

SUMMARY

News is being broadcast every day around the world in the form of news articles, television, and newspapers, which supply people with the latest information. Searching and categorizing the news is becoming a bigger problem since news is created at all times. Topic modeling is an approach that takes a set of documents and generates topics that can be used for categorizing text. We specifically look at extending the latent Dirichlet allocation (LDA) model and the Pachinko Allocation Model (PAM) with metadata.

In this paper we answer these research questions:

- *How can we establish models that incorporate metadata from the Nordjyske dataset?*
- *How does including metadata within such models impact the resulting topics?*

We work with a dataset from Nordjyske, a Danish news agency. This dataset contains 248,385 articles from 2017 to 2019. Basic preprocessing is done to make the data more applicable for topic models. Each article includes three types of metadata: author, category, and taxonomy, that we chose to extend our topic models with.

The author metadata contains the name of the author, who has written the article. This metadata is fully observed within the dataset, and each article only has a single author, so we do not account for multiple authors. There are 227 different authors within the dataset and after preprocessing 184 remain.

The category metadata describes a variety of different aspects, but the categories are generally either about a specific newspaper or an overall subject. This metadata is also fully observed within the dataset, and there are 58 unique categories in the dataset before processing, and 34 categories after preprocessing.

The taxonomy metadata describes a hierarchical sequence of the topical or geographical subjects associated with the article. Each sequence consists of several taxonomy entries. This metadata is only partially observed within the dataset, with $\sim 25\%$ of the articles containing this metadata. There are 1135 different taxonomy entries and after preprocessing 355 remain.

The models we have made that include metadata for our evaluations are called the author-topic, category-topic, and taxonomy-topic model. We also use a standard LDA model as a baseline for the performance of our models. In LDA, D is the number of documents in the corpus, N_d is the number of words in document d , and K is the number of topics. Topics are represented as distributions over words and documents are represented as distributions over topics.

In the author-topic and category-topic models, there are no document-topic distributions θ . Instead, each of the A authors and C categories have their own topic distribution. This is based on the assumption that authors prefer to write about specific topics and that categories of the articles were chosen based on the content of the finished article or that local newspapers have their own unique topic preferences. For our category-topic model and author-topic model, each document d is associated with one category c_d from a set of categories C and one author a_d from a set of authors A , which is used when drawing a word topic.

To have our taxonomy-topic model handle the hierarchical structure of the taxonomy metadata, we use a hierarchical topic model called the Pachinko Allocation Model (PAM). PAM generalizes LDA, making it possible to construct topic hierarchies based on any Directed Acyclic Graph (DAG) structure. PAM is a topic model focusing on finding topics of different abstraction levels and modeling the correlations between these topics.

Without modification, PAM finds topics with the same structure as our taxonomy, but the taxonomy entries would be disregarded during training since it would generate new taxonomy sequences. However, in our case, we have a dataset with a partially observed taxonomy metadata ($\sim 25\%$ of the documents), and we want to use the existing metadata information to estimate the topics quicker and more accurately.

To account for this, we only sample the unobserved nodes within the topic sequences. For some of the dataset only the fourth layer is unobserved, but for the $\sim 75\%$ documents without a taxonomy sequence, all layers are unobserved. The observed taxonomy sequences are never sampled, hence they are 'locked' in place. This creates a constant context for the taxonomy topics, which the documents with unobserved taxonomy sequences are fitted around.

The main metric used, in our evaluation, is the topic coherence metric C_v . This metric indicates how semantically similar the top words within each topic are, and is an indication of the quality of the topics of a topic model. The second metric used in our evaluation is topic difference. The topic difference is another metric that is used to check the quality of the topic model. It is based on the assumption that a good topic model has little overlap between topics.

Before fitting our models, we search for the optimal hyperparameters, since these can vary based on the dataset. The hyperparameters we are testing in this grid search are the number of topics K and the α and η parameters. We only run the grid search on the standard LDA model, with the assumption that the number of topics that perform well for this model, also performs well for the metadata models, when the same dataset is used. Based on the topic coherence of the model we choose $K = 90$, $\alpha = 0.01$, and $\eta = 0.1$ as the hyperparameters for all models in our evaluation.

For the results, the topic coherence of LDA is 0.520, author-topic is 0.335, category-topic is 0.370, and taxonomy-topic is 0.660. This shows that the author-topic and category-topic models performed the worst in topic coherence, whereas the taxonomy-topic model is outperforming all other models. However, the elapsed time of the taxonomy-topic model is much higher than the standard LDA.

In the analysis of the models, we explored the highest probable words from each model that would appear in a arbitrarily chosen article. It is seen that there is a large amount of overlap between the models, but the words that best summarized the article came from the LDA and taxonomy-topic models.

For the author-topic and category-topic models, we are also able to analyze the similarity between author pairs and category pairs. This similarity is calculated using symmetric Kullback-Leibler divergence of either the author-topic distributions or the category-topic distributions. This gives the intuition of how similar authors are based on the topics they write, and categories in the topics they cover. It is seen that it is difficult to find specifically why authors are similar because they often write about many different subjects. From the category similarities, there are obvious high similarity pairs, such as 'Sport-avis' and 'Morsø Sport' where they both include sports articles.

For the taxonomy-topic model, we found that the topics, in general, were the most understandable of all topic models, which made sense with it having the highest topic coherence. The taxonomy-topic model was also able to separate words without meaning into their own topics, which allows the model to apply an extra layer of preprocessing, automatically filtering away irrelevant words into topics.

As further experimentation and exploration, we tested what would happen if we did further preprocessing. We also tested the effect of including multiple topic distributions in our models and what would happen if we used the author and category metadata in the PAM.

For further preprocessing, we tested what effect including stemming in our dataset would have on the models. With stemming on the LDA model, it was seen that there were a lot fewer unique words. When looking at the topics of the model, they generally did not change much, but may be slightly more understandable with fewer words with the same meaning.

One of the new models, that we made is called the author-category model. This model includes both an author-topic distribution and a category-topic distribution and uses both to draw word topics. This model has only a slightly higher topic coherence of 0.390 compared to the author-topic and category-topic model, and the topics are still not very understandable.

We also made two other models called Author-Doc and Category-Doc. These models are made by combining the LDA model with the author-topic and category-topic model, and also use both distributions to draw word topics. We run these models using the same hyperparameters as all the other models and they get similar results to the standard LDA model. Author-doc gets a topic coherence of 0.543 and category-doc gets 0.530, which is slightly higher than the LDA model.

Finally, we tested what would happen if we ran PAM with just the author and category metadata. A Four-Level PAM was used for both of these models, but otherwise, the same structure is used as with the taxonomy-topic model. The author PAM gets a topic coherence of 0.598 and the category PAM gets 0.585. These models achieved better results than the previous author-topic and category-topic models and the LDA model, but lower topic coherence compared to the taxonomy-topic model.

Integrating News Article Metadata into Topic Models

Rasmus Engesgaard Christensen
Department of Computer Science
Aalborg University
Aalborg, Denmark
rech16@student.aau.dk

Peter Langballe Erichsen
Department of Computer Science
Aalborg University
Aalborg, Denmark
perich16@student.aau.dk

Dennis Højbjerg Rose
Department of Computer Science
Aalborg University
Aalborg, Denmark
drose16@student.aau.dk

Abstract—Topic models are used to find underlying topics in a set of documents. Integrating metadata into topic models can improve their performance. We introduce models that extend latent Dirichlet allocation (LDA) to include author and category metadata information and a model which integrates taxonomy metadata into the Pachinko Allocation Model (PAM). The author-topic and category-topic models are based on the author-topic model with modifications, and the taxonomy-topic model is based on PAM. To make the PAM include the metadata information, a novel topic locking mechanism is created. The results show that for a news article dataset, our taxonomy-topic model integrates the metadata well and improves the elapsed time in comparison to the original PAM. The taxonomy-topic model has a higher topic coherence and more understandable topics than LDA. Our results show that integrating metadata can improve topic modeling in various ways.

Index Terms—Machine learning, Natural Language Processing

I. INTRODUCTION

News is being broadcast every day around the world in the form of news articles, television, and newspapers, which supply people with the latest information. Searching and categorizing the news is becoming a bigger problem since news is created at all times. Topic modeling is an approach that takes a set of documents and generates topics that can be used for categorizing or annotating text documents, such as news articles [3].

Latent Dirichlet allocation (LDA) is a well-cited topic model which generates topics and topic distributions for documents based on the words in documents [6]. Extensions of LDA have also been proposed to model various other information sources to generate better topics and/or topics, however, with different focuses and potential uses. author-topic model by Rosen-Zvi et al. [15] and the MetaLDA model by Zhao et al. [17] being notable examples.

The author-topic model by Rosen-Zvi et al. [15] combines the LDA model with author information to model the relationship between authors and the documents they have written. This is based on the assumption that most authors usually write about only a few different topics. They show that the author-topic combination yields better and more coherent topics, which begs the question of whether any other document-related data can be applied similarly. However, they only test

their algorithm on a dataset of scientific papers, where the authors usually only write about a small set of subjects (their research field), which might not be the case for other fields, like journalism.

We have a dataset from a Danish media group, called Nordjyske, with three years of article data. The dataset contains a variety of different metadata, which have the potential to improve topic models in the same way as Rosen-Zvi et al. [15]. In this paper, the dataset is used with a focus on topic modeling and the incorporation of metadata.

The dataset used with a topic model can have a large impact on the topics generated. When referring to a dataset, we talk about a dataset including metadata information. An example of metadata could be the publication date of a text document. In most contexts, it is not needed to use and understand the document; however, it provides more context to the main content of the document.

We want to construct different topic models that incorporate various metadata, to see whether incorporating this information changes the topic quality. We also investigate differences between the produced topics for various models, to evaluate their potential uses.

We define the following research questions to investigate in this paper:

- *How can we establish models that incorporate metadata from the Nordjyske dataset?*
- *How does including metadata within such models impact the resulting topics?*

Our work is similar in goal to that of Zhao et al. [17] since we work with metadata information and are applying it to an LDA model. However, Zhao et al. [17] learn a specific Dirichlet prior based on the metadata given in their dataset, which affects the document-topic distribution used with the LDA. Instead of using the document-topic distribution for drawing word topics, we want to create a new metadata-topic distribution that influences the drawn topics.

We investigate the effect that creating a metadata-topic distribution for any specific metadata information, such as the author information, has on the resulting topics. The intuition behind this is to create new topic models, which describe the data in a different way using topics that are influenced by metadata. For example, if the metadata describes something

about geographic locations, location-specific topics are generated, which could be useful in many cases.

The metadata, within the Nordjyske dataset, includes author information, as well as higher-level categorical information, which is described in Section II. These categorical metadata are the basis for the novelties in this paper, which is our category-topic model that makes a topic distribution for each category in the dataset and using the Pachinko Allocation Model (PAM) on a hierarchical metadata (taxonomy) by applying a locking mechanism.

The paper is organized as follows: Section II describes the dataset and the metadata used in the evaluation. In Section III, we explore related work within topic modeling using metadata. Section IV gives preliminary knowledge about the topic models we adapt, and Section V describes our metadata topic models and shows the plate notation. In Section VI, we set up an evaluation to test the performance of the different metadata topic models and present the results. In Section VII, we analyze and discuss the results of our topic models. Finally, in Section VIII, conclusions and future work are given.

II. DATASET AND PREPROCESSING

Nordjyske is a Danish news agency that maintains multiple newspapers, radios, and other news sources throughout north Jutland, a region in Denmark. They store their news articles in a non-public database, where each article contains multiple metadata types which describe aspects of the data, e.g., the author and publication date. The dataset we use ranges from 2017 to 2019 and contains 248,385 articles.

