Predicting Text Difficulty

Matthew Zimolzak, Thomas Friss, Josh Koboldt
University of Michigan

Author Note:

This project was developed in fulfillment of an educational requirement as specified by the SIADS 694/695 milestone course offered through the Master of Applied Data Science program by the School of Information at the University of Michigan. The authors Matthew Zimolzak, Thomas Friss, and Josh Koboldt may be contacted with inquiries via email at zimolzak@umich.edu, tfriss@umich.edu, and jkoboldt@umich.edu and respectively.

Table of Contents

Motivation

In many real-world applications, there is a need to make sure textual information is comprehensible/readable by audiences who may not have high reading proficiency. This might include students, children, adults with learning/reading difficulties, and those who have English as a second language. The simple Wikipedia (https://en.wikipedia.org/wiki/Simple_English_Wikipedia), for example, was created exactly for this purpose. Before the editors spend a lot of effort to simplify the text and increase its readability, it would be very useful to suggest to them which parts of an article's text might need to be simplified. We approach this issue as a binary classification problem – whether a sentence needs to be simplified or not – which inherently falls under the scope of supervised learning. In an attempt to bolster our efforts, we also employ a few unsupervised learning techniques with the ultimate goal of improving classification accuracy.

Data Source

Throughout the course of this project, we utilize three datasets which are detailed below.

WikiLarge_Train.csv:

- As provided by the course instructional team, this csv file contains 416,768 records with the following variables of interest:
  - *id* – a unique integer identifier for each record
  - *text* – a free-form sentence extracted from a Wikipedia article in the form of a string
  - *label* – the class label of the text
    - 0: the sentence **does not** need to be simplified
    - 1: the sentence **does** need to be simplified

dale_chall.txt:

- This is the *Dale Chall 3000 Word List*, which is one definition of words that are considered "basic" English. A summary is at https://www.readabilityformulas.com/articles/dale-chall-readability-word-list.php.

AoA_51715_words.csv:

- This file csv contains "Age of Acquisition" (AoA) estimates for about 51k English words, which refers to the approximate age (in years) when a word was learned. Early words, being more basic, have lower average AoA. The two main variables of interest are as follows:
  - *Word* – the word in question
  - *AoA_Kup_lem* – Estimated AoA based on Kuperman et al. study lemmatized words.

The datasets are utilized in both components of the project but the unsupervised learning portion relies mainly on the WikiLarge_Train.csv dataset.

Supervised Learning

Methods and Evaluation

We explore a variety of feature representations and models. To start, we begin by tokenizing the text of each sentence using regular expressions. One tokenization was performed on word characters only while the other was performed on any non-whitespace character. Once tokenized, the tokens were then vectorized so that we could use them as input for our selected models. Four different vectorizations were used: simple term frequency (TF), term frequency-inverse document frequency (TF-IDF), TF-IDF 2-grams and 3-grams at the word level, and TF-IDF 2-grams and 3-grams at the character level. In some cases, English stop words were removed while in others they were not. In most cases, the vocabulary was built on the entire corpus with the one exception being a vocabulary established by the *Dale Chall 3000 Word List*. Other features considered include the proportion of sentence words present in the *Dale Chall 3000 Word List*, the proportion of sentence words present in the AoA file, and the AoA sum of sentence words (where individual AoA is given by the variable *AoA_Kup_lem)*.

In classifying the sentences, the models used are as follows: uniform dummy, uniform most frequent, logistic regression (LR), multinomial naive bayes (MNB), and random forests (RF). The first three are intended to be used as baseline classifiers while the latter two represent more sophisticated classification models. Initially, no hyperparameter tuning was performed on any of the models – we simply trained the models on the input features and computed classification accuracy using a 75-25 train-test split. The results are as follows:

