Word Vector Analysis

Yijia Dai

March 3, 2022

The preprocessing of the description of courses is done through converting to lowercase, removing punctuations and characters, and stripping.

Then, we vectorize each course's description into a word vector. The dimension of the word vectors is 5640 as the number of words in the given common words text file. Each dimension represents the words count for the specific word represented by that dimension. We visualize the density of our word vector to check if the original dictionary is good enough to represent the descriptions.

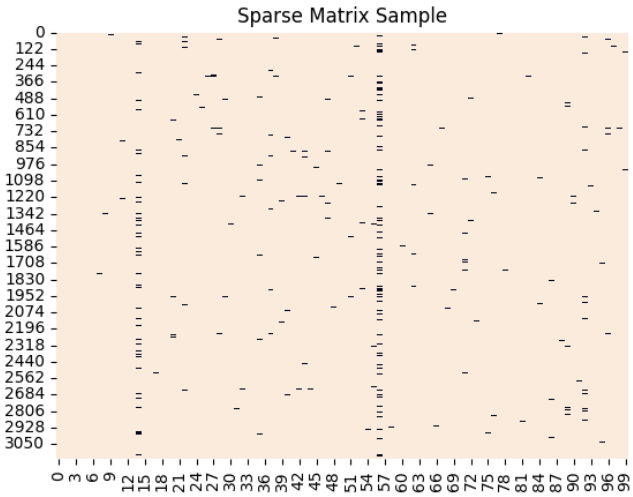

Figure 1: Sparse Matrix Sample.

As shown above, the represented vectors are relatively dense, indicating that the common words in the give text file is a good choice as the dictionary of the word vectors.

Next, we want to correctly classify the word vectors into either engineering or humanity. The 3162 data points are split into a 7:3 training set versus testing set ratio. For each word vector x_i, it is paired with a label y_i which is referenced by the given subject of the course. We utilized three models, Multinomial Naive Bayes, Support Vector Machine, and Binomial Logistic Regression, to classify course descriptions into engineering or humanity. As linear classifiers, they allow us to precisely find $\mathbf{y}$ vector and θ value.

The Naive Bayes classifier is

$$P(c = 1 \mid \mathbf{x}) = \sigma\left(\sum_i \log \frac{p(x_i \mid c = 1)}{p(x_i \mid c = 0)} + \log \frac{p(c = 1)}{p(c = 0)}\right)$$

where σ is the logistic function. With the $p(x_i \mid c)$ defined as an exponential, the Multinomial Naive Bayes classifier $P(c = 1 \mid \mathbf{x})$ would be a linear classifier in the form $P(c = 1 \mid \mathbf{x}) = \sigma\left(\sum_i \mathbf{w}_i^\top \phi_i(x_i) + b\right)$. Thus, the $\mathbf{y}$ vector and θ value is corresponding to $\mathbf{w}$ and $\sigma(b)$.

The SVM classifier is a classic linear classifier. However, it would not converge if the data points are not linearly separable. In our case, since the dimension is relatively high, we can still expect good performance with SVM. And, indeed, during the training process, we do receive warnings that after

10,000 iterations the classifier is still not converged. Thus, we add a soft margin, a tolerance for the stopping criteria. The SVM classifier, as a result, is simply

$$class(\mathbf{x}) = sign\left(\sum_i \mathbf{w}_i^\top \phi_i(x_i) + b\right)$$

Here, $\phi(x)$ is the kernel applied to the original vector which allows the classifier to be non-linear. However, as we are seeking yx, we shall simply set linear kernel. Thus, the $\mathbf{y}$ vector and θ value is corresponding to $\mathbf{w}$ and $-b$.

The Binomial Logistic Regression classifier is similar to the Multinomial Naive Bayes in terms of its classifier equation as a sigmoid function. And the corresponding $\mathbf{y}$ vector and θ value are $\mathbf{w}$ and $\sigma(b)$.

Before implementing the model, prepared as a comparison, the top 30 common words for Engineering and Humanity are found and shown.