We perform basic preprocessing to make the data more applicable for topic models. Firstly, because the dataset includes articles from multiple cities and regions, duplicates do occur in the dataset. These duplicates are removed, so only unique articles are kept. After this, words that appear in less than 10 articles and words that appear in more than 10% of articles are filtered out. This is done to keep words that are used enough to find patterns in topics and to remove words that are similar to stop words. Finally, after the words are filtered out, the empty documents are removed. After preprocessing, the dataset contains 139,060 articles that use a vocabulary of 69,192 unique words.

In the following sections, we describe each of the metadata types which are analyzed. Further details about these metadata types can be seen in Appendix Section A. The inclusion of a stemming process has also been tested and is described in Appendix Section E.

The metadata types that we are working with do have some problems that can be mitigated to a degree by preprocessing. These problems are all related to some metadata values only being used in a few documents. Since the metadata values are used to group documents together and find common topics and words within grouped documents, metadata values that group too few documents are not very relevant and are therefore combined or removed.

A. Author

The author metadata contains the name of the author, who has written the article. Each article only has a single author, so we do not account for multiple authors, whereas Rosen-Zvi et al. [15] account for multiple authors. This metadata is fully observed within the dataset, meaning that every article has an author. Originally there were 227 different authors within the dataset. After combining authors that have written less than 14 documents ($\sim 0.001\%$ of the total document set) into a 'misc' author of size 204, 184 authors remain.

B. Category

The category metadata describes a variety of different aspects. One part of the categories contains which specific newspaper the article belong to, e.g., 'Aalborg-Newspaper'. Another part of the categories describes the overall subject of the document, such as 'Culture' and 'Sports-newspaper'. However, there are also nonsensical categories such as '53. Frederik', that do not seem to describe the subject of the document. This metadata is fully observed within the dataset and there are originally 58 different categories in the dataset. However, while most of these categories cover a significant number of documents, some categories are only used by a few documents. After combining all categories covering less than 140 documents ($\sim 0.01\%$ of the total document set) into a single new 'misc' category of size 229, 34 categories remain. All of the category labels can be seen in Appendix Table X and the statistics in Appendix Table XIII.

C. Taxonomy

The taxonomy metadata describes a hierarchical sequence of the topical or geographical subjects associated with the article. Each taxonomy sequence consists of several taxonomy entries. This metadata is only partially observed within the dataset, which means that $\sim 25\%$ of the articles contain this metadata. It is also possible for articles to contain multiple taxonomy sequences. General patterns for taxonomy sequences include:

- PLACES/Country/Region/Town
- TOPICS/Sub-Topic/Subsub-topic

Examples of these sequences are:

- PLACES/Danmark/Nordjylland/Aalborg/Lillevorde
- TOPICS/Religion/Christianity

About 80% of the observed sequences contain the 'PLACES' variable and 20% use the 'TOPICS' variable. There are also a few other top-level taxonomy entries; however, they are not as informative and are very rarely used. Originally there were 1135 different taxonomy entries; however, after removing taxonomy entries used by less than 14 documents ($\sim 0.001\%$ of the total document set), only 355 remain.

III. RELATED WORK

Rosen-Zvi et al. [15] have developed an LDA model called the author-topic model, which incorporates authorship information in the LDA model. Specifically, each document has a group of authors, where for each word an author is chosen

uniformly at random. The author is then used in combination with an author-topic distribution to choose a specific topic that this author writes about, and the word is then generated from this topic. The purpose of using authorship information this way is to show patterns in which topics an author usually writes about, and be able to explore how related authors are to each other, based on what they write about. Rosen-Zvi et al. also show that the combination of authorship and LDA yield more coherent topics. However, the author-topic model is applied to scientific article datasets, where the authors usually write about the same subject. We apply this model to the Nordjyske news article dataset to see whether similar performance can be obtained.

Zhao et al. [17] present a model, called MetaLDA, which can incorporate both metadata information and word embeddings within a topic model. Since the field of incorporating word embeddings within generative topic models has gained popularity[8], Zhao et al. [17] show how to use word embeddings for a variety of different datasets. They also compare against a list of other models that take either metadata information or word embeddings into account when doing inference. We take inspiration from Zhao et al. [17], but we do not employ the model they present. We adapt the author-topic model instead of MetaLDA because, in MetaLDA, each document has a specific Dirichlet prior for its topic distribution, which is computed from the metadata of the document, making it difficult to analyze the effect of a specific metadata type in a model that includes multiple metadata types.

Chang et al. [7] present different methods for evaluating probabilistic topic models. An important observation they made is that a good held-out likelihood, normally called perplexity, infers less semantically meaningful topics. Due to this, we do not use perplexity as an evaluation metric. Röder et al. [14] introduce new measures for evaluating topic models, where some of them use the co-occurrence or conditional probability of words within topics to measure how coherent the topics are. In order to verify that these metrics work, they conduct a large user study in conjunction with these metric evaluations. We use an evaluation metric called "Topic coherence" presented by Röder et al. [14] to evaluate the topic quality of each of our models.

Li and McCallum [9] present a Directed Acyclic Graph (DAG) structured topic model called the Pachinko Allocation Model (PAM), where topics are in a hierarchical structure, which allows it to find two types of topics, namely super-topics and sub-topics. PAM is a generalization of the LDA model, where super and sub-topics are used to correlate topics within the model, e.g., a football topic being part of a sports topic. We have a hierarchically structured taxonomy metadata within our dataset, which fits well with the hierarchical structure of this model. We create a modified PAM to account for this metadata and investigate whether this type of information can improve the topic quality.

There also exist a variety of models that look at either document or word metadata. Some examples of models that incorporate document-level metadata are: Supervised LDA

(sLDA) by Blei and McAuliffe [5], Labelled LDA (LLDA) by Ramage et al. [13], and the Dirichlet Multinomial Regression (DMR) model by Mimno and McCallum [10]. SLDA learns a model given the restriction of only having one label per document, while LLDA allows multiple labels per document, though it requires the number of topics to be the same as the number of unique metadata labels. DMR handles metadata similarly to MetaLDA [17] by incorporating labels on the prior of the documents' topic distributions. Examples of models that incorporate word-level metadata are: WF-LDA by Petterson et al. [12] and LF-LDA by Nguyen et al. [11]. WF-LDA extends LDA by using word features to make a prior for the topics. LF-LDA takes the approach of replacing LDA's topic-word Dirichlet multinomial component with a two-component mixture of a topic-word Dirichlet multinomial component and a latent feature component. We focus specifically on document-level metadata since this is easily available in our dataset.

Some of these works have shortcomings that we want to account for. The model may not be built around incorporating metadata, such as the PAM [9], which then has to be modified. The metadata may be incorporated in the model in such a way that a deeper analysis of the becomes difficult, such as with the MetaLDA model [17]. It may also simply be that the work does not explore multiple types of metadata for their models, which is the case for the author-topic model [15].

From these works, we adapt the concepts from Rosen-Zvi et al. [15] for new models that can handle the unique characteristics of our dataset and support a deeper exploration of the models. As mentioned earlier, one of the main reasons why we use the concepts from the author-topic model instead of the newer MetaLDA model [17] is because each document in MetaLDA has a topic distribution that is based on the document's metadata. Analyzing such a model that includes multiple metadata can be complicated, while in the author-topic model, other metadata can be included as their own meta-topic distributions and be analyzed further. Also, since MetaLDA uses the document-topic distribution as its base, we would not be able to explore other interesting connections, such as the connection between learned category-topic distributions and the topics that are most probable for specific categories. We also explore whether adapting the PAM [9] to work with a hierarchically structured taxonomy metadata, described in Section II-C, gives the model more context and improves the topic quality or performance.

IV. PRELIMINARIES

Here we present the foundation for the two models that we are adapting in this paper.

A. Latent Dirichlet allocation

The purpose of LDA, and topic models in general, is to create a tool for exploring collections of text. Topic models do this by uncovering the underlying semantic structure of a text collection by using hierarchical Bayesian models. LDA

uncovers this semantic structure by discovering patterns of word use in documents and finding topics based on these [4].

The standard LDA by Blei et al. [6] can be described by the following generative process, which is the way the model assumes the documents arose: D is the number of documents in the corpus, N_d is the number of words in document d , V is the size of the vocabulary, and K is the number of topics. Topics are represented as distributions over words and documents are represented as distributions over topics. LDA assumes that the topics are shared across the corpus, while the document-topic distributions are unique for each document. For each topic $k \in \{1, \dots, K\}$ a topic-word distribution β_k is sampled from a V -dimensional Dirichlet distribution parameterized by η . That is, K topics $\beta_{1:k}$ are sampled, each being a distribution over the vocabulary, written as: $\beta_k \sim \text{Dir}(\eta)$. Likewise, for each document $d \in \{1, \dots, D\}$ a document-topic distribution θ_d is sampled from a K -dimensional Dirichlet distribution parameterized by α . For each word $n \in \{1, \dots, N_d\}$ in each document d , a topic $z_{d,n}$ is sampled from a K -multinomial distribution θ_d , and then a word $w_{d,n}$ is sampled from a V -multinomial distribution $\beta_{z_{d,n}}$. The generative process for each document is seen in these steps:

- A) Draw topic proportion $\theta_d \sim \text{Dir}(\alpha)$
- B) For each word n in the document:
 - a) Draw topic assignment $z_{d,n} \sim \text{Mult}(\theta_d)$
 - b) Draw word $w_{d,n} \sim \text{Mult}(\beta_{z_{d,n}})$

The result is K topics, based on D documents. These topics are represented by a $K \times V$ matrix of topic-word distributions and a $D \times K$ matrix of document-topic distributions. The plate notation for LDA can be seen in Figure 1.

B. Author-topic LDA

Rosen-Zvi et al. [15] present the author-topic model. It seeks to find topics based on author metadata, and is based on the assumption that authors prefer to write about specific topics. In this model, there are no document-topic distributions θ , but rather one author-topic distribution for each author. The reason for this is to find the interest of authors within the documents that we are analyzing. The generative process for the author-topic model is similar to that of the LDA model with a few minor changes. The model assumes that there are multiple authors for each document which is modeled by the \mathbf{a}_d and x variables in Figure 2a. Before drawing a word topic, we need to select an author x from \mathbf{a}_d . The generative process is seen in the following steps:

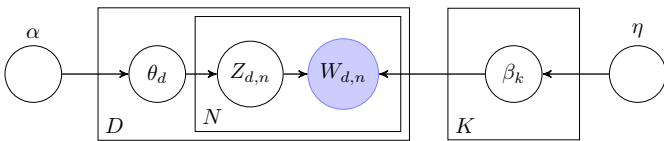


Figure 1: Plate notation for latent Dirichlet allocation (LDA).

- A) For each author, draw topic proportion $\theta_a \sim \text{Dir}(\alpha)$
- B) For each word n in the document:
 - a) Draw author assignment $x \sim U(\mathbf{a}_d)$
 - b) Draw topic assignment $z_{d,n} \sim \text{Mult}(\theta_x)$
 - c) Draw word $w_{d,n} \sim \text{Mult}(\beta_{z_{d,n}})$

Here $U()$ denotes the discrete uniform distribution.

Rosen-Zvi et al. [15] describe the author-topic model where for each document d , they assign a vector of authors \mathbf{a}_d from a set of authors A , and for each word uniformly choose an author x at random from this vector. The original plate notation for the author-topic model can be seen in Figure 2.