	wc w/o stop	ws w/o stop	wc w/ stop	ws w/ stop	DC_3000	ws w/ stop + dc_prop	ws w/ stop + dc_prop + AoA
Random	0.49888	0.49888	0.49888	0.49888	0.49888	0.49888	0.49888
Most Frequent	0.49825	0.49825	0.49825	0.49825	0.49825	0.49825	0.49825
LR, count	0.65809	0.68131	0.67456	0.69370	0.65119	0.69599	0.70217
LR, TF-IDF	0.66298	0.68611	0.68529	0.70284	0.65132	0.70292	0.70102
LR, TF-IDF n-gram	0.54527	0.70561	0.65906	0.71829	0.49825	0.71768	0.71714
LR, TF-IDF n-gram characters	0.70651	0.70651	0.70651	0.70651	0.70651	0.70639	0.70479
MNB, count	0.57572	0.59470	0.59480	0.60961	0.61876	0.61139	0.62157
MNB, TF-IDF	0.57698	0.59158	0.59507	0.60896	0.62005	0.60947	0.64220
MNB, TF-IDF n-gram	0.51056	0.62069	0.57544	0.63547	0.49825	0.63619	0.60668
MNB, TF-IDF n-gram characters	0.63106	0.63106	0.63106	0.63106	0.63106	0.63176	0.62990
RF, count	0.62143	0.64437	0.64000	0.65057	0.63775	0.65327	0.65725
RF, TF-IDF	0.62622	0.65555	0.64804	0.65723	0.63796	0.66578	0.66315
RF, TF-IDF n-gram	0.53893	0.66884	0.60195	0.66261	0.49825	0.66459	0.65911
RF, TF-IDF n-gram characters	0.68582	0.68582	0.68582	0.68582	0.68582	0.68656	0.68522

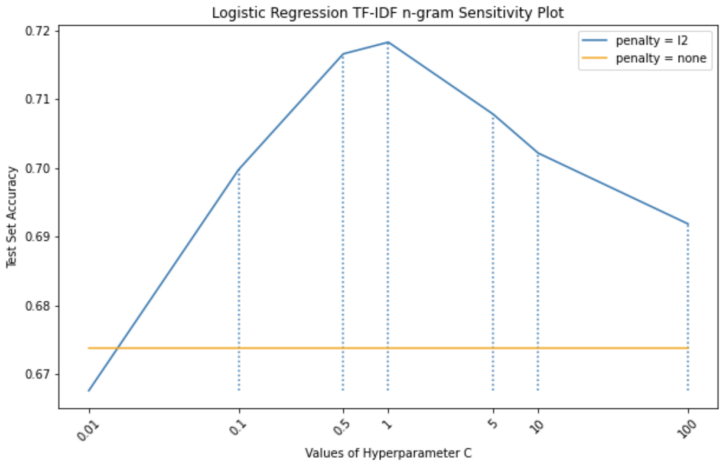
1. wc is an acronym for word characters while ws is an acronym for whitespace.
2. w/o stop and w/stop indicate the removal and the inclusion of stopwords respectively.
3. DC_3000 was built on non-whitespace tokens including stop words while using the *Dale Chall 3000 Word List* as the vocabulary.
4. dc_prop is the proportion of sentence words in the *Dale Chall 3000 Word List* while AoA is the proportion of sentence words in the AoA file and the sum of their respective *AoA_Kup_lem* value (2 features). Non-whitespace tokens were used and stop words were included for each.

Here we observe that the single highest accuracy score of 0.71829 is produced by the logistic regression model where the text is tokenized on non-whitespace characters, including stopwords, and vectorized with TF-IDF 2-grams and 3-grams. Additionally, the same model, regardless of feature space, appears to perform the best with an average accuracy of 0.70625. It is necessary to point out that this value is weighed down a bit by the poor performances on word character tokens without stop words and the vocabulary created by the *Dale Chall 3000 Word List*. The former suggests that word character tokens without stop words do not adequately capture the signal in the text. The latter can be attributed to imposed vocabulary – 2-grams and 3-grams do not exist when the feature space is limited to 1-grams. This is true for all models using n-gram vectorizations in the table above as they perform at the level of the most frequent dummy classifier.

