university of groningen

faculty of science and engineering

# Topic Modeling Evaluations: The Relationship Between Coherency and Accuracy

Alfiuddin R. Hadiat

university of groningen / faculty of science and engineering

**University of Groningen**

# Topic Modeling Evaluations: The Relationship Between Coherency and Accuracy

**Master's Thesis**

To fulfill the requirements for the degree of
Master of Science in Computational Cognitive Science
at the University of Groningen under the supervision of
Joost Doornkamp (Artificial Intelligence, University of Groningen)
and
Prof. dr. Jennifer Spenader (Artificial Intelligence, University of Groningen)

**Alfiuddin R. Hadiat (s2863685)**

August 31, 2022

# Contents

# Acknowledgments

I thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster.

I thank Joost Doornkamp for patiently addressing all my worries and anxieties while providing the supervision necessary to keep this project on track and on time.

I thank Dr. Jennifer Spenader for her scepticism at the beginning of the project and her enthusiasm at the end, providing a voice of reason necessary to steer this project in the right direction.

I thank Lars Cordes and Sebastian Prehn for holding nothing back in their thoughts regarding my project and for keeping me humble throughout our friendship.

I thank Peter Varga, Alex Hill, and Roy David for being a daily reminder that Logistic Regression will remain eternally relevant.

Finally, I thank my family for constantly providing the care and support I needed throughout this process and for attending my final presentation.

# Abstract

Topic models are generally evaluated using coherency measures. These measures calculate the frequency of co-occurrence between all the representative words of a topic. Research shows that coherence correlates well with human judgment. However, no research has looked into the correlation between coherence and classifier accuracy. Can we use coherence for topic model selection when it is used for a prediction problem? To fill this gap, this project conducts two experiments that investigate this correlation. Two topic models (LDA and BERTopic) are trained and evaluated with four different coherence measures (UCI, UMass, NPMI, and CV). Classifiers (Logistic Regression and Decision Trees) are trained using topic model features to predict corpus categories. Accuracies are then correlated with the coherence measures. The results found the classifiers significantly correlated with UMass and NPMI. However, the UMass correlations were problematic for being inconsistent. Therefore, only NPMI could be considered generalizable for classifier performance estimation. The results also showed a difference between classifier performance using different topic models. That is, though BERTopic had higher coherence scores, LDA led to better logistic regression classifiers.

# 1 Introduction

Text data often contain a large number of different words and symbols. A simple and popular processing method is to transform text data into term-frequency (TF) matrices. TF matrices represent documents as counts of all words in a corpus. For example, a corpus containing 10,000 word types will represent each document using 10,000 features. While TF matrices can be used for statistical and machine-learning models, they are often very inefficient. Each additional word token results in another feature, leading to further complexity and resource use. They also do not represent documents intuitively, providing little insight into possible underlying patterns. This is where topic modelling comes in.

Topic modeling generates latent variables – or "topics" – from documents in the corpus. Topic models can represent documents as a collection of topics. This results in a smaller and more efficient corpus representation. It also results in more interpretable features. An early form of topic modeling that is consistently referred to in the literature is Latent Dirichlet Allocation (Blei et al., 2003). LDA has been applied to many different domains like scientific journals and historical newspapers (Griffiths & Steyvers, 2004; Yang et al., 2011). LDA's popularity inspired other topic modeling techniques as well as additional domains of application.

Topic models are, by nature, unsupervised models. This begs the question: how are they evaluated? Initially, topic models were evaluated using extrinsic measures. Wei and Croft (2006) evaluated them based on their performance in information retrieval tasks. Wallach et al. (2009) evaluated them based on the model's ability to predict whether a document belonged to a corpus. Other evaluations involve human judgment, where people would read the words of a topic and judge them. These evaluation methods, while valid, are unscalable. They take time and resources to conduct. "Coherence" measures would prove to be the solution. Coherence measures how "fitting" all top words are to a topic. The top words are the most frequently appearing words in a given topic, where frequency is calculated differently depending on the topic model. How "fitting" top words are together refer to how well they correlate with human judgment. A coherent topic could contain "brother, father, sister" because they cohere to family. An incoherent topic could contain "pizza, book, submarine" because their connection is unclear. Different coherent measures calculate these scores differently.

The first two coherence metrics ever developed were UCI (Newman et al., 2010, 2011) and UMass (Mimno et al., 2011). In brief, UCI calculates topic coherence by measuring its pointwise mutual information (PMI). PMI measures the rate of co-occurrence between two terms across documents. UCI finds the PMI of terms using an external resource – Wikipedia specifically. UMass also calculates the rate of co-occurrence between two terms. However, it calculates the rate differently and only uses the training corpus. Lau et al. (2014) shows that both methods correlate well with human judgment.

Little is known if coherence also predicts feature quality for machine-learning. Researchers have already used topic models to create well-performing classifiers (Hall et al., 2008; Sarioglu et al., 2012). They often use coherence to determine how many topics a model should have. However, they use coherence for interpretability, not knowing whether it leads to stronger classifiers. Knowing the relationship between coherence and classifier accuracy would be useful for machine-learning. If a correlation exists, coherence can predict topic feature importance. If a correlation does not exist, we know only to use coherence for interpretability. This research project examines whether this correlation exists.

## 1.1 Research Questions

This project asks whether there is a relationship between coherence and classifier accuracy. There exist many topic models, coherence measures, and classifiers. More detailed questions with these things in mind are necessary to better understand the relationship. As such, this project poses the following questions:

1. Is there a significant correlation between average model coherence and classifier accuracy?
2. Is there a significant correlation between individual topic coherence and feature importance?
3. Are significant correlations consistent across topic models?
4. Are significant correlations consistent across coherence measures?
5. Are significant correlations consistent across classifiers?

The first question looks at the relationship between coherence and accuracy directly. It aims to see if coherence could be used to predict classifier performance before training. If no correlations are found, then it is clear that coherence cannot predict classifier accuracy. However, if a correlation is found, then the remaining four questions can determine the direction of this relationship.

The second question looks at the relationship between coherence and classifier accuracy on a smaller scale. Topic model classifiers effectively use topics as features for classification. Coherency measures compute an individual score for each topic in a topic model. The average coherence score estimates the general performance of a topic model. Some classifiers have feature importance metrics that show how much a given feature contributes to its predictions. We can correlate individual topic coherence scores and the feature importance metrics that evaluate the topics as features. Doing this lets us see the relationship between coherency and classifier performance on a smaller scale. A significant correlation would suggest that classifiers use individual topics for prediction. A non-correlation, on the other hand, would imply that classifiers use all or several topics for prediction. This point, of course, depends on whether average coherency correlates with classifier accuracy. If so, this would suggest that topics are more predictive if they are used as a group.

The three remaining questions investigate the generalizability of these correlations. The correlation necessitates three different parts: the topic model, the coherence measure, and the classifier. A correlation may be only found for a specific constellation of those parts. The three separate questions look at whether there are parts that generalize. The third question looks at topic models. If only LDA coherence correlates with classifier accuracy, then the correlation is only applicable for LDA. If another topic model finds a correlation, then we can generalize the prediction.

The fourth question looks at coherence measures. It is possible not every coherence measure would correlate with classifier accuracy. Every measure may correlate with classifier accuracy. If we know which one correlates, then we know which measure predicts classifier performance. Inconsistency across measures can also tell us more about these measures. For instance, only UMass may correlate with accuracy because it does not use an external source for calculation.

The final question looks at classifiers. Different classifiers use different methods to predict. We want to know if coherence can predict feature quality across different classifiers. Inconsistent correlations across classifiers mean some use topics better than others. It also means we can only predict classifier performance for those classifiers. Consistent correlations mean coherence predicts classifier performance in general.

# 2 Theoretical Framework

## 2.1 Topic Models

An early topic model, Latent Semantic Index (LSI), works by using singular value decomposition on a TF matrix (Deerwester et al., 1990). SVD decomposes a matrix into smaller matrices that explain the largest amount of variance. The decomposed matrix is more efficient than a TF matrix because it has significantly less features. The algorithm generates latent variables by using SVD on these matrices.

LSI, however, isn't as commonly used today. LSI suffers from polysemy, words that have multiple meanings. The word "saw" could be used as the verb "to see" or as a tool, but not both. Probabilistic LSI (pLSI) solved polysemy by using the latent variables as an intermediary between words and documents (T. Hoffman, 1999). The researchers called these latent variables "topics". Topics would be represented as a distribution of words. Documents would be represented as a distribution of topics. These two forms of representations are still used in current topic modeling technique research.

Most current topic models generate topics through latent variable generation. But each model generates these models differently. LDA, as stated before, remains the most known and used topic model. The model is consistently used as a baseline comparison against other topic models. It follows pLSI's principle, though its additions are substantial. We will explore LDA more in-depth in a future section. LDA uses a probabilistic method to generate topics. Following the developments in machine learning and NLP, other topic modeling techniques have also been developed.

One recent topic modeling development has been to combine them with word embeddings. Several topic models are combinations of LDA with word embeddings. The differences between these models lie in how they are combined. Das et al. (2015) transforms LDA topic-word distribution into Gaussian mixtures instead of the original Dirichlet. A Gaussian distribution means that the topics are, effectively, centroids in a word embedding. That is, topics are represented by the words surrounding the centroid. Batmanghelich et al. (2016) does something similar by using a von Mises-Fischer distribution. The vMF distribution transforms the word embedding space such that it handles directional data better. Sia et al. (2020) applies a k-means algorithm on a word embedding and uses the generated clusters as topics.

Neural networks have also been used to develop topic modeling techniques. The general idea of neural topic models has been to use neural network representation of documents to find topics. For instance, variational auto-encoders (VAEs) can transform documents into efficient forms of distributions. Neural variational document modeling applies a Gaussian softmax on auto-encoded documents (Miao et al., 2016, 2017). It finds averages in the representation and uses them as topics. ProdLDA extends this idea further by using products-of-experts (Srivastava & Charles, 2017; Hinton, 2017). Word embeddings trained from neural networks can also be used directly to find topic models. In this case, the word embeddings act as the neural network document representation. BERTopic, for example, applies clustering methods to SBERT word embeddings to generate topics (Grootendorst, 2022).

Though neural topic models yield competitive coherence scores, probabilistic techniques remain superior (Doan & Hoang, 2021). Nevertheless, this project's experiment will include a neural topic model. The project includes both a probabilistic and a neural topic model to account for each method. LDA will represent probabilistic topic models for their popularity and competitive performance. BERTopic will represent neural topic models for its use of a neural network and word embeddings. Furthermore, compared to other neural topic models, BERTopic competes with LDA's coherency. The upcoming sections examine their training methods in detail.