C. Pachinko Allocation Model

The Pachinko Allocation Model (PAM) [9] is a topic model that generalizes LDA, making it possible to construct topic hierarchies based on any DAG structure. The model focuses on finding topics of different abstraction levels, as well as modeling the connections between these topics.

Each node in the DAG structure represents a topic in the pachinko allocation model. However, unlike LDA where topics are distributions over words, in PAM topics are distributions over words and/or other topics further down the hierarchy of the DAG structure.

Li and McCallum [9] present a Four-Level PAM, which is a layered PAM meaning that the DAG structure is divided into layers, with every node in a layer being fully connected to every node in the next layer of the hierarchy. It consists of L layers of varying sizes S_0, S_1, \dots, S_{L-1} . The first layer consists of only the root node, a topic which all documents are part of. The bottom layer consists of leaf nodes, which represent the words in the vocabulary. The rest are middle layers representing topics of different abstraction levels.

$T = t_1, t_2, \dots, t_s$ is the set of topics in the PAM. Each topic t_i is associated with a Dirichlet distribution $g_i(\alpha_i)$ based on a vector α_i which has the same dimension as the number of children in t_i . While it is possible to use different α values for each topic, as shown below, we found through experimentation that using the same α value for all layers still provided good results.

The generative process for each document d in PAM consists of the following steps, as described by Li and McCallum [9]:

- A) Sample $\theta_{t_1}^{(d)}, \theta_{t_2}^{(d)}, \dots, \theta_{t_s}^{(d)}$ from $g_1(\alpha_1), g_2(\alpha_2), \dots, g_s(\alpha_s)$, where $\theta_{t_i}^{(d)}$ is a multinomial distribution of topic t_i over its children.
- B) For each word $w \in d$,
 - Sample a topic path \mathbf{z}_w of length $L_w : < z_{w1}, z_{w2}, \dots, z_{wL_w} >$. z_{w1} is always the root and z_{w1} through z_{wL_w} are topic nodes in T . z_{wi} is a child of $z_{w(i-1)}$ and it is sampled according to the multinomial distribution $\theta_{z_{wL_w}}^{(d)}$.
 - Sample word w from $\theta_{z_{wL_w}}^{(d)}$.

The intuition behind this generative process is to create all possible topic sequences and combine these into a multinomial

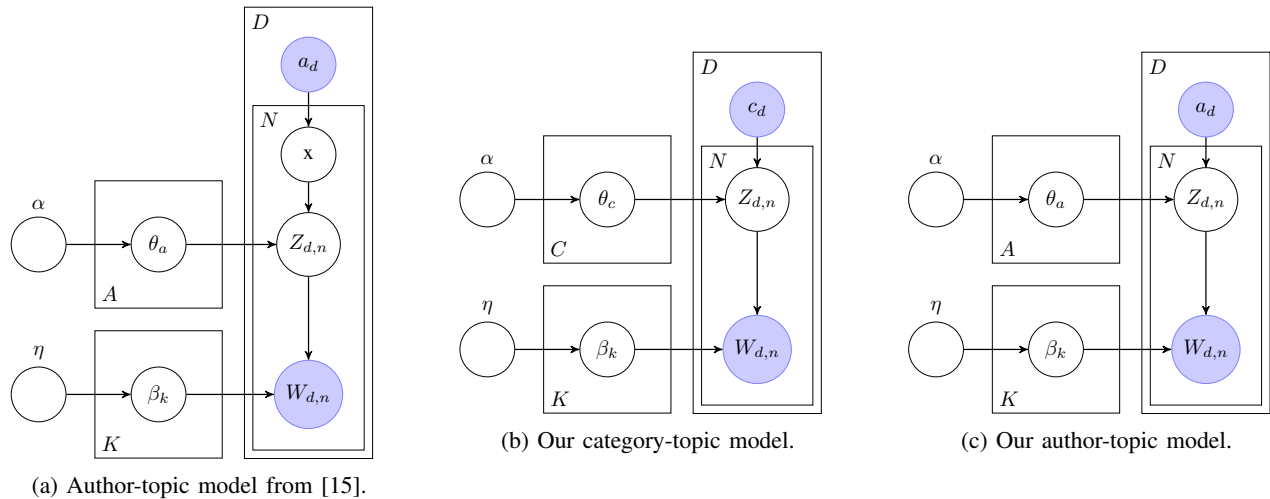


Figure 2: Plate notation for the metadata LDA models.

distribution to draw a topic sequence from. Otherwise, the Gibbs sampling is the same as with the LDA model.

The plate notation from Li and McCallum [9] and our modification can be seen in Figure 3.

V. PROPOSED TOPIC MODELS

In this section, we detail the modifications that we have implemented to describe our proposed topic models outlined in Section IV. We present three models, one for each of the metadata types detailed in Section II. Two of the models, the author-topic model and category-topic model, are based on the author-topic model explained in Section IV-B. The taxonomy-topic model is based on PAM, which is explained in Section IV-C.

A. Author-Topic model and Category-Topic model

We model both the author and category metadata types similarly to the model by Rosen-Zvi et al. [15]. The category-topic model is based on the assumption that categories of the articles were chosen based on the content of the article and that local newspapers have their own unique topic preferences. We find this model structure to be generally applicable to most metadata information, assuming that the metadata influences the text of the document or that the text of the document influences the metadata. The author metadata influences the text of the documents, as each author has their own writing style and subjects they write about. The text of the documents influences the category metadata, as it is assumed that the category is chosen after the document has been written.

For our category-topic model and our author-topic model, each document d is associated with only one category c_d from the set of all categories C and only one author a_d from the set of all authors A . This is different from Rosen-Zvi et al. [15], where each document has a vector of authors. This is due to our dataset never having more than one author or category for each document. For the remainder of this paper, 'author-topic model' refers to our modified topic model, rather than

the one presented in Rosen-Zvi et al. [15]. The plate notation for our category-topic and author-topic models can be seen in Figure 2. The Gibbs sampling algorithm we have implemented and use for LDA, the author-topic model, and the category-topic model is described in Appendix Section G.

B. Pachinko Allocation

In order to handle the hierarchical structure of the taxonomy metadata, we use a hierarchical topic model, namely the Pachinko Allocation Model (PAM) from Li and McCallum [9]. Pachinko allocation generalizes LDA, making it possible to construct topic hierarchies based on any DAG structure. PAM is a topic model focusing on finding topics of different abstraction levels and modeling the connections between these topics.

Each node in the DAG structure represents a topic in the pachinko allocation model. However, unlike LDA where topics are distributions over words, in PAM topics are multinomial distributions over words and/or other topics further down the hierarchy of the DAG structure. Figure 4 illustrates an example of the DAG structure used in this paper. The idea behind the DAG structure is to be able to model correlations between topics and in turn make more coherent topics.

In this paper, we use a layered PAM, as in [9], meaning that the DAG structure is divided into layers where each layer is fully connected to the next layer. However, Li and McCallum [9] use four layers where we use five to capture more of the underlying information in the taxonomy metadata.

We construct some layers in our DAG structure based on the structure from the taxonomy metadata in our dataset, having some nodes represent a topic based on a specific taxonomy entry. An example of this can be seen in Figure 4, where we have the node "STEDER", and this is connected to "Danmark" in the third layer. To make the algorithm construct the topics to be based on our taxonomies, we introduce a novel locking mechanism for the Gibbs sampler which we use to run PAM. This mechanism is discussed further at the end of this section.

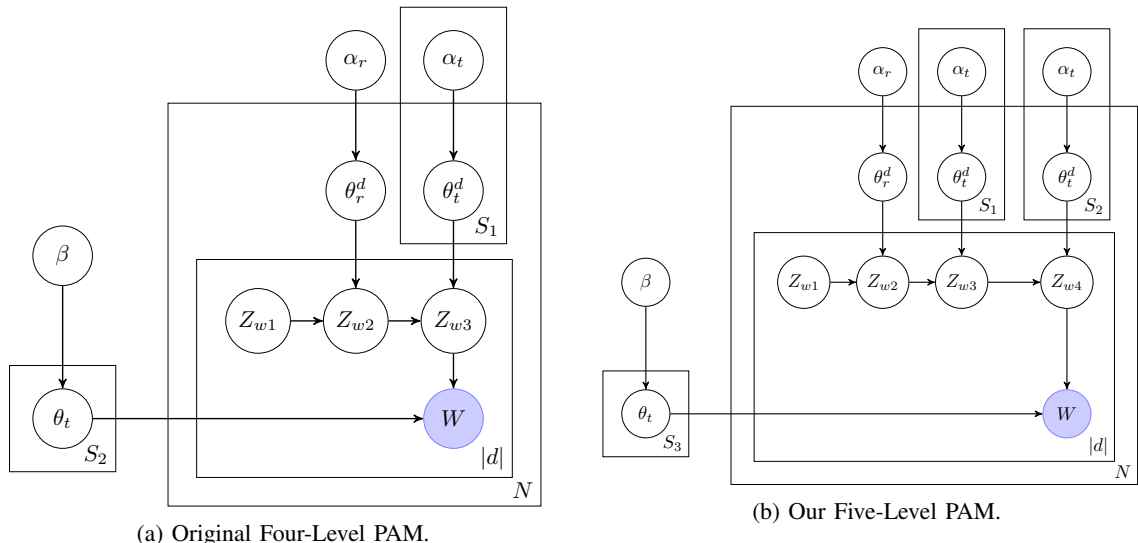


Figure 3: Plate notation for the original Four-Level PAM and our Five-Level PAM.

We use a Five-level pachinko tree structure, following the format presented by Li and McCallum [9]. The first layer is the root layer which all topics are a part of. The last layer is the word layer consisting of one node for each word in the vocabulary of our corpus.

The second and third layers is constructed based on the entries of the first two positions in our taxonomy metadata, meaning there is one node for each unique sub-taxonomy entry that is in the first or second position in the taxonomy sequence (e.g., "STEDER" and "Danmark", which is taken from "STEDER/Danmark/Aalborg", but not "Aalborg" since it is in the third position). We only use the first two layers for this, because introducing even more layers would slow down the training significantly, since the probability of all possible combinations of topic sequences needs to be sampled for every word during training. From our experiments, the training time for 50 epochs increases from 12 hours to 130 hours between Four-Level and Five-Level pachinko.

The fourth layer consists of $K = 90$ 'blank' topics, where the value of 90 comes from a grid search described in Section VI-B. This layer is added so that the model can construct topics based on the higher-level topics learned from our taxonomy metadata.

Normally when working with topic modeling, one does not know which topics are present in the document set before training the model. However, the taxonomy metadata provide some general subject names of different levels of abstraction and some amount of documents attached to these subject names. This provides a unique opportunity for using the existing taxonomy entries as higher-level topics. Without modification, PAM finds topics with the same structure as our taxonomy, but the taxonomy values would be disregarded during training since it would generate new taxonomy sequences. However, in our case, we have a dataset with a partially observed taxonomy metadata ($\sim 25\%$ of the documents), and

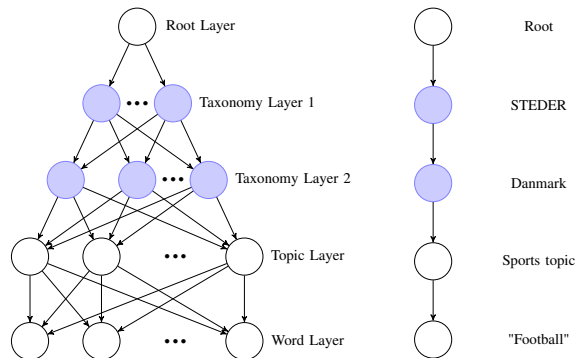


Figure 4: Illustration of the DAG structure for the Five-Level PAM, where the general structure is on the left and an example of a path from the root to a word is on the right. The two observed variables are from our taxonomy metadata and can be found in Appendix Table XII.

we want to use the existing metadata information to estimate the topics quicker and more accurately. To account for this, we only sample the unobserved nodes within the topic sequences. For some of the dataset only the fourth layer is unobserved, but for the $\sim 75\%$ documents without a taxonomy sequence, all layers are unobserved. The observed taxonomy sequences are never sampled, hence they are 'locked' in place. This creates a constant context for the taxonomy topics, which the documents with unobserved taxonomy sequences will be fitted around. It also makes the algorithm run faster, as fewer sequences need to be sampled. We go into more detail about the difference in elapsed time in Appendix Section I.