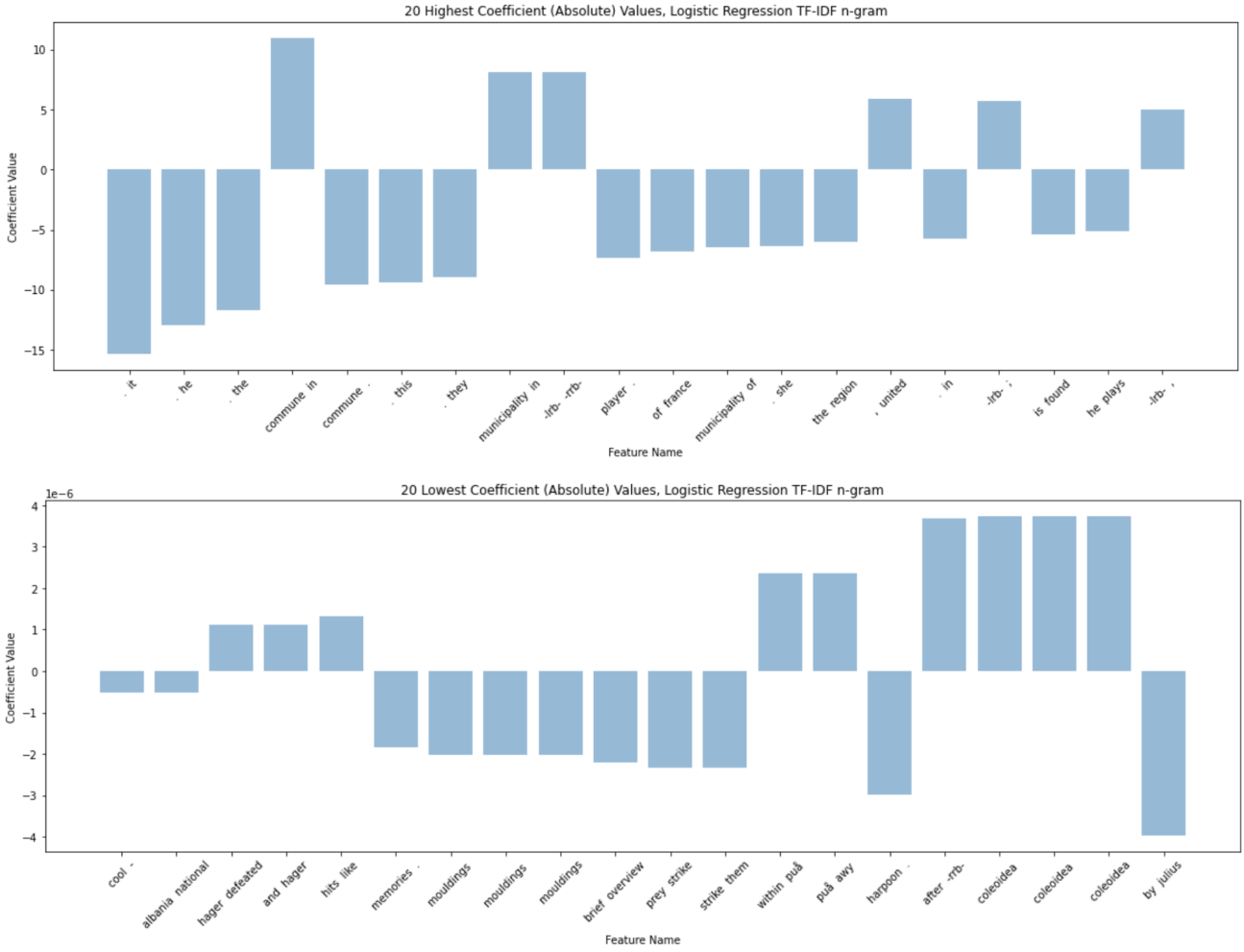
It is expected that the dummy classifiers yield the same accuracy score for each feature space as the two do not consider the text vectors. A keen eye might also notice that the performance of models using 2-gram and 3-gram vectors at the *character level* produce the same accuracy score until additional features are added. This is because the character level tokenization setting is not designed to consider regular expressions or stop words – it produces the same text vectors each time.

