A Poisson Multigraph Model For Clustering Scientific Abstracts

Eve Fraczkiewicz

Department of Statistics, University of California, Riverside

August 30, 2024

Abstract

Generative AI models like ChatGPT are notorious for regurgitating false information, and as they become more sophisticated, it becomes increasingly difficult to differentiate between AI-generated and “human-generated” text. As more evidence of these models being used in academic writing comes to light, it is important to develop methods for detecting AI-generated text to protect academic integrity. We propose a Poisson multiplex graph model that clusters scientific abstracts into three groups: AI-generated, human-written, or ambiguous, by utilizing n-grams. AI-generated text, like human-written text, tends to have distinct features. By observing shared n-grams across and between generated and written text, we can find these unique features and use them to cluster abstracts into their respective groups. To model these relationships, we construct a multiplex graph model in which abstracts are nodes connected by the number of their shared n-grams. We tested our method on two datasets, the first being a collection of 28,000 abstracts from COVID-19 research papers, and the second being a collection of 12,000 abstracts of various topics scrapped from *Nature*. To assess the quality of our clustering, we calculated the Jaccard indices and F-1 scores for each n-gram type. We successfully built an accessible model that directly accounts for n-gram statistics and is able to achieve reasonable clustering into generated and written abstracts. In the future, we anticipate using our insights to pool information across layers in the multiplex graph to guide the design of classification algorithms.

Introduction

The use of generative AI and large language models (LLMs) in scientific literature has been rapidly increasing. In 2023 alone, at least 60,000 articles, about 1% of all articles published that year, were found to be LLM-assisted (Gray, 2024). While some argue that they can be useful tools when used properly, the fact that they tend to output false information puts the integrity of the scientific community at risk. Thus, we set out to build a model that could differentiate between AI-generated and human-written text in scientific abstracts. We chose to focus on abstracts because it's reasonable to assume that most people using LLMs are using them to aid in summarizing their work, rather than writing the full paper.

We utilized a multiplex graph as the foundation for our model. A multiplex graph is an extension of the simple graph that allows for multiple edges between nodes and different types of edges within the graph. This allows us to capture the complex relationships found within real-world networks.

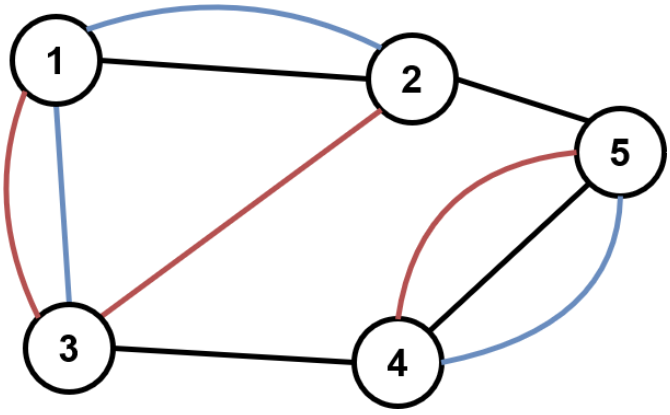


Figure 1. Simple representation of a multiplex graph. The red, blue, and black lines represent different types of edges.

To build this multiplex graph we extract multiple n-grams from each abstract. An n-gram is a collection of n-sized consecutive word groupings in a piece of text. For

example, a 1-gram list for “I like graphs” would be “I”, “like”, “graphs”, a 2-gram list would be “I like”, “like graphs”, and so on.

Observing n-gram statistics between abstracts can allow us to discover patterns and features that are unique to human-written and AI-generated abstracts and cluster them appropriately.

Methods

Building the Multiplex Graph

To build the graph for the model, we assign each node to be an abstract from an article in the dataset and the edges connecting them to be their shared n-grams. The number of edges connecting each node pair directly corresponds to the number of n-grams that they share. The different types of edges in this graph are the different types of n-grams. For simplicity the graph is represented in the form of a count matrix, where each row and column is an abstract and the values are their shared n-gram counts. A separate count matrix is made for each n-gram type.

