Word Importance Classification for Debate Evidence Cards

Quentin Clark, Esha Vaishnav, Wiktor Lidwin
{qtcc,eshav,wlidwin}@bu.edu

1. Introduction

In competitive Policy and Lincoln-Douglas debate at the high school and collegiate level in the United States, students present evidence in what are called "Cards" (fashioned after notecards used in research). To do this, students take an existing snippet of an article and "cut" it, taking the raw text of a section of an article and underline, highlight, and bolding certain sections of it as those sections relate to their importance to the overall argument the article makes.[1,2] We examined using machine learning approaches to automate this process, so that our final model could cut cards approximating human produced ones.

2. Literature Review

Existing literature on our narrow problem statement (machine aided card production) is nonexistent, and authors are only aware of rumored efforts towards such an approach from first-hand discussions with members of the debate community. If such a study on our problem statement has been performed, it is not widely available nor are the results accessible. One partial implementation is the "auto-underline" feature in Verbatim, the most widely used Word add-on for debate, but the details of how that works are ill-known and its performance poor enough that the vast majority of the community is unaware of its existence and no one to our knowledge actually uses it to aid in evidence processing.

However, the broader field of Natural Language Processing (NLP) is rich with problems regarding text processing and a deep history of people inventing effective solutions to those problems. As such we looked into the NLP field for works solving problems similar to ours. Even within this context, debate cards are a somewhat unique subclass of summarization in the sense that you want to produce a summarization *from the words of the text, presented in the order of the original text*, and that there are other meaningful measures like bolding and underlining text. The majority of summarization approaches in NLP generate a key word or phrase from an article, which would not apply for card production. We found a small handful of potential solution approaches from similar problems. Most solutions reframe these kinds of problems as a *per word classification* task, generating some feature vector from each word and classifying it into "buckets" of importance, but even within the rough area of "importance classification in text" they operate on a per-sentence or paragraph level, and not the word level like we want for card production (for example, sentiment analysis in Twitter posts). The previous work closest to our case was a paper on deaf/hard of hearing subtitle enhancement algorithms[3], which is a similar problem to ours, as a deaf user processes a text sequentially while being informed of the important words highlighted by the text similar to how a debater will cut a card by reading an article and then highlighting relevant words and passages. This work influenced a great deal of our problem approach.

3. Problem Formulation and Solution Approaches

Inspired by that previous work and conversations with professors, we decided to approach the problem as a supervised per-word classification problem. For training data, we used the College Debate Open Source Wiki, where most teams voluntarily "disclose" their speech documents to an online open-source repository, containing the cards they read in each debate. We used evidence from the 2021-2022 year to produce a substantial dataset. The topic that year was about economic Antitrust, but due to the nature of Collegiate debate evidence was taken from a wide variety of sources including economics journals, news articles, the critical theory field, and others.[4] We downloaded all the speech documents from that debate season in their native .docx Word format, and then wrote and ran a parser that converted the cards into a Python list of tuples, where each tuple has a string with the body text of the card and a list of class values equal to the length of the text split into words. Repeat cards were filtered out using a Bloom Filter, and problematic cards or documents with unusual formatting were skipped. This produced around 300,000 unique cards (many cards show up in multiple documents, which were filtered out) with around ~850 words per card on average.

To formulate our problem as a classification problem, we devised a class schema for each word as follows:

0 - Normal text with no processing applied
1 - <u>Underlined Text</u>

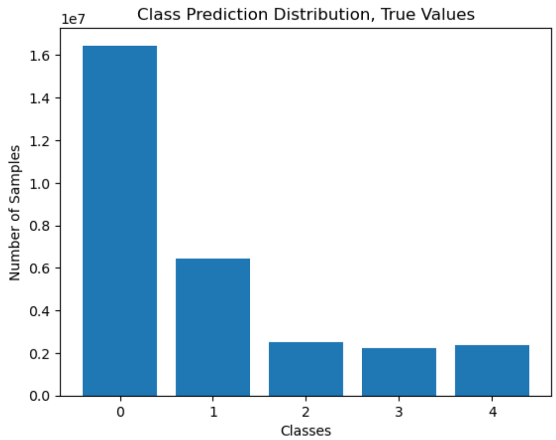
---

[1] See Appendix Fig. 1 for example.
[2] For more information, see Phillips.
[3] Amin et al.
[4] The full topic was "Resolved: The United States Federal Government should substantially increase prohibitions on anticompetitive business practices by the private sector by at least expanding the scope of its core antitrust laws."

2 - **Underlined and bolded text**
3 - Underlined and highlighted text
4 - **Underlined, bolded, and highlighted text**

This schema covers the whole range of possible processing values for text in debate cards, and is ordered roughly from least important to most important.

The nature of our training data, and the problem itself, presented some difficulties. One is the distribution of classes in our training data: it was severely imbalanced towards the 0 class. The full distribution is shown below:



This disproportionality is important to consider in order to avoid the algorithm from overfitting all the word labels as 0 or 1 due to most labels being those values.

Another is the inherently subjective nature of card-cutting as an activity. Debaters are most often not attempting to extract the "true" argument an article is making, but are instead trying to selectively emphasize certain parts of an article to support a particular viewpoint, often called "spin".[5][6] Examining the merits or detriments of such an activity in what is theoretically supposed to be a pedagogical valuable activity are beyond the scope of this work. Regardless, it presents an odd challenge because several cuttings of a card can be "correct", and the cards in our database cover a wide range of interpretive viewpoints, as it includes cards from every team and thus many stylistic and argumentative viewpoints. As such we expect our data to be very noisy, and practically found it to be hard to work with.