When considering the performance of all models relative to individual featurizations, there are a few findings of note. For the feature space *without* stop words, we see an average accuracy of 0.59548 for word character tokens and an average accuracy of 0.63352 for non-whitespace tokens. For the feature space *with* stop words, we see an average accuracy of 0.62105 for word character tokens and an average accuracy of 0.63999 for non-whitespace tokens. This suggests two things. First, non-word character tokens (mainly, punctuation) capture some of the signal in the text as the non-whitespace featurizations lead to better results regardless of stop words. Second, stop words also capture some of the signal as each tokenization produces better results with stop words included than without. Furthermore, we see that models perform better with the inclusion of non-text features: an average accuracy of 0.64137 is exhibited when including the proportion of sentence words in the *Dale Chall 3000 Word List* and an average of 0.64195 with both the Dale Chall and AoA features. It is important to note, however, that this is not the case for all models – our top performing model experienced a slight decrease in performance when including non-text features.

Aiming to further improve the performance of our best model, the logistic regression TF-IDF n-gram model, we set out to optimize the choice of hyperparameters. Using grid search with stratified five-fold cross validation, we cycle through a range of values for the regularization hyperparameter and find that the default settings originally used, where C = 1 and penalty = l2, work best. This is reinforced with a sensitivity plot relative to test set accuracy. Note that the x-axis is denoted in



logarithmic scale for illustrative purposes. A horizontal line for test set accuracy is exhibited when the penalty parameter is set to none because the regularization parameter is ignored in this case. In an attempt to better understand which features the model deems important and unimportant, we consider the feature coefficients with the highest and lowest absolute values respectively. One observation that follows is the number of features containing some form of a



punctuation mark or stop word. For the 20 features with the highest absolute coefficient values, we see that 13 contain a punctuation character and 14 contain a stop word. For the 20 features with the lowest absolute coefficient values, these counts are 4 and 4 respectively. This appears to coincide with our previous supposition that punctuation marks as well as stop words do indeed provide some signal for the task at hand. Lastly, we have a training set accuracy score of 0.80775, which is roughly 0.08946 higher than the test set accuracy. This may indicate some level of overfitting but it is nothing egregious as the model will generally perform better for the data on which it was trained.

Failure Analysis

	Predicted True	Predicted False
Actually True	0.368838	0.132908
Actually False	0.148802	0.349451

Here we have the confusion matrix for the test set predictions of our logistic regression TF-IDF n-gram model where true refers to a sentence that **does** need to be simplified and false refers to a sentence that **does not** need to be simplified. Note that the figures given are the proportion of each subset out of 104,192 samples. It is clear that samples predicted correctly as true (38,430 samples) or false (36,410 samples) occur in relatively equal proportions, as do samples that are false positives (15,504 samples) and false negatives (13,848 samples). The precision and recall of this model are calculated as 0.71254 and 0.73511 respectively. Consequently, it seems that the model struggles similarly for each class.

We now turn our analysis towards the features that the model deems the most important – those features with the highest (positive) coefficients and those with the (lowest) negative coefficients. We consider the top 50 of each. For the sentences that were correctly predicted true, the ratio of the mean feature value for the highest coefficients to the mean feature value for the lowest coefficients is 3.89. This makes intuitive sense – the values of the features that most positively influence the model's decision are nearly four times larger, on average, than those of the features that most negatively influence the model's decision. As a result, these samples will tend to be further above the decision threshold and are consequently predicted as true. Compare this to the false negatives – those samples that were supposed to be labeled as true. Their ratio of the mean feature value for the highest coefficients to the mean feature value for the lowest coefficients is 1.24 – roughly one third of the same ratio for those samples correctly predicted true. This suggests that for the false negative samples, the associated feature values for positive coefficients do not outweigh those for negative coefficients to the point where the sample is predicted as true. In fact, the mean feature value associated with the highest coefficients for the samples correctly predicted as true is nearly 2.5 times larger while the mean feature value associated with the lowest coefficients for the false negatives is actually *greater* by about 30%.