### 2.1.1 LDA

LDA is a topic model that uses Bayesian probability to generate topics. A specific number of topics K are first set. Each topic is defined by a multinomial distribution of all words in the corpus. The distribution of all topics follows a Dirichlet distribution, a non-binary distribution for probabilities. LDA generates the topics with the following process:

1. set $k$ amount of topics
2. Randomly generate a document-topic distribution $\beta \sim$ Dirichlet for each document
3. Randomly generate a topic-word distribution $\theta \sim$ Dirichlet
4. For each word $w$ in each document $d$:
    - randomly select topic $k$ given the document's topic distribution
    - Assign word $w$ to topic j given $p(w, j) = p(k|d) * p(w|k)$

Following this, documents are represented as topic mixtures and topics are represented as word mixtures. The random document-topic distribution generation means that each word in a document is first assigned to a topic at random. Then, for each word in a document, it randomly selects a topic given the document's topic distribution. It calculates the probability of the chosen topic given the document-topic distribution. After, it calculates the probability of the observed word's appearance in the selected topic. The word is finally assigned to the topic the word is most likely to appear in given the product of the previous two probabilities. With each document, words become more associated with certain topics over others. Another way to understand this is by summing the assignments:

$$p(w_d|\beta_d, \theta) = \Sigma_k \beta_{dk} \theta_{kw} \tag{1}$$

The summation comes in a form corresponding to principal component analysis. PCA yields smaller matrices that explain the variance of another matrix. In this sense, LDA is a "PCA" for term-frequency matrices (i.e. count of words) that yield two variance-explaining matrices: a document-topic matrix $\beta$ and a topic-word matrix $\theta$.

These posterior distributions act as the latent variables of a corpus that we use for prediction and analysis. The posterior cannot be computed directly (Blei et al., 2003) and is usually approximated. One common method of approximation uses a Markov Chain Monte Carlo (MCMC) method called Gibbs sampling (Griffiths & Steyvers, 2004). Another common method is variational inference, which will be outlined below. While both are effective, both suffer computationally when facing large datasets. A solution for this is to use an online version of variational inference: online Variational Bayes.

Traditional Variational Bayes approximates the posterior distribution $p(w|\theta, \beta)$ using a simpler distribution: $q(z, \theta, \beta)$. For distribution $q$, $z$ refers to the topic assignments of each word. This simpler distribution can optimized by maximizing the Evidence Lower Bound (ELBO):

$$logp(w|\alpha, \eta) \geq \mathcal{L}(w, \phi, \gamma, \lambda) = \mathbb{E}_q[logp(w, z, \theta, \beta|\alpha, \eta)] - \mathbb{E}_q[logq(z, \theta, \beta)] \tag{2}$$

where w = word count, $\phi$ = word-topic assignments, $\gamma$ = document-topic weights, and $\lambda$ = topic distributions. ELBO finds the lower bound where evidence for the posterior distribution is highest. It does so by minimizing the Kullback-Leibler divergence between the two posterior distributions. A smaller KL divergence means a smaller difference in expectation between the two distributions, as

seen in Formula 2. A smaller difference in expectation means the two distributions are more similar. Thus, minimizing the difference in expectation allows distribution $q$ to approximate distribution $p$.

By factorizing the sum of expectations and letting it be a function of the ELBO, we get

$$\mathcal{L} = \Sigma_d \ell(n_d, \phi_d, \gamma_d, \lambda) \tag{3}$$

This formula not only finds the ELBO for distribution $q$, it also only requires the word counts of each document. Thus, solving Formula 3 lets us find the posterior distribution that becomes the topics we search for.

One final addition is made on top of the VB formula above. While the VB algorithm is faster than Gibbs sampling, it can be further optimized for much larger datasets. In online VB, topics $\lambda$ are recalculated after computing $\phi$ and $\gamma$. The algorithm's goal is to maximize:

$$\mathcal{L} = \Sigma_d \ell(n_d, \phi_d(n_d, \lambda), \gamma_d(n_d, \lambda), \lambda) \tag{4}$$

To maximize the formula above, $\tilde{\lambda}$ is computed, which are the topics $\lambda$ optimized if the whole corpus consisted of a single document repeated N number of times (where N = corpus length). $\lambda$ is then updated by averaging its previous value and $\tilde{\lambda}$. The online version of the VB algorithm converges much faster than the traditional version. M. Hoffman et al. (2010), the proposer of online VB, shows that it converges much faster than traditional VB for larger datasets. The faster convergence is why this project uses the online VB solver for LDA.

### 2.1.2 BERTopic

BERTopic is a topic model that utilizes the Bi-directional Encoder Representation transformer to generate topics. However, the topic model does more than just use BERT. BERTopic create topics from corpora through three steps. First, it transforms documents into embedding representations. Second, it reduces the dimensionality of the embeddings and clusters them. Third, it extracts the topics using modified version of TF-IDF.

BERTopic transforms documents into embeddings using a specific version of BERT called Sentence BERT (Reimers & Gurevych, 2019). The purpose of the document embedding is to compare the documents in vector space. SBERT uses sentences instead of words for document embedding. This let's the model capture the semantics of arguments made in prose. SBERT achieves competitive performance for sentence embedding tasks (Reimers & Gurevych, 2020; Thakur et al., 2020). BERTopic only uses these embeddings for clustering, thus any embedding technique could be used. This means clustering can be improved by simply using a better language model.

The embedding clustering is done in two parts. The embedding dimensions are first reduced before clustering. Large dimensions, which SBERT embeddings have, can lead to problems in clustering (Beyer et al., 1999; Aggarwal & Yu, 2001). Larger dimensions lead to farther distances between points, causing difficulty for nearest-neighbor problems. Though there are clustering algorithms that can circumvent this dimensionality issue (Steinbach et al., 2004; Pandove et al., 2018), the simpler solution, which BERTopic uses, is to reduce the dimensions.

BERTopic uses Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction (McInnes et al., 2018). Other possible dimensionality reduction techniques, like PCA or t-distributed stochastic neighbor(t-SNE) were considered. UMAP, however, has three clear advantages. It preserves local and global features better than other techniques. Its computational efficiency makes it viable for general dimensionality reduction. And it has been shown to improve the performance of clustering algorithms (Allaoui et al., 2020).

In brief, UMAP does two things to reduce dimensionality. It first constructs a high dimensional representation of the data. It does this by creating a "fuzzy simplicial complex" representation. In short, this means each data point connects to other points where their radii overlap. Thus, two points with overlapping radii are connected. This can, then, be thought of as a graph. Choosing the right radius length is important. A larger radius can lead to too many connections, while a smaller radius can lead to no connection. UMAP solves this by choosing a radius locally, setting the distance according to each point's nearest neighbors. The resulting graph is made fuzzy by reducing the chances of a connection as the radius grows. Finally, local structures are guaranteed by forcing each point to connect with its closest neighbor. Once the high dimensional graph is made, UMAP approximates it with a lower dimensional graph. The lower dimensional representation achieves the same representation performance while being more efficient.

The UMAP-reduced embeddings are then clustered using Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) (McInnes et al., 2017). HDBSCAN works by using density to cluster data points together. The algorithm measures density using a mutual reachability distance, which is defined as:

$$d_{mreach-k}(a,b) = max\{core_k(a), core_k(b), d(a,b)\} \tag{5}$$

Function $d(a,b)$ measures the distance between $a$ and $b$. Function "core" measures the distance between a given point and its furthest neighbor within k-closest neighbors. For example, if k = 5, then the core distance of each point is the distance to the 5th closest neighbor. With this metric, dense points remain the same distance from each other while spare points are pushed away through their core distance. The algorithm then clusters groups together using a density threshold. That is, a group of points are only considered a cluster if their density is above a certain threshold.

The final step comes in the hierarchical addition to DBSCAN. Due to varying densities, global thresholds can yield bad clusters. The density of points can be thought of as a "mountain" with multiple peaks. "Peaks" are regions in the data that are denser compared to the "base" of the mountain. Incorrect thresholds can yield bad clusters when there are more points in a base than its peaks or vice versa. To solve this, HDBSCAN split clusters depending on local density. Simply put, clusters are combined when the base has more points than the peaks, or they are split if the peaks have more points. This automates the threshold for clustering and also yields better clusters. These clusters are the bases for the topics.

One final step is necessary before the clusters can be used as topics: generate the topic-word distribution. BERTopic does this by using a modified term-frequency inverse-document-frequency (TF-IDF) function. TF-IDF itself is a modified term-count matrix, and is defined as:

$$W_{t,d} = tf_{t,d} \cdot log(\frac{N}{df_t}) \tag{6}$$

Term frequency $tf_{t,d}$ is a matrix of counts of all terms that occur in each document. The inverse document frequency $log(\frac{N}{df_t})$ measures how much a given term provides to a document by calculating the logarithm of a terms count in all documents. The IDF function ensures that rare terms are not overpowered by frequent terms.

BERTopic applies TF-IDF to the clusters of documents to generate the topics. In this case, the clusters can be considered as a corpus and each cluster as a single document. To do the latter, all documents in a cluster are concatenated together. The result is a cluster-based TF-IDF that measures the importance of words in clusters rather than individual documents. The final product results in a topic-word distribution for each cluster of documents.

## 2.2 Coherence Measures

Researchers first evaluated topic models using extrinsic methods. These methods did not use the topic models themselves and required other resources. For example, they were evaluated on their performance in summarizing or information retrieval. Wallach et al. (2009) presented the first intrinsic method by evaluating topic models with perplexity. This measured the topic model's probability for unseen documents held out from the training corpus. Better models would have higher probabilities for these unseen documents. Perplexity was used up until human judgment measures appeared.

Chang et al. (2009) developed the first human evaluation of topic models. Human participants were presented with a list of five topic words from topic models. Each list switched one word with a word from another topic. Participants were asked to identify the switched word. Better topics were those that had the switched words consistently identified. Interestingly, they also found that better topics had higher perplexity. This suggested that perplexity does not necessarily lead to more interpretable topics.

Newman et al. (2010) developed the first coherency measure. They aimed to correlate different evaluations with the human evaluation method from Chang et al. (2009). They found that pointwise mutual information (PMI) correlated the most with human evaluation. Lau et al. (2014) corroborated this finding by also comparing human correlations with other coherences measures. Due to its strong human correlation, most topic models are evaluated using coherence measures. This project focuses on four coherence measures: UCI, UMass, NPMI, and Roder's CV. These measures were chosen because they currently correlate the most with human judgment and are currently most commonly used when evaluating new topic models (Röder et al., 2015).