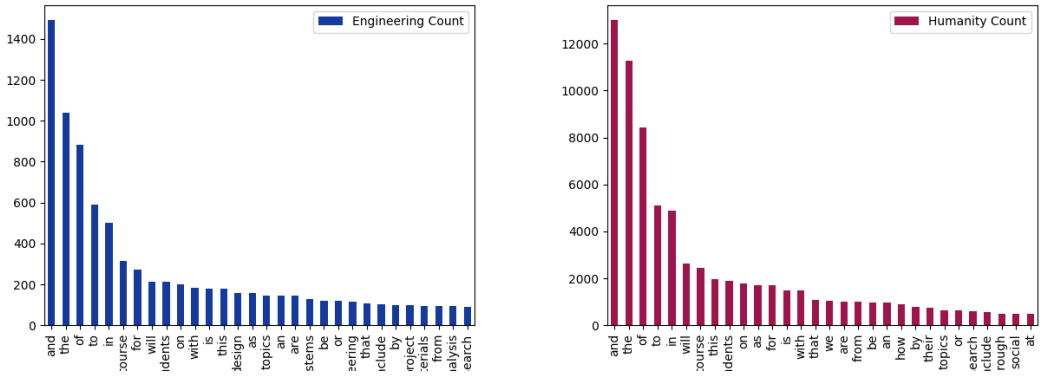

Figure 2: Common Words for Engineering (Left) and Humanity (Right).

Notice that two classes share many common words like "and", "the", and "or". But there are still many great words to distinct identity like "design" and "projects" belonging to engineering, and "social" belonging to humanity. There are potentials in using PCA in reducing dimensions. However, since there are still distinctions on the relative frequency of using the common words, we keep all dimensions for maintaining the features of data.

Now, let's see the results. The log probability of the word vector $\mathbf{y}$ given by the Naive Bayes classifier gives us the following result in Figure 3. We can see the similarity between the heated words between the trained $\mathbf{y}$ and the original data set within each class. The accuracy rate for this classifier has an average about 0.88, which, later we would see, is relatively low comparing to the other models.

Figure 4 gives us the resulted word vector $\mathbf{y}$ calculated by the Linear Support Vector Machine. The direction of the Engineering vectors is very different from the previous result as seen in the figure. There is more emphasis on the keywords that differentiate the classes. It is a result of the property of separating hyperplane since it is aiming for the largest margin for the support vectors. Also, since the algorithm not always converge, the models trained in three cases are different, but showing relatively stable accuracy of avereage 0.925.

The Logistic Regression gives us an accuracy of around 0.927. Here, we show the coefficient assigned to each dimension in the regression model. The assigned weights differ in each iteration of training. As you can see in Figure 5, words with strong characteristics, either in engineering or humanity, are assigned with high weights.

For each of the courses, we have a confidence level of its classification. If we rank, for both Engineering and Humanity, the most confident confident courses to the least, it would provide interesting insights into both the course description and our model. The overview over all courses are shown in Figure 6.

The courses that attain the maximum and minimum values of $yx - \theta$ are shown as follow in Figure 7. Some of the most confident Engineering classes are AEP 4450, ECE 4380, etc. And for Humanity

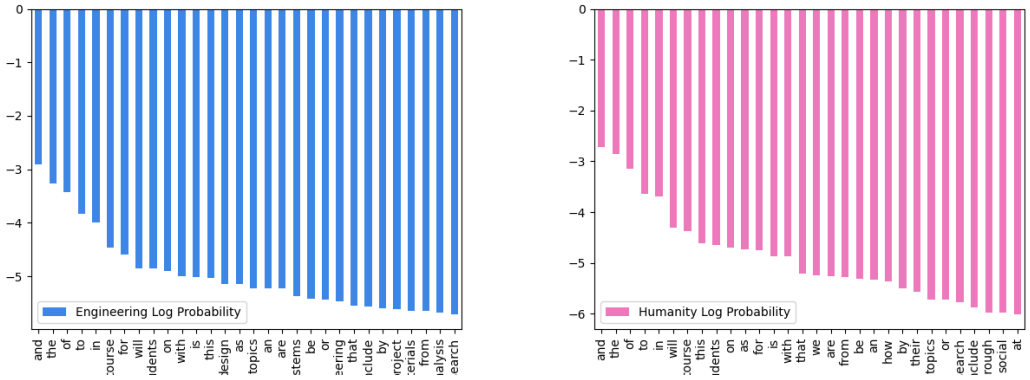

Figure 3: Common Words for Engineering (Left) and Humanity (Right) given by trained Naive Bayes Classifier.