We see the opposite when comparing the samples correctly predicted as false with the false positives. For the sentences that were correctly predicted as negative, the ratio of the mean feature value for the highest coefficients to the mean feature value for the lowest coefficients is 0.68. Again, this makes intuitive sense – the features that most positively influence the model's decision are outweighed by the features that most negatively influence the model's decision. For the false positives, the ratio of the mean feature value for the highest coefficients to the mean feature value for the lowest coefficients is 3.02. In this case, the positive outweighs the negative in a way that renders the model unable to produce a prediction of false.

Unsupervised Learning

Methods

We utilized latent Dirichlet allocation (LDA), non-negative matrix factorization (NMF), and k-means clustering in an effort to improve our classification efforts. For LDA and NMF, we tokenized the text on non-whitespace characters with stop words included and vectorized the tokens using simple term frequency. Using 10 and 20 components each, we selected the highest probability topic for each sentence and used that as a label for the sentence. For k-means clustering with 10 and 20 clusters, we tokenized the text similarly and vectorized the tokens using TF-IDF. The labels produced from LDA, NMF, and k-means were then used as additional features for our logistic regression classifiers. The number of components/clusters for each were chosen uniformly.

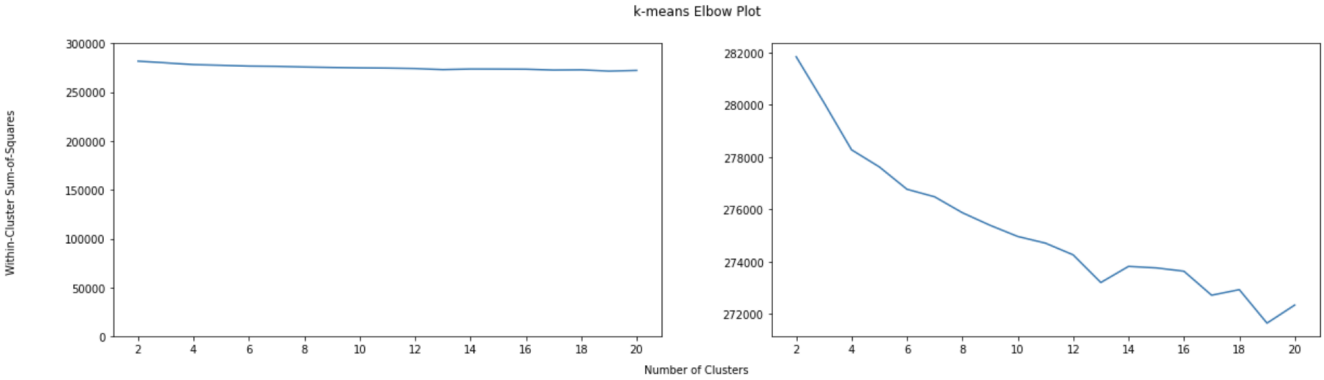
Evaluation

Being that our original goal was to improve the accuracy of our classification efforts, our first method of evaluation was to check how the inclusion of these new features affected classification performance. The feature space included text vectors (tokenized on non-whitespace characters including stop words), the non-text dc_prop and AoA features, and the labels produced from our unsupervised techniques. Comparing these results with the initial performance for each model, we see that the logistic regression TF-IDF model exhibited a slight increase in classification accuracy while the rest exhibited a slight decrease in classification accuracy. We speculate that there are a few reasons why this did not improve our models as originally hoped.

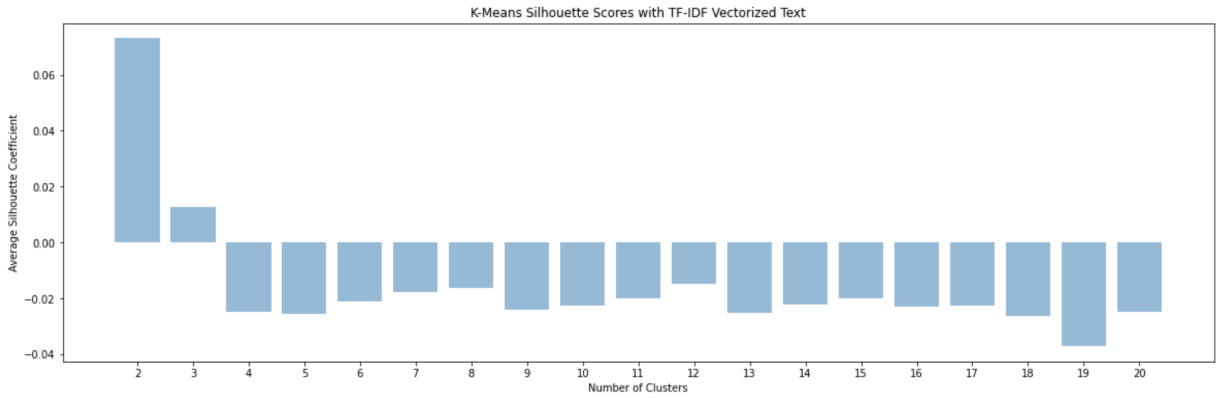
	Accuracy
LR, count	0.69895
LR, TF-IDF	0.70225
LR, TF-IDF n-gram	0.71058
LR, TF-IDF n-gram characters	0.70322