To generate feature vectors for our words, we looked at using Word2Vec[7] and Large Language Models (LLMs) based on Transformer architecture, specifically Google's BERT-BASE.[8] The key difference between Word2Vec and BERT is that BERT is a bidirectional encoder, which allows it to generate relative importance values in its output feature vector representing the relationship between that word and other words in the same passage, which can change with context. Word2Vec cannot do that because it assumes a fixed meaning for every word, which it learns from context, but cannot assume multiple meanings. Once the two models were chosen, an additional concern to be aware of is the parameters to implement in the NLP algorithms (i.e. Word2Vec). The main parameter that affects the behavior of the Word2Vec model would be the window of words that are weighted in a neural network to create a feature vector. The window of words that would be ideal would be the one that creates the lowest amount of loss after a chosen number of epochs. For BERT, the chosen headings for the classification models are also important to set as parameters for the model, to create multiple feature vectors for each word. In our use case, we used BERT without one of the typical "heads" (linear layers) on top of the hidden layers, using them only to generate feature vectors for our words.

3.2 Regression

An OLS(Ordinary Least Squares) linear regression model was trained on a fraction of the dataset. Each entry consisted of a 500 length scope vector and a 100 length current word vector, forming a 600 length entry vector. The scope vector is the past 5 highlighted word vectors. The predicted label of the entry depends on the following equation:

$$\hat{y} = \beta_0 + \beta_1 x; \quad argmin_{\beta_0,\beta_1} \sum (y_i - \hat{y}_i)^2$$

Due to the nature of the numerical labels, we believed a regression model would be able to capture the significance relationship of each label. We were unable to use the entire dataset for this model due to time and memory constraints. Likewise, polynomial regressions was unpractical because of the constraints.

3.3 Online SVM

A linear SVM (Support Vector Machine) model can take advantage of batch training due to SGD (Stochastic Gradient Descent) following the formulas:

$$\hat{y} = sign(w^T x_{ext}); \; [x]_+ = max(0, x)$$

$$argmin_w \left(\sum_{i=0}^{n} [1 - y_i w^T x_{ext}]_+ + \lambda ||w||^2\right)$$

$$w \leftarrow w - \eta_t \left(\lambda w - y_t x_{ext,t} 1_{\{y_t w^T x_{ext,t} < 1\}}\right)$$

---

[5] See Figure 8 for an example.
[6] For a more detailed, debate contextual example, see: https://hsimpact.wordpress.com/2020/05/26/spin-class-how-does-corona-impact-cjr/
[7] Mikolov et al.
[8] Devlin et al.

This algorithm was chosen because of its high success rate in general classification. Specifically, 3 SVM models were trained, zeroVsAll (segregating 0 class from the other 4 classes), oneVsTop (segregating 1 class from {2,3,4}), and 2vs3vs4. Running through the entire dataset with batches of 30 took ~30 hours to train. Similarly, the SVM entries were calculated identically to the regression entries. Additionally, an approximate RBF (Radial basis function) kernel was implemented by transforming the entries.

4. Implementation

4.1 Word2Vec Implementation

The initial algorithms used in this experiment were based in Word2Vec because the model for Word2Vec is simpler to implement than BERT and easy to apply regression and SVM models to. Word2Vec is a natural language processing model that trains itself by selecting a word as an input, looking at surrounding words using a given window that the user provided, and generating a single feature vector as an output, assigned to a single word. Word2Vec has some interesting applications tied to it, such as similarity commands that output similarity scores based on words (for example, "good" and "asteroid" has a score of 0.057 while "good" and "bad" has a score of 0.60, meaning that it is more common to see "good" with the word "bad" than "asteroid"). Utilizing feature vectors that can be transformed to contextualize words, the Word2Vec model can be used to apply linear regression of SVM models to highlight important words and boost the highlighted areas based on the similarities of all the words given their feature vectors.

4.2 BERT Implementation

The theory behind using BERT to generate features for our words is that transformers are trained in an unsupervised setting to find dependencies between words, which could potentially be useful in finding words of global importance to a text as we are trying to do. Using the "21-22 Antitrust Topic" data set, D=712 feature vectors for each word were generated by running each card through Google's existing BERT-BASE (breaking up cards longer than 500 words into 500-word chunks) and saving them in a new file. BERT-BASE was trained unsupervised on a large online corpus, which we assumed would generalize well to our dataset also taken from internet English language sources. This gave a total n= 34,088,809, for a total processed data size of ~90 GB. This is too large for training methods which require the entire training set to be kept in-memory, so we utilized online/batch methods to help train our model.

Simply applying multiclass classification to the problem proved unsuccessful, as the severe class imbalance negatively affected performance. As such, we looked at two approaches: an ensemble model of three separate classifiers, and a regression model.

The classification model split the problem into three smaller problems: zeroVsAll, oneVsTop, and 2vs3vs4. Because each of these splits are roughly proportional, the idea was that each classifier would perform better than training a single multi-class classifier in one run.

The regression problem attempts to utilize the inherent "ordering" of the class labels by training a regressor over the dataset, treating the class labels as regression labels. Once a model is produced and outputs generated, you could sort the resulting "lossy floats" into class buckets either with hard thresholds or by sorting the values and splitting them into buckets via some user-defined proportion.[9] This approach is useful because it is more likely to misclassify words in the labels *around it,* producing cards of closer qualitative similarity to human-produced ones, and lets the user adjust the amount of highlighting on a *per-card basis*, whereas the ensemble classification models cannot adaptively highlight.[10] This approach is similar to the previous work on hard of hearing transcription importance, except they used fixed cutoffs whereas we used this sorting-based approach.

4.3 Regression Implementation

Initially, the data was filtered to only contain valid examples that had an equal number of labels and words. From there, the data was split into the training and testing data. Next, Word2Vec was trained on the data. After, we iterated through each card, splitting it into words. When parsing a new card, the scope vector was reset to a 0 5x100 matrix (5 being the scope and 100 being the word vector size). Continuing, at each word an entry was created by concatenating the flattened current scope vector with the current word vector. The label was simply the label of the current word. If the label was a 3 or 4, the 0th row of the scope vector would be deleted and the word vector would be append to the scope vector. Once every entry and label is added to the list of entries and labels, a sklearn linear regression model is fitted to those entries.[11]

4.4 SVM Implementation

As in the regression model, the data was filtered and separated into training and testing, and Word2Vec was trained. Also, the scope vector implementation was

[9] See Figure 5.
[10] For an example, see Figure 14.
[11] See Figure 15.

identical to the regression model. The SVM implementation uses 3 SGDClassifier models: zeroVsAll, oneVsTop, and twoVsThreeVsFour. Each responsible for the decisions outlined in the classification model. Furthermore, the SVM algorithm separates the training dataset into batches of size 30, training the models at each iteration. This training is done by first calling the "parseCards" function. This function returns the entries and labels corresponding to each model.[12] As seen in Figure 16, these returned entries and labels are feed into the "partial_fit" function of the SGDClassifier.