Instead of just using pure n-gram counts, we also want to include n-grams with similar semantic meaning in the shared counts. To do this, we calculate the sentence embeddings of each n-gram in the dataset. Sentence embeddings are formed from word embeddings, which are real-valued vectors containing the semantic and contextual information of a word. Sentence embeddings are calculated by computing the weighted average of the words in an n-gram for each n-gram:

$$v_s = \frac{1}{|s|} \sum_{w \in s} \frac{\alpha}{\alpha + p(w)} v_w$$

and then removing the projections of the new average vectors on their first singular vector, in what's called “common component removal” (Arora et al., 2017):

$$v_s = v_s - uu^T v_s$$

After the sentence embeddings are calculated, we compare every possible abstract pair and calculate the cosine distance between every sentence embedding. If the distance is less than 0.1, then we count them as “similar” and add it to the count matrix.

Building the Poisson Clustering Model

To build our model, we first assume that the number of edges between nodes, which is our parameter of interest, follows a Poisson distribution. A Poisson assumption is ideal for our model because it is simple and flexible: it only has one parameter and it can approximate a binomial or normal distribution (Ranola et al., 2010).

To approximate the expected number of edges between nodes, we introduce several parameters to the model. For every node *i*, we assign a propensity p_i to form edges with other nodes. Thus, for every node *i* and *j* the mean number of edges between them is the product of their propensities, $p_i p_j$. To account for clustering, we assign a cluster assignment indicator c_i to each node and a cluster similarity matrix *R* whose entries quantify the relationships between clusters (Ranola et al., 2013). This culminates into the Poisson log-likelihood function below:

$$Poisson(c, p, R) = \sum_i \sum_{j \neq i} [x_{ij} \ln(r_{c_i c_j} p_i p_j) - r_{c_i c_j} p_i p_j - \ln(x_{ij}!)]$$

Since our model is dealing with thousands of nodes and edges, estimating parameters can be computationally expensive. Fortunately, we can use the MM (minorize-maximize) principle to iteratively maximize the likelihood function (Ranola et al., 2013).

Mutual Ranking

To determine statistically significant connections between nodes, we use what is called mutual ranking. To calculate mutual ranks, we take each node and all of its possible node pairs and calculate the residuals:

$$R_{ij} = h(Z_{ij}) - [h(\hat{\mu}_{ij}) + \tfrac{1}{2}h''(\hat{\mu}_{ij})\hat{\mu}_{ij}]$$

Here, the residuals are calculated by subtracting the observed edge counts to the expected edge counts. The expected edge counts follow a Poisson distribution, and since they are defined on non-negative integers, the p-values may be poor. To alleviate this, we use the delta method to update the residual calculation. We then calculate the negative log p-values of these residuals and rank them from smallest to largest.

Results

We tested our model on two datasets, the first being a collection of 28,000 Covid-19 related articles, and the second being a collection of 12,000 articles web scraped from the *Nature* journal, of various topics. For both datasets half of the articles are human-written while the other half are AI-generated. We used the purely Covid-19 dataset to control for topic variability and the scraped dataset as a more realistic example. The abstracts were pulled from each article and cleaned and the n-grams were extracted from each abstract.

