ArXiv Data and Article Classification

Junwon Choi, Reuben Rosenberg, Mabel Huynh

Dataset Description

- ArXiv Dataset from Cornell University
- Over 1.7 million research papers and articles
- Contains: article titles, authors, **categories**, **abstracts**, full text PDFs, and more
- 158 subcategories in the system
- Each article can be placed into multiple categories

We only pulled articles from 2018 to 2020 (inclusive):

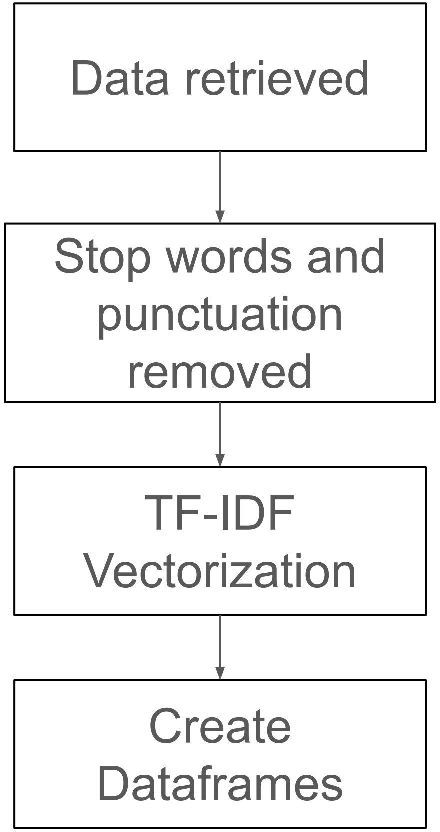
- 15,771 articles
- 3764 unique category combinations
- 158 unique subcategories
- 20 unique main categories

Questions/Motivations

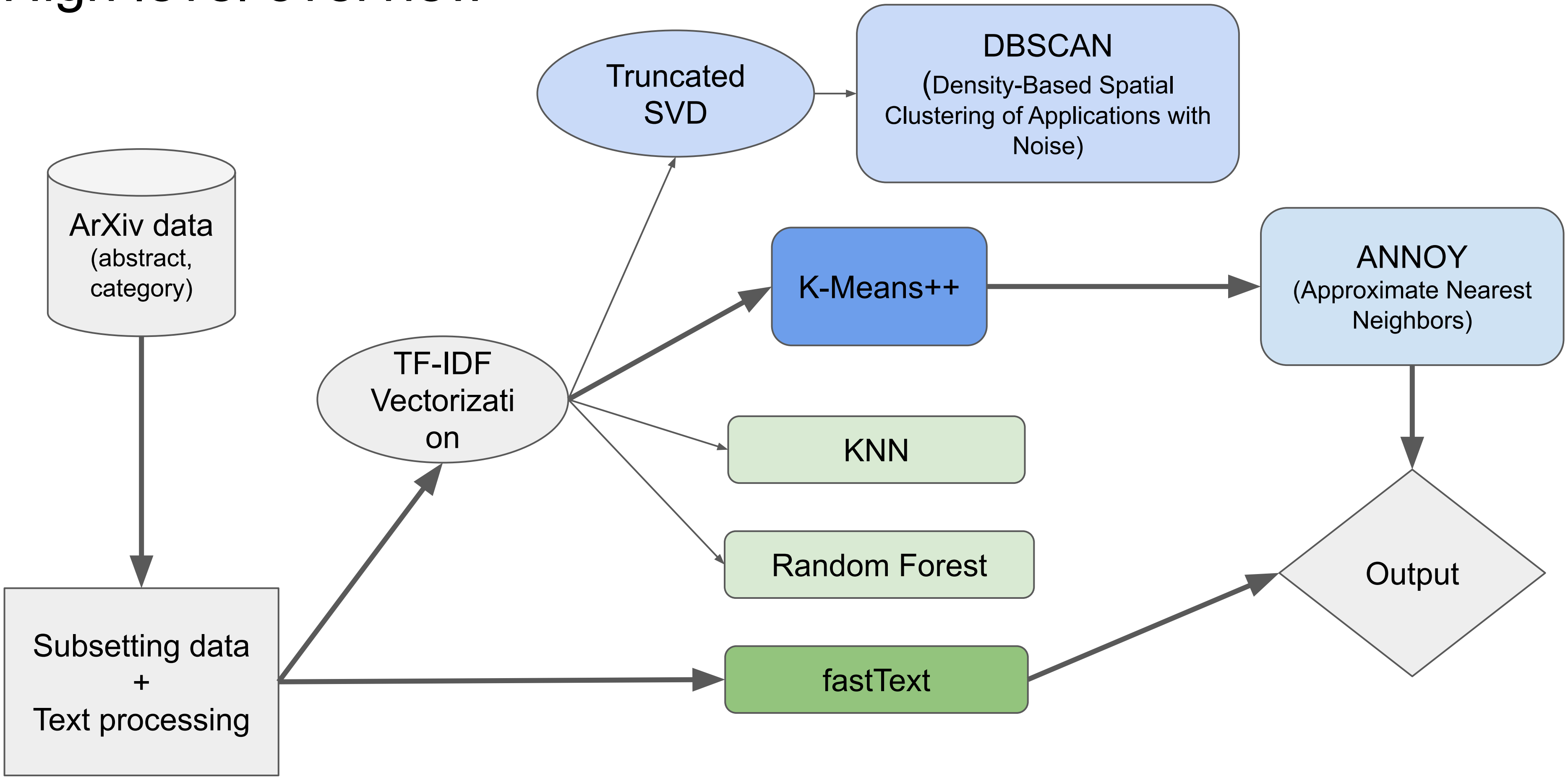
1. How can one be expected to effectively navigate such a large number of classes when performing research?
2. How does one know which categories to assign to their own papers?
3. Is there an efficient way to organize such a great variety of papers, while reducing the number of difficult to navigate categories?