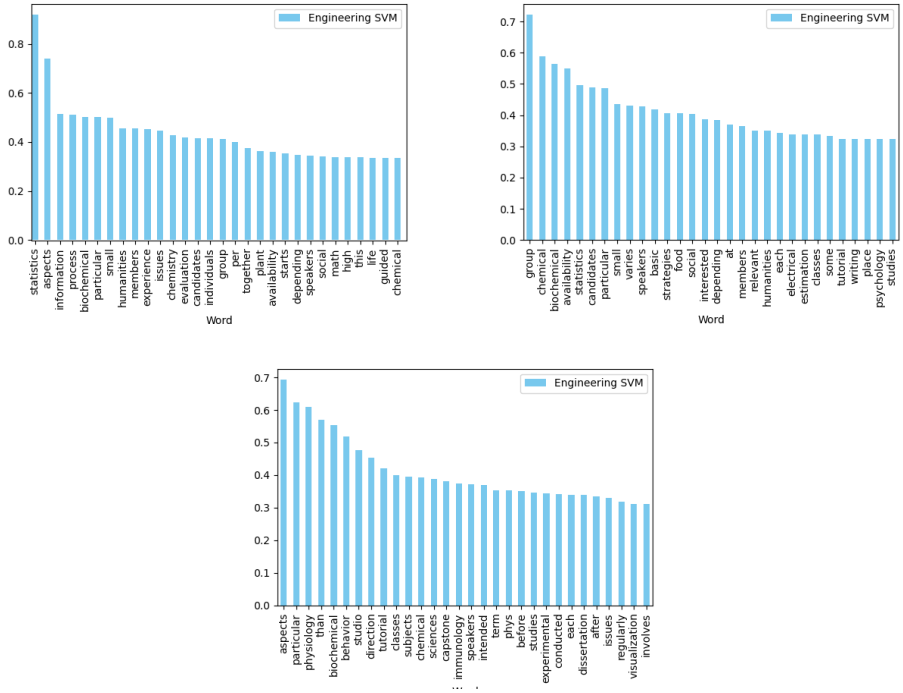

Figure 4: SVM for Engineering Vectors in three training cases.

are YORUB 1109, LAW 7840, etc. And for the least confident courses, we can perceive plenty of the misclassified ones.

Overall, these three models approach the classification task with different means and techniques. And as seen, all of them give persuasive results after iterations of training. And, in the future, allowing us to further optimize the models combining multiple advantages inherent by different models.

Some of the improvements we can make in the next step are PCA to decrease dimensions, bagging and cross-validation to better utilize the given data, and applying kernels and utilizing nonlinear models to train a more flexible classifier.