Covid dataset:

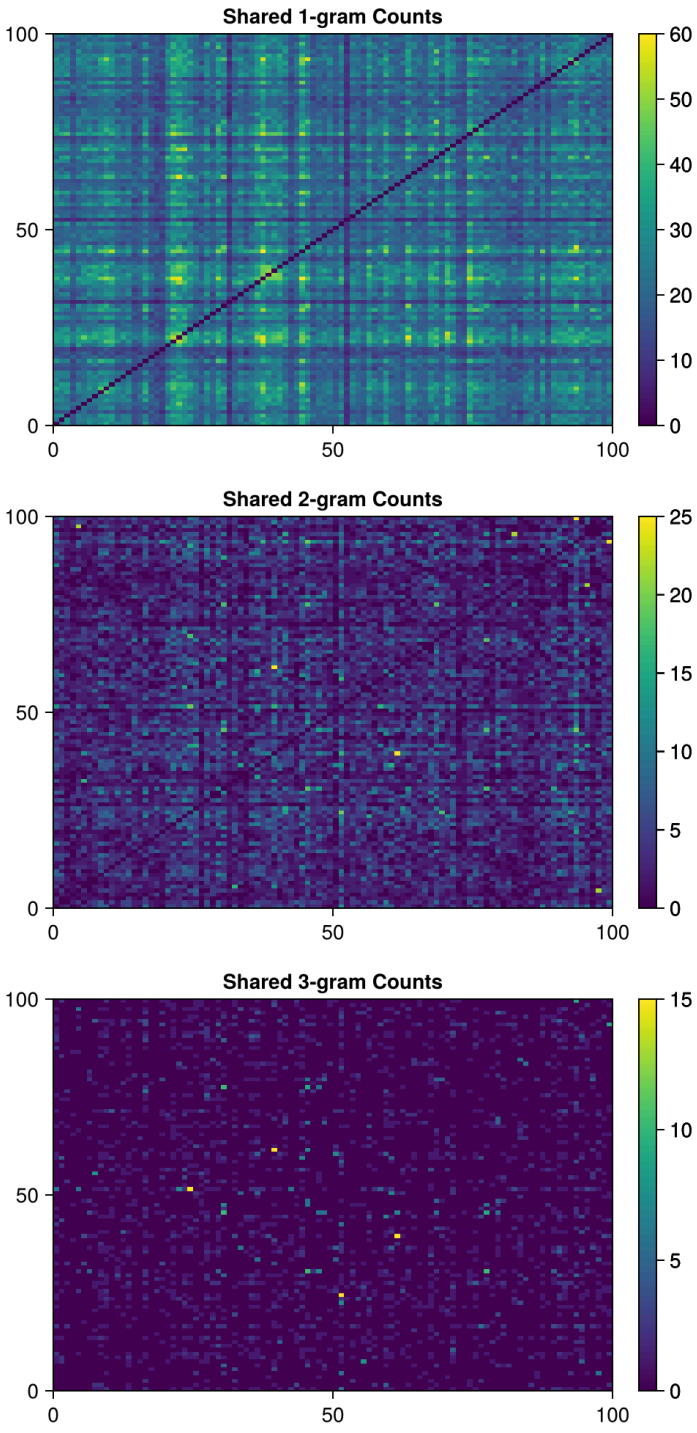


Figure 2. Heat maps for the shared n-gram counts for 100 random abstracts in our dataset.

