, and lemmatizing
Nletters_abv_mid_count	✓		The number of AoA matched words in a sentence that have number of letters above the median number of letters in AoA dataset
Nletters_blw_mid_count	✓		The number of AoA matched words in a sentence that have number of letters below the median number of letters in AoA dataset
Nletters_hi_count	✓		The number of AoA matched words that have number of letters greater than the 75% of number of letters in AoA dataset
Nletters_lo_count	✓		The number of AoA matched words that have number of letters less than the 25% of number of letters in AoA dataset
Nphon_abv_mid_count	✓		The number of AoA matched words that have number of phonemes above median number of phonemes in AoA dataset
Nphon_blw_mid_count	✓		The number of AoA matched words that have number of phonemes below median number of phonemes in AoA dataset

Nphon_hi_count	✓		The number of AoA matched words that have number of phonemes greater than the 75% of number of phonemes in AoA dataset
Nphon_lo_count	✓		The number of AoA matched words that have number of phonemes less than the 25% of number of phonemes in AoA dataset
AoA_abv_mid_count	✓		The number of AoA matched words that have AoA above median AoA in AoA dataset, per AoA_Kup_lem value.
AoA_blw_mid_count	✓		The number of AoA matched words that have AoA below median AoA in AoA dataset, per AoA_Kup_lem value.
AoA_hi_count	✓		The number of AoA matched words that have AoA greater than the 75% of AoA in the AoA dataset, per AoA_Kup_lem value.
AoA_lo_count	✓		The number of AoA matched words that have AoA less than the 25% of AoA in the AoA dataset, per AoA_Kup_lem value.
Conc_abv_mid_count	✓		The number of Concreteness matched words that have scores above median of Concreteness dataset
Conc_blw_mid_count	✓		The number of Concreteness matched words that have scores below median of Concreteness dataset
Conc_hi_count	✓		The number of Concreteness matched words that have scores greater than the 75% of scores in the Concreteness dataset.
Conc_lo_count	✓		The number of Concreteness matched words that have scores less than the 25% of scores in the Concreteness dataset.
Simple_word_count	✓		Count of simple words in a sentence
pct_simple		✓	Percentage of words in a sentence that are simple
Freq_abv_mid_count	✓		The number of AoA matched words that have frequency scores above median frequency score of AoA dataset.
Freq_blw_mid_count	✓		The number of AoA matched words that have frequency scores below median frequency score of AoA dataset.
Freq_hi_count	✓		The number of AoA matched words that have frequency scores greater than the 75% frequency score of the AoA dataset
Freq_lo_count	✓		The number of AoA matched words that have frequency scores less than the 25% frequency score of the AoA dataset
Article	✓	✓	The number of AoA matched words that were classified as article
Noun	✓	✓	The number of AoA matched words that were classified as noun
Verb	✓	✓	The number of AoA matched words that were classified as verb
Name	✓		The number of AoA matched words that were classified as name
Adjective	✓	✓	The number of AoA matched words that were classified as adjective
Adverb	✓	✓	The number of AoA matched words that were classified as adverb
Preposition	✓	✓	The number of AoA matched words that were classified as preposition

Abbreviation	✔		The number of AoA matched words that were classified as abbreviation
Interjection	✔		The number of AoA matched words that were classified as interjection
Determiner	✔	✔	The number of AoA matched words that were classified as determiner
Conjunction	✔	✔	The number of AoA matched words that were classified as conjunction
Unclassified	✔	✔	The number of AoA matched words that were classified as unclassified
Pronoun	✔	✔	The number of AoA matched words that were classified as pronouns
Total_matched	✔		Total number of words in a sentence that matched to any of the words in the supplemental datasets. This was only used to stratify the dataset and was removed from the analysis.
Percentages of count features	✔		The corresponding percentages for each of the above features relating to counts
mean_freqs		✔	Mean frequency of usage in English for words of sentence
mean_pct_known		✔	Mean percentage of people that know each word