Building A Curriculum for Robo-Poets

W. Yandell

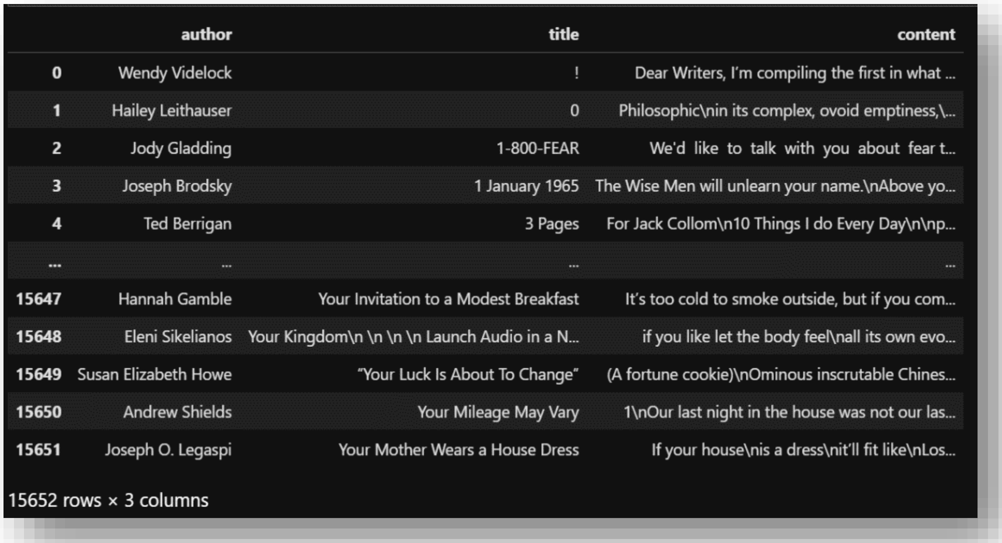
University of Denver

Building A Curriculum for Robo-Poets

In this first stage of research into the Poetry Foundation dataset consisting of 15,600+ poems, the primary goals have been as follows: clean (carefully), vectorize (expansively), and explore (poems in their entirety and as lines for composition and construction respectively). These map onto the larger goal of constructing text generators capable of producing poems which are coherent and similar to those in the dataset.

Data

The data of interest comes from the Poetry Foundation website by way of John Hallman on Kaggle (https://www.kaggle.com/johnhallman/complete-poetryfoundationorg-dataset). This dataset retains 15,638 unique instances of strings labeled 'content' as well as columns denoting the 'author' (3310 different authors represented), 'title', along with the Poetry Foundation ID and an index for each document. The following figure gives a snapshot of the raw data:



	author	title	content
0	Wendy Videlock	!	Dear Writers, I'm compiling the first in what ...
1	Hailey Leithauser	0	Philosophic\nin its complex, ovoid emptiness,\...
2	Jody Gladding	1-800-FEAR	We'd like to talk with you about fear t...
3	Joseph Brodsky	1 January 1965	The Wise Men will unlearn your name.\nAbove yo...
4	Ted Berrigan	3 Pages	For Jack Collom\n10 Things I do Every Day\n\np...
...	...	...	...
15647	Hannah Gamble	Your Invitation to a Modest Breakfast	It's too cold to smoke outside, but if you com...
15648	Eleni Sikelianos	Your Kingdom\n \n \n \n Launch Audio in a N...	if you like let the body feel\nall its own evo...
15649	Susan Elizabeth Howe	"Your Luck Is About To Change"	(A fortune cookie)\nOminous inscrutable Chines...
15650	Andrew Shields	Your Mileage May Vary	1\nOur last night in the house was not our las...
15651	Joseph O. Legaspi	Your Mother Wears a House Dress	If your house\nis a dress\nit'll fit like\nLos...

15652 rows × 3 columns

The variety of the data cannot be understated even compared to other corpora. While a standard corpus may have some number of unique words and stylistic grammars, for poems these are regularities or commonalities in the dataset that must be carefully engaged. Many common transformations such as lemmatization, stemming, or even certain types of filtering could incidentally remove characteristic elements which may be confounding in another problem but necessary to one like this.