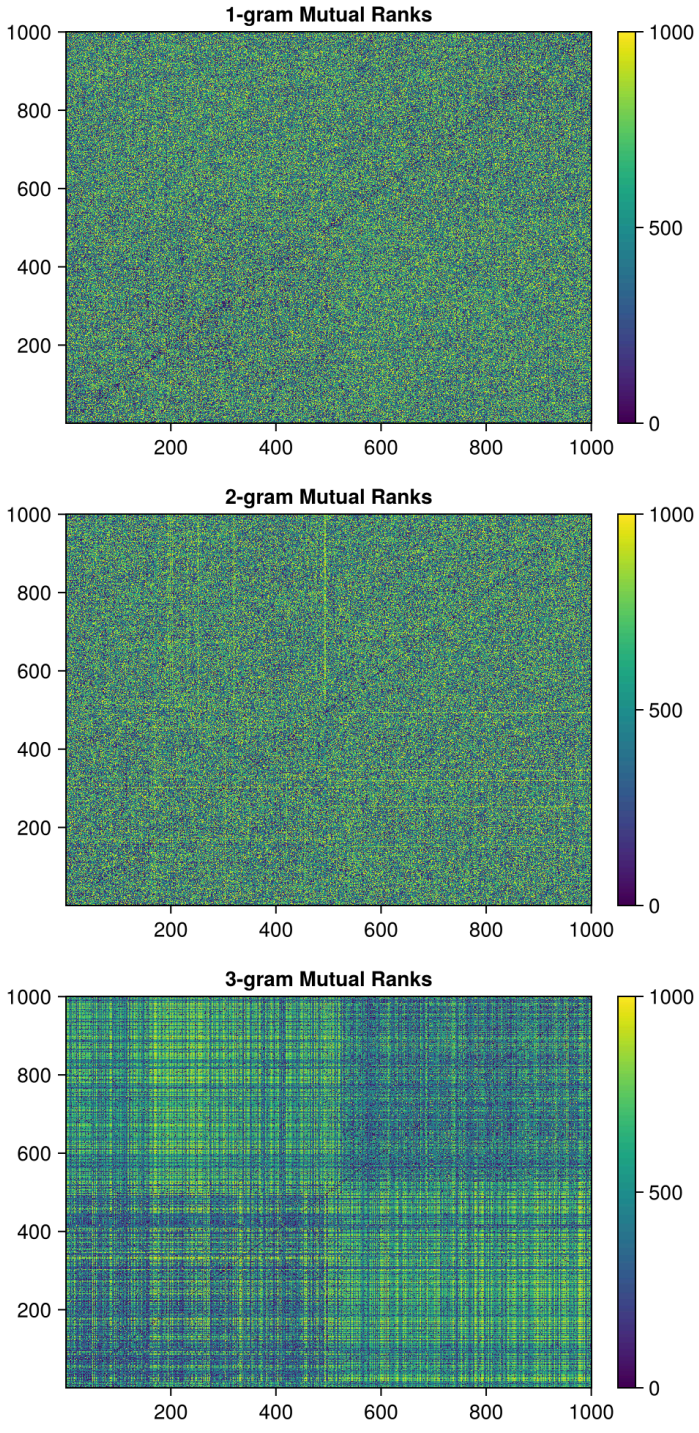


Figure 3. Heat maps for the mutual ranks for 1,000 random abstracts in our dataset. The abstracts were sorted so the first 500~ are all human-written abstracts, and the last 500~ are AI-generated abstracts.

a)

Row	n_gram	cluster_label	ground_truth	j_score
	String	String	String	Float64
1	1-gram	human	human	0.567839
2	1-gram	human	machine	0.0873309
3	1-gram	machine	human	0.201292
4	1-gram	machine	machine	0.609682
5	2-gram	human	human	0.882246
6	2-gram	human	machine	0.0270552
7	2-gram	machine	human	0.0400411
8	2-gram	machine	machine	0.873294
9	3-gram	human	human	0.710824
10	3-gram	human	machine	0.101751
11	3-gram	machine	human	0.0948181
12	3-gram	machine	machine	0.680357

Row	n_gram	f1_score
	String	Float64
1	1-gram	0.757519
2	2-gram	0.932362
3	3-gram	0.809777

b)

Row	n_gram	cluster_label	ground_truth	j_score
	String	String	String	Float64
1	1-gram	ambiguous	human	0.287066
2	1-gram	ambiguous	machine	0.164634
3	1-gram	human	human	0.449912
4	1-gram	human	machine	0.0589041
5	1-gram	machine	human	0.103651
6	1-gram	machine	machine	0.574733
7	2-gram	human	human	0.842756
8	2-gram	human	machine	0.042061
9	2-gram	machine	human	0.0510417
10	2-gram	machine	machine	0.829828
11	3-gram	human	human	0.667758
12	3-gram	human	machine	0.0963719
13	3-gram	machine	human	0.128962
14	3-gram	machine	machine	0.657095