### 2.2.1 UCI

UCI is the first coherence measure developed by Newman et al. (2010). The measure uses the top N words of each topic to calculate coherency. UCI is defined as:

$$UCI = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} PMI(w_i, w_j) \tag{7}$$

N = length of top N words and is chosen. PMI is the rate of co-occurrence between words in a given corpus. It is defined as:

$$PMI(w_i, w_j) = log \frac{P(w_i, w_j) + \varepsilon}{P(w_i) \cdot P(w_j)} \tag{8}$$

Word probabilities $P$ are estimated using a sliding window moving through an external corpus. $P(w)$ is the probability of observing a single word as the window slides through each document. $P(w_i, w_j)$ is the probability of observing words $w_i$ and $w_j$ co-occurring as the window slides through each document. Simply put, as words co-occur more, the PMI increases. The original UCI sets $\varepsilon$ = 1 as logarithm requires non-zero probabilities. UCI, as can be seen, is the average PMI of the top N words in each topic.

### 2.2.2 UMass

UMass, developed by Mimno et al. (2011), also computes co-occurrence between pairs of words. However, it does not use an external corpus and only uses the model's training documents. UMass is defined as:

$$UMass = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N} \sum_{j=1}^{i-1} log \frac{P(w_i, w_j) + \varepsilon}{P(w_j)} \tag{9}$$

As with UCI, N = length of top N words and is chosen. Though the formula between UCI and UMass look similar, they are computed differently. Word probability $P(w_j)$ is calculated using the frequency of word presence in a document. That is, the count of how many documents have word $w$. $P(w_i, w_j)$ is the probability of words $w_i$ and $w_j$ co-occurring in the same documents. It uses the count of how many documents have both words $w_i$ and $w_j$. As with UCI, an increase in co-occurrence results in better UMass. This measure, however, looks for co-occurrence in documents rather than sliding windows. UMass averages the log document co-occurrence between all words for each topic.

### 2.2.3 NPMI

NPMI, introduced by Aletras and Stevenson (2013), replaces the PMI in UCI with normalized PMI. Normalized PMI (Bouma, 2009) rescales the probabilities of word co-occurrence and is defined as:

$$NPMI(w_i, w_j) = \frac{log \frac{P(w_i, w_j) + \varepsilon}{P(w_i) \cdot P(w_j)}}{-log(P(w_i, w_j) + \varepsilon)} \tag{10}$$

NPMI applies a negative log co-occurrence probability to each PMI result to normalize them. Aletras and Stevenson (2013) found that UCI performed better using NPMI. Lau et al. (2014) showed that NPMI correlated better with human judgment. As with UCI, the NPMI between all top word pairs in a topic are averaged for the final score.

### 2.2.4 Roder's CV

Roder's CV was developed within the coherence measure framework under Röder et al. (2015). The authors took the different elements in coherence calculations to develop a single framework. In that framework, they developed new coherence measures, one of which was labelled $C_v$. For readability, this project labels this measure as CV. Their results showed that CV correlated with human evaluations the most.

CV, in contrast with the previous three measures, does not use co-occurrence frequency. Rather, it uses context vectors. Context vectors count the words that appear within $\pm$ five tokens of a given top word. CV restricts this count to only the topic's top words so that vectors share the same length and words. Thus, a context vector is a count of all top topic words seen within $\pm$ five tokens of all observations of the given word in all documents. The context vector then computes the NPMI between the given word and the count of top topic words in the vector.

For example, given a topic whose top three topics are "game, sport, team", the vector for game $\vec{v}_{game}$ would be:

$$\vec{v}_{game} = [NPMI(game, game), NPMI(game, sport), NPMI(game, team)] \tag{11}$$

Once the elements of each context vector are computed, the similarity between vectors is used for coherence. CV uses cosine similarity:

$$cos(\vec{u}, \vec{w}) = \frac{\sum_{i=1}^{|W|} u_i \cdot w_i}{||\vec{u}_2|| \cdot ||\vec{w}_2} \tag{12}$$

where W = all top word context vectors. Vectors $\vec{u}$ and $\vec{w}$ refer to the context vectors being compared. The theory behind context vector similarity over co-occurrence counts is in its indirectness. It is possible that some words are semantically similar but rarely co-occur. Some top words may rarely co-occur directly. Checking the similarity in co-occurrence with all other top words allows more sensitivity to the semantics.

## 2.3 Classifiers

Two requirements had to be met for the classifiers. The first was that they could predict multiple classes. The second was that they could produce a feature importance metric. The second research question asks if individual topic coherence correlates with feature importance. This requires us to measure a topic's influence on a classifier's prediction. Doing so is simple for classifiers that can yield feature importance metrics. For this reason, logistic regression and decision trees are used as classifiers. What follows is a brief reminder of both classifiers and how they can calculate feature importance.

### 2.3.1 Logistic Regression

Logistic regression is a modified version of linear regression. The latter is a predictor of continuous values. It works under the assumption that target values are a linear combination of all features. For instance, when plotted for two features, a linear regression generates a line. In general, linear regression predictions are defined as:

$$\hat{y}(w,x) = w_0 + w_1x_1 + ... + w_px_p \tag{13}$$

where $\hat{y}$ = predicted values, $x$ = input features, and $w$ = weight coefficients with $w_0$ = intercept. Regression training involves adjusting $w$ such that the predicted values are as close to the actual values as possible.

Linear regression can generalize for binary and multi-class prediction. Because the project is only interested in multi-class prediction, we will focus on multinomial logistic regression. In logistic regression, each class $k$ gets its own coefficient vector. Thus, each feature weight is different depending on the class. The regression model predicts a class that has the highest probability given the input features. It calculates probability as follows:

$$\hat{p}_k(X_i) = \frac{\exp(X_iW_k + W_{0,k})}{\sum_{k=0}^{K-1}\exp(X_iW_l + W_{0,l})} \tag{14}$$

where $X$ = input features, $W_k$ = weight coefficients for class $k$, $W_l$ = weight coefficients for not-class $k$, and $W_0$ = intercept. In short, the probability computes the likelihood that the input features predict class $k$ given $k$'s coefficients compared to the other classes and their coefficients.

Logistic regression models are evaluated using the log-loss functions (also known as the cross-entropy loss). Log-loss is a measure of how much predicted probabilities deviate from true values. For multinomial logistic regression, the log-loss is defined as:

$$L_{log}(\hat{y},\hat{p}) = -\sum_{i=1}^{n}\sum_{k=0}^{K-1}[y_i = k]\cdot log(\hat{p}_k(X_i)) \tag{15}$$

where $\hat{y}$ = predicted values, y = true values, n = length of input data, K = classes, and $\hat{p}$ is computed using Formula 14. The lower the log-loss, the better the classifier's performance.

Log-loss can be used to determine the importance of a given feature. For binary logistic regression models, the weights can be directly taken. However, the multiple coefficients used to calculate probabilities in multinomial models prevent us from doing this. Instead, we can compare the log-loss between models with and without a given feature. This difference measures how much contribution a given feature adds to the model. We use this method to measure the feature importance of a given in a logistic regression classifier.

### 2.3.2 Decision Trees

Decision trees are simple and intuitive. They classify by splitting the input dataset into smaller datasets. The splitting results in a tree graph where the leaf nodes – the final nodes – are classifications. The optimization goal of the decision tree is to keep splitting the dataset until each leaf is as "pure" as possible.

The classifier splits the dataset using impurity measures. Specifically, it measures the impurity of a feature at each split. There are two possible impurity measures: Shannon's Entropy and Gini Impurity. Decision trees achieve approximately the same performance with both measures, but Gini Impurity is faster for computation. Gini impurity is defined as:

$$Gini = 1 - \sum_{i=1}^{n} p_i^2 \tag{16}$$

where $p_i$ = the probability of each value in a given feature. For instance, given a feature with values $[a, a, b]$, the probability of $a$ = 2/3 and of $b$ = 1/3. The Gini index of this example feature is then 0.444. At each split, the Gini impurity of every feature is computed. The feature with the smallest Gini impurity is then selected for the split. The dataset is then split depending on whether the feature was categorical or continuous. Given that the input data of our experiments are continuous probabilities (i.e. the topic distributions of each document), the split would be binary. The algorithm finds a criterion that yields the lowest Gini impurity for a given continuous feature. We use the Gini impurity to measure the importance of a given topic for the decision trees classifier.

# 3 Methods

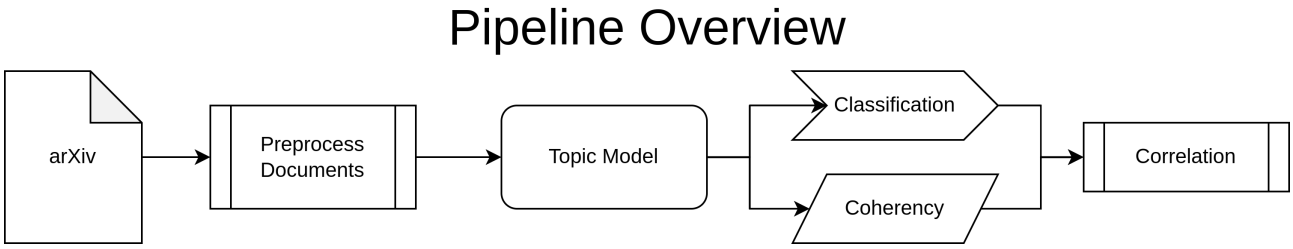


Figure 1: Experimental pipeline used to determine the correlation between topic model coherency and classifier accuracy.

Each experiment requires four necessary parts: a corpus, topic models, coherence measures, and classifiers. Topic models must be trained on the corpus. Coherence measures must measure the performance of the topic models. And classifiers must be trained using the topics of the topic models. The experiments correlate the coherence measures and the classifier performance at two levels.

Figure 1 shows the pipeline for the experiments. As can be seen, there are five factors involved in the experiment. Each of these factors is detailed further in this section. Each factor has a different number of elements. The corpus, "arXiv", and its preprocessing only have one as they do not change. The topic models have two elements: LDA and BERTopic. The classification has two elements as well: logistic regression and decision trees. Finally, the coherence measures have four elements: UMass, UCI, NPMI, and CV.

## 3.1 Experiments

Two experiments were designed to answer the research questions. The first experiment investigates the relationship between topic model coherency and the accuracy of classifiers using topics as features. Topic model coherence refers to the average coherence used to approximate its performance. This experiment aims to answer the first question: whether there is a significant correlation between a coherence measure and classifier accuracy. This correlation would show that a change in coherence follows a change in accuracy. A variation in coherence is, therefore, required.