5. Experimental Results

To compare our experimental results, we created a performance metric that created a "penalty value" dependent on the kind of misclassification that occurred, and then took the average of all samples examined.[13] These penalty values were higher for what we subjectively perceived to be more egregious misclassifications: for example, highlighting {3,4} versus not highlighted, and underlined vs not underlined were penalized more heavily compared to other misclassifications.

5.1 Word2Vec Results

When looking for the best window of words to use as a parameter for the Word2Vec algorithm, the following method was implemented: run 30 epochs of each model with a different window parameter, and choose the window with the least loss fluctuation, especially in the last epochs (epochs 25 to 30). The outcome was that the window with 5 words was the best option as shown in Figure 2 in the Appendix, and this finding was utilized in the implementation of the regression model.

After a model was trained using Word2Vec and a context window of 5, a linear regression algorithm was then trained, alongside an SVM model using a radial basis kernel. Both of these methods produced inconsistent results that mostly predicted 0 value labels with the highest CCR, as shown in figures 3 and 4, which show the confusion matrix results of the regression and SVM models, respectively. As shown in the graphs, regression has more variability and a higher chance of correctly classifying labels that are not 0 values, compared to the SVM label.

5.2 BERT Results

Before more complex methods on our BERT data were performed, we applied simple transformations based on the approaches investigated in Amin et al. In that work they examined the relationship between statistical quantities of each feature vector and applied a linear model to achieve impressive results. Unfortunately, such results did not bear fruit in our study. Plots of the L2 Norm[14] and mean[15] versus the class values did not reveal a significant relationship, and linear classification approaches via logistic regression on that data performed no better than random chance.

Neural Networks also showed some promise, but training even in batch over our data took long enough on a consumer graphics card to make hyperparameter tuning and architecture experiments difficult, even on a toy sample of the data. When training a multi-class classifier Neural Network, we ran into the same difficulties with our imbalanced dataset. We attempted many things: different optimizers, learning rate, loss functions (cross-entropy loss with carefully selected update weights, for instance) but all failed to produce desirable results (by far the most common was the classifier classifying every output result as 0).

We found the most success using gradient boosted decision trees, as implemented in the XGBoost Python package. XGBoost seeks to minimize an objective function containing some performance metric (MSE by default and in our use case) and a penalty term for net tree complexity (roughly defined by the net number of decisions made by each tree for all the training data) to reduce overfitting:

$$\text{obj}(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \omega(f_k)$$ [16]

Weights for those trees are then updated via a Taylor expansion approximating the loss/penalty gradient. Individual Trees are pruned/optimized via a Gain function that rewards splitting samples along different classes and penalizes splitting between the same class.

XGBoost has support for hardware acceleration and ScikitLearn integration, which makes training in batch efficient and hyperparameter tuning efficient. For the following section, the first 30 million samples were used for training, and next 1 million for validation. Training over the sample data took around half an hour for one epoch. All future models referenced were trained for one

[12] Figure 16
[13] See Figure 9 for pseudocode.

[14] Figure 6.
[15] Figure 7.
[16] Source and more details at XGBoost documentation at https://xgboost.readthedocs.io/en/stable/tutorials/model.html.

epoch, as the training time per epoch increased linearly[17] and did not show improved performance in our case.

To tune hyperparameters, we did a grid search on the following XGBoost Regressor/Classifier parameters: {max_depth, min_child_weight, objective}. These were selected because the other hyperparameters are for controlling resistance to overfitting, which we found the default configuration was not prone to, so we tried to increase model complexity via these parameters. We found no appreciable difference on holdout data performance when changing them against the default configuration.[18]

Confusion matrices and performance metrics for the classification and regression models are here.[19] The BERT XGBoost Classifier achieved the highest performance of 2.24 on the test data out of the total four models tested.

One interesting behavior revealed by the confusion matrices is that the regression models are more likely to misclassify in class buckets similar to the true class bucket even when its overall classification rate was lower, whereas the classification ensemble model was more random. This was expected and is arguably desirable, as we discussed above.

5.3 Qualitative Results

To produce real-world samples, we applied the full data processing pipeline to two example cards for the purposes of examination in this work. This included turning article selections into a Python string, generating feature vectors, running them through our model, applying post processing techniques, and saving them to a Word document with card-like formatting with the model's suggested cutting.

For the sake of making our software more feature-complete, we also included an abstract summarizer from Facebook's BERT model trained on abstract summarization from CNN news articles, which is a roughly approximate task to what tagging (reading a summary/shortened version of the argument before the card text) does. We found the summarizer using as input all the text from the card *not* predicted to be class 0 to capture the most relevant parts of the card compared to the whole text or just highlighted sections.

We also applied a simple smoothing algorithm to the final predicted class values that changed each class to take a max of its surrounding *n* neighboring class values.[20] We found that with a window of n=2, it produced more natural looking evidence. This whole process, for one piece of evidence, takes around 20 seconds on a consumer graphics card, and could probably be cut down if cutting several cards in batch (which would prevent needing to load BERT/the CNN summarizer several times).

We ran this on two samples: the Skelley evidence in the "card spin" example and the Oduntan evidence read in the 2022 LD TOC finals. Both come from fields outside the training set's antitrust area (electoral politics and geospatial intellectual property), which we hoped would provide a challenge to our model and represent real-world use cases well. We found that the regression model performed better, due in no small part because the user can hand-select the class distribution proportions. The classification ensemble produced a proportion of labels on the training data similar but significantly skewed from the original classes[21] and as mentioned previously does not allow for per-card fine tuning. We found that the distribution of the training set worked well for Oduntan, a longer piece of evidence, but proportions skewed towards highlighted classes worked better for Skelley, which only the regression model allowed us to adjust.