Row	n_gram	f1_score
	String	Float64
1	1-gram	0.831403
2	2-gram	0.907001
3	3-gram	0.793068

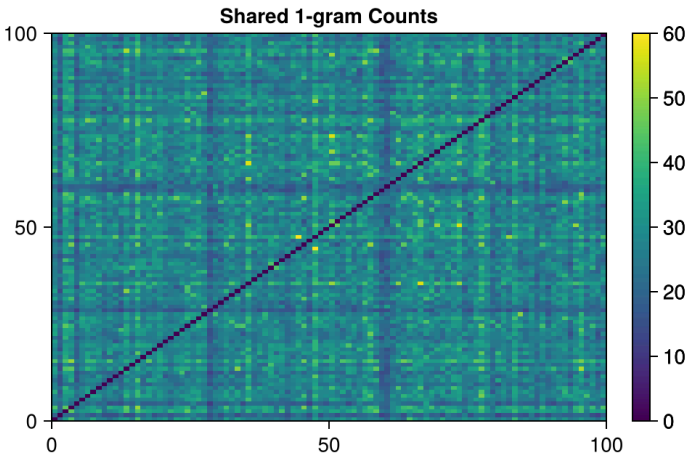
c)

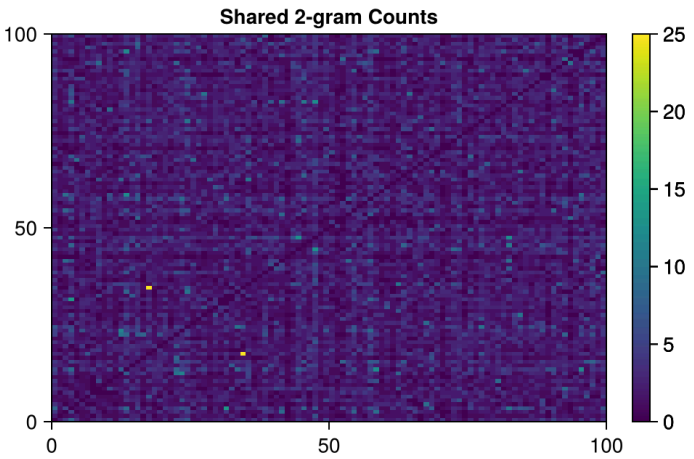
Row	n_gram	cluster_label	ground_truth	j_score
	String	String	String	Float64
1	1-gram	ambiguous	human	0.299051
2	1-gram	ambiguous	machine	0.159879
3	1-gram	human	human	0.422339
4	1-gram	human	machine	0.0656425
5	1-gram	machine	human	0.112161
6	1-gram	machine	machine	0.564148
7	2-gram	human	human	0.87406
8	2-gram	human	machine	0.00638978
9	2-gram	machine	human	0.0613682
10	2-gram	machine	machine	0.874766
11	3-gram	ambiguous	human	0.150794
12	3-gram	ambiguous	machine	0.182777
13	3-gram	human	human	0.623746
14	3-gram	human	machine	0.0850059
15	3-gram	machine	human	0.0703883
16	3-gram	machine	machine	0.56015

Row	n_gram	f1_score
	String	Float64
1	1-gram	0.818878
2	2-gram	0.9332
3	3-gram	0.820937

Figure 4. Jaccard indices and F-1 scores: a) for 2 clusters, b) for 5 clusters, and c) for 8 clusters.