To achieve this, each topic model is trained to have varying amounts of topics: 25, 50, 75, and 100 topics. Furthermore, for each topic amount, five topic models were created. This resulted in 20 trained models with varying topic amounts for each topic model. Spearman rank correlations are computed between the coherency measures of each topic model and the accuracy of their corresponding classifiers. Each coherency measure will have a separate score to determine its correlation with a given classifier's accuracy.

The second experiment investigates the relationship between individual topic coherency and classifier feature components. This experiment aims to answer the second question: whether there is a significant correlation between individual topic coherency and feature importance scores. Coherence measures compute a score for each topic in a model. Classifiers use each topic for prediction. The feature importance score measures how informative a given feature is for the prediction. A correlation between coherence and feature informativity would show that the latter approximates how useful a single topic is for a classifier.

## 3.2 arXiv Dataset

ArXiv is an open-access digital archive hosting over 1.9 million scientific articles. The archive primarily contains papers from physics, mathematics, and computer science. The experiment uses a dataset containing metadata from roughly 1.7 million arXiv articles. The data contains fields such as the title, author(s), category, and abstract. Though the dataset gives access to each article's full text, only the abstract was used. The abstracts were enough to generate the topic models. Furthermore, due to limited time and resources, using the full texts would have resulted in fewer papers sampled. Sampling many papers was preferred to generate broader topics that better approximate the whole corpus.

Not all 1.7 million articles were used in the experiments. Only articles published after 2018 were used. This was done for two reasons. The first was to ensure that scientific terminology shared across papers was similar. ArXiv contains papers from the 1990's up until 2022. The language across the domain may have evolved over the 30-year range. To minimize variance and use of resources, a smaller time range was chosen. This yielded 530,710 articles for sampling.

The topic models and the classifiers were trained with separate amounts of articles. The topic models were trained using all 530,710 articles while the classifiers were trained using 200,000. This was primarily done to address the category imbalance. The classifier learns to predict the category of an arXiv paper given its topics. The arXiv subset used in the experiments contains 23 different categories, each varying in the paper count. Computer science, the category with the highest count, has 158,368 papers. Functional analysis and algebraic geometry, in contrast, have one paper each.

To solve this imbalance, only the categories with at least 40,000 papers were kept. Furthermore, only 40,000 papers were kept from each category. This resulted in a dataset with 200,000 documents split evenly across five different categories: computer science, mathematics, physics, condensed matter, and astrophysics. The topic models are trained with as many documents as possible to approximate the arXiv corpus.

The dataset was preprocessed in the same way for both model and classifier training to ensure validity. Preprocessing was also shared between LDA and BERTopic. Each word of each abstract was tokenized and lowercased. Stopword tokens existing in Gensim's English corpus were removed. Each tokenized document was then transformed into a bag-of-words (Rehurek & Sojka, 2011). Tokens that appeared in 2% of the documents were removed for being too rare. Those in more than 95% were removed for being too common. These thresholds were recommended by the BERTopic author and extended for LDA to ensure validity.

## 3.3 Topic Models

### 3.3.1 Latent Dirichlet Allocation

LDA modeling was done using the Scikit-Learn Python package (Pedregosa et al., 2011). Sklearn uses the online variational Bayes algorithm outlined in the theoretical framework section. Online VB's efficiency is useful to handle the 530,710 documents. For the experiment, four different amounts of topics were created: 25 topics, 50 topics, 75 topics, and 100 topics. The different amounts of topics were built to generate noise in the coherence scores. Five models were trained for each topic amount, resulting in a total of 20 LDA models.

LDA has a number of parameters capable of changing the training results. However, for simplicity, the parameter settings set to default in the Sklearn package were used. The document-topic and topic-word prior distributions ($\alpha$ and $\beta$ parameters in (M. Hoffman et al., 2010, 2012)) were kept to

1/n, where n = number of topics. Each LDA model took roughly 20 minutes to train, thus taking around six and a half hours in total.

### 3.3.2 BERTopic

BERTopic modelling was done using the original developer's Github implementation (Grootendorst, 2022). As with LDA, four kinds of BERTopic models were built that varied in topic amounts. Due to BERTopic's dynamic topic modelling, it trains until it converges to a specific amount of topics. After convergence, it can then be reduced to a specified amount. With the 530,710 documents, each BERTopic model would train to have between 750-850 topics before reduction. Reduction involves clustering topics together until the specified amount.

Default parameters set by the author were used for training. A summary of the model's method is required to understand the parameters. The algorithm uses language models to create document embeddings. These embeddings are then reduced in dimensions and clustered. Each cluster is assumed to represent a topic. Finally, the word-topic distribution for each cluster is computed using c-TF-IDF.

The first parameter is the minimum topic size. This parameter is the threshold for how many documents are required before a cluster is considered a topic. A larger minimum size results in a smaller number of topics and vice versa. It also determines how fast a BERTopic model takes to train. The author recommends a topic size of 50 as a trade-off between time limitation and topic approximation. Thus, each BERTopic model used a minimum topic size of 50.

The second parameter is the top $n$ words. This parameter refers to how many words represent each given topic. For example, if set to five, then each topic is represented by its top five words. The top words are those that score the highest c-TF-IDF. Computing this metric takes time; the larger the $n$, the longer it takes. Based on the author's recommendation, an $n$ of 10 is set.

Each BERTopic model took around nine hours to train. With a total of 20 models, this resulted in around 180 hours of training. Each model was trained in parallel to speed up this process.

## 3.4 Coherence Measures

Coherency scores for each model were measured for the experiments. As previously stated, four different coherency scores are used: UCI, UMass, NPMI, and CV. In total, 40 topic models had to be measured four times each, which results in a total of 160 measurements. The coherency scores were computed using Gensim's coherence model. BERTopic uses the coherence model directly right while LDA uses "tmtoolkit". Tmtoolkit is a package that computes the coherence model in parallel, which was necessary for LDA's measurements.

Coherency is measured using a specified number of a topic's top words. The larger the number of selected top words, the longer computation takes. Röder et al. (2015) found that more top words correlate stronger with human rating of top words. To balance between coherence quality and computation time, five top words were chosen for measurement. Five top words correlate with human judgment and are more efficient than ten or more.

## 3.5 Classifiers

The classifiers used for the experiments are Logistic Regression and Decision Trees. These classifiers are preferred over other classifiers for their interpretability. That is, compared to other competitive classifiers, these classifiers provide simple feature-level metrics that measure the importance of a given feature. For the logistic regression model, this comes in the form of the log-loss difference. For

decision trees, this comes in the form of the Gini index. Both scores measure a feature's contribution to the classifier's prediction. The second experiment correlates these metrics with topic coherence. The first experiment correlates the classifier accuracy with average topic coherence.

Each classifier predicts an arXiv category using the word-topic distributions. The topic models transform documents into a word-topic distribution. That is, a model transforms a document into a topic distribution using the document's words. A single datapoint is, therefore, represented as an array of a topic distribution. With 40 total topic models and two different classifiers, 80 fitted classifiers were created for the experiment. Five-fold cross-validation was used to measure the accuracy of each classifier.

Both classifiers are implemented using Sklearn. Logistic regression classifiers were trained with an L2 regularization to prevent overfitting. It used the "lbfgs" solver (Liu & Nocedal, 1989) to optimize the multinomial predictions efficiently. For the decision trees, Gini impurity was chosen as the criterion for splitting. Gini impurity results in approximately the same performance as Shannon's entropy but is faster. No depth limit was specified to ensure each feature would receive a Gini index score. All other parameters were set to default.

# 4 Results

## 4.1 Data Loss

Four of the LDA models returned invalid values as coherence scores. The invalid values were caused one topic per faulty model. Three of the 100-topic models and one of the 75-topic models had this issue. The invalid coherence scores were all measures that used an external corpus (i.e. Wikipedia). The issue may be caused by a topic containing a token that is not present in the external corpus. The token could be a consistent error (e.g. formatting symbols) or an incorrectly processed word.

To fix this issue, the faulty topic was removed and each coherence measure was recomputed for the faulty model. This was done in both models to ensure a valid comparison. This should not affect the outcome because model coherence is only an average of all topic coherence scores. As such, the comparisons between faulty and non-faulty models should remain valid.

## 4.2 General Results

Figure 2 shows the coherency scores as a barplot and Table 4 (in the appendix) shows the scores in table format. Both the table and figure show us that the BERTopic coherency scores appear to be better than the LDA coherency scores. This can be seen by the lower UMass score and the higher NPMI, UCI, and CV scores, all of which translate to having better topic coherency in general.

Interestingly, and already suggestive for our experiments, the other direction is apparent for the classifier accuracies. Table 1 shows accuracy scores of each classifier and Figure 3 (in the appendix) shows this in barplot form. The results show us that the LDA classifiers have higher accuracy scores than the BERTopic classifiers. This would suggest that LDA topics result in better classifiers than BERTopic topics.

Properly answering the research questions of this paper requires an inspection of the experiments. Before examining them, however, a brief recap of the research questions is required. The most general question is whether there is a correlation between model coherency and classifier accuracy at all. The other questions ask what kind of relationship between the two scores.

For one, is there also a correlation between individual topic coherency and topic feature importance? This question is investigated in experiment 2. Is there a difference in correlation between the topic models? This question checks whether the correlation is consistent among the models or dependent on them. The final question asks whether there is a difference in correlation between the classifiers. This question is similar to previous in that it checks to see if the relationship is consistent or dependent on the classifier being used.

## 4.3 Experiment 1

The first experiment looks at the correlation between model coherency and classifier accuracy. Table 2 show these correlations between all model classifiers and coherency measures. Figure 4 show scatterplots between accuracy and coherency for LDA, and Figure 5 show scatterplots for the metrics for BERTopic. In this experiment, we examine whether there are any significant correlations, whether such correlations are different in significance or direction between classifiers and/or topic models. Furthermore, we also closely look at which measures have significant correlations. Spearman correlations were done to determine the relationships between coherency scores and classifier accuracies. The following correlations are for $n = 20$.

| Mean Accuracy | | | | | |
|---|---|---|---|---|---|
| Models | | Topic Amount | | | |
| Topic Model | Classifier | 25 | 50 | 75 | 100 |
| LDA | Log. Reg. | 0.8431 | 0.8576 | 0.8663 | 0.8694 |
| | Dec. Trees | 0.7908 | 0.8020 | 0.8049 | 0.8004 |
| BERTopic | Log. Reg. | 0.7117 | 0.7265 | 0.7489 | 0.7622 |
| | Dec. Trees | 0.7668 | 0.7858 | 0.8006 | 0.8100 |

Table 1: Average accuracy for each classifier using different topic models and topic amount; standard errors were within 0.005 or lower; Figure 3 shows barplots for these accuracies and is available in the appendix.

