Aidan Tambling, Sharan Majumder

CIS4930

12/2/2023

Final Project Report

Link to Video Tour

https://youtu.be/274-y6Pm-8k

Motivation

Email service, as a widely depended-upon communication medium, is increasingly relevant in our technologically oriented world. It is important to be able to sort emails according to their content to provide a more robust service to users. Therefore, our model seeks to classify emails according to their body content into several categories.

Proposed Implementation vs Actual Implementation

Upon selecting the focus of our semester project, we developed an implementation timeline. We intended to spend September compiling and cleaning data from a representative email dataset in order to use it for our model. In October, we would implement the model itself and begin the process of training it. And in November, we would test, evaluate, and integrate the model.

Our implementation timeline was not realized as we had envisioned. In September, we ran into trouble in trying to compile and clean an email dataset, which set back our progress on the project. We found it difficult to find a large, representative dataset of emails which we could use to train our model. This setback made October more challenging, as we had to make up for lost time in implementing and training the model. However, we were able to successfully test and evaluate the model in November, keeping us on schedule.

Implementation Specifics

To implement our model, we chose to use the Enron Corpus, a database of over 500,000 emails, as our dataset. The emails are sourced from former Enron email servers and were generated or received by various employees of Enron in the early 2000s.

Due to the expansive nature of the Enron Corpus, we chose to take random subsets of emails from the database of 500,000. We extracted both a training and testing subset, each of size 10,000, using a random sample of the '.csv' file which represented the Corpus in its entirety.

We chose to train the model based on the body content of emails. Although header content, such as sender, recipient, and date, could be relevant in email classification, we felt that the nature of the dataset might lead to overgeneralization in this regard. For example, by including emails' "sender" data in the training process, the model might cluster emails together based on irrelevant information. Indeed, the email dataset has only ~150 employees associated

with it, all of whom have email addresses ending in "@enron.com." This extraneous data could potentially compromise the model and lead to inaccurate classifications.

In implementing the model, we use an unsupervised approach. The Enron Corpus contains only emails, not any sort of classifiers or indices. Therefore, in order to classify these emails, we must let the machine learning model cluster the data according to its own understanding of the data.

We optimized the model to use a number of clusters that would yield the most accurate results in classification. Incidentally, this optimal number was seven.

The program can be explained in a sequence of steps:

1. Import relevant libraries for machine learning applications
2. Split excessive dataset into reasonably-sized training and testing datasets
3. Train the model
    a. Read in the training dataset
    b. Parse each email for its header content and especially its body content
    c. Vectorize the email body content container and train the k-Means model on it
    d. Optimize the k-Means model in terms of number of clusters (optimal count = 7)
    e. Associate each email with its assigned cluster
4. Test the model
    a. Read in the testing dataset
    b. Parse each email for its header content and especially its body content
    c. Vectorize the email body content container and generate predicted clusters with the k-Means model
    d. Associate each email with its predicted cluster
5. Evaluate the model
    a. Visually examine samples from the generated clusters and confirm their association
    b. Generate silhouette score for both training and testing cluster assignments
    c. Generate cohesion and separation scores for the model
    d. Examine cluster centroid distances in the form of a matrix

Limitations

The unsupervised approach to this classification problem is limited in that the clusters are unlabeled. Therefore, it may be difficult to understand why an email is classified into a given cluster and what the significance of this classification may be.

Another limitation to this classification model is the dataset that was used to both train and test the model. The emails that are interpreted by the model were sourced from a corporate domain. As a result, these emails might include mostly (or even exclusively) content that relates to such an atmosphere. This limitation could impact both the model's generalizability and applicability to real-world problems.

Performance

The model categorized the training emails into seven clusters. A drawback of unsupervised learning is that the model's results are not labeled, so speculation is required to make meaning of what these clusters might represent. Below is an excerpt of an email body from each of the classifier's clusters, with a potential manually-generated label for that cluster indicated.

Cluster 0 (Newsletters/Advertisements?)	"NETWORK WORLD NEWSLETTER: NEAL WEINBERG on PRODUCT REVIEWS 01/15/02"
Cluster 1 (Casual/Informal?)	"just chill girl. -----Original Message----- From: Erin Richardson"
Cluster 2 (Legal/Administrative?)	"Dan - here are the special provisions that were provided by EnergyUSA."
Cluster 3 (Scheduling?)	"Dayton Power is away from the office on Friday and Monday for the new year holiday."
Cluster 4 (Internal Requests?)	"Please share this with your staff, and ensure everyone who needs to attend gets signed up for a time next week."
Cluster 5 (Market/Trading?)	"We received a copy of an Assignment, effective as of May 1, 2000, between= =20 Mega Natural Gas Company"
Cluster 6 (System-Generated?)	"Start Date: 12/29/01; HourAhead hour: 23; No ancillary schedules awarded. No variances detected."

The model was evaluated according to its silhouette score, separation and cohesion scores, and its centroid distance matrix.

Silhouette Score

Silhouette Score is a measurement which ranges from -1 to 1. It indicates both the separation and distinguishment of a given model's clusters. A score of 1 indicates that the clusters are well separated and very much distinguished from one another; a score of -1 indicates the opposite.

Training Emails Result: 0.031757380960461254

Testing Emails Result: 0.03219699861039162

Separation Score

Separation score measures the distance between a model's generated clusters. A higher separation score indicates that the clusters are well-spaced, a lower score indicates the opposite.

Result: 1.55355737642792

Cohesion Score

Cohesion score measures the closeness of data within a given cluster. A higher cohesion score indicates that a cluster's data points are spread apart, whereas a lower score indicates that the points are concentrated together.

Result: 8665.704858490091

Centroid distance matrix

The centroid distance matrix displays the distances between the centroids of each cluster. The higher each distance in the matrix is, the more spread apart those centroids are.

Result:

0	0.38698652	0.38095415	0.36446778	0.60067753	0.76904884	0.88584401
0.38698652	0	0.37160375	0.29390371	0.5857317	0.7545845	0.85862804
0.38095415	0.37160375	0	0.32049645	0.50151117	0.74582664	0.86957666
0.36446778	0.29390371	0.32049645	0	0.54022415	0.71017291	0.80089481
0.60067753	0.5857317	0.50151117	0.54022415	0	0.86052284	0.9685976
0.76904884	0.7545845	0.74582664	0.71017291	0.86052284	0	1.04870977
0.88584401	0.85862804	0.86957666	0.80089481	0.9685976	1.04870977	0

Discussion and Conclusion

In general, the performance results indicate that the model performed somewhat well, but not excellently. The Silhouette Scores for both the training and testing email datasets indicate that the model's clusters, while not overlapping, are in close proximity to each other; they are not entirely distinguished from one another. The high cohesion score and low separation score also imply that the model's clusters are close to one another and that, within a cluster, the data may be more spread out than is desirable. The centroid distance matrix demonstrates a similar pattern, with the distances between each centroid being relatively small. In essence, the model's clusters are distinct, and do not overlap, but they are in close proximity to each other, and their boundaries are not defined ideally.

We believe that these results can be partially explained by the given dataset. Given the corporate atmosphere that the emails were sourced from (the Enron workplace), it might be harder for a classifier to neatly separate emails. The model might benefit from a more representative dataset.