Articles dataset:





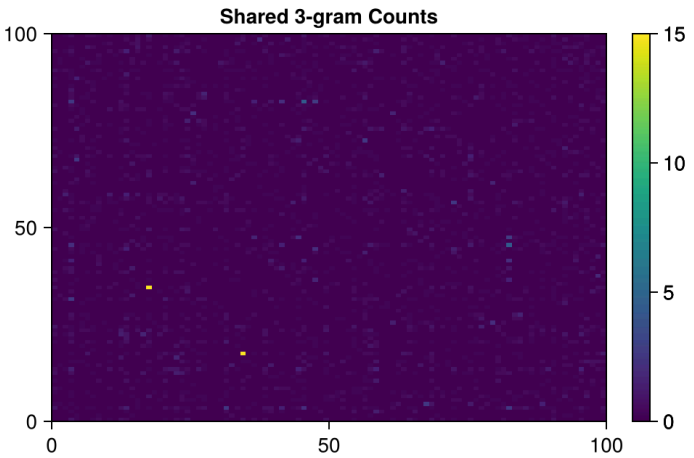
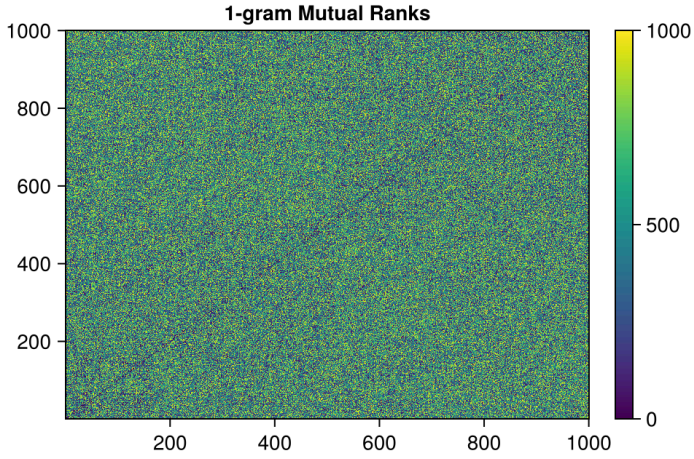
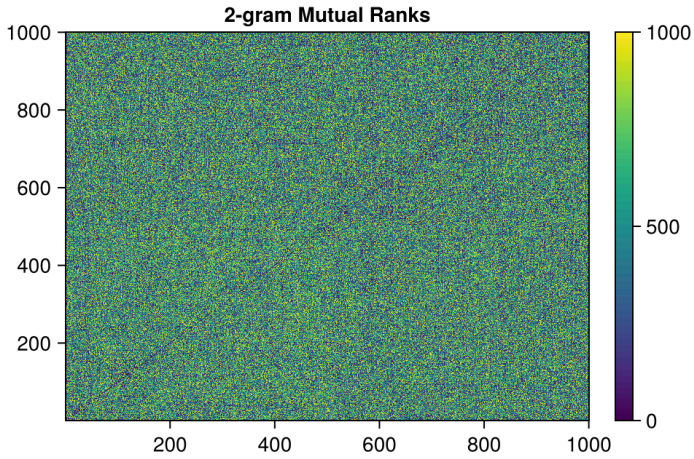


Figure 5. Heat maps for the shared n-gram counts for 100 random abstracts in our dataset.





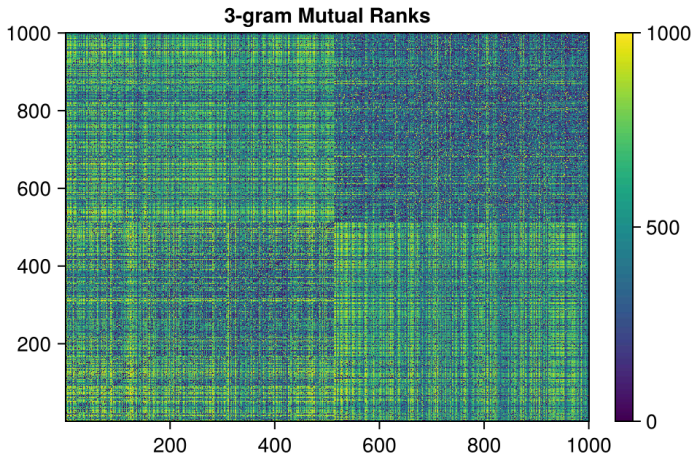


Figure 6. Heat maps for the mutual ranks for 1,000 random abstracts in our dataset. The abstracts were sorted so the first 500~ are all human-written abstracts, and the last 500~ are AI-generated abstracts.