For LDA, two of the measures significantly correlated with the classifier accuracies and the others did not. The UMass measure had a strong negative correlation with logistic regression ($r_s$ = -.90, p $<$ .001) and decision trees ($r_s$ = -.62, p = .004). The NPMI measure also had a moderate positive correlation with logistic regression ($r_s$ = .53, p = .017) and decision trees ($r_s$ = .47, p = .036). The UCI measure had an insignificant and weak relationship with logistic regression ($r_s$ = .30, p = .200) and decision trees ($r_s$ = .40, p = .081). The CV measure similarly had an insignificant and weak relationship with logistic regression ($r_s$ = .34, p = .139) and decision trees ($r_s$ = .36, p = .115).

For BERTopic, all the measures significantly correlated with the classifier accuracies. For UMass, there was a positive moderate relationship with logistic regression ($r_s$ = .57, p = .009) and decision trees ($r_s$ = .50, p = .024). For UCI, there was a very strong positive relationship with logistic regression ($r_s$ = .87, p $<$ .001) and decision trees ($r_s$ = .80, p $<$ .001). For NPMI, there was a very strong positive relationship with logistic regression ($r_s$ = .82, p $<$ .001) and decision trees ($r_s$ = .89, p $<$ .001). For CV, there was a strong positive relationship with logistic regression ($r_s$ = .66, p = .002) and decision trees ($r_s$ = .69, p $<$ .001).

| Coherency-Accuracy Correlations | | | | | |
|---|---|---|---|---|---|
| Models | | Coherency Scores | | | |
| Topic Model | Classifier | UMass | UCI | NPMI | CV |
| LDA | Log. Reg. | **-0.904** | 0.299 | **0.528** | 0.343 |
| | Dec. Trees | **-0.615** | 0.400 | **0.471** | 0.364 |
| BERTopic | Log. Reg. | **0.570** | **0.872** | **0.821** | **0.662** |
| | Dec. Trees | **0.504** | **0.802** | **0.892** | **0.690** |

Table 2: Spearman's Rho correlations between coherency scores and classifier accuracy using topics; Significant correlations are in bold

## 4.4 Experiment 2

The second experiment looks at the correlation between individual topic coherency and classifier accuracy. This experiment looks at the relationship between coherency and its influence on classifier accuracy by examining its effect on the classifier directly. Table 3 shows the correlations across topic

coherence scores and classifier feature importance scores. Figure 6 shows scatterplots of topic coherence scores against feature importance scores within LDA, and Figure 7 shows the same scatterplots but for BERTopic.

Spearman correlations were also computed to determine the significance in correlation between topic coherency and feature importance. As a reminder, log-loss difference measures the difference in log-likelihood caused by the removal of a topic in the classifier, and Gini impurity measures how well a given topic can be used for classification. In short, a bigger (negative) log-loss difference and higher Gini impurity means stronger feature importance. The following correlations are for $n = 100$.

For LDA, four out of the eight correlations are significant. UMass has a moderate negative correlation with log-loss difference ($r_s$ = -.55, p $<$ .001) and a weak positive relationship with Gini impurity ($r_s$ = .38, p $<$ .001). UCI have non-significant correlations with log-loss difference ($r_s$ = -.19, p = .059) and Gini impurity ($r_s$ = .15, p = .15). NPMI has a weak negative correlation with log-loss difference ($r_s$ = -.22, p = .030) and a non-significant correlation with Gini impurity ($r_s$ = .16, p = .118). CV has a weak negative correlation with log-loss difference ($r_s$ = -.36, p $<$ .001) and a non-significant correlation with Gini impurity ($r_s$ = .19, p = .054).

For BERTopic, none of the feature importance scores correlate with either feature importance scores. UMass has no significant correlation with log-loss difference ($r_s$ = .19, p = .065) or Gini impurity ($r_s$ = .10, p = .340). UCI has no significant correlation with log-loss difference ($r_s$ = .06, p = .559) or with Gini impurity ($r_s$ = -.14, p = .17). NPMI has no significant correlation with log-loss difference ($r_s$ = -.01, p = .930) or with Gini impurity ($r_s$ = -.10, p = .306). And CV has no significant correlation with log-loss difference ($r_s$ = -.05, p = .647) or with Gini impurity ($r_s$ = -.02, p = .877).

| Feature Importance - Topic Coherency Correlations | | | | | |
|---|---|---|---|---|---|
| Model | | Coherency Scores | | | |
| Topic Model | Feature | UMass | UCI | NPMI | CV |
| LDA | Loss Dif. | **-0.550** | -0.189 | **-0.217** | **-0.359** |
| | Gini Imp. | **0.382** | 0.147 | 0.157 | 0.1935 |
| BERTopic | Loss Dif. | -0.185 | 0.059 | -0.0089 | -0.046 |
| | Gini Imp. | 0.096 | -0.137 | -0.103 | -0.016 |

Table 3: Spearman Rho correlations between coherency scores and feature importance scores of each topic; using first 100-topic models for each topic model; significant correlations are in bold

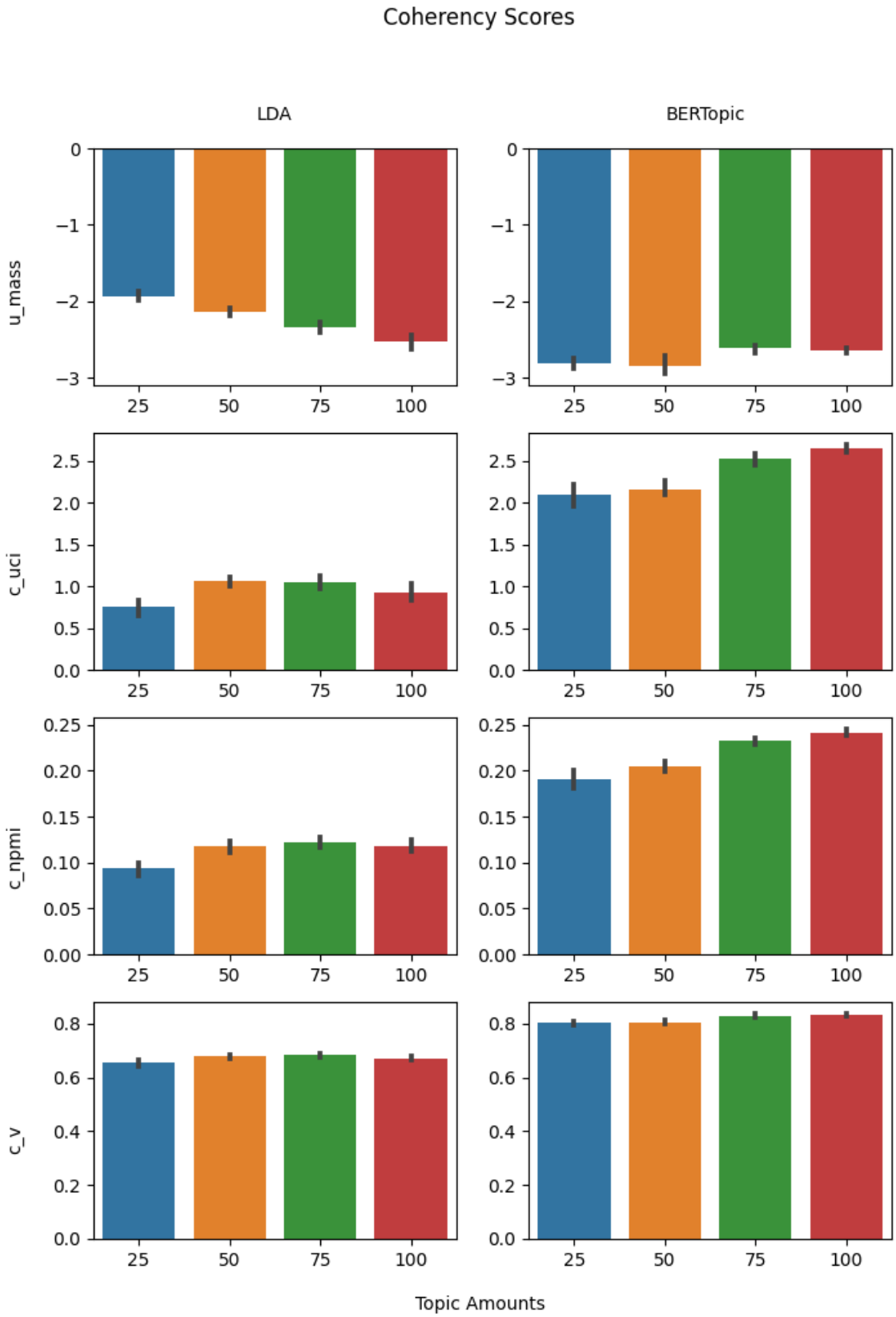


Figure 2: Coherency score averages across topic models and topic amounts; lower UMass means better coherence while higher UCI, NPMI, and CV mean better coherence; the graph suggests that BERTopic models tended to have better coherence across all coherence measures

# 5 Conclusion

## 5.1 Discussion

This project investigated the relationship between topic coherence and classifier accuracy. The main focus is whether topic coherence would predict the performance of a classifier using topics. To comprehensively explore this question, the project asked five specific questions:

1. Is there a correlation between model coherence and classifier accuracy?
2. Is there a correlation between individual topic coherences and feature importance?
3. Are correlations between coherence and accuracy/feature importance consistent across topic models?
4. Are correlations between coherence and accuracy/feature importance consistent across coherence measures?
5. Are correlations between coherence and accuracy/feature importance consistent across classifiers?

The first two questions investigate the relationship at two different levels. The first question looks at whether the coherence of a whole model predicts the performance of a classifier that used all the topics. The second question looks at whether classifiers primarily use topics with the most coherence or not. The three remaining questions investigate whether these correlations can be generalized across the three separate parts – those parts being the topic models, coherence measures, and classifiers.

The first experiment showed that there were significant correlations between model coherence and classifier accuracy. Most coherence measures correlated with classifier accuracy. With LDA, UMass had strong correlations with the classifiers and NPMI had moderate correlations. LDA classifiers had no significant correlation with UCI or CV. With BERTopic, the classifiers correlated with all coherence measures. In particular, UCI and NPMI had the strongest correlations with the classifiers. UMass and CV had moderate relationships.