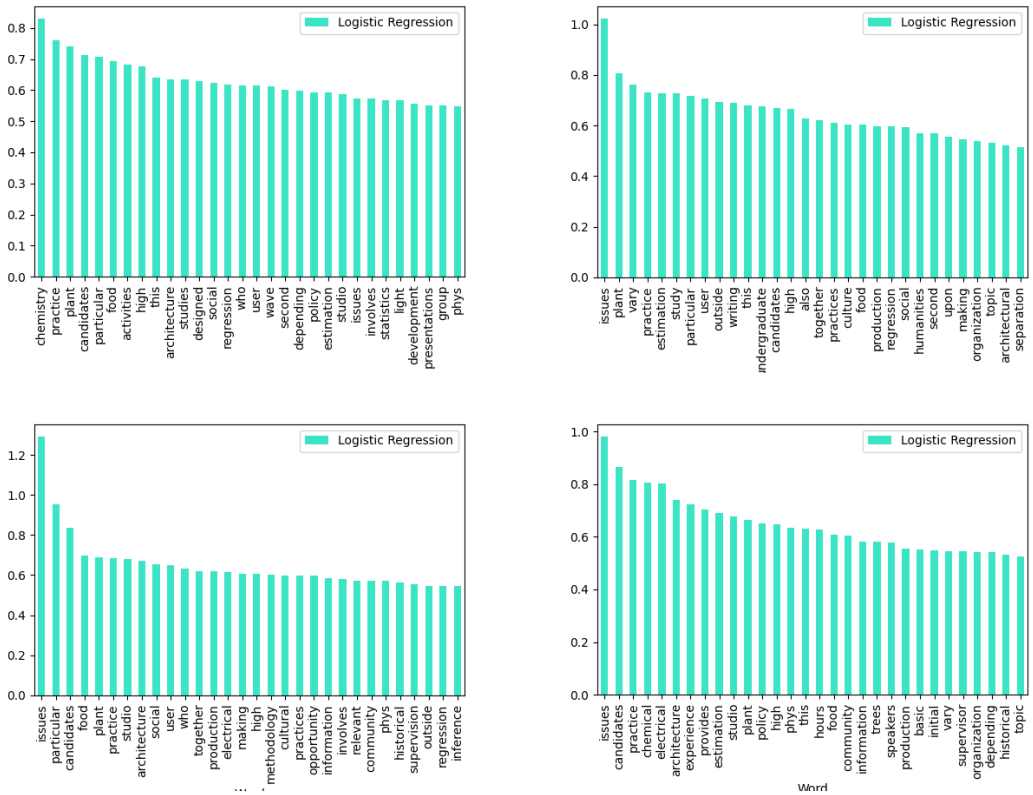

Figure 5: Dimension Assigned Weights for Logistic Regression.

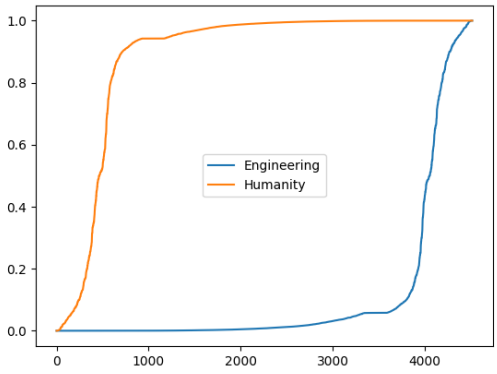

Figure 6: Sorted Courses According to Confidence Level.

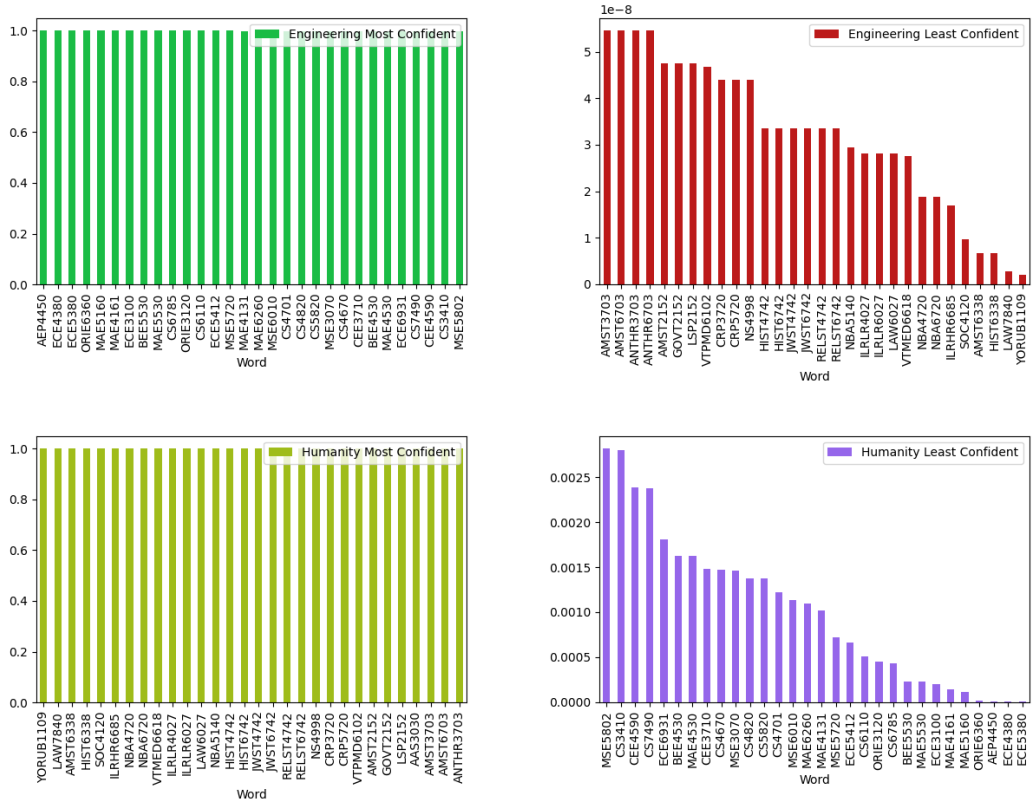

Figure 7: Most and Least Confident Courses for Engineering and Humanity.