Preprocessing

- Each article has a lot of information, we only care about abstracts and classifications
- Similar to homework 3, we must remove all stop words and punctuation
- Use TF-IDF vectorization to convert each abstract into a vector
- Create a dataframe with each abstract vector and its associated category list



High level overview

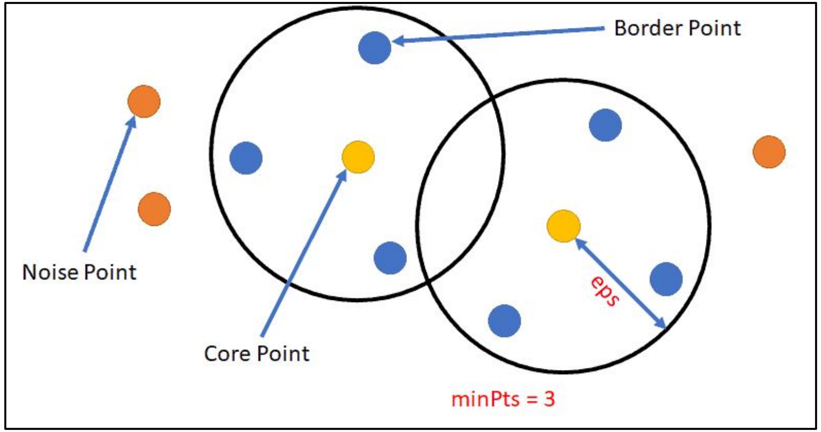


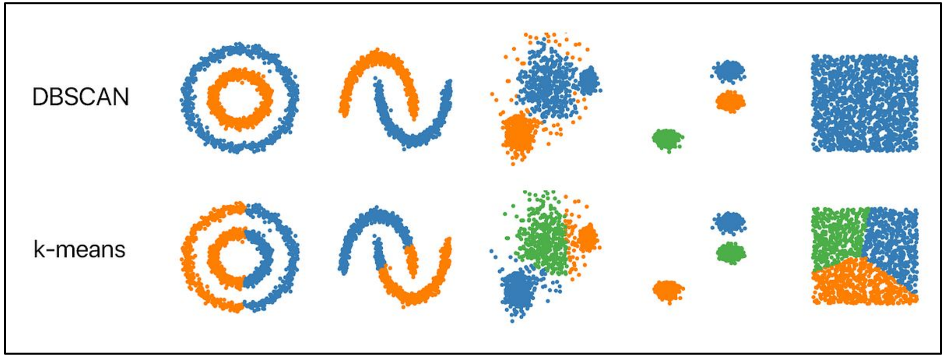
Unsupervised Learning

- Goal
  - Determine if an organization system with fewer categories is possible
  - Create a clustering system to pass into the paper retrieval tool
- Used 2 different clustering algorithms
  - DBSCAN
  - K means
- Models are evaluated by silhouette score, a measure of how well-defined and distinct the clusters are