At surface level, these results suggest we use UMass for LDA classifiers and UCI or NPMI for BERTopic. However, the correlation between UMass and LDA classifier performance is negative. Theoretically, an increase in UMass is an increase in coherency. Thus, a negative correlation means classifiers perform better as coherency decreases. In contrast, BERTopic's correlation with UMass is positive.

This effect is best explained by how UMass is calculated and how each model generates topics. Recall that UMass calculates the log probability of two words co-occurring in a document. With this in mind, recall that BERTopic's method specifically employs document clustering. This clustering results in topic words that are more likely to co-occur in documents. LDA's method, on the other hand, does not use the document explicitly. For one, this would explain why LDA's UMass score decreases as topic size increases while BERTopic's UMass score increases as topic size increases (see Figure 2). The UMass-accuracy correlation could then be better explained by the variable topic amounts. As such, UMass should not be used as a predictor for classifier performance due to its use of document co-occurrence. Rather, NPMI is better suited as a general predictor for classifier performance. The measure has a significant moderate-to-strong correlation with both topic models' classifiers.

The second experiment found significant correlations between individual topic coherency and feature importance. As a brief reminder, a lower (more negative) log-loss difference implies stronger feature importance. A higher Gini index implies stronger feature importance. The only significant correlations were found in LDA. BERTopic topic did not correlate with any feature importance measures.

Interestingly, LDA's UMass negative correlation with log-loss difference would imply a higher-coherence higher-effect relationship. This is roughly in line with LDA's NPMI and CV correlation as well. LDA UMass also correlated with Gini index, while no other coherence measure did. This would imply that, for individual topics in a single model, higher UMass topics are more useful for LDA classifiers.

That being said, the second experiment mainly shows that individual topic coherence does not correlate with feature importance. Very few coherence measures correlated with either classifier. Furthermore, most correlations were weak, and only one was moderate. This suggests that topic coherences cannot predict how a classifier will use the topics. It tells us very little about the classifier's performance. Thus, a topic model's average coherence is the better method to gauge classifier performance.

The results of the first experiment answer the three remaining questions. The third and fourth questions can be answered together. Some coherence measures significantly correlated with model classifiers. Specifically, NPMI and UMass. Given the issues with UMass discussed previously, NPMI should be considered the only consistent measure. UCI and CV only correlated with the BERTopic classifiers. Interestingly, UCI and NPMI have similar correlations strength with the BERTopic classifiers. Nevertheless, NPMI's consistent correlation with both model classifiers suggests it can be generalized for performance prediction.

The previous answer already tells us the answer for the fifth question. However, special attention is warranted for this answer. The general answer for NPMI's consistent correlation between model classifiers is possible because classifier correlations were also consistent. That is, correlations within models and between classifiers share strength and direction. For instance, NPMI finds strong positive correlations in both BERTopic classifiers. UMass, though theoretically problematic, also shows these tendencies within each topic model. For LDA, it finds moderate-to-strong negative correlations with classifiers, and for BERTopic, it finds moderate positive correlations. This would suggest that significant correlations can generalize across classifiers. More specifically, NPMI, due to its consistency, can be used to predict the performance of a classifier using topics.

There is, however, a caveat to this. The significant coherence measures (NPMI specifically) do not outright predict classifier performance. For example, LDA's models tend to have lower NPMI than BERTopic's (see Figure 3). Yet LDA's logistic regression models have higher accuracy, and both model's decision trees perform roughly the same. This means coherence should only be used to compare the accuracy of classifiers using the same topic model.

## 5.2 Conclusion

The experiments tell us that we can predict the performance of classifiers using coherence. In particular, only NPMI can be used to predict classifier performance. UCI and NPMI were not consistently correlated between the two topic models. Though UMass was consistent in significance, its directionality was not. Furthermore, the measure's dependence on document co-occurrence results in problematic interpretation. Finally, the correlation only works when comparing classifiers using the same topic models. NPMI does not directly predict the accuracy of a classifier. Therefore, NPMI can be used to predict the relative performance of a topic model classifier, but only when compared within

the topic model.

## 5.3 Limitations and Future Work

The experiments involved different replaceable factors so that the results could be generalized. Nevertheless, only a limited set of models and measures were used for each factor. Limiting the parts was done to ensure timely results; implementing each part took time. Still, these results may be only applicable to the models involved. As such, future work looking into this relationship should include other topic models and classifiers. Other coherence measures could also be included. However, it is important to note that the four measures used in the experiments currently correlate the most with human judgment. As such, any other measure – such as the others in Lau et al. (2014) – would test with lesser human judgment correlation. Arguably, the ideal coherence measure would be one that is capable of estimating its contribution to classifier accuracy and, at the same time, correlate with human judgment. Doing so would be helpful during model training and feature interpretation.

The experiments only use the arXiv corpus. This corpus primarily constitutes scientific and mathematical documents. Such documents share a number of commonalities (e.g. statistics, mathematical notation), which is expected when topic modeling. However, other corpora with varying amounts of shared commonalities may influence the correlations. Arguably, a different corpus may not be different at all from our results given the nature of topic modeling. In fact, it is most likely that a corpus with less commonality would be captured by the coherence scores as topic models would struggle to find good topics. Nevertheless, using another corpus would give us a better idea of how generalizable these correlations in different documents.

For future work, training data should be shared between topic models and classifiers. During implementation, the category imbalance issue only became apparent after topic model training. It became unfeasible to retrain the models using the reduced number of training data. It is possible that the topics generated from the balanced dataset would have led to different classifiers. Nevertheless, this does not invalidate the results. The focus was on the influence of coherence on classifier accuracy. That the results found a correlation implies that the same would be found with differently trained topic models. Still, future experiments should ensure this training data sharing. NLP pipelines that use topic modeling for feature engineering will most likely do the same.

Finally, the second experiment may not be feasible enough for its purpose. Classifiers use features differently for prediction. Figure 6 shows that, for LDA, the decision trees mostly used 10 features to predict the categories. In contrast, logistic regression uses all features to predict the categories. The classifier-specific metrics may be too different to determine per-topic coherence correlations. SHAP values could prove to be the solution to this problem (Lundberg & Lee, 2017). SHAP values are a recent development in the machine-learning domain. In brief, they use game theory to measure the general contribution of a feature in a classifier. Theoretically, they are applicable to any machine-learning model. This means correlation comparisons are much easier to make between different classifiers. This would also mean more competitive classifiers could be used as well. As such, future work could replace the classifier-specific feature metrics with SHAP values.

# Bibliography

Aggarwal, C. C., & Yu, P. S. (2001). Outlier detection for high dimensional data. In *Proceedings of the 2001 acm sigmod international conference on management of data* (pp. 37–46).

Aletras, N., & Stevenson, M. (2013). Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th international conference on computational semantics (iwcs 2013)–long papers* (pp. 13–22).

Allaoui, M., Kherfi, M. L., & Cheriet, A. (2020). Considerably improving clustering algorithms using umap dimensionality reduction technique: a comparative study. In *International conference on image and signal processing* (pp. 317–325).

Batmanghelich, K., Saeedi, A., Narasimhan, K., & Gershman, S. (2016). Nonparametric spherical topic modeling with word embeddings. In *Proceedings of the conferenece. association for computational linguistics. meeting.* (Vol. 2016, pp. 530–539).

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful? In *International conference on database theory* (pp. 217–235).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Machine Learning Research*, *3*, 993-1022.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, *30*, 31–40.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J., & Blei, D. (2009). Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, *22*.

Das, R., Zaheer, M., & Dyer, C. (2015). Gaussian lda for topics models with word embeddings. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing* (Vol. 1: Long Papers, pp. 795–804).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391–407.

Doan, T. N., & Hoang, T. A. (2021). Benchmarking neural topic models: An empirical study. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 4563–4368).

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *PNAS*, *101*, 5228-5235.

Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Hall, D., Jurafsky, D., & Manning, C. D. (2008, October). Studying the history of ideas using topic models. In *Proceedings of the 2008 conference on empirical methods in natural language processing* (pp. 363–371). Honolulu, Hawaii: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D08-1038`

Hinton, G. E. (2017). Training products of experts by minimizing contrastive divergence. *Neural computation*, *14*(8).

Hoffman, M., Blei, D. M., & Bach, F. (2010). Online learning for latent dirichlet allocation. In *Proceedings of the 23rd international conference on neural information processing systems - volume 1* (p. 856–864). Red Hook, NY, USA: Curran Associates Inc.

Hoffman, M., Blei, D. M., Wang, C., & Paisley, J. (2012). *Stochastic variational inference.* arXiv. Retrieved from `https://arxiv.org/abs/1206.7051` doi: 10.48550/ARXIV.1206.7051

Hoffman, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual acm conference on research and development in information retrieval, 1999* (pp. 50–57).

Lau, J. H., Newman, D., & Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th conference of the european chapter of the association for computational linguistics* (pp. 530–539).

Liu, D. C., & Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, *45(1-3)*, 503-528.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

McInnes, L., Healy, J., & Astels, S. (2017). hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, *2*(11), 205.

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Miao, Y., Grefenstette, E., & Blunsom, P. (2017). Discovering discrete latent topics with neural variational inference. In *34th international conference on machine learning.*

Miao, Y., Yu, L., & Blunsom, P. (2016). Neural variational inference and learning in belief. In *Proceedings of icml.*

Mimno, D., Wallach, H., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 262–272).

Newman, D., Bonilla, E. V., & Buntine, W. (2011). Improving topic coherence with regularized topic models. *Advances in neural information processing systems*, *24*.

Newman, D., La, J. H., Grieser, K., & Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics* (pp. 100–108).

Pandove, D., Goel, S., & Rani, R. (2018). Systematic review of clustering high-dimensional and large datasets. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *12*(2), 1–68.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Rehurek, R., & Sojka, P. (2011). Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, *3*(2).

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Reimers, N., & Gurevych, I. (2020). Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth acm international conference on web search and data mining* (p. 399–408). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2684822.2685324 doi: 10.1145/2684822.2685324

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth acm international conference on web search and data mining* (pp. 399–408).

Sarioglu, E., Choi, H. A., & Yadav, K. (2012). Clinical report classification using natural language processing and topic modeling. In *2012 11th international conference on machine learning and applications* (Vol. 2, pp. 204–209).

Sia, S., Dalmia, A., & Mielke, S. J. (2020). Tired of topics models? clusters of pretrained word embeddings make for fast and good topics too! *arXiv preprint arXiv:2004.14914*.