First, with such a massive feature space – ranging from roughly 35,000 features to nearly two million, depending on the vectorization method used – we speculate that the addition of only a few more features may not be perceptible to the model. Second, we feel that the labels gathered from the various techniques are neither coherent enough nor discriminative of the text. When examining the top tokens for each topic/cluster we see that there are some similarities between topics/clusters, and that there may be a few with some semantic meaning, but none appear to be particularly informative. Words for each topic when the component/cluster argument is set to 10 can be viewed in the appendix. For example, topic 5 from LDA, topics 2 and 7 from NMF, and clusters 4 and 7 from k-means all seem to refer to France but the exact context is unclear. Moreover, they are redundant which raises concerns about each method's ability to discern groups in this instance and ultimately, their usefulness as features. There is also considerable overlap between topics/clusters – consider the tokens rrb, llb, and s. Some are even nonsensical. Consider topic 5 from NMF – it is simply a bunch of single character tokens.

In an effort to more formally evaluate the appropriate number of clusters for k-means, we turn to the elbow plot method. Here we see that the within-cluster sum-of-squares is large for all iterations up to 20 clusters and that any semblance of an elbow is evident only when the y-axis is restricted. Given the plot on the right, the optimal number of clusters may very well be



somewhere in the four to six cluster range but having to zoom in to make this distinction does not feel appropriate. Aiming to further understand the optimal number of clusters for k-means, we turn to silhouette scores which are used to assess cluster quality. Using euclidean distance and sample sizes of 10,000, we find that the average silhouette score for 2-20 clusters is near zero.



This is indicative of overlapping clusters that are not well defined. This might suggest that the selected feature space does not reasonably exhibit considerable differences between samples, there are too many samples for clusters to be inferred, or both. It is entirely possible that k-means could still be useful for the task but with this representation of the data, it has proven unfeasible.

Discussion

In terms of the supervised component of the project, we learned a few things. First, we were surprised to see how well the logistic regression model worked for this task given our feature spaces. We had originally intended for logistic regression to provide some sort of baseline measure but in the end, it performed the best even with the default parameter settings. We were doubly surprised to see that it outperformed the random forest models that we had used but this makes sense as random forests do not do well with high dimensional data. In fact, the random forest model that performed the best actually had the smallest feature space by a considerable margin. We were also surprised to see how little the results vary with addition of non-text features. Moreover, we experienced first hand the difficulty of interpreting why a model makes certain decisions. With such a large feature space and relatively short document

lengths, it feels that for any given document, the prediction output from the model is still somewhat of a mystery.

Given more time, we would certainly focus on refining our feature selections, rather than model selections, as we feel that we have a good start with our logistic regression classifiers. One avenue to explore would be alternative vector representations of the text. For example, limiting the vocabulary of the various vectorizers with a hard feature limit or with minimum/maximum term/document frequency settings or using different vectorizers altogether. Another avenue to explore would be the inclusion of additional non-text features that may capture whether certain documents need to be simplified. Examples of this include sentence length/word count, part of speech tagging and counting, and the use of statistical parsers to evaluate difficulty of a sentence in other ways (related to grammar, for example). Effectively improving these two aspects of the feature space may ultimately better illustrate why the models make the predictions that they do.