In its preprocessed state, the dataset's relevant features are unstructured or latent and thus ill-suited for the modeling required to create a robust and coherent generator. Thus bad data must be wrung out, new features must be created, and the tokens must be vectorized to manage the space complexity while retaining as much information as possible about the latent structural and semantic spaces of the documents.

Preparation/Processing

Because the relevant machine learning algorithms require numeric features, the content of each poem must be parsed, cleaned, aggregated, and vectorized. The complete processing pipeline includes the following: removing rows with duplicate 'content' values, splitting content into lines, clearing noise from lines, tokenizing poems/lines, aggregating, exploring, tagging, outlier detection and removal, and vectorization.

Cleaning too aggressively or at the wrong step could erase relevant structures, such as line breaks, though cleaning insufficiently could result in spurious results from distinctions without difference. Thus, multiple options were explored across the Spacy, NLTK, regular expressions, and genism packages with a final result that balances processing concerns and project goals such that each poem is split into lines prior to heavy-duty text cleansing applying regular expressions, natural python expressions, and tokenization by the Natural Language Toolkit. Of course, other relevant and arguably as or more consequential information will almost certainly be lost as is the nature of loss in processing. Human poets too are challenged by incomplete information as well as loss, so this shouldn't be an overwhelming obstacle.

In early explorations, filtering was comparatively minimal to avoid incidences of false identity, however this introduced merely alternate forms of false identity, as in situations like the use of a contraction being converted to a non-relevant signifier merely by the brute force but restriction on punctuation and non-ASCII alone or being transformed in such a way that the grammar and structure of the line made for a completely different effect. This exploration of punctuation as token may have to occur at some point, in order to produce coherent and similar works to the underlying texts behind any given text transformation result. Further, important syntactical analysis constructs can be complicated by the nuances of parsing such that lexical analysis in particular becomes wrought with error, producing a trade-off dynamic for each of the particular goals.

A sequential ordering between preparation and analysis becomes challenging when they have such a recursive relationship. Exploratory analysis provides insight which feeds back to processing which feeds forward to new analysis. For instance, there were some texts in this dataset which had only one line of a thousand words and others with thousands of lines caused by scraping error. Some poems, after origin look-up, were revealed to be missing the entirety of their content save the epigraph. This plastic boundary is more complicated by the fact that much of what is being created is merely training data for a number of deep learning models for predicting the structure of ‘the next line’ and then generating a line that meets that standard or for detecting whether or not the generated text fits in with the others.

In any case, the processing pipeline prior to vectorization at present offers two less noisy versions of the raw data, one for entire poems to train document generation (or line prediction if you rather), and one for lines to train text generation. In the poem dataset, there remain 14,620 instances to train the Doc2Vec model so that the data may be passed to additional machine learning algorithms for categorization and eventually generation. The Doc2Vec model converts these documents as word lists into 14620 vectors compressed to 2000 numeric features under a vocabulary of over 106,000 words.