A small number of documents in the dataset also have multiple taxonomy sequences. For these documents, one of the taxonomies is chosen randomly for each word in the document. A further description of the Gibbs sampling for our PAM implementation is given in Appendix Section H.

VI. EVALUATION

In this section, we describe the evaluation metrics and how hyperparameters for the models are chosen. Lastly, an overview of the results is given.

For our evaluation, we train four models on our news article dataset from Nordjyske. The standard LDA model is used as a baseline, and our models: the author-topic model, the category-topic model, and the taxonomy-topic model, are tested. The main differences between the models are how they draw a specific topic for a word, and in the taxonomy-topic model, a hierarchical structure is used.

A. Evaluation Metrics

The main metric used in this evaluation is topic coherence[16]. This metric indicates how semantically similar the top words within each topic are and can therefore be used as an indication of the topic quality within a model [14]. There are different ways of calculating topic coherence. We are using C_v topic coherence, for this paper. The intuition is to calculate the degree of semantic similarity between highly probable words in a topic. Topic coherence ranges between zero and one. This evaluation metric is explained further in Appendix Section B.

The second metric used in the evaluation is topic difference. Topic difference is another metric that is used to check the quality of the topic model. It is based on the assumption that a good topic model has little overlap between topics. It is not the best measure of the final quality of a topic model as non-coherent topics can have little overlap. However, a low topic difference can be an indicator of potential problems within a model. Topic difference is calculated using the following equation:

$$\text{TopicDifference} = \frac{1}{K \cdot K} \sum_{i=1}^K \sum_{j=1}^K JS(\beta_i, \beta_j) \quad (1)$$

where JS is the Jensen-Shannon distance, K is the number of topics, and β_k is the topic-word distribution for topic k . Topic difference ranges between zero and one.

B. Grid Search

To find the optimal hyperparameter values for the models, we run two rounds of grid searches. To find the best-performing model we test different values for the number of topics K and the two Dirichlet priors α and η , for the document-topic/metadata-topic distributions and topic-word distribution, respectively. In the first round, the number of topics K we test are the values of K_1 , as seen in Table I, with a single randomly chosen α and η value for each K value. This creates much fewer runs of the grid search to start with and eliminates hyperparameter values that give worse models. In the second round, the number of topics K we test are the values of K_2 with all combinations of α and η except for those with values of 0.001, since these models gave much worse scores.

Table I: Tested hyperparameter values for the grid search. K_2 are the K values used for the grid search in conjunction with the bolded values in α and η .

Parameter	Tested Values
K_1	10, 20, 30, . . . , 100, 150
K_2	50, 60 70, 80, 90, 100
α	0.1 , 0.01 , 0.001
η	0.1 , 0.01 , 0.001

We only run the grid search on the standard LDA model, with the assumption that the number of topics that perform well for this model also performs well for the metadata models when the same dataset is used.¹ To evaluate the LDA models during the grid search, we measure the topic coherence of a model after training it on the dataset for 50 epochs. The hyperparameters of the model with the highest score are then used for the models in the rest of the experiment.

Based on the topic coherence of the best-performing model, we choose $K = 90$, $\alpha = 0.01$, and $\eta = 0.1$ as the hyperparameters for all models in the experiment. More details on how we chose these values are given in Appendix Section C.

C. Overview of results

From Table II, we can see that the author-topic and category-topic models are performing the worst, whereas the taxonomy-topic model is outperforming all other models. However, the elapsed time of the taxonomy-topic model is worse than the standard LDA. It takes about 6-8 hours to compute 50 epochs for the LDA model, depending on the CPU. The taxonomy-topic model running a 5 layered PAM took ~ 132 hours before completing the 50 epochs. Analysis of the topics in the trained models is given in Section VII. Extended analysis and other models are investigated in Appendix Section I, Section J, and Section K.

VII. MODEL ANALYSIS

In this section, we investigate the different models to see how each metadata affects the resulting topics. Firstly, we want to investigate the most probable topic words within each model. We have chosen an article arbitrarily from the dataset and visualized how the topics differ between the models. Before investigating the article, we define a specific color scheme for each model, which is seen in Table IV.

¹Initial testing on the category-topic model indicates that this assumption holds true.

Table II: Overview of topic coherence and topic difference results from our models. Bolded results are the highest within each column.

Topic Model	Topic Coherence	Topic Difference
Latent Dirichlet allocation	0.520	0.575
Author-topic model	0.335	0.615
Category-topic model	0.370	0.560
Taxonomy-topic model	0.660	0.709

Table III: Top 10 words of top 3 most occurring topics, within the article in Figure 5, for each model used in Table IV.






Topic	Top 10 words for LDA
1	sæby, a, direktør, frederikshavn, virksomheden, hans, medarbejdere, pedersen, firmaet, procent
2	området, boliger, natur, naturen, ligger, du, vand, dyr, a, skov
3	nielsen, arets, prisen, dansk, mors, jensen, løgstør, aars, vm, thy
Topic	Top 10 words for author-topic
1	du, procent, unge, børn, arige, hans, dansk, mig, thisted, mener
2	sine, skriver, mig, børn, seneste, land, dansk, kommuner, andersen, formand
3	set, glas, odense, vesthimmerland, leth, markedet, trump, ni, regionerne, prins
Topic	Top 10 words for category-topic
1	du, hans, børn, mig, thisted, procent, bedre, a, kr, maske
2	klar, haft, fem, hjørring, ham, nyt, formand, min, aften, sagen
3	du, min, gode, gamle, ad, henrik, eu, finde, sat, hobro
Topic	Top 10 words for taxonomy-topic
1	aab, jacob, kasper, rasmus, jakob, pedersen, andersen, friis, minut, allan
2	landbrug, landbruget, landmænd, vand, miljø, affald, bedre, natur, vandløb, fødevarer
3	virksomheden, millioner, a, direktør, procent, medarbejdere, selskabet, overskud, ansatte, virksomhed

In Figure 5, we have highlighted the highest probable words within the three most probable topics in the article. The article is about agriculture and how farmers opening their doors to the public. It also mentions a few different farms in the Northern part of Jutland and describes these in various ways.

To compare these models, we take the top 200 words of the topic-word distributions within each model and mark them in the article. We take 200 words since we want to see how intertwined the models are. Since the author-topic and category-topic models do not have a document-topic distribution we can not look at the specific document, but instead, we have marked the words from the given category- and author-topic distribution for the document's category and author to see what the difference in topics are. For the taxonomy model, we find the most probable topics within the article by inspecting the topic-word distribution and marking the words within Figure 5.

Overall, we see that there is a large amount of overlap between the models, which is interesting since the models

Table IV: Color scheme for each model.

Topic Model	Color
Latent Dirichlet allocation	
Author-topic model	
Category-topic model	
Taxonomy-topic model	
Word appearing in 3+ models	

kig på grise, køer og kyllinger¹⁰ nordjyske bedrifter åbner søndag for stalddøre landbruget åbner søndag 16. september ladeporte og stalddøre for offentligheden. 52 danske bedrifter er med i årets "åbent landbrug". i det nordjyske kan man kigge forbi på 10 forskellige landbrug. blandt de nordjyske deltagere er der mulighed for at få indsigt i både kvæg- og svinebedrifter, ligesom en producent af slagtekyllinger byder velkommen. sidstnævnte kan opleves hos rokkedahl i farstrup. de er tre familier med i alt seks børn, der sammen driver rokkedahl landbrug med slagtekyllinger og planteproduktion samt rokkedahl energi, som laver energioptimering. herudover har de eget slagteri, hvor ca. 35 af deres i alt 65 medarbejdere arbejder. familien rokkedahl har arbejdet med kyllinger siden 1963 og er tredje generation. i staldene og i de omkringliggende folde har de både fritgående og økologiske slagtekyllinger. velfærdskyllingerne går i flokke og har adgang til store folde. på årsbasis opdrætter rokkedahl otte millioner kyllinger som enten slagtes på deres eget slagteri eller sælges til eksterne slagterier. på de 1350 hektar har de hvede, byg raps, havre, rug, arter og hestebønner. det anvendes primært til foder til velfærdskyllingerne. de dyrker jorden primært økologisk og anvender halmen til opvarmning af staldene. de har varmevekslere på alle stalde for at minimere varmeforbruget og ammoniakudledningen til omgivelserne. britt og klaus kristiansen på solbakken agri ved aabybro er klar til vise en stor, dansk mælkeproduktion frem. familien tæller også de fire børn, maria på 18 år, daniel på 16 år, kamilla og laura på 13 år, og de er sjette generation på gården, som de overtog i 2013. solbakken har 600 økologiske malkekøer, som tilsammen giver 17.000 liter mælk om dagen. den bliver hentet og kørt til et af arlas mejerier, hvor den bliver anvendt til økologiske mejeriprodukter. 575 hektar land tilhører gården, og her producerer familien foder til deres dyr samt andre fødevarer. i himmerland kan man besøge sanne og ole mathiasen, der driver nørregaard på braulstrupvej 9 i suldrup. her kan man se søer, smågrise og slagtesvin i staldene og høre om produktion af velfærdsgrise, se maskinerne, få smagsprøver fra danish crown og på lokale fødevarer, og høre om biavl for børnene er der leg i korncontainer og halm, pedaltraktorbane og ponytrækketure. der er kaffe og kagebord. åbent landbrug foregår søndag fra klokken 10 til 16. det er gratis at deltage. sidste år deltog 96.000 danskere i åbent landbrug.

Figure 5: An article chosen arbitrarily from our dataset where words within top 200 of the top 3 topics within each model are highlighted.

use different metadata information to create the various topic distributions. This indicates that the models share many of the top words, while also indicating a slight deviation between the models due to the metadata information. The LDA model and the taxonomy model show words like "landbrug" (agriculture) and "produktion" (production), and "hektar" (acre) which is what the article is mostly about. Author-topic specific words are not very present and are only showing three unique words: "byder", "hører", and "børnene". This indicates that the author-topic model has trouble generalizing what the author of this article (Peter Tordrup Larsen) is writing about. This might be because he has written 5002 articles in our dataset and generalizing that many articles is a challenge. Another aspect of the author-topic model is that the authors writing these

Table V: Top 10 author pairs based on the symmetric KL divergence between authors.

Author pair	KL
Lars Termansen (328) & Mikkel Færgemann Viken (91)	1.50
Morten Nis Klenø (17) & Anne Helene Thomsen (606)	1.72
Lars Termansen (328) & Lars Christensen (1293)	2.43
Esben Heine Pedersen (1689) & Caspar Birk (71)	2.47
Lars Christensen (1293) & Poul Christoffersen (65)	2.53
Lone Beck (92) & Max Melgaard (587)	2.74
HANNE Lindblad Jensen (27) & Peter Tordrup Larsen (5002)	2.94
Søren Kjær (95) & Carl Chr. Madsen (785)	2.98
Heidi Majgaard B. Pedersen (244) & Lisbeth Helleskov (361)	3.05
Lars Termansen (328) & Morten Lind (413)	3.16
Maximum	34.51
Median	24.20

articles most likely do not write about just one subject, which explains why there are only three less important words marked here. The category-topic model only shows three unique words: "klaus", "kamilla", and "leg" (play). These words are also very abstract and can be used in many different scenarios.