Overall, the models performed reasonably well, capturing important phrases (like "**Obviously, this circumstance could have grave consequences…** US national security" in Oduntan) and skipping over much of the not-underlined prose. However, the model was unable (although not expected) to produce strings of highlighting with rough semantic sense like how human highlighting does, which is a limitation of our approach.

6. Conclusion

Out of the Word2Vec linear regression, Word2Vec SVM classifier, BERT XGBoost Regressor, and BERT XGBoost Classifier, the model with the best performance metric was the BERT XGBoost Classifier, with a performance of 2.243. Generally, the label of words as "unlabeled", or 0, was the label that gave the highest CCR in comparison to the other words. Our qualitative examination found regression models capable of producing crude but better than random cuttings of new evidence.

There are several avenues for potential future work. One would be fine-tuning BERT for our context on our input data, instead of simply using the existing model to generate feature vectors. Several state of the art papers do this for real-world tasks, but we lacked the time or expertise. Another suggestion we frequently encountered was using some kind of Markov decision process like a

[17] Figure 17.
[18] Table 1.
[19] Figure 10 and 11.
[20] Figure 18.
[21] See Figure 12.

Markov Random Field, which would recontextualize the classification problem into a probabilistic markov decision process (given previous words and their class values, what is the probability the next given word will have each class value)? This seems promising, as it would theoretically capture more of the behavior real people use when cutting cards (ie, the odds a word is cut similarly to its neighbors is always higher than what that word's importance might represent in the abstract) but we were warned off of using them in this project for its technical difficulty and our short time span for project completion. A similar way to capture recurrent dependence might be a RNN.

Further feature data processing might help as well. A convolutional neural network (CNN), usually used for image processing, might aid processing the large, feature complex BERT vectors. Including a window of BERT features like we did in Word2Vec might help improve performance as well.

Lastly, there is room for more intelligent post processing methods that would further help the cards resemble human produced ones. Our model's post processing helps produce human-like cards, but that still needs manual processing to fix grammar. A model that can recognize semantic correctness and fine-tune our model's broad importance distribution into something debate-ready out of the box.

7. Description of Individual Efforts

Quentin - wrote all parsing/writing methods. Performed all BERT related experiments. Wrote processing methods (performance function, post-processing method, Word interface methods, etc.). Wrote sections 1, 2, 3, 5.3, 6, and all sections regarding BERT experiments.

Esha - Studied how the Word2Vec and BERT algorithms worked functionally in preliminary research. Specifically changed and compared loss graphs for different windows of a Word2Vec model's training data and chose the ideal window parameter. Worked on sections 3, 4.1, 5.1, and 6.

Wiktor - Wrote linear regression and SVM files, and performed all related experiments.
Wrote sections: 3.2, 3.3, 4.3, 4.4

8. References and Acknowledgements

Amin, Akhter Al, et al. "Using BERT Embeddings to Model Word Importance in Conversational Transcripts for Deaf and Hard of Hearing Users." *arXiv preprint arXiv:2206.12368* (2022).

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

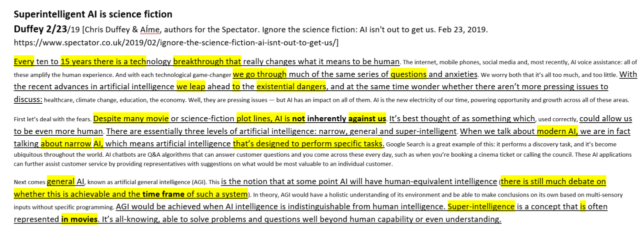
Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

Phillips, Scott. "Debate 101 Cutting a Card." YouTube, 1 June 2020, www.youtube.com/watch?v=VRPZDt-EKX0.

8.1 Acknowledgements

Thanks to Boston University professors Prakash Ishwar, Ashok Cutkosky, Brian Kulis, and Alex Olshevsky, and graduate students Vaibhav Choudhary and Chris Schwarze for input and advice.

9. Appendix



Superintelligent AI is science fiction

Duffey 2/23/19 [Chris Duffey & Aíme, authors for the Spectator. Ignore the science fiction: AI isn't out to get us. Feb 23, 2019. https://www.spectator.co.uk/2019/02/ignore-the-science-fiction-ai-isnt-out-to-get-us/]

Every ten to 15 years there is a technology breakthrough that really changes what it means to be human. The internet, mobile phones, social media and, most recently, AI voice assistance: all of these amplify the human experience. And with each technological game-changer we go through much of the same series of questions and anxieties. We worry both that it's all too much, and too little. With the recent advances in artificial intelligence we leap ahead to the existential dangers, and at the same time wonder whether there aren't more pressing issues to discuss: healthcare, climate change, education, the economy. Well, they are pressing issues — but AI has an impact on all of them. AI is the new electricity of our time, powering opportunity and growth across all of these areas.

First let's deal with the fears. Despite many movie or science-fiction plot lines, AI is **not inherently against us**. It's best thought of as something which, used correctly, could allow us to be even more human. There are essentially three levels of artificial intelligence: narrow, general and super-intelligent. When we talk about modern AI, we are in fact talking about narrow AI, which means artificial intelligence that's designed to perform specific tasks. Google Search is a great example of this: it performs a discovery task, and it's become ubiquitous throughout the world. AI chatbots are Q&A algorithms that can answer customer questions and you come across these every day, such as when you're booking a cinema ticket or calling the council. These AI applications can further assist customer service by providing representatives with suggestions on what would be most valuable to an individual customer.