Srivastava, A., & Charles, S. (2017). Autoencoding variational inference for topic models. *ArXiv preprint arXiv:1703.01488*.

Steinbach, M., Ertöz, L., & Kumar, V. (2004). The challenges of clustering high dimensional data. In *New directions in statistical physics* (pp. 273–309). Springer.

Thakur, N., Reimers, N., Daxenberger, J., & Gurevych, I. (2020). Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv preprint arXiv:2010.08240*.

Wallach, H. M., Murray, I., Salakhutdinov, R., & Mimno, D. (2009). Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning* (pp. 1105–1112).

Wei, X., & Croft, W. B. (2006). Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international acm sigir conference on research and development in information retrieval* (pp. 178–185).

Yang, T.-I., Torget, A., & Mihalcea, R. (2011, June). Topic modeling on historical newspapers. In *Proceedings of the 5th ACL-HLT workshop on language technology for cultural heritage, social sciences, and humanities* (pp. 96–104). Portland, OR, USA: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W11-1513

# Appendices

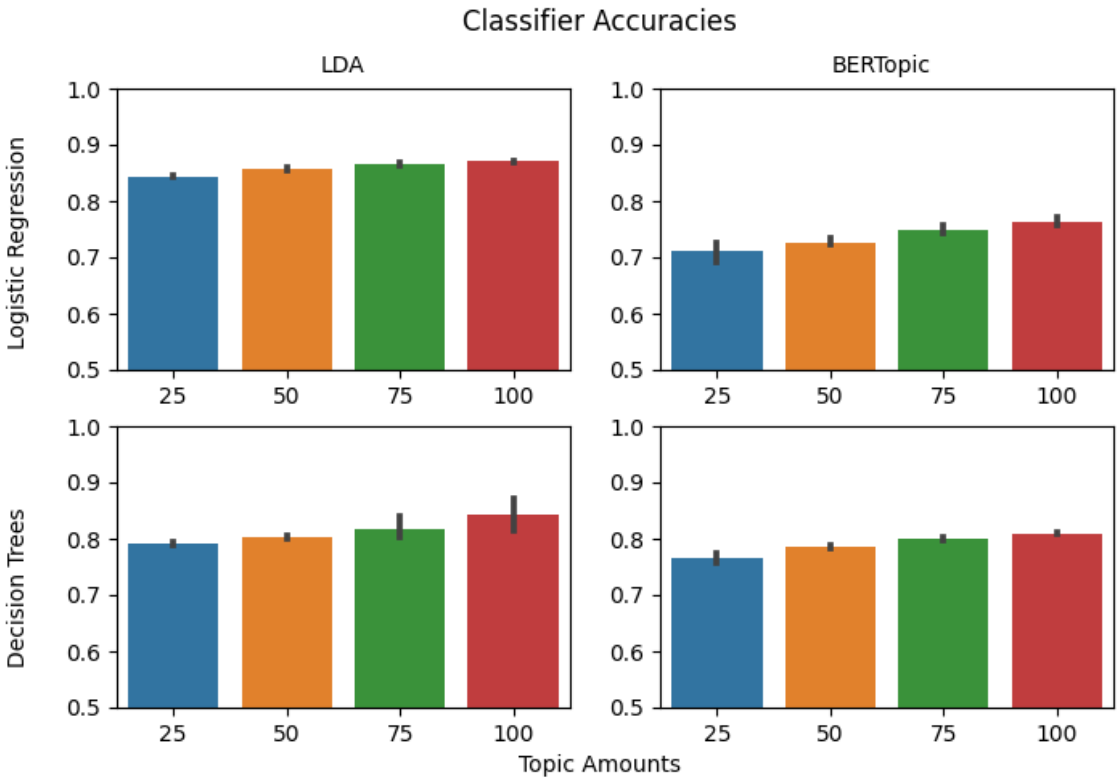


Figure 3: Average classification accuracy across topic models and classifiers; closer inspection suggests that classifiers using LDA topics yield better results compared to those using BERTopic topic

| Mean Coherency | | | | | |
|---|---|---|---|---|---|
| Models | | Topic Amount | | | |
| Topic Model | Measure | 25 | 50 | 75 | 100 |
| LDA | UMass | -1.9270 | -2.1328 | -2.3218 | -2.4478 |
| | UCI | 0.7561 | 1.0646 | 1.0286 | 0.9278 |
| | NPMI | 0.0936 | 0.1182 | 0.1198 | 0.1190 |
| | CV | 0.6527 | 0.6784 | 0.6795 | 0.6745 |
| BERTopic | UMass | -2.8043 | -2.8331 | -2.6125 | -2.6399 |
| | UCI | 2.0944 | 2.1605 | 2.5305 | 2.6501 |
| | NPMI | 0.1913 | 0.2048 | 0.2330 | 0.2414 |
| | CV | 0.8030 | 0.8050 | 0.8295 | 0.8329 |

Table 4: Average coherency for each topic model with different topic amounts; standard errors were within 0.15 or lower

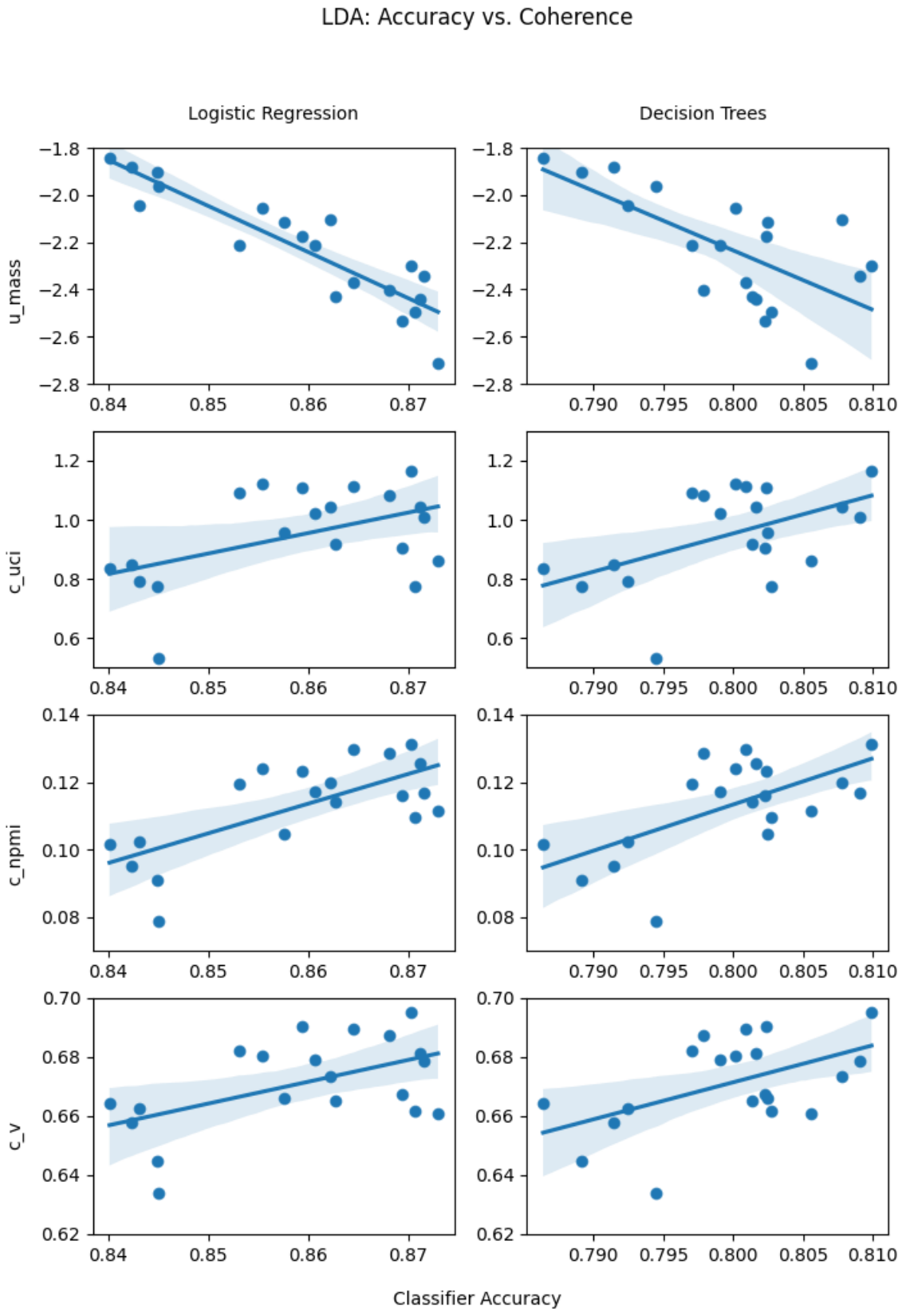


Figure 4: Scatterplots placing coherency scores against classifier accuracies for LDA; the left column shows all the plots for Logistic Regression and right column shows all the plots for the decision trees; each row show the plots of each different coherence measure