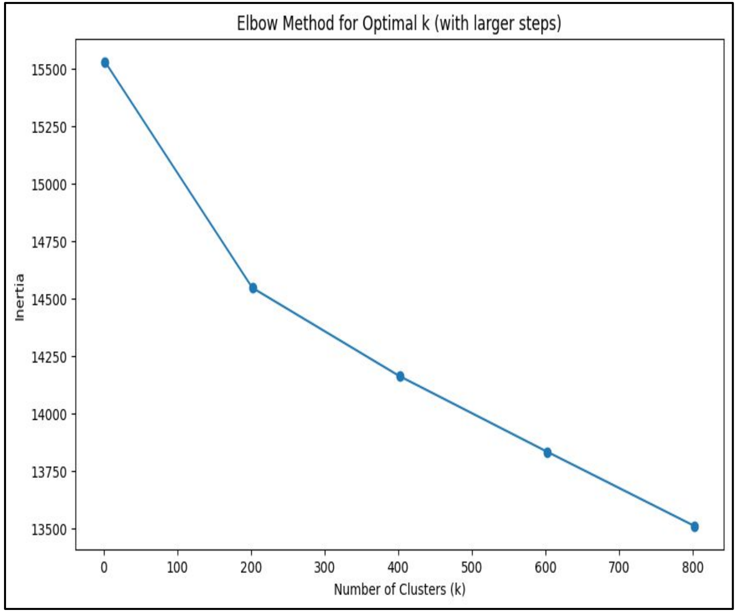
DBSCAN

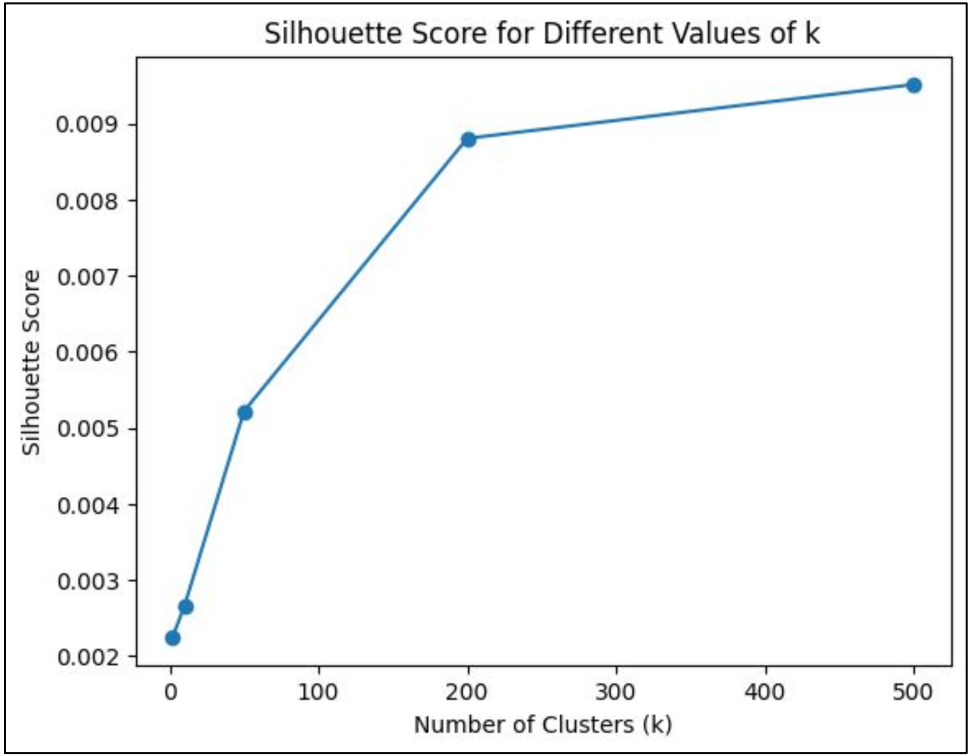
- Density Based Spatial Clustering in Applications with Noise
- Advantages
  - Can learn arbitrary shapes
  - Number of clusters is not specified
- Disadvantages
  - Slow with high dimensional data
    - Solved by using Truncated SVD
- Best silhouette score of 0.34, but classified large number of points as noise. Data structure does not work well with this model





K-Means (K-Means++)