Next comes general AI, known as artificial general intelligence (AGI). This is the notion that at some point AI will have human-equivalent intelligence (there is still much debate on whether this is achievable and the **time frame** of such a system). In theory, AGI would have a holistic understanding of its environment and be able to make conclusions on its own based on multi-sensory inputs without specific programming. AGI would be achieved when AI intelligence is indistinguishable from human intelligence. Super-intelligence is a concept that is often represented **in movies**. It's all-knowing, able to solve problems and questions well beyond human capability or even understanding.

Figure 1: Example of a Card. Text at the top is a "tag", or user written summary of argument card makes. Next section is the citation, usually in MLA or similar format. Next section is the "card body" which is composed of some or all paragraphs from the article in the citation, with certain words underlined, bolded, and/or highlighted to signify importance. Highlighted text is read in-round. Text that is bolded (called "emphasized" in community) or highlighted must also be underlined, for a total of 5 possible classes for each word.

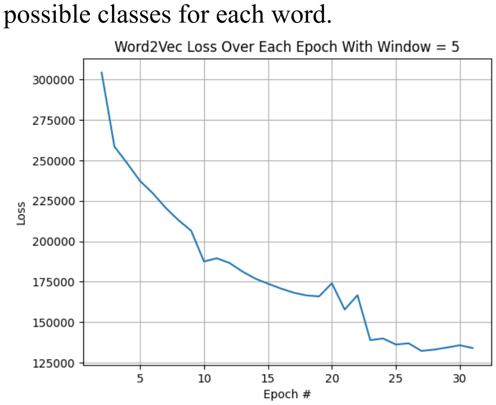


Figure 2: Word2Vec loss word ideal window parameter (5 words).

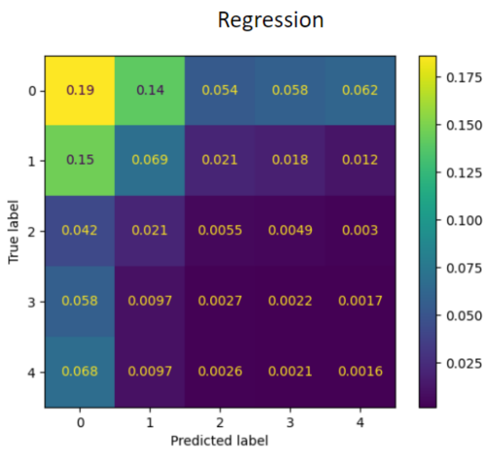


Figure 3: Confusion matrix for the regression algorithm, based on the Word2Vec model(Performance=2.6)

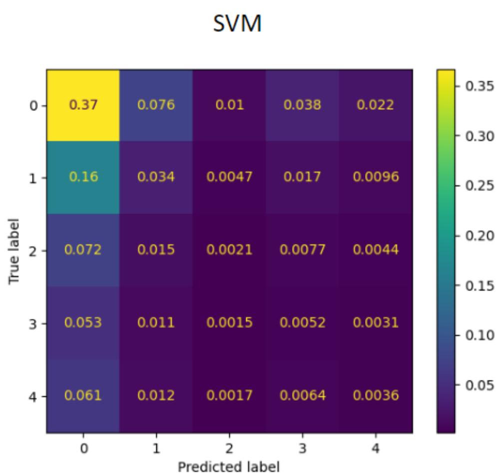


Figure 4: Confusion matrix for the SVM algorithm, based on the Word2Vec model(Performance=2.3)

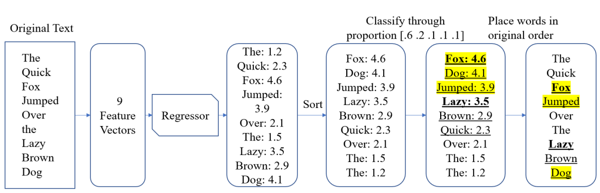


Figure 5: Regression to Classes Algorithm Example

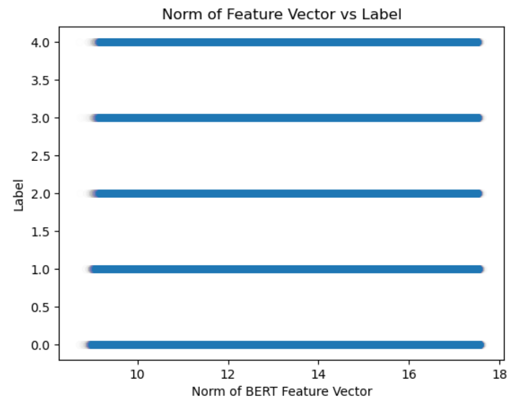


Figure 6: L2 Norm of BERT Feature Vectors vs Class Labels

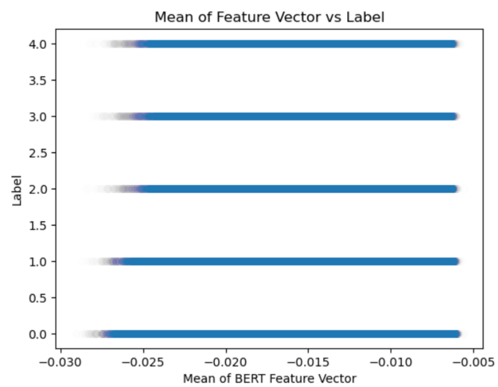


Figure 7: Mean of BERT Feature Vectors vs Class Labels

Warnock run-off win good for Democrats

Skelley 12/7 Skelley, Geoffrey. Geoffrey Skelley is a senior elections analyst at FiveThirtyEight. "Republicans Have A Clear Path To Retaking The Senate In 2024." FiveThirtyEight, 7 Dec. 2022, fivethirtyeight.com/features/republicans-senate-2024-map.