If these models were to be deployed for the specific use case intended in this project, potential ethical issues that may arise are hard to imagine as they are intended to expand inclusivity by catering to those with learning disabilities, those who are minimally educated, or those with English as a second language for example. Any such ethical issues would certainly be further filtered by a human in the loop – the person that attempts to rewrite a given sentence in a way that is more easily understood. However, ethical issues may arise if the models are repurposed. For example, consider an online community or a mobile application that aims to filter out posts that are considered "troll" posts and/or "incomprehensible" with the goal of ensuring a friendly and productive environment. Such a use case may inadvertently exclude the aforementioned groups or other populations by not allowing them to share content by virtue of their writing abilities. This could be limited by defining explicit guidelines and use cases for which the models are appropriate and inappropriate.

In terms of the unsupervised component of the project, we also learned a few things. First, we were surprised to see how the various techniques honed in on similar aspects of the text vectors. We also learned of the difficulties in topic modeling and clustering – mainly, producing labels that are discriminative and coherent in some capacity. Many of the topic-word vectors across techniques were related and had overlapping tokens which suggest that the text representation requires further refinement. Given more time, any further efforts would aim to address these issues. This would likely involve further preprocessing of the text (removing common/overlapping/uninformative tokens), further limiting the size of the feature space or choosing different representations, including non-text features, better tuning/selection of technique parameters, or even considering alternative techniques altogether. As the output from the unsupervised component of the project is intended solely to boost the efforts of the supervised component, any ethical issues that may arise closely mirror those associated with the supervised component. As an example – the topics/clusters output from the various techniques may inadvertently identify text of or related to certain populations which could exhibit harm when input to a supervised classifier. Again, this could be limited by defining explicit guidelines and use cases for which the models are appropriate and inappropriate.

Statement of Work

Matthew contributed to writing the main code base, writing the report, and creating the visualizations for the report.

Thomas contributed to the project proposal and in investigating neural networks.

Josh contributed to the project by investigating models with non-text features.

Appendix

LDA, 10 Components:

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
	s	team	lrb	music	lrb	france	rrb	s	lrb	city
	united	football	rrb	band	rrb	department	lrb	series	rrb	states
	university	national	used	album	war	region	born	lrb	district	united
	states	league	s	new	s	commune	1	rrb	o	county
	states	season	used	released	world	calais	football	film	province	s
	school	lrb	called	s	years	pas	player	book	located	river
	government	rrb	water	rock	species	north	american	television	municipality	game
	new	world	number	song	people	la	2	known	town	people
	president	played	usually	tropical	century	saint	4	written	language	north
	college	club	term	hurricane	family	northern	january	john	city	largest

NMF, 10 Components:

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
	rrb	s	france	united	born	1	city	pas	new	football
	lrb	u	department	states	american	4	county	calais	world	national
	o	album	commune	county	player	2	capital	nord	known	team
	d	women	region	kingdom	january	ð'	state	region	york	player
	â	state	la	president	september	3	north	department	used	league
	known	band	calvados	iowa	march	â	located	france	war	club
	called	second	normandie	states	july	î	district	north	time	played
	km	film	basse	canada	april	5	largest	commune	called	japanese
	german	children	aisne	america	february	0	south	l	south	plays
	english	death	northern	nations	footballer	ñ	area	ã	best	hockey

k-means, 10 Clusters:

	Clusteer 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9
	city	football	district	s	department	series	rrb	pas	united	released
	used	player	municipality	u	france	television	lrb	calais	states	album
	new	born	canton	rrb	commune	game	born	nord	city	tropical
	people	rrb	switzerland	lrb	region	character	o	department	county	hurricane
	world	lrb	province	world	aisne	s	american	region	iowa	storm
	references	japanese	located	state	calvados	animated	known	france	kentucky	band
	called	team	belgian	city	normandie	tv	1	commune	state	single
	known	national	aargau	new	basse	american	d	north	kingdom	studio
	time	club	pakistan	county	picardie	rrb	â	ã	illinois	atlantic
	1	plays	ticino	time	gironde	lrb	english	l	florida	song