Analysis

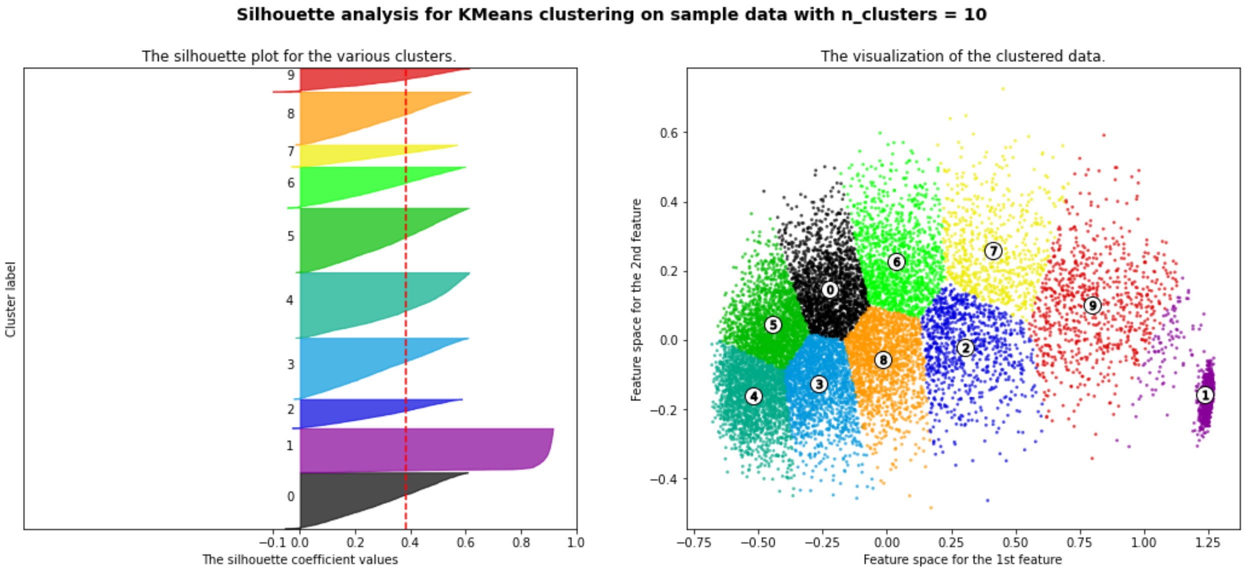
After vectorizing, leaving the two deep learning models and frequency distribution features for further semester research, I performed clustering via K-Means, DBSCAN, Hierarchical, and Gaussian Mixture Model algorithms. Because clustering tends to perform best with only two features, I used Principle Component Analysis on the Doc2Vec vectors for the poems.

To tune the K-Means model, I iterated over various numbers of clusters after performing scree and knee analysis then compared the silhouettes scores and plots to determine the best number of clusters wherein the thickness of the plot denotes homogeneity across clusters and the score for each cluster exceeds the average. For DBSCAN, I constructed a gridsearch across candidate epsilon informed by knee analysis on pairwise distances and a range of minimum sample values. For hierarchical clustering, I used cosine cosine similarity and manual iteration, and for Gaussian Mixture Modeling, I used BIC over a grid of covariance types. Graphic outputs of these may be found in the appendix with the full process found in the accompanying Jupyter Notebook.

While none of these clustering algorithms provided ideal results (silhouette score < 0.5 for nearly all with more than 2 clusters) on just the text vectors, these may provide assistive labels for future classification, clustering, or generation alone or combined with the elided features.

Appendix I: Visual Outputs

K-Means Clustering

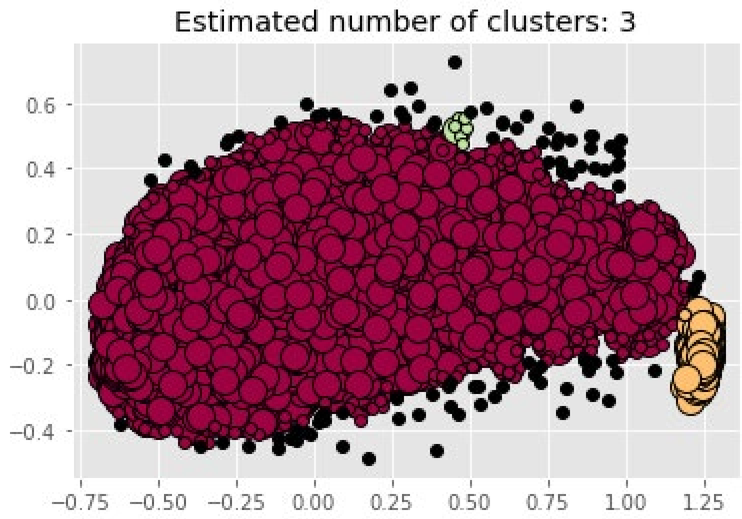


Silhouette analysis for KMeans clustering on sample data with n_clusters = 10

The silhouette plot for the various clusters.