In a bit of electoral déjà vu, Democratic Sen. Raphael Warnock **won Georgia's runoff on Tuesday**, almost two years after he won a special election runoff to help hand Democrats a narrow 50-50 majority in the Senate via Vice President Kamala Harris's tie-breaking vote. This time around, Warnock topped Republican Herschel Walker to earn a full six-year term, which will have **major ramifications for** how the **new Senate will conduct business in January**. Warnock's win gives Democrats 51 seats — including independent Sens. Angus King of Maine and Bernie Sanders of Vermont — so the Democratic caucus will **no longer have to constantly rely on Harris to break ties**. Democrats will also now have majorities on each committee and will be able to more easily confirm President Biden's judicial appointments. Yet if we look even further into the future, it turns out the Georgia outcome could also play a role in deciding which party controls the Senate after the next election. The good news for Democrats is that they will have **51 seats instead of 50**, which **gives them a chance to maintain control even if they lose one seat**, depending on whether the next vice president is a Democrat or Republican. The good news for Republicans, however, is that the 2024 Senate map puts them in a better position to take control of the chamber than it does for Democrats to hold onto it.

Warnock run-off win good for Republicans

Skelley 12/7 Skelley, Geoffrey. Geoffrey Skelley is a senior elections analyst at FiveThirtyEight."Republicans Have A Clear Path To Retaking The Senate In 2024." FiveThirtyEight, 7 Dec. 2022, fivethirtyeight.com/features/republicans-senate-2024-map.

In a bit of electoral déjà vu, Democratic Sen. Raphael Warnock won Georgia's runoff on Tuesday, almost two years after he won a special election runoff to help hand Democrats a narrow 50-50 majority in the Senate via Vice President Kamala Harris's tie-breaking vote. This time around, Warnock topped Republican Herschel Walker to earn a full six-year term, which will have major ramifications for how the new Senate will conduct business in January. Warnock's win gives Democrats 51 seats — including independent Sens. Angus King of Maine and Bernie Sanders of Vermont — so the Democratic caucus will no longer have to constantly rely on Harris to break ties. Democrats will also now have majorities on each committee and will be able to more easily confirm President Biden's judicial appointments. Yet if we look even further into the future, it turns out the **Georgia outcome could** also **play a role in deciding** which party **controls the Senate after the next election**. The good news for Democrats is that they will have 51 seats instead of 50, which gives them a chance to maintain control even if they lose one seat, depending on whether the next vice president is a Democrat or Republican. The **good news for Republicans**, however, is that the 2024 Senate map puts them in a **better position to take control of the chamber than** it does **for Democrats to hold onto it.**

Figure 8: Example of card “spin”: both cards are cut from the same FiveThirtyEight article about the Georgia special election, the top is meant to convey a pro-Democrat viewpoint and the bottom pro-Republican.

```
total_penalty = 0
for real_class_val and pred_class_val in
real_class_values and predictions:
    penalty = 0
    for difference between
    (real_class_val and pred_class_val):
        if 0-1: add 3
        if 1-2: add 1
        if 2-3: add 2
        if 3-4: add 1
     total_penalty += penalty
return total_penalty/n
```

Figure 9: Performance Function Pseudocode

	Default	max_depth = 8	min_child_weight = 0.25	objective = ‘log_loss’
Test Performance	2.391	2.397	2.391	2.404

Table 1: Parameter tuning XGBoost Regression

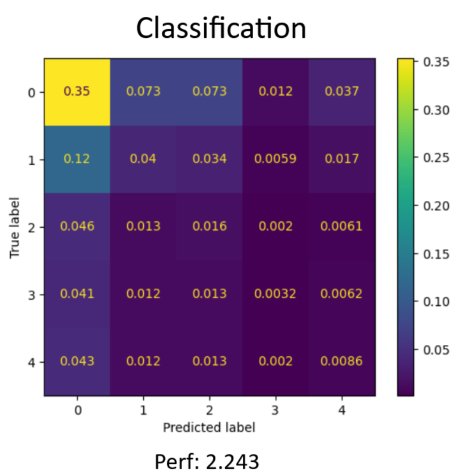


Figure 10: BERT Classifier Ensemble Confusion Matrix and Performance

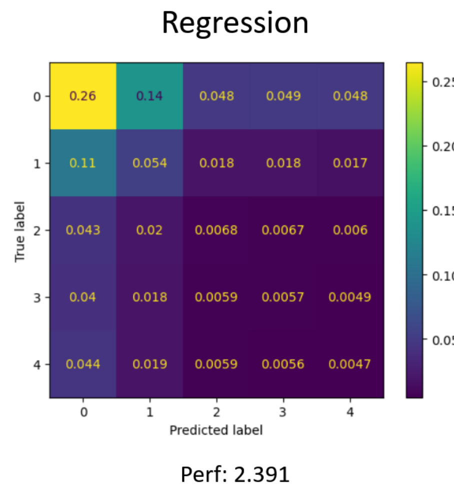


Figure 11: BERT Regression Confusion Matrix and Performance

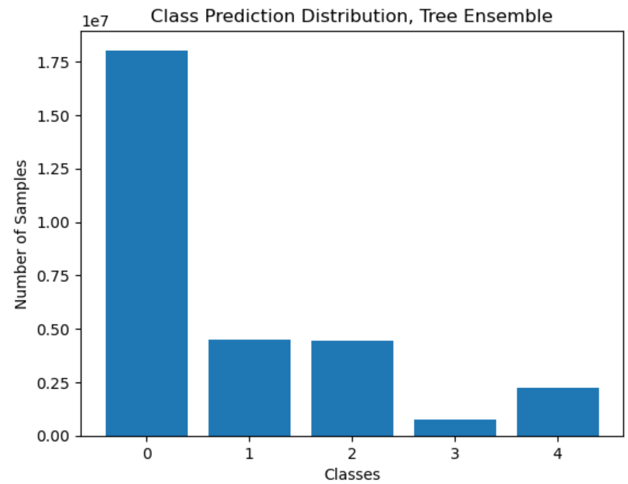


Figure 12: Classification Ensemble Predicted Clases Proportions

Original:

Private remote sensing proliferation causes military and archeological vulnerability – domestic laws don't solve because states can only prosecute when violators enter domestic territory – an international framework providing legal certainty solves

- Sensing across borders causes sovergenty probelsm – ie, American sat sensing in brazil – can't persecute and so no framework for protection
- [which answers the like, make the aff's bad stuff illegal cp]

Oduntan 19 Oduntan, Gbenga, sociolegal scholar and critical legal studies academic, Ph.D. (Law, (Kent)), MA, LLB (hons) BL (hons) BA (hons), ACIArb. "Geospatial sciences and space law: Legal aspects of earth observation, remote sensing and geoscientific ground investigations in Africa." Geosciences 9.4 (2019): 149.