a)

Row	n_gram	cluster_label	ground_truth	j_score
	String	String	String	Float64
1	1-gram	ambiguous	human	0.512
2	1-gram	ambiguous	machine	0.488
3	1-gram	human	human	0.0
4	1-gram	human	machine	0.0
5	1-gram	machine	human	0.0
6	1-gram	machine	machine	0.0
7	2-gram	human	human	0.876173
8	2-gram	human	machine	0.0219895
9	2-gram	machine	human	0.0459653
10	2-gram	machine	machine	0.876173
11	3-gram	human	human	0.729875
12	3-gram	human	machine	0.0524554
13	3-gram	machine	human	0.109129
14	3-gram	machine	machine	0.744932

Row	n_gram	f1_score
	String	Float64
1	1-gram	NaN
2	2-gram	0.934
3	3-gram	0.853824

b)

Row	n_gram	cluster_label	ground_truth	j_score
	String	String	String	Float64
1	1-gram	human	human	0.895753
2	1-gram	human	machine	0.00630252
3	1-gram	machine	human	0.0482897
4	1-gram	machine	machine	0.899254
5	2-gram	ambiguous	human	0.167213
6	2-gram	ambiguous	machine	0.166102
7	2-gram	human	human	0.758285
8	2-gram	human	machine	0.00114025
9	2-gram	machine	human	0.0233074
10	2-gram	machine	machine	0.764244
11	3-gram	human	human	0.722772
12	3-gram	human	machine	0.101512
13	3-gram	machine	human	0.0816777
14	3-gram	machine	machine	0.701068

Row	n_gram	f1_score
	String	Float64
1	1-gram	0.99
2	2-gram	0.972781
3	3-gram	0.860656

c)

Row	n_gram	cluster_label	ground_truth	j_score
	String	String	String	Float64
1	1-gram	human	human	0.906764
2	1-gram	human	machine	0.0355691
3	1-gram	machine	human	0.0165803
4	1-gram	machine	machine	0.89881
5	2-gram	ambiguous	human	0.047619
6	2-gram	ambiguous	machine	0.0661479
7	2-gram	human	human	0.847145
8	2-gram	human	machine	0.0327004
9	2-gram	machine	human	0.0278075
10	2-gram	machine	machine	0.822957
11	3-gram	ambiguous	human	0.139535
12	3-gram	ambiguous	machine	0.157343
13	3-gram	human	human	0.667254
14	3-gram	human	machine	0.0645905
15	3-gram	machine	human	0.057377
16	3-gram	machine	machine	0.636872

Row	n_gram	f1_score
	String	Float64
1	1-gram	0.946708
2	2-gram	0.936877
3	3-gram	0.86692

Figure 7. Jaccard indices and F-1 scores: a) for 2 clusters, b) for 5 clusters, and c) for 8 clusters.

Discussion

We tested our model on 2 through 8 different types of clusters. We decided to test more than 2 clusters to see if it would add more flexibility to our model, and to detect any hierarchical clustering, or clustering happening on a different level (i.e. clustering based on topic, sentence structure, etc.). For more than 2 clusters, after labeling the clusters as human, machine, or ambiguous, we would aggregate all same-labeled clusters into one.

Figures 2 and 5 show shared n-gram counts for 100 abstracts. As we get past 1-grams the shared counts start to rapidly fall off, so we chose to only go up to 3-grams. Figures 3 and 6 depict mutual ranks for 1,000 abstracts for every n-gram. The data is sorted so that the first 500 or so abstracts are human-written and the last 500 or so are AI-generated. A mutual rank closer to 0 means stronger associations between abstracts on a relative scale. For 1 and 2-grams it's difficult to see any patterns, but when we get to 3-grams it becomes a lot more apparent. The dark squares show us that there are stronger associations within human-written and AI-generated abstracts than between them, which tells us that there is significant clustering occuring.