The visualization of the clustered data.

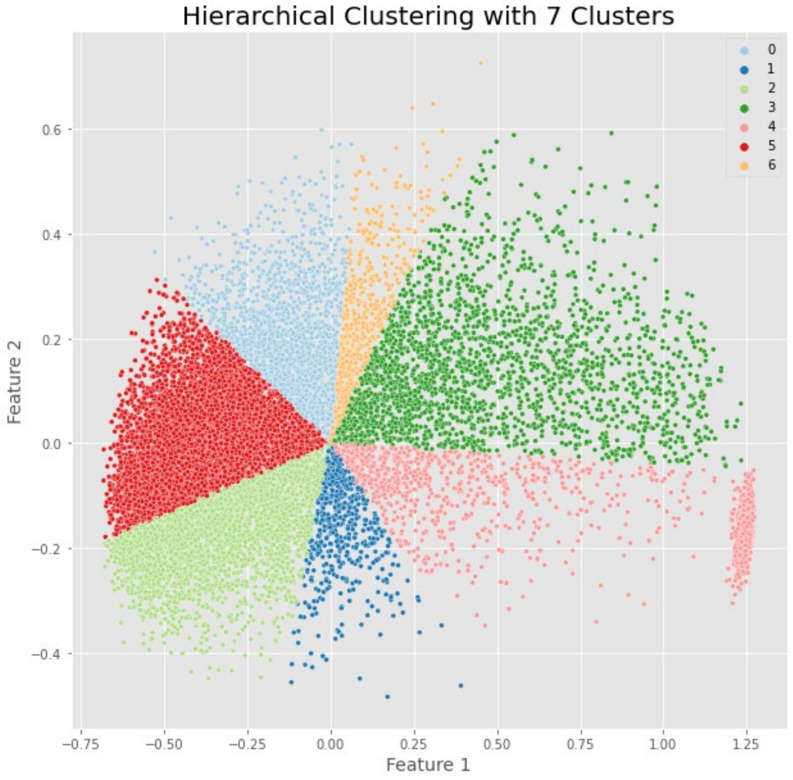
DBSCAN Clustering



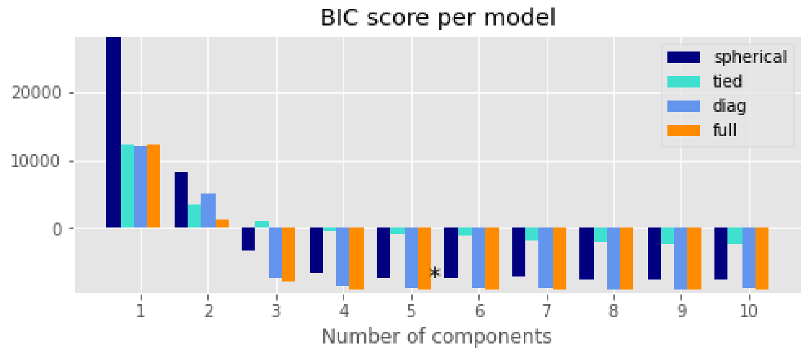
Estimated number of clusters: 3

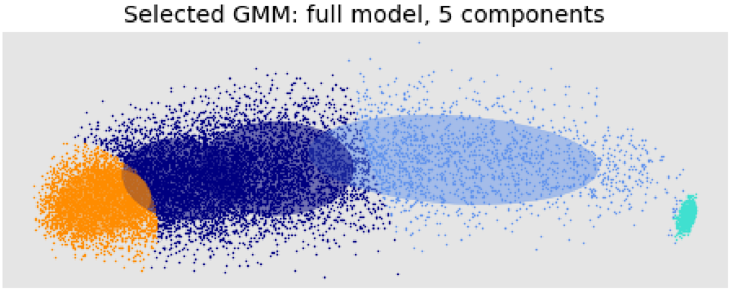
*black points are noise

Hierarchical Clustering



Gaussian Mixture Model Clustering


BIC score per model
spherical
tied
diag
full
*
Number of components


Selected GMM: full model, 5 components