As technological prowess and commercial successes of private entities becomes more democratized, **even powerful states are vulnerable**. A writer admits this conclusion when he wrote: "There is a **dark side,** however. Just as the military will have access to high-resolution commercial imagery, so too will the general public and foreign entities, allies and adversaries alike. Without **proper protections**, military movement and build-up, the lay-out of military facilities and even the locations of individual pieces of military equipment could be made available to the public eye within a matter of hours. Obviously, this circumstance could have **grave consequences for military operations and U.S. national security**" [118]. One of the options open to states that are apprehensive of spying over their **archaeological and heritage sites** from above is that they should adopt domestic laws with extraterritorial effects that make the remote sensing of certain places illegal. This may include the prohibition of dissemination of any unauthorised information derived therefrom. Presumably such laws will apply to offenders as Cheng said "wherever in the world the offences may have been committed" although the ability to enforce jurisdiction will be severely limited if perpetrators do not come within territorial jurisdiction of the offended states [106]. Furthermore the ability of a state to enforce it such cases will inevitably depend on its diplomatic, political and economic clout. Cheng one of the fathers of modern day international space law himself had noted the limitations provided by the Lotus case (1927) to the extent that the case while seemingly allowing states to criminalize acts committed outside territory does so upon the proviso that enforcement may be attempted **only when such persons come within their territorial boundaries,** Cheng introduced two helpful concepts of enduring importance. Jurisfaction he says denotes the normative element of jurisdiction and it represents the powers a state has to adopt valid and binding legal norms and to concretise them with binding effect through its appropriate organs, whether judicial or otherwise. The spheres of validity or operative force of these norms may be realised ratione loci (territorial), ratione instrumenti (quasi territorial) or ratione personae (personal). Jurisaction on the other hand, is the formal element of state jurisdiction and it encompasses the powers a state possesses to, at any place or time, physically perform the acts of making, concretising or enforcing laws. That is it can hold legislative assembly, set up courts or tribunals or even arrest wanted persons. From this point of view, "the validity of jurisaction presupposes jurisfaction, but it is possible to have jurisfaction without jurisaction" [119]. Cheng has persuasively argued that; "Military reconnaissance satellites have not only become a fact of international life that states just have to learn to live with but also a vital instrument in the process of arms control and the preservation of international peace" [106]

Regression Model:

Without proper protections, military the lay-out facilities and even the of could be made to public within of hours

Oduntan 19 Oduntan, Gbenga, sociolegal scholar and critical legal studies academic, Ph.D. (Law, (Kent)), MA, LLB (hons) BL (hons) BA (hons), ACIArb. "Geospatial sciences and space law: Legal aspects of earth observation, remote sensing and geoscientific ground investigations in Africa." Geosciences 9.4 (2019): 149.

As technological prowess and commercial successes of **private** entities **becomes more democratized, even** powerful states are vulnerable. A writer admits this conclusion when **he wrote:** "There is a dark side, however. Just as the military will have access to high-resolution commercial imagery, so too will the general public and foreign entities, allies and adversaries alike. Without proper protections, military movement and build-up, the lay-out of military facilities and even the locations of individual pieces of military equipment could be made available to the public eye within a matter of **hours. Obviously, this circumstance could have grave consequences** for military operations and U.S. national security" [118]. One of the options open to states that are apprehensive of spying over their archaeological and heritage sites from above is that they should adopt domestic laws with extraterritorial effects that make the remote sensing of certain places illegal. This may include the prohibition of dissemination of any unauthorised information derived therefrom. Presumably such laws will apply to offenders as Cheng said "wherever in the world the offences may have been committed" although the ability to enforce jurisdiction will be severely limited if perpetrators do not come within territorial jurisdiction of the offended states [106]. Furthermore the ability of a state to enforce **it** such **cases will** inevitably depend on its diplomatic, political and economic clout. Cheng one of the fathers of modern day international space law himself had noted the limitations provided by the Lotus case (1927) to the extent that the case while **seemingly allowing** states to criminalize acts committed outside territory does so upon the proviso that enforcement may be attempted **only when such** persons come within their territorial boundaries, Cheng introduced two helpful concepts of enduring importance. Jurisfaction he says denotes the normative element of jurisdiction and it represents the powers a state has to adopt valid and binding legal norms and to concretise **them with** binding effect through its appropriate organs, whether judicial or otherwise. The spheres of validity or operative force of these norms may be **realised ratione loci (territorial), ratione** instrumenti (quasi **territorial)** or ratione personae (personal). Jurisaction on the other hand, **is the formal** element of state jurisdiction and it encompasses the powers a state possesses to, at any place or time, physically perform the acts **of** making, concretising **or** enforcing laws. That is it can hold **legislative assembly, set up courts or** tribunals or even arrest wanted persons. From this point of view, "the **validity of jurisaction presupposes jurisfaction, but it is possible** to have jurisfaction **without** jurisaction" [119]. Cheng has persuasively argued that; "Military reconnaissance satellites have not only become **a fact of international life** that states just have to **learn** to **live with** but also a vital instrument in the process of arms control and the preservation of international **peace"** [106].

Figure 13: Oduntan Card, Original and Regression-Model Cut

Regressor

Democratic Sen. Warnock won Georgia's Tuesday, almost years won a special election runoff Democrats narrow in

Skelley 12/7 Geoffrey Skelley is a senior elections analyst at FiveThirtyEight. https://fivethirtyeight.com/features/republicans-senate-2024-map/, Dec. 7, 2022.