An interesting part of this analysis is the words appearing in three or more models. Some notable words within this category are: "medarbejder" (co-worker), "arbejdet" (worked), "jorden" (earth), "land", "nordjyske" (North Jutland), and "dyr" (animals). These words are fairly representative of the content of the article.

A general pattern which can be seen in Figure 5 is that the LDA model and the taxonomy-topic model are marking many of the same words, where the taxonomy-topic model only occurs together with other models. This makes sense because these two models have the best topic coherence in our results.

There is also the possibility that choosing another random article would give completely different numbers of marked words per model because this highly depends on the article's author and category. In Appendix Section D, we investigate how the coloring applies to two other articles.

A. Author-topic model

Interesting observations can also be made specifically for the author-topic model. The similarity of authors is a good example, which can be measured by taking the distance between their topic distributions. In this model, the author-topic distribution defines the probabilities of topics being written by a specific author. Then, just as Rosen-Zvi et al. [15], the similarity of authors can be found by calculating the symmetric Kullback-Leibler divergence:

$$sKL(i, j) = \sum_{t=1}^T \left[\theta_{it} \log \frac{\theta_{it}}{\theta_{jt}} + \theta_{jt} \log \frac{\theta_{jt}}{\theta_{it}} \right] \quad (2)$$

where θ is the posterior distribution of the model, and θ_{it} is the probability of author i having written about topic t , and the same for θ_{jt} with author j .

In the context of using these similarities for recommendation, knowing how similar authors are, gives the opportunity to recommend new authors to readers, while the articles are about

Table VI: Top 10 category pairs based on the symmetric KL divergence between categories.

Category pair	KL
misc (292) & Friii (2333)	3.65
Friii (2333) & Debat (10075)	3.80
Feature (188) & Hjørring-avis (4235)	4.22
Sport-avis (10941) & Morsø Sport (2350)	5.04
Indsigt (984) & Perspektiv (613)	5.27
53. Frederik (203) & Navne (3749)	5.69
Rebild-avis (4415) & Bo Godt (1447)	5.78
Nordjyske Biler (1400) & Thisted-avis (11473)	5.91
misc (292) & Debat (10075)	6.67
Frederikshavn-avis (4325) & Bo Godt (1447)	6.81
Maximum KL divergence	35.62
Median KL divergence	27.09

similar topics. In Table V, the top 10 author pairs, based on this similarity measure, are shown. A smaller KL value means the authors are more similar. The number in parenthesis next to each author is the number of articles they have written in our dataset.

In general, for these pairs, there does not seem to be a correlation between a high similarity and the categories of the articles they have written. While one author in a pair might have mainly written for the sports category (Sport-avis) the other author might not have written for this category at all. This can also be seen for categories that cover geographic locations, where one author might have written for Aalborg (Aalborg-avis) and the other author can have written for Thisted (Thisted-avis).

From sample documents written by the most similar author pair (Lars Termansen & Mikkel Færgemann Viken), we find that both authors write a mix of regular news and sports articles. Their high similarity could be due to the ratio between news and sports news for both authors being similar, and possibly also because of the types of news they write about. Another interesting observation is that for the second

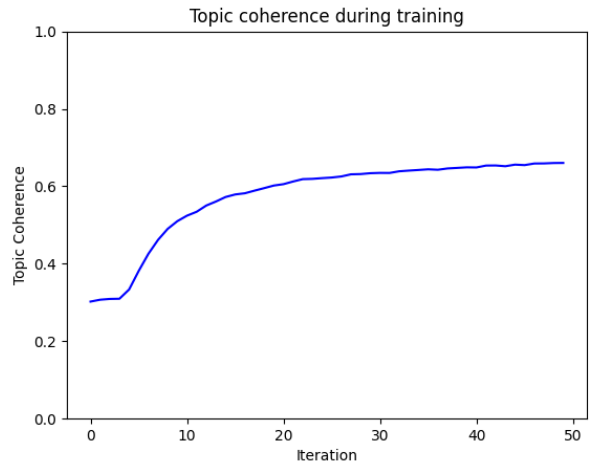


Figure 6: Topic coherence during training of the taxonomy-topic model.

Table VII: A selection of authors and categories and the top 10 words from their most probable topic.

Author	Topic ID	Top 10 words
Birgitte Bové	41	millioner, eu, hans, større, bedre, formand, kr, nordjyske, taget, skriver
Kirsten Østergaard	50	du, thisted, unge, mig, børn, procent, hans, hver, penge, hjørring
Pauline Bülow	3	procent, bag, rigtig, lave, dansk, formand, gode, klar, svært, plads
Karen Marie Foldbjerg	13	du, sine, formand, seneste, jensen, hvert, nyt, hvordan, finde, kommunen
Claus T. Kræmmergård	88	du, procent, unge børn, arige, hans, dansk, mig, thisted, mener
Hanne Lindblad Jensen	2	du, thisted, procent, mig, børn, hans, unge, dansk, mener, a
Ole Jensen	50	du, thisted, unge, mig, børn, procent, hans, hver, penge, hjørring
Category	Topic ID	Top 10 words
Frieord	37	du, thisted, mig, hans, procent, børn, a, kr, arige, unge
Bo Godt	7	du, børn, mig, hans, unge, procent, mener, politiet, hvordan, thisted
WEEKEND	31	du, hans, børn, mig, thisted, procent, bedre, a, kr, maske
Mariagerfjord-avis	39	du, thisted, dansk, unge, mig, børn, a, hans, procent, arbejde
Aalborg-avis	63	børn, hver, thy, rigtig, millioner, synes, mennesker, ham, mand, dansk
Navne	14	hver, haft, bedre, ham, thy, mener, hans, nordjylland, plads
Sport-avis	39	du, thisted, dansk, unge, mig, børn, a, hans, procent, arbejde

most similar author pair (Morten Nis Klenø & Anne Helene Thomsen) the difference in the number of articles written is significant. Here Morten Nis Klenø has written just 17 articles while Anne Helene Thomsen has written 606 articles. This suggests that some part of why these authors' similarity is high, simply depends on the types of news the authors have written, no matter the amount.

It is also worth noting that while authors that write scientific papers usually write in just a few subject areas, the scientific area they work in, this is not necessarily the case for news article authors. In our dataset, this can be seen in the fact that the authors have written for 7.86 categories on average, with 7 categories as the median. This can make it more difficult for the author-topic model to find patterns in what the authors write about, especially since each category can cover multiple topics.

A random selection of authors from the dataset and the top words from their most probable topic can be seen in Table VII.

B. Category-topic model

Specific observations for the category-topic model can also be made. As with the author-topic model, the similarity between pairs of categories can be calculated. Because topic distributions are generated for each category, category similarity can also be calculated using Equation 2 where i and j are categories instead of authors. In Table VI, the top 10 category pairs, based on symmetric KL divergence, are shown.

The second most similar pair, 'Friii' and 'Debat', is interesting to look at since 'Friii' does not seem to have a theme in the articles written, articles with the 'Debat' (debate) category seem to mostly cover themes that can bring differing opinions and articles with interviews. This indicates that the model does not find these deeper thematic differences in articles or that it finds other patterns that are difficult to see.

It is also interesting that the 'misc' category is seen twice in the top 10 ranking even though it is made up of many smaller categories with no connection to each other. Though, it is not surprising that this thematically mixed category is quite

similar to 'Friii' and 'Debat', which are more thematically wide categories.

It is also clear that some of the topics that the model has learned fit well with how some categories are used in the dataset. For example, the 4th ranking pair 'Sport-avis' and 'Morsø Sport' are clearly correlated by their category names covering sports news and the similarity of the topic distributions learned for both categories indicates that the model has learned these sports topics correctly.

Finally, it is worth noting that there are no category pairs, where both categories are based on geographic locations, in the top 10 pairs. This may indicate that each city or municipality in Denmark does have some differences in which topics are written about in general.

A random selection of categories from the dataset and the top words from their most probable topic can be seen in Table VII.

This knowledge about categories can support recommendation in multiple ways. An example is, that while news sites often have the possibility to filter articles based on categories, knowing which categories are similar gives further opportunities for recommending related or similar articles.

C. Taxonomy-topic model

Finally, we also want to analyze the taxonomy-topic model, especially since this model has the highest topic coherence out of all the tested models. Table VIII shows selected lowest level topics from the the taxonomy-topic model. As can be seen from the figure most of the topics are quite understandable. However, there are some topics (e.g., topics 19 and 42) that consist entirely of words that provide little context or semantic meaning. This indicates that the model has learned to group words that do not belong to any good topics. This is a good feature that allows the model to apply an extra layer of preprocessing, automatically filtering away irrelevant words into topics. This feature is also seen in some other topic models, such as in the hierarchical LDA (hLDA) by Blei et al. [2] and the embedded topic model (ETM) by Dieng et al. [8], but the LDA does not seem to have this feature.

Table VIII: Top 10 words of selected topics from our taxonomy-topic model. Labels have been manually added to the topics to increase readability.

Topic	Label	Top 10 words
8	politics	venstre, valg, valget, partiet, partier, parti, stemmer, mette, politik, regering
9	money	procent, viser, tal, antallet, milliarder, pct, seneste, penge, millioner, indland
19	filler	mig, maske, du, folk, synes, ting, faktisk, nogen, altid, tror
21	university	unge, uddannelse, studerende, gymnasium, elever, uddannelser, universitet, procent, uddannelsen, nordjylland
41	academic research	universitet, professor, forskere, forskning, forskerne, viser, verden, institut, procent, aarhus
42	filler	mig, min, mit, ham, aldrig, gik, lille, maske, mine, altid
45	wildlife	dyr, naturen, natur, ulve, fugle, ulven, arter, dyrene, vilde, ulv
47	church	kirke, kirken, sognepræst, præst, søndag, koret, gudstjeneste, aften, kor, organist
59	music concerts	musik, koncert, sange, spiller, koncerter, band, koncerten, festival, musikken, publikum
60	buisness	virksomheden, millioner, a, direktør, procent, medarbejdere, selskabet, overskud, ansatte, virksomhed
69	primary school	elever, unge, skole, eleverne, skolen, skoler, klasse, børn, folkeskolen, lærere
74	elder care	ældre, borgere, kommunen, millioner, penge, nordjylland, plejehjem, borgerne, kommunens, budget
75	filler	du, din, dig, dit, altsa, dine, maske, nemlig, bruge, hvordan
79	filler	mig, min, hendes, hende, rigtig, arige, altid, arbejde, mor, mine
86	sports	handbold, mors, thy, hold, kamp, sæson, kampe, kampen, point, holdet

Table IX: IDs of the 5 most occurring fourth layer topics for each third layer topic from the taxonomy-topic model. See Table VIII and Appendix Table XVI for the most occurring words for each ID.