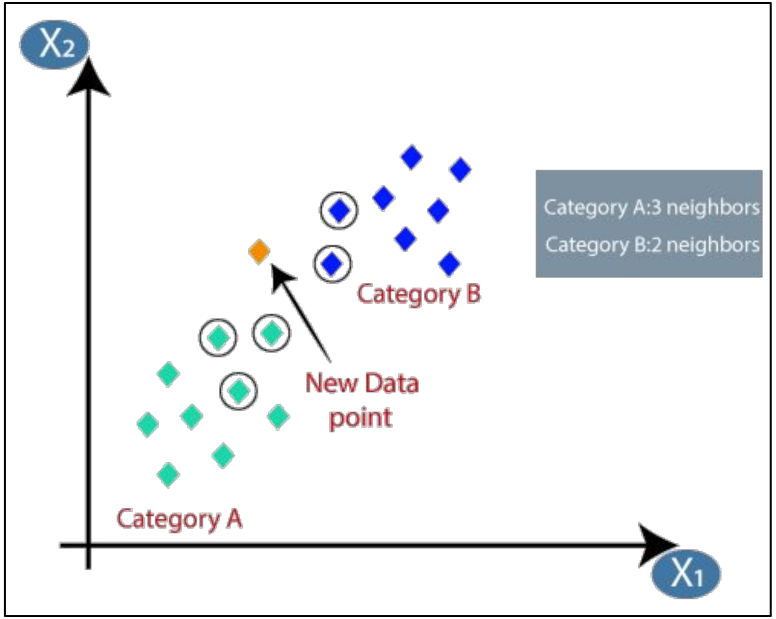


K-Means (K-Means++)

- Best Silhouette Score ~ 0.01
- Worse score, but we believe it fits the data more appropriately
- Using our trained K means on 500 clusters, we pass it to ANNOY model
  - Based on inertia values (elbow method) & silhouette scores, we found the model preferred greater amounts of clusters (future improvement)

Paper Retrieval Tool

- Implement ANNOY model with the K means results
- Approximate Nearest Neighbors (Oh Yeah)
- Rather than using it as a classification tool, use it to return similar papers



Paper Retrieval Tool Results

"Testing High-dimensional Covariance Matrices under the Elliptical Distribution and Beyond" (math.ST - Statistics Theory)

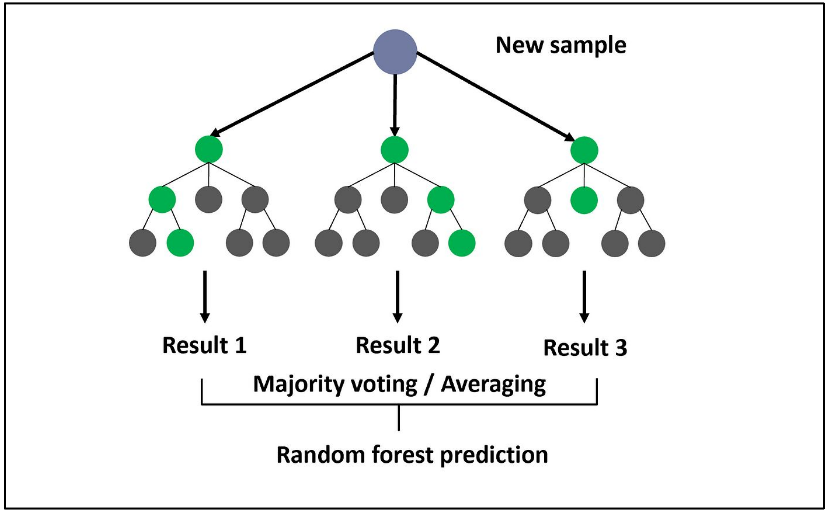
id	title	abstract	category	year
1511.05710	Complex-Valued Gaussian Processes for Regression	in this paper we propose a novel bayesian sol...	[cs.LG]	2018
1808.01123	Covariance Matrix Estimation from Linearly-Cor...	covariance matrix estimation concerns the pro...	[cs.IT, math.IT]	2019
2001.09187	Certified and fast computations with shallow c...	many techniques for data science and uncertai...	[math.NA, cs.LG, cs.NA, stat.CO]	2020
1805.07460	Fast Kernel Approximations for Latent Force Mo...	a latent force model is a gaussian process wi...	[stat.ML, cs.LG]	2018
1811.04956	Recovery Map for Fermionic Gaussian Channels	a recovery map effectively cancels the action...	[quant-ph]	2019
1604.03182	Cascade and locally dissipative realizations o...	this paper presents two realizations of linea...	[quant-ph, cs.SY, math.OC]	2018
2006.01448	Sparse Cholesky covariance parametrization for...	the sparse cholesky parametrization of the in...	[stat.ML, cs.LG]	2020
1802.01513	Covariance Matrix Estimation for Massive MIMO	we propose a novel pilot structure for covari...	[cs.IT, math.IT]	2018
1708.06296	Spiked sample covariance matrices with possibl...	in this paper we study the convergent limits ...	[math.PR]	2020
1602.05522	Central limit theorems for functionals of larg...	in this paper we consider the asymptotic dist...	[math.ST, stat.TH]	2019