In a bit of electoral déjà vu, Democratic **Sen. Raphael Warnock** won Georgia's runoff on Tuesday, almost two years after he won a **special election runoff** to help hand Democrats a narrow 50-50 majority in the Senate via Vice **President Kamala** Harris's tie-breaking vote. This **time around, Warnock** topped Republican Herschel Walker to earn **a full six-year term,** which will have major ramifications for how the new **Senate will conduct** business in January. Warnock's win gives Democrats 51 seats — including independent Sens. Angus King of Maine and Bernie **Sanders** of **Vermont** — so the Democratic caucus will no longer have to constantly rely on Harris to break ties. Democrats will also now have majorities on each **committee and will** be able to more easily confirm President Bidens judicial appointments. Yet if we look even further into the future, it turns out the **Georgia outcome could also** play a role in deciding which party controls the Senate after the next election. The good news for Democrats is that they will have 51 seats instead of 50, which gives them a chance to maintain control even if they lose one seat, depending on whether the next vice president is a Democrat or Republican. The good news for Republicans, however, is that the 2024 Senate map puts them in a better position to take control of the chamber than it does for Democrats to **hold** onto it.

Regressor, Adjusted Proportions

Democratic Sen. Warnock won Georgia's Tuesday, almost years won a special election runoff Democrats narrow in

Skelley 12/7 Geoffrey Skelley is a senior elections analyst at FiveThirtyEight. https://fivethirtyeight.com/features/republicans-senate-2024-map/, Dec. 7, 2022.

In a bit of **electoral** déjà vu, Democratic Sen. Raphael **Warnock won Georgia's runoff** on Tuesday, almost two years after he won **a special election runoff** to help hand Democrats a narrow 50-50 majority in the Senate via Vice **President Kamala Harris's tie-breaking vote.** This time around, Warnock topped Republican Herschel Walker to earn a full six-year term, which will have major ramifications for how the new **Senate will conduct** business in January. Warnock's win gives Democrats 51 seats — including **independent Sens. Angus King of** Maine and **Bernie Sanders of Vermont** — so the Democratic caucus will no longer have to constantly rely on Harris to break ties. Democrats will also now have majorities on each **committee and will** be able to more easily confirm President Bidens judicial appointments. Yet if we look even further into the future, it turns out the Georgia outcome could also play a role in deciding which party controls the Senate after the next election. The good news for Democrats **is** that they will have 51 seats instead of 50, which gives them a chance to maintain control even if they lose one seat, **depending on** whether the next vice president is a Democrat or Republican. The good news for Republicans, however, **is that the** 2024 Senate map puts them in a better position to take control of the chamber than it does for Democrats to hold onto it.

Figure 14: Skelley Card, Classification Model, Regression Model with Original Proportions, Regression model with Modified Proportions (for original(s), see Figure 7).

Classifier

The win gives 51 seats including King of Maine Sanders of Vermont — the Democratic no longer have to constantly rely to

Skelley 12/7 Geoffrey Skelley is a senior elections analyst at FiveThirtyEight. https://fivethirtyeight.com/features/republicans-senate-2024-map/, Dec. 7, 2022.

In a bit of **electoral déjà vu, Democratic** Sen. Raphael Warnock won Georgia's runoff on Tuesday, almost two years after he won a special election runoff **to** help **hand Democrats** a narrow 50-50 majority in the Senate via Vice President Kamala Harris's tie-breaking vote. This time around, Warnock topped Republican Herschel Walker **to earn** a full **six-year term, which will** have major ramifications for how the new Senate will conduct business in January. Warnock's win gives Democrats 51 seats — including independent Sens. Angus King of Maine and Bernie Sanders of Vermont — so the Democratic caucus will no longer have to constantly rely on Harris to break ties. Democrats will also now have majorities on each **committee and** will **be able** to **more easily** confirm President Bidens judicial appointments. Yet if we look even further into the future, it **turns** out the Georgia outcome could also play a role in deciding which party controls the Senate after the next election. The good news for Democrats is that they will have 51 seats **instead of 50, which gives them** a chance to **maintain** control even if they lose one seat, depending on whether the next vice president is a Democrat or Republican. The good news for Republicans, however, is that the 2024 Senate map puts them in a better position to take control of the chamber than it does for Democrats to hold onto it.

```
for card in train_data:
    for sentence in card_sentences:
        for word in sentence_words:
            entries.append(current_scope_vector+current_word_vector)
            labels.append(label)
            current_scope_vector = new_scope_vector
model = LinearRegression().fit(entries, labels)
```

Figure 15: Linear regression model *pseudo*-code

```
for batch in data:
    entries,labels = parseCards(batch)
    model1.partial_fit(entries[0], labels[0])
    model2.partial_fit(entries[1], labels[1])
    model3.partial_fit(entries[2], labels[2])
```

Figure 16: SVM model *pseudo*-code

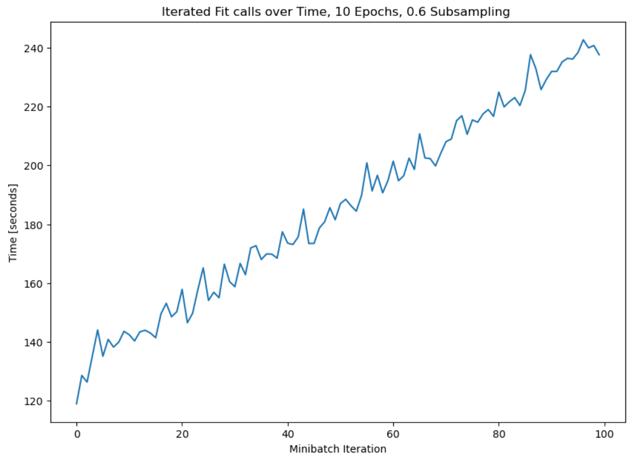


Figure 17: Time to train mini batches versus minibatch iterations, over 10 epochs.

```
x = user defined window
new class_vals = array of len(predicted_class_values)
for value in predicted_class_values:
    {n-x,...,nx} = 2x class values around value
    class_vals[value.index] = most popular of {n-x,...,n}
    OR maximum value of most popular {n-x,...,nx} if tied
return class_vals
```

Figure 18: Smoothing algorithm pseudocode