Taxonomy Name	Top 5 Topic IDs	Taxonomy Name	Top 5 Topic IDs	Taxonomy Name	Top 5 Topic IDs
Danmark	8, 42, 82, 59, 79	Udland	42, 79, 59, 8, 32	Kultur	9, 42, 79, 19, 8
Landbrug	42, 79, 8, 9, 19	Kriminalitet	42, 75, 60, 8, 86	Socialstof	42, 9, 79, 86, 8
Arbejdsmarked	42, 79, 59, 8, 9	Økonomi	79, 75, 74, 42, 9	Sundhed	8, 32, 42, 9, 19
Politik	42, 75, 9, 19, 74	Musik	75, 42, 59, 11, 79	Sport	42, 75, 8, 59, 52
Bolig	75, 42, 86, 79, 8	Videnskab	42, 8, 52, 79, 19	Trafik	42, 74, 8, 52, 32
Erhverv	42, 8, 59, 32, 79	Uddannelse	42, 9, 75, 32, 74	Energi	42, 8, 79, 19, 86
Ulykker	42, 75, 9, 79, 32	Fritid	42, 8, 75, 82, 79	Socialt	42, 75, 79, 59, 9
Dyr	86, 42, 79, 52, 9	Natur	42, 52, 9, 32, 79	Miljø	8, 42, 75, 52, 59
Familie	79, 8, 42, 59, 32	Politi	42, 75, 79, 8, 59	Byggeri	75, 42, 79, 77, 59
Etik	79, 42, 8, 86, 74	Religion	42, 79, 8, 59, 32	Kommunalvalg	42, 8, 75, 79, 32
Nordjyske Plus	42, 86, 9, 79, 74	DF	42, 8, 59, 52, 19		

Since this model deals with more topic distributions than the other models, it is worth checking whether it also converges within the first 50 epochs, as with LDA. This does seem to be the case, as indicated by Figure 6. Here it can be seen that the topic coherence curve has flattened significantly, and thus additional epochs would yield diminishing returns.

Table IX gives an overview of how the taxonomy topics in the third layer of the taxonomy-topic model, are connected to the fourth layer topics that were generated by the model. Some of these connections make a lot of sense, such as the 'Økonomi' (Economy) taxonomy entry topic which has the three filler topics (i.e., topics which consists entirely of words with little semantic value) within the top 5 most probable topics: 79, 75, and 42, as well as two topics which are about money: 74 and 9. Table VIII shows the top words for each of these topics. However not all the connections between higher and lower level topics are as understandable as these. For example, the 'Kriminalitet' (Crime) taxonomy entry has two filler topics within the 5 most probable topics: 42 and 75, one topic about economy: 60, one topic about politics: 8, and one topic about sports: 86. Table VIII shows the top words for the topics mentioned in this section, and Appendix Table XVI shows top words for all topics in the taxonomy-topic model.

Having the layered structure of the PAM gives many possibilities for recommending new articles to readers. There is the possibility of exploring the similarity of taxonomies at the same layer and using this to recommend new articles with

similar subjects. For example, if an article is about 'Miljø' (environment) similar taxonomies might be 'Natur' (nature) and possibly 'Etik' (ethics), 'Trafik' (traffic), and 'Energi' (energy).

VIII. CONCLUSION

Utilizing information to its full potential can be a complicated task, especially within the field of topic modeling. Multiple approaches have been proposed to improve the incorporation of metadata [17][15], but tailoring the models to benefit from the existing metadata can be a difficult task. In this paper, we have explored the possibilities for incorporating news article metadata from Nordjyske into existing topic models, such as LDA and PAM. We have constructed models based on three different types of metadata: author, category, and taxonomy, each of which represents the data in a different way. We evaluated our models based on topic coherence and found that the taxonomy-topic model was the best-performing model for our dataset.

From the topic coherence results, shown in Table II, the Pachinko Allocation Model (PAM) using the taxonomy metadata gets the best results. The author-topic and category-topic models based on LDA are the worst-performing, where the topic coherence is much lower than the other models. This can be due to the authors within journalism covering a broader range of topics than within scientific literature, which can negatively impact the topic coherence.

We want to answer the research questions we stated in Section I:

- *How can we establish models that incorporate metadata from the Nordjyske dataset?*
- *How does including metadata within such models impact the resulting topics?*

We incorporate metadata into our topic models in multiple ways. The LDA model is used as a baseline, which indicates whether incorporating metadata can improve the topic quality of our topic models. Based on the results of the author-topic and category-topic models, we see that only using the metadata for word topic assignment within LDA can hurt the topic quality. Other studies have shown that only using the metadata for word topic assignments can improve performance [17] [15]. This might be due to the particularities of our dataset, such as authors not usually writing about the same subject within the news environment.

We use the PAM to incorporate a hierarchically structured taxonomy metadata, where we use a novel locking mechanism to lock the observed metadata's topics into place. Using this model and technique, we can get better topic quality within PAM compared to the LDA model. However, the elapsed time of the algorithm is quite slow compared to the LDA.

Topic modeling can be used to support recommendation in different ways. We can use the topic distributions from our models to compare articles based on their similarity in topics. For example, the author-topic model's topic distribution can be used to recommend similar authors, while for the topic distributions of the taxonomy-topic model, there is the possibility of looking at topics from different layers. This information can be used in a content-based filtering approach to recommend similar articles. Due to the promising results provided by the modified PAM, investigating this model further might be beneficial to incorporating metadata into topic models. However, testing these models on multiple datasets needs to be accounted for since the generalizability of these models is not explored within this paper. Using word embedding to further improve the performance of models can also be viewed as a possible next step for this project since a wide number of papers are using this technique to improve topic modeling[17][8].

It would also be interesting to see how we could incorporate these models into an existing IT infrastructure for a news site such as Nordjyske. The next step in that process would be to investigate which part of their infrastructure could benefit from the use of topic modeling, whether it is for recommendation or automatic tagging of articles. We have written about a few more potential use cases of our project in Appendix Section L.

REFERENCES

- [1] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletos, Rohit Singh, Paul A. Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20:28:1–28:6, 2019. URL <http://jmlr.org/papers/v20/18-403.html>.
- [2] David Blei, Thomas Griffiths, Michael Jordan, and Joshua Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in Neural Information Processing Systems*, 16, 05 2004.
- [3] David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133826. URL <https://doi.org/10.1145/2133806.2133826>.
- [4] David M Blei and John D Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10 (71):34, 2009.
- [5] David M. Blei and Jon D. McAuliffe. Supervised topic models. In *Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS'07*, page 121–128, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.
- [6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
- [7] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, 2009.
- [8] Adji B Dieng, Francisco JR Ruiz, and David M Blei. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453, 2020.
- [9] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577–584, 2006.
- [10] David M Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*, volume 24, pages 411–418. Citeseer, 2008.
- [11] Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313, 2015. doi: 10.1162/tacl_a_00140. URL <https://www.aclweb.org/anthology/Q15-1022>.
- [12] James Petterson, Wray Buntine, Shравan Narayana-murthy, Tibério Caetano, and Alex Smola. Word features for latent dirichlet allocation. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL <https://proceedings.neurips.cc/paper/2010/file/db85e2590b6109813dafa101ceb2faeb-Paper.pdf>.
- [13] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256, Singapore, August 2009. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/N09-1022>.

org/anthology/D09-1026.

- [14] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, page 399–408, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450333177. doi: 10.1145/2684822.2685324. URL <https://doi.org/10.1145/2684822.2685324>.
- [15] Michal Rosen-Zvi, Thomas L. Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. *CoRR*, abs/1207.4169, 2012. URL <http://arxiv.org/abs/1207.4169>.
- [16] S. Syed and M. Spruit. Full-text or abstract? examining topic coherence scores using latent dirichlet allocation. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 165–174, 2017. doi: 10.1109/DSAA.2017.61.
- [17] H. Zhao, L. Du, W. Buntine, and G. Liu. Metalda: A topic model that efficiently incorporates meta information. In *2017 IEEE International Conference on Data Mining (ICDM)*, WSDM '15, pages 635–644, 2017. doi: 10.1109/ICDM.2017.73.

APPENDIX

In the main paper, we initialize and describe our problem with a focus on results and analysis. Within this appendix, we are expanding on many aspects of the aforementioned paper and new extensions to the models. Following is an overview over each section and what it investigates.

A) Metadata labels

- The metadata is shown in tables for all three metadata types, and observations about the metadata labels are described.

B) Topic coherence

- We explain the purpose and mathematical ideas behind the evaluation metrics we use in the paper.

C) Grid search

- Our grid search process is described further on how we chose our hyperparameters.

D) Coloring articles

- Two more articles are highlighted the same way as in Section VII and are analyzed.

E) Stemming the dataset

- An experiment using a stemmed dataset is described, and the results are shown.

F) Pyro model implementation

- We describe the probabilistic programming language Pyro which we explored before choosing to work with Gibbs sampling.

G) Gibbs sampling

- The code for the Gibbs sampler is explained and investigated. A parallel Gibbs sampling method is also mentioned.

H) Pachinko implementation

- The implementation of our Pachinko Allocation Model (PAM) model and how it differs from the Gibbs sampling method is explained.

I) Category and author PAM

- We go into detail about how we are using author and category metadata in the PAM and what results were achieved.

J) The author-category model

- We also combine the author-topic and category-topic models into one model with two topic distributions. This model is also analyzed.

K) The author-doc and category-doc models

- We create two new models called the author-doc and category-doc model. These models are a combination of the standard latent Dirichlet allocation (LDA) and the author and category metadata.

L) Applications of our models

- We explore the application possibilities of our project and propose various applications which could be implemented at news sites such as Nordjyske.

A. Metadata labels

In this section, the different types of metadata used in our evaluations are explored. Here the focus is on the observations related to the labels of the metadata.

1) *Category*: There are 58 different category labels present within the dataset. Categories with less than 140 documents ($\sim 0.1\%$ of the number of documents) are removed as part of preprocessing and replaced with a single miscellaneous category 'misc'. This reduces the number of categories to 34, while only replacing 292 documents to have the 'misc' category. This preprocessing also makes the size of the remaining categories more evenly distributed, as can be seen in Figure 7. The smallest category consists of 188 documents, and for all categories, the median category size is 3022. Figure 8 shows an overview of the size of the categories after filtering. The categories from the 3rd quartile and up do become much larger compared to the median, and there is an outlier category with 20,241 documents. These larger categories seem to be about topics that are written about often or categories that cover a specific newspaper.

As mentioned in Section II, some category labels seemingly have no relevance to the documents using the labels. Specifically, the category labels '26. Frederik' and '53. Frederik' do not indicate what they cover, since Frederik is simply a normal Danish name. Since they are not filtered into the 'misc' category during preprocessing, it is worth looking into what documents have these categories.

By looking at a random selection of documents from these categories, there is no clear pattern to be found. The topics, within these categories, can be about anything from sports to news from anywhere around the world or locally in Denmark. The articles from '26. Frederik' appear all years in our dataset, while the articles for '53. Frederik' seem to be just from 2019,

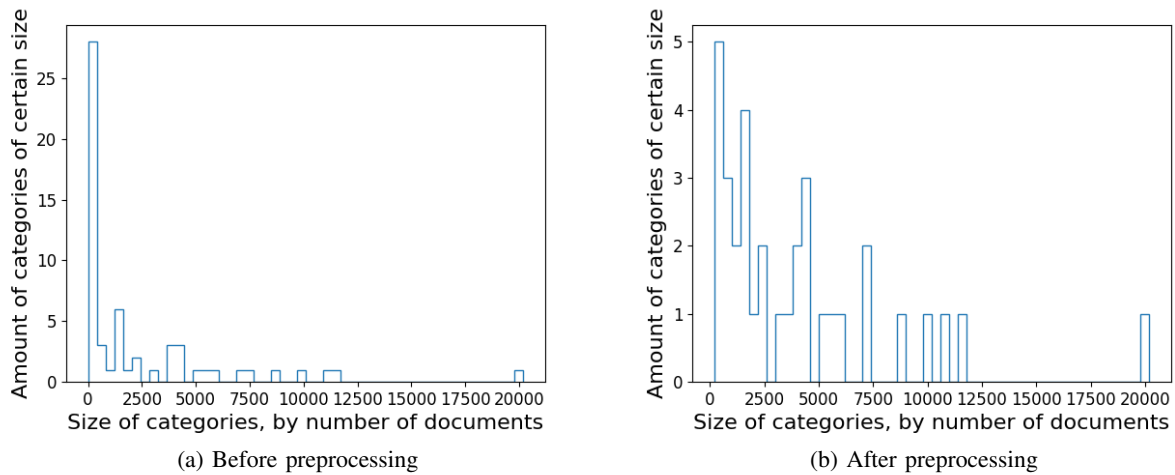


Figure 7: Histogram over the number of categories for different number of documents, before and after preprocessing. Categories on the x-axis are grouped into 50 columns.

the last year in our dataset, but evenly distributed over the whole year. Curiously enough, most of the documents from these categories seem to be written by Anders Kjærgaard, with only a few other authors. For '26. Frederik' there are just 5 unique authors and for '53. Frederik' there are just 4 unique authors, where Morten Kyndby Holm, Jens Fogh-Andersen, and Anders Kjærgaard have written for both categories.