To assess the quality of our model, we calculated the Jaccard indices and F-1 scores for each n-gram for all the different numbers of clusters. The Jaccard index is a measure of similarity between two sets. In this case, we measured the similarity between the abstracts in the clusters created and labeled by our model and the "ground truth" list of abstract labels. For example, in figure 4 a) row 1, we compared the abstracts from the human labeled cluster to the list of all the actual human abstracts. Essentially, we're determining how many human-written abstracts were grouped into a

human cluster and how many AI-generated abstracts were grouped into an AI cluster. The F-1 score is a measure of predictive performance calculated by a combination of recall and precision. We calculated the F-1 score for each n-gram in each cluster number.

For the Covid dataset 2-grams generally performed the best with a Jaccard index around 0.86 and an F-1 score around 0.90. The Jaccard indices for 1-grams and 3-grams were sub-par, but the F-1 scores were still quite high. This tells us that although not all the human-written abstracts are in the human cluster and not all the AI-generated abstracts are in the AI cluster, the clusters we do have are quite accurate.

The articles dataset performed even better than the Covid one, with 1-grams performing the best with a Jaccard index of about 0.90 and an F-1 score at around 0.96. One interesting thing to note is that in figure 7 a) the Jaccard indices and F-1 score for 1-grams were 0 or NaN. This is because our model labeled both clusters as “ambiguous”. Our cluster label rule used a 2:1 ratio, if there were at least twice as many human-written abstracts in the cluster than AI-generated, then the cluster would be labeled as “human”, and vice versa. If there is less than a 2:1 ratio, then the cluster is labeled as “ambiguous”. Despite this, our model performed really well with the articles dataset.

While our model performed well, there are many improvements to be made. In the future, we hope to improve the initial clustering and investigate any possible hierarchical clustering. We will also experiment with using adjective data to improve our model accuracy. Ultimately, we hope to use what we’ve learned from our clustering model to guide in the design of a full classification model.

References

Gray, Andrew. "ChatGPT" contamination": estimating the prevalence of LLMs in the scholarly literature." *arXiv preprint arXiv:2403.16887* (2024).

Arora, Sanjeev, et al. “A Simple but Tough-to-Beat Baseline for Sentence Embeddings”, 5 May 2023, https://openreview.net/forum?id=SyK00v5xx.

Ranola, John M., et al. “A Poisson model for random multigraphs.” *Bioinformatics*, vol. 26, no. 16, 16 June 2010, pp. 2004–2011, https://doi.org/10.1093/bioinformatics/btq309.

Ranola, John Michael et al. “Cluster and propensity based approximation of a network.” *BMC systems biology* vol. 7 21. 14 Mar. 2013, https://doi.org/10.1186/1752-0509-7-21.

Acknowledgements

This research and paper was greatly supported and guided by Dr. Alfonso Landeros. Funding and support for this research was made possible by RISE.

Reflection

I am extremely grateful to have been given this opportunity by RISE to engage in this hands-on research program. I was incredibly nervous about it because this was my first time doing any proper research, but the RISE staff and my faculty mentor made this an incredibly smooth and fun experience. Their encouragement and guidance greatly boosted my confidence in my research and data science skills. The workshops and peer mentor groups also helped with the research process a lot. Being able to interact with other scholars and knowing I’m not the only one who was new to everything really helped me feel more comfortable. Researching AI and text analysis has piqued my

interest in the topic, so I'm definitely considering pursuing it to hopefully make a more positive and ethical impact in the field. These 10 weeks have been an invaluable experience, and I hope many others like me will be able to have the same experience too.