Supervised Learning

- How can we effectively classify our own paper?
- Implement 3 supervised models
  - K Nearest Neighbors
  - Random Forest
  - FastText
- Checked classification on 21 broad categories
  - Math, Physics, Statistics, etc.
- And 157 subcategories
  - Math- Machine Learning, Statistics - Theory, etc

Random Forest

- Train many decision trees, and output the aggregate decision
- Specific Categories: 11.525%
- Broad Categories: 50.845%

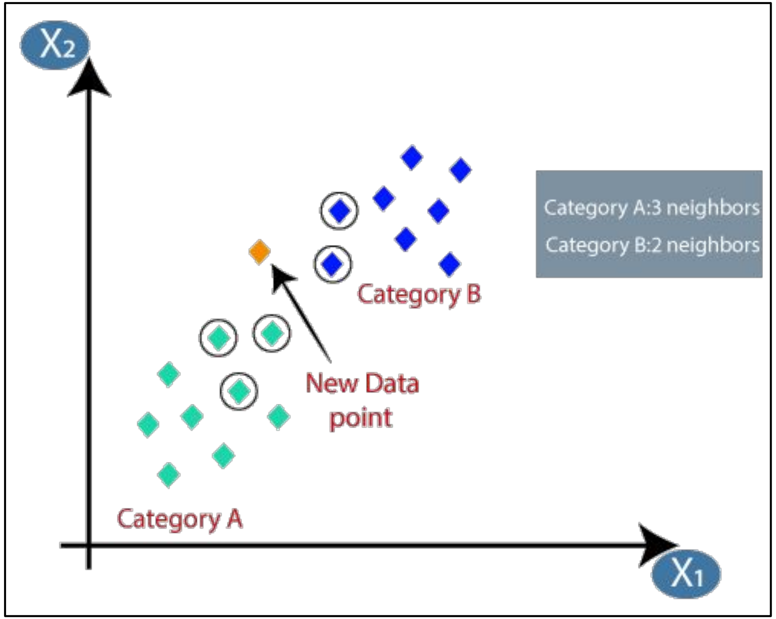


- Worst performance in both specific and broad categories

K-Nearest Neighbors

- Select the category that is most common among the k nearest neighbors
- Specific Categories: 20.342%
- Broad Categories: 56.4452%

- Best performance in broad categories



fastText

Efficient text classification model developed by Facebook AI

Key features:

- Word embeddings
  - For both words & sub-words
- N-gram features (best N-gram value was 1 for our dataset)
- Hierarchical softmax
  - Logarithmic reduction in the number of computations needed to compute the softmax probabilities. No need to compute probabilities for every label!
  - Instead, we follow a binary tree
- Faster than linear classifiers + complex NN

Results:

- Specific Categories: 27.179%
- Broad Categories: 54.892%

Continuous Bag of Words (CBOW):

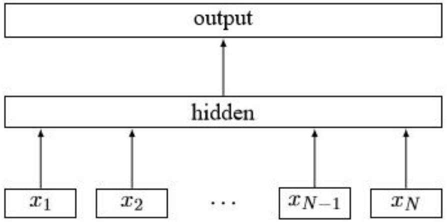
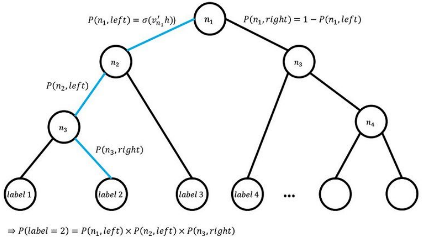


Figure 1: Model architecture of `fastText` for a sentence with N ngram features $x_1, \ldots, x_N$. The features are embedded and averaged to form the hidden variable.



Sample 1 Input Actual Category: **[math.st | econ.EM | stat.ME]**
Sample 1 Output Prediction: **[state.ST]** 19.7% | **[stat.SH]** 19.96% | **[stat.ME]** 17.6%

Conclusion

- Introduced 2 new tools to help navigate the ArXiv repository
- Future research could involve applying these processes to other large datasets. Or expanding on the number of tools
- Ideally these allow for faster research, and an easier way to upload papers and articles
    - Almost instant retrieval once clustering models are trained!
- Proposed new and efficient method for contextual search & retrieval
    - More tailored information than simple search using keywords in search bar

High level overview