From the exploration of these two categories, we can not say with certainty why these categories exist, or why multiple authors have chosen to write for these categories. We continue to use these categories in our experiment, for the possibility of making other observations through the topic models.

2) *Author*: There are a total of 227 authors within the dataset where each author on average have written 757 articles from 2017 to 2019. An interesting fact is that the median is 323 which is much lower than the average which is visible in Figure 10b. This shows that while most authors have written just a few hundred articles, there is a small number of authors that have written thousands of articles, increasing the average. The minimum number of articles that have been written by an author is 1 and the maximum number is 9906. When we look at the top two most writing authors, we get these:

- Ove Nørhave (9893 articles)
 - A well-known journalist, who has been at Nordjyske for over 25 years.
- System Administrator (9038 articles)

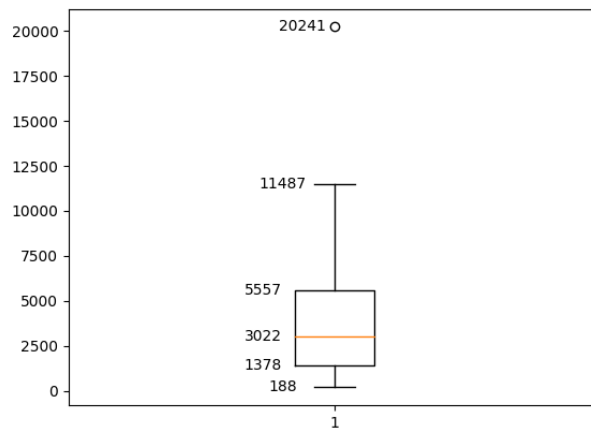


Figure 8: Boxplot over the size of the categories after filtering. Size meaning the amount of documents with the same category.

Table X: Number of documents for each of the 58 categories within the Nordjyske dataset. The bolded categories are combined during preprocessing.

Category	Number	Category	Number	Category	Number	Category	Number
Fælles	20204	Navne	3749	Kram	244	Østvendssyssel Avis	4
Thisted-avis	11473	Kultur	3012	53. Frederik	203	DF Motor Biler	3
Sport-avis	10941	Morsø Sport	2350	Feature	188	Nyhedsmotoren-net	3
Debat	10075	Friii	2333	Aalborg:nu	73	Plus Publicering	3
Udland-avis	8855	Bagside	1933	Erhvervsnavne	39	RB	3
Erhverv-avis	7356	MitLiv	1519	Newspack	35	Sport-net	3
Mariagerfjord-avis	7241	WEEKEND	1493	DF Søfart	32	Thisted-net	3
Morsø-avis	5959	Bo Godt	1447	Morsø Ugeavis	27	Hanbo-bladet	2
Aalborg-avis	5544	Nordjyske Biler	1400	DF Licitation Byggeri	14	Brugermappe	1
Vesthimmerland-avis	5131	Morsø Debat	1375	Biler	13	Brønderslev-net	1
Rebild-avis	4415	Frieord	1341	Samfund	9	Lokalavisen	1
Frederikshavn-avis	4325	Indsigt	984	Nordjyske Plus	6	Mariagerfjord-net	1
Hjørring-avis	4235	Thisted sport	698	Oplandsavisen	6	Morsø-net	1
Brønderslev-avis	3857	Perspektiv	613	INFOMAKER PRINT	5		
Jammerbugt-avis	3791	26. Frederik	484	DF Licitation Diverse	4		

– We are not sure why this has been used. It could be a placeholder.

In Figure 9a, we see that the vast majority of authors have written under 2000 articles within the three years. All authors and the number of articles they have written can be seen in Table XI.

Unlike with the category metadata, the author metadata does not have a natural threshold, with authors that have values in a specific lower range, followed by more evenly distributed values. Instead, the vast majority of the values fall in a lower range, meaning that setting the threshold too high would result in removing a large portion of the data. We instead choose a lower threshold, keeping most of the authors, except the ones that contained so few documents, that finding common topics would be inefficient. Authors, who have written less than 14 documents ($\sim 0.01\%$ of the number of documents), are removed as part of preprocessing. This removes 43 out of 227 authors, combining them into a single 'misc' author. A total of 204 documents are assigned to the 'misc' author.

Other than the 'System Administrator' author, there are a few other authors that seem different since they do not have the name of a person. These are: 'SAXoTECH Systembruger' with 1 article, 'JSLbruger Nordjyske', 'testbruger', and 'Danske Fagmedier master' with 2 articles each, and 'AAArtikler parkeret' with 51 articles. After preprocessing, only 'AAArtikler parkeret' is not put into the 'misc' author, since it has more than 13 articles. From their names, they seem to be test users or authors that cover specific articles. From looking at some articles these authors have written this seems to be the case since there are no obvious patterns in the articles written. The name 'AAArtikler parkeret' also indicates that these articles, at least at some point, were unfinished and put under this author's name.

These articles are kept in our dataset, even though we can not be certain who wrote them.

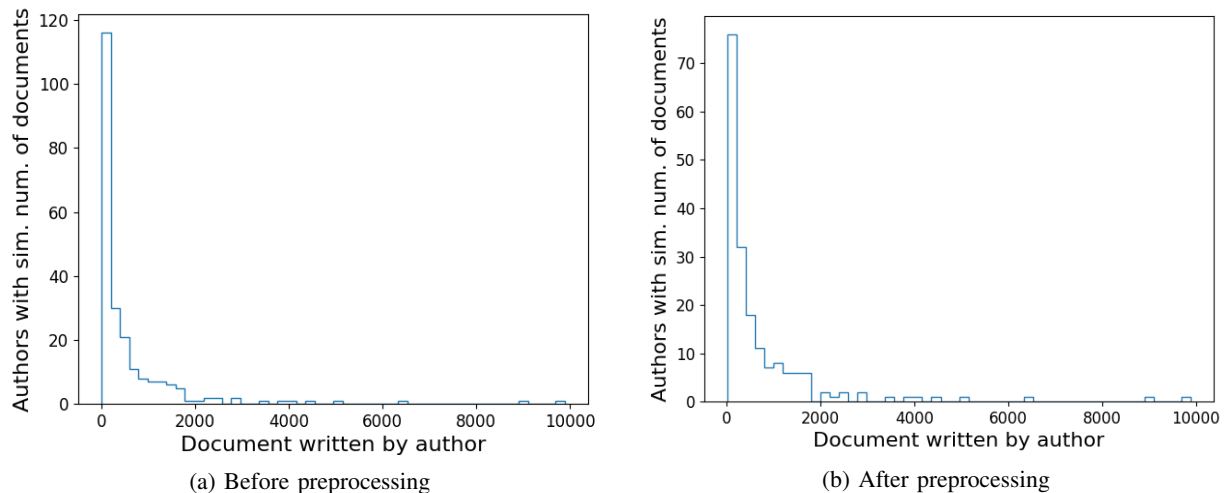
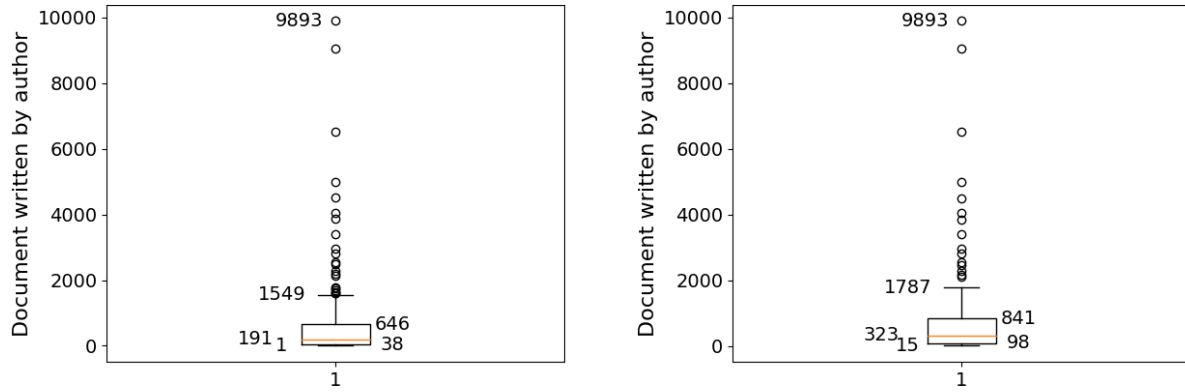


Figure 9: Histogram over amount authors who have written certain number of documents, before and after preprocessing. Authors on the x-axis are grouped into 50 columns.



(a) Before preprocessing

(b) After preprocessing

Figure 10: Boxplot over the amount of documents written by each author, before and after preprocessing.

3) *Taxonomy*: The taxonomy metadata is fundamentally different from the other metadata types. It is not fully observed with only roughly 25% of documents having a taxonomy sequence. It is hierarchical, with each taxonomy containing a sequence of taxonomy entries, such as: 'STEDER/Danmark/Nordjylland/Aalborg'. Taxonomy sequences can have varying length, ranging from 1 to 5. It is also possible for each document to have multiple taxonomy sequences, such as having both "STEDER" and "EMNER" in two separate taxonomies. Like with authors, we remove any taxonomy entries that are used in less than 14 taxonomy sequences ($\sim 0.01\%$ of the number of documents). Out of 1135 taxonomy entries, 779 are removed during this preprocessing, with 355 remaining. A subset of the taxonomy entries and the number of articles they appear in can be seen in Table XII. The taxonomy sequences for all the documents in the dataset can be combined into a tree of taxonomy entries. This taxonomy tree has the following layer sizes: 4, 32, 80, 99, and 290. The layers of size 4 and 32 are used in the second and third layer of the Directed Acyclic Graph (DAG) structure for the taxonomy-topic model.

For the labels in Table XII, the counts show the labels' sizes. It is worth mentioning that, while the counts of the labels are directly linked to the number of documents in the whole dataset, they are also connected to the size of the previous layer's taxonomy entry. For example, the label 'Danmark' is in the layer below the 'STEDER' label, where the 26145 documents of 'Danmark' are a subset of the 29535 documents of 'STEDER'.

There is also a much larger number of 'STEDER' (places) documents compared to 'EMNER' (subjects) documents, respectively 29535 compared to 5449. In Table XII, the first label on a lower layer that comes from 'EMNER' is 'Sport' with 408 documents. This indicates that the subject taxonomies, in general, are much smaller in size. The PAM is mostly influenced by the location information compared to actual topical labels.