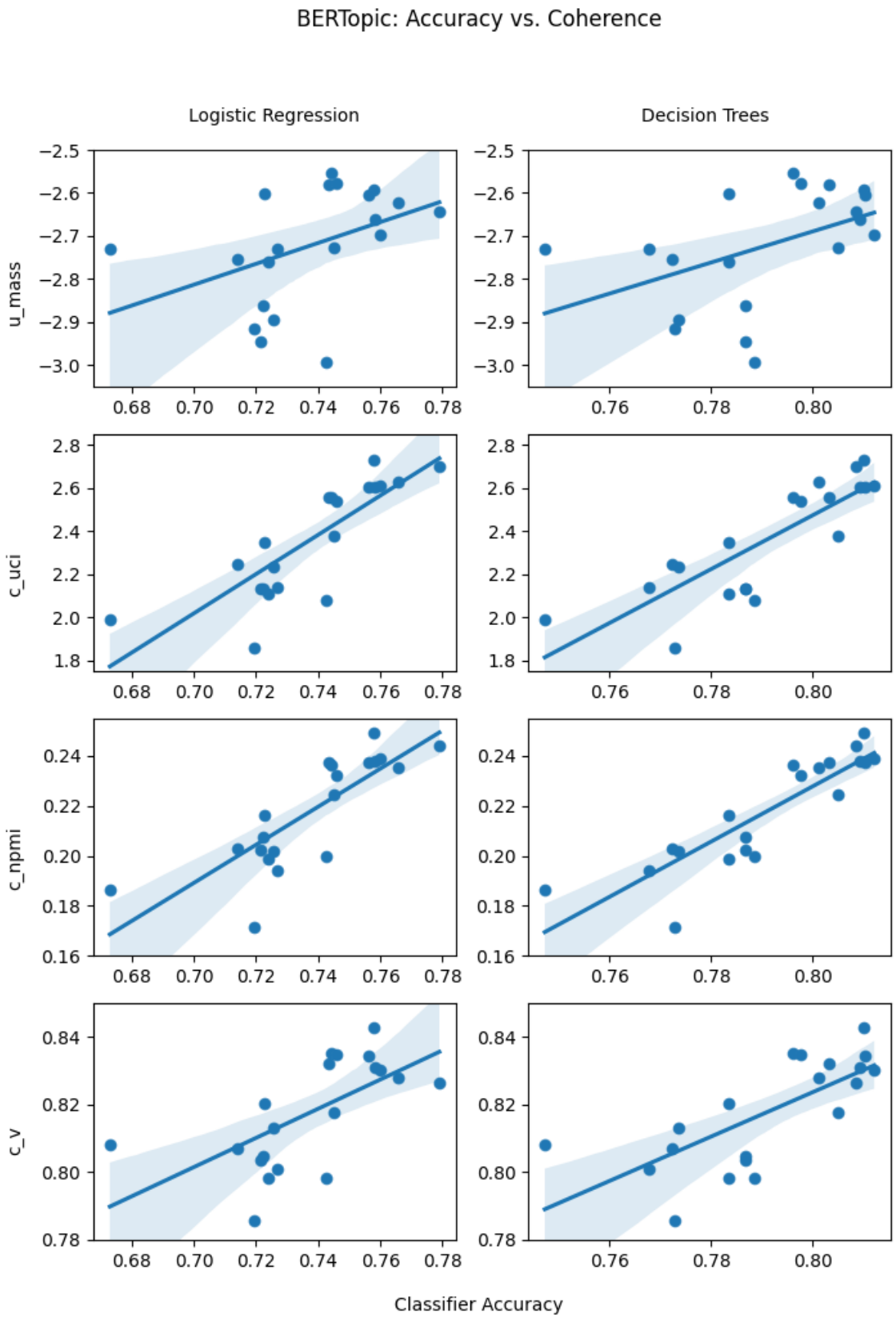


Figure 5: Scatterplots placing coherency scores against classifier accuracies for BERTopic; the left column shows all the plots for Logistic Regression and right column shows all the plots for the decision trees; each row show the plots of each different coherence measure

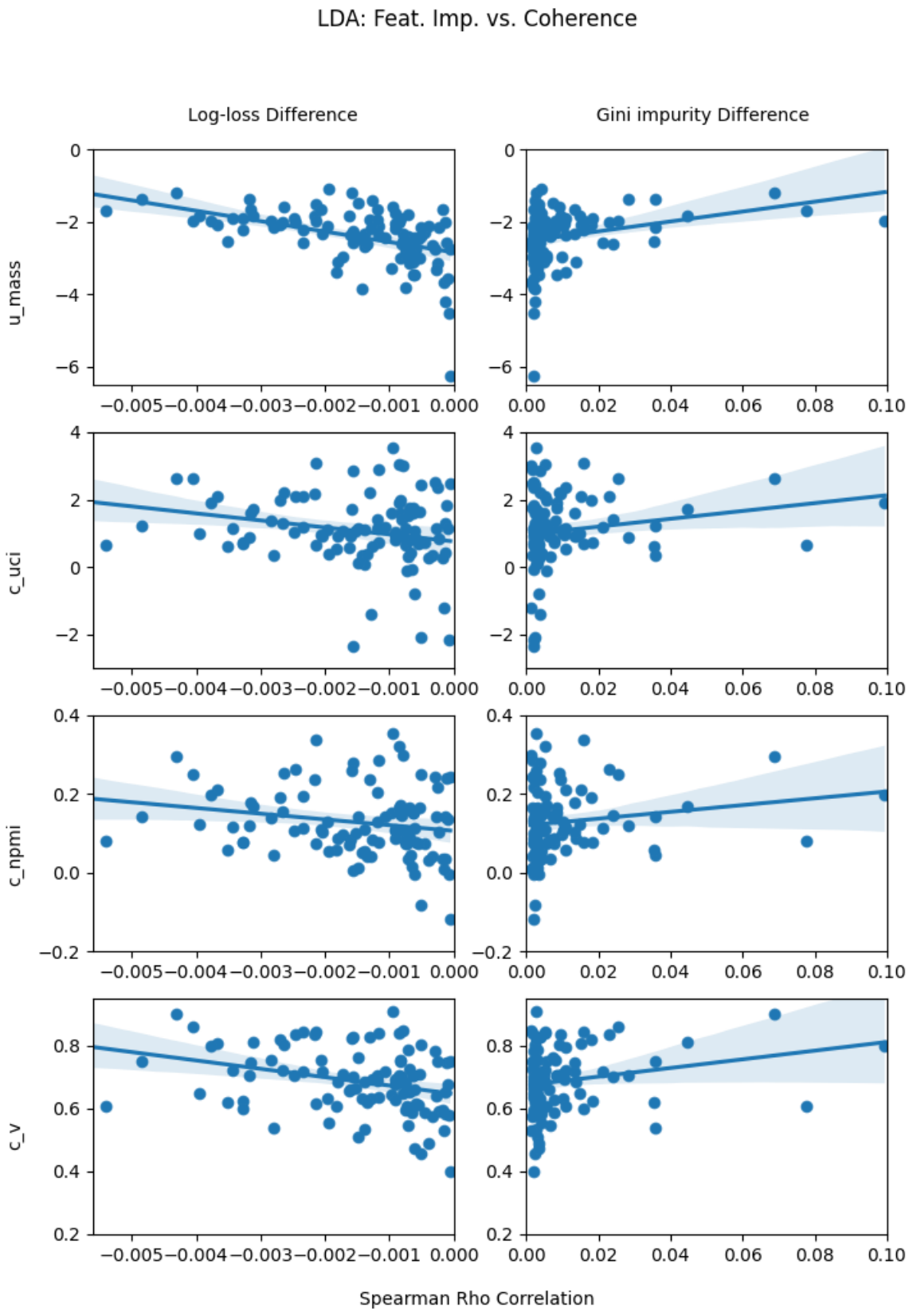


Figure 6: Scatterplots placing topic coherency scores against feature importance scores for LDA; the left column shows all the plots for Logistic Regression's log-loss difference and right column shows all the plots for the decision tree's Gini impurity; each row show the plots of each different coherence measure

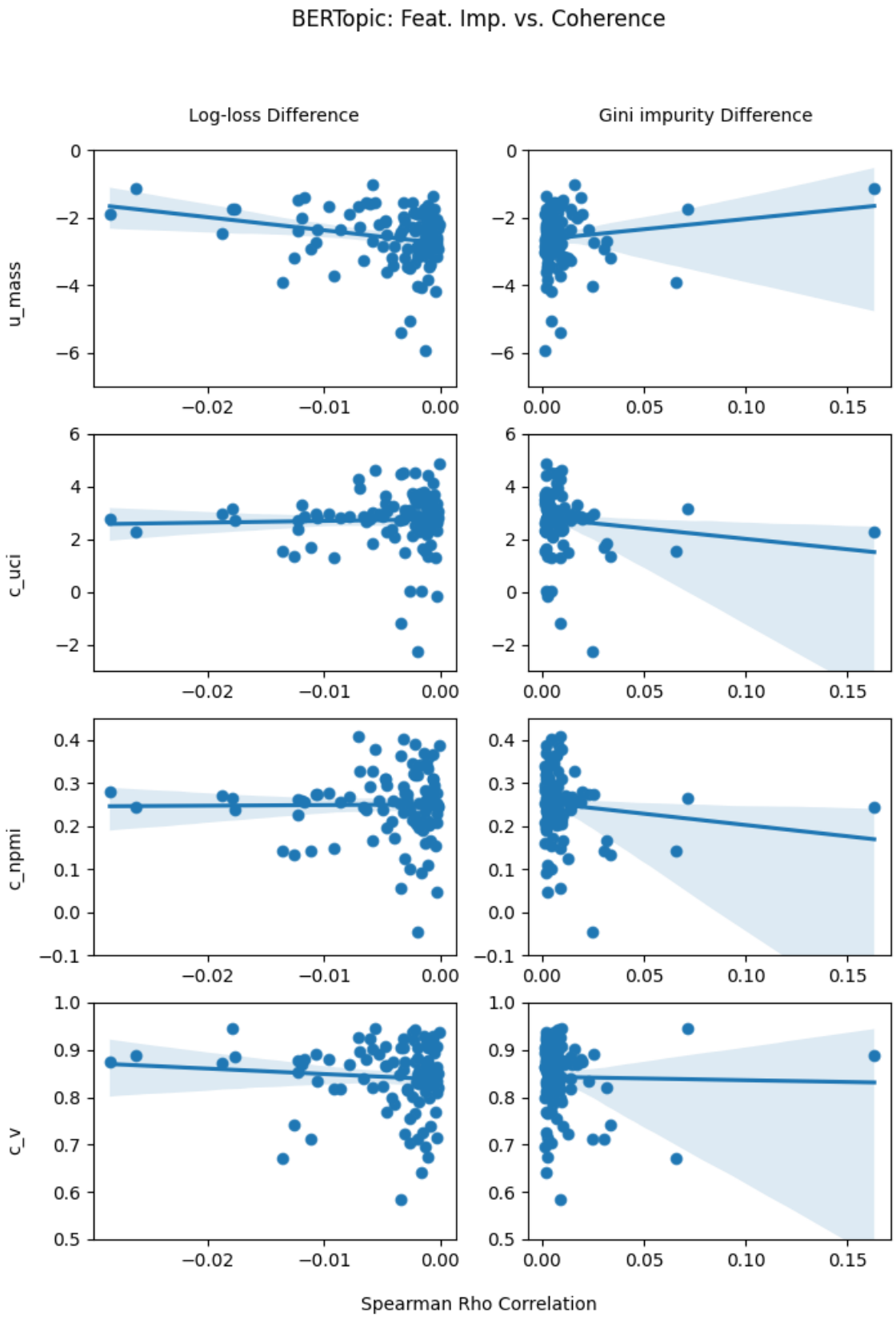


Figure 7: Scatterplots placing topic coherency scores against feature importance scores for BERTopic; the left column shows all the plots for Logistic Regression's log-loss difference and right column shows all the plots for the decision tree's Gini impurity; each row show the plots of each different coherence measure