The labels removed during preprocessing appear, as expected, to be places or subjects that have been written about just once or a few times. We could have combined them into a 'misc' taxonomy as we did for the category and author metadata, but decided against it because it would just be a large collection of mostly random documents. The taxonomy sequences that had some taxonomy entries removed, also still contain other taxonomy entries so they are not entirely without information as with the 'misc' author and category.

Statistics over the three metadata types, before and after preprocessing, can be seen in Table XIII.

B. Topic coherence

The equations, explanations, and values of the hyperparameters in this section are based on Syed and Spruit [16] and the *gensim* python package². Calculating topic coherence requires the following steps:

- A) Topic-word segmentation into word set pairs
- B) Word and word pair probability calculation
- C) Word set confirmation measure
- D) Aggregation of confirmation measures

For segmentation, a set of word pairs S is created, which pairs each word in the top-N most probable words W in a specific topic t with all other words in W . S is defined by Equation 3.

²<https://radimrehurek.com/gensim/>

Table XI: Number of documents written by each author in the Nordjyske dataset. The bolded authors are combined during preprocessing.

Author	Number	Author	Number	Author	Number	Author	Number
Ove Nørhave	9893	Bent Stenbakken	646	Katrine Schousboe	189	Sarah Sandhøj	35
System Administrator	9038	Lars Hofmeister	628	Mathias Majlund Laursen	178	Suzanne Tram	34
Ole Fink Mejlgaard	6518	Anne Helene Thomsen	606	Anne Brik Jensen	177	Sebastian Engelberth Hansen	33
Peter Tordrup Larsen	5002	Max Melgaard	587	Peder Pedersen	166	Anna Østergaard Bjørn	29
Kim Juhl Andersen	4506	Karen Marie Foldbjerg	580	Carl Åge Østergaard	152	Michael Sand Andersen	27
Jeppe Damsgaard	4057	Lise Larsen	575	Mette Siggaard	150	HANNE Lindblad Jensen	27
Jørn Larsen	3863	Asbjørn Hansen	566	Karen Keinicke	150	Mathilde Juul Back Jensen	25
Jørn Eriksen	3395	Peter Witten	544	Tune Kristensen	149	Allan Bauer	19
Anders Kjærgaard	2960	Allan Vinding Sørensen	534	Morten Brændstrup	146	Linse Daugaard	18
Søren Beukel Bak	2811	Dorrit Gap Jensen	530	Emil Halkier	143	Morten Nis Klenø	17
Søren Olsson	2558	Hans Christensen	500	Britt Kristensen	135	OLE SANVIG KNUDSEN	16
Jens Peter Svarrer	2480	Lars Høj	493	KAREN Marie Foldbjerg	132	Frederik Siiger	15
Flemming Kristensen	2282	Jesper Ramsing	469	Claus Smidstrup	128	Kim Lesanner	15
Thomas Jasper	2186	Jens Hukier	464	Sarah Thun Madsen	127	Pia Haugaard	13
Bente Lembo	2128	Svend Ole Jensen	447	Jakob Kanne Bjerregaard	126	MERETE HORN	12
Helle Møller Larsen	1787	Martin Frandsen	437	Lars Teilmann	122	Regitze Ørnstrup Christensen	12
Claus Jensen	1739	Tobias Brandt	423	Natasha Jahanshahi	117	Inge Steen Sørensen	11
Esben Heine Pedersen	1689	Andrea Jessen Jakobsen	423	Jens Ole Pedersen	116	Anika Thorø Møller	11
Margit Sig	1632	Carsten Søgaard Jensen	420	Pauline Bülow	116	Tim Søgaard	11
Jens Fogh-Andersen	1614	Morten Lind	413	Christoffer Green Sørensen	115	Flemming Haslund	10
Ole Jensen	1611	Esben Agerlin Olsen	406	Flemming Junker	103	Henrik Nordstrøm Mortensen	10
David Højmark	1549	Carsten Hougaard	406	Niklas Grønberg	103	Emil Halkær	9
Niels Hansen	1512	Dorit Glintborg	405	Sofie Møller	99	Katrine Hjulmann Nielsen	9
Line Lykkegaard Skou	1504	Karin Pedersen	397	Michael Strandfelt	98	John Jensen	8
Lars Bang Bertelsen	1443	Kasper Ørkild	393	Anders Sønderup	95	Jacob Eggert Kabel	8
Villy Dall	1408	Marianne Isen	387	Søren Kjær	95	Søren Dietrichsen	7
Hans Peter Kragh	1403	Jakob Gammelgaard	385	Lone Beck	92	Nicolai Østergaard	6
Claus T. Kræmmergård	1354	Dorte Geertsen	383	Torben O. Andersen	91	Søren L. Hviid	6
Lars Christensen	1293	Torben Duch Holm	364	Mikkel Færgemann Viken	91	Kathrine Lykkegaard Jeppesen	5
Rasmus Skovbo	1253	Henrik Strømgaaard	362	Hans Henrik Rasmussen	90	Ursula Rechnagel Taylor	5
Anders Abildgaard	1229	Lisbeth Helleskov	361	Steffen Bek	89	Jens Otto Barsøe	4
Lasse Damsgaard	1209	Niels Brauer	358	Simon Kjær Jensen	86	Maria Berg Badstue Pedersen	4
Søren Østergaard	1207	Jesper Poulsen	348	Michael Sand	85	Jacob Andersen	3
Hanne Lindblad Jensen	1191	Lars Termansen	328	Julian Drud Sørensen	84	Christian Brahe-Pedersen	3
Charlotte Bøje	1117	Mikkel Eklund	328	Tina Larsen	82	Helle-Lise Ritzau Kaptain	3
Morten Kyndby Holm	1102	Susie Skov	323	Nils Rasmussen	79	Jane Schmidt Klausen	3
Lars Aare Jensen	1084	Birgitte Sonne	321	Katrine Hjulmann	73	Ebbe Fischer	3
Hanne Overbye	1075	Anders Andersen	315	Mathias Lykke	72	Stefan Buur Hansen	3
Lise Stenbro	1029	Gunnar Onghamar	309	Caspar Birk	71	Camilla Pehrson	2
Lone Lærke Krog	1005	Carl Emil Nielsen	305	Anna Bech Sørensen	69	Danske Fagmedier master	2
Carsten Tolbøll	1004	Emma Tofte Lund Poulsen	299	Fie Dømler	65	testbruger	2
Mette Møller	974	Ole Sanvig Knudsen	287	Poul Christoffersen	65	Anders Fuglsang	2
Ida Smith	973	Charlotte Rørth	276	Søren Skov	56	Peter Kargaard	2
Merete Horn	929	Morten Appel	273	Nana Sofia Hansen	54	Morten Munk Andersen	2
Pernille K. Damsgaard	922	Kristian Gull Pedersen	266	Tobias Refstrup Rasmussen	53	JSLbruger Nordjyske	2
Lars Løcke	873	Birgitte Bové	262	AAArtikler parkeret	51	Jan M. Jensen	1
Thomas Nielsen	841	Helle Madsen	258	Camilla Gammelgaard Johansen	50	Tom Andersson	1
Jakob Frey Ahrentzen	832	Kasper Ørkild Hansen	255	Klaus Færch Gjerulff	49	Jørgen la Cour-Harbo	1
Lise Myrup Lassen	793	Emil Abkjær Kristensen	253	Ida Thorsen	48	Rene Sonne	1
Carl Chr. Madsen	785	Nanna Holm Hansen	252	Mathias Overgaard	46	SAXoTECH Systembruger	1
Inge Nørregård	746	Bine Martine Gori	247	Kirsten Østergaard	45	Andreas K. Wielandt	1
Jesper Schouenborg	724	Heidi Majgaard B. Pedersen	244	Gerda Buhl Andersen	44	Per Lyngby	1
Jens Sønderup	709	Daniel Vendner	211	Pia Christensen	44	Michael F. Nørfelt	1
Birgit Eriksen	700	Ida Marie Kristensen	204	Dorte Rohde	42	Simon Dinsen Hansen	1
Søren Wormslev	681	Kirsten Pilgaard	197	Anna Grethe Jensen	41	Mads Skov Aagurd	1
Knud Labohn	653	Marianne Dyhrberg Cornett	192	Henrik Schulz	38	Benita Dreyer-Andersen	1
Hans Jørgen Hansen	646	Susanne Justsen	191	Lone Heilskov	37		

$$S = \{(W', W^*) | W' = \{w_i\}; w_i \in W; W^* = W\} \quad (3)$$

Before calculating word probabilities, a sliding window of size s where $s = 110$ is used to create a set of subdocuments D_s over the document set D , by creating one subdocument for every window of size s by sliding over each document at a rate of one word per step. So document $d = \{w_1, w_2, \dots, w_l\}$, would be converted into $d_1 = \{w_1, w_2, \dots, w_s\}, d_2 = \{w_2, w_3, \dots, w_{s+1}\}, \dots, d_{l-s} = \{w_{l-s}, w_{l-s+1}, \dots, w_l\}$. If l is smaller than s only a single subdocument is made. This also means that the value of s has an impact on how much influence each document has on the metric. We choose not to change the sliding window size s to be able to compare against other papers, but this is a hyperparameter, which could provide better results for our dataset if changed. These subdocuments are used rather than the normal documents to capture some degree of word proximity. Word probabilities are calculated based on how many documents, within the document set D_s , they occur in. $p(w_i)$ is the number of subdocuments in which the word w_i occurs divided by $|D_s|$. $p(w_i, w_j)$ is the number of subdocuments in which both words occur divided by $|D_s|$.

Table XII: Number of documents that contain each taxonomy entry in the Nordjyske dataset. After sorting by the number of documents, only the first 100 and last 100 taxonomy entries are shown. The bolded taxonomy entries are filtered out during preprocessing.

Taxonomy Entry	Number	Taxonomy Entry	Number	Taxonomy Entry	Number	Taxonomy Entry	Number
<i>No taxonomy</i>	103928	Farsø	182	Mallorca	1	Mali	1
STEDER	29535	Skørping	177	Kloning	1	Godstransport	1
Danmark	26145	Hurup	169	Hjørring revyen	1	Energiforbrug	1
Nordjylland	23274	Dronninglund	167	Fredensborg	1	Gedser	1
EMNER	5449	Fjerritslev	165	Iværksættere	1	Mylund	1
Thisted	3592	Aabybro	163	Tall ships races	1	Bygge- og anlægsbranchen	1
Udland	3390	Frankrig	159	Sproget	1	Kunstig intelligens	1
Aalborg	3311	Sverige	158	Grurup	1	Nielstrup	1
Hjørring	2146	Paris	157	Hørning	1	Kristiansand	1
Frederikshavn	1997	Fyn	156	Hem	1	Nordborg	1
Mariagerfjord	1987	Rusland	151	Floorball	1	Uggerhalne	1
Brønderslev	1969	Ulykker	150	Store Brøndum	1	Barmer	1
Vesthimmerland	1660	Aarhus	145	Korup	1	Narkomisbrug	1
Hovedstadsområdet	1380	Tyskland	144	Fødevarsikkerhed	1	Adoption	1
Rebild	1289	Moskva	143	Hammershøj	1	Fødevarerindustri	1
Jammerbugt	1198	Arden	142	Hjernesgade	1	Smugling	1
København	1125	Politik	139	CATEGORY	1	Skikke og traditioner	1
Hobro	996	Morsø	137	AaB Plus	1	Kigali	1
Aalborg og omegn	833	Berlin	132	Ekstrem sport	1	Holtet	1
Thisted og omegn	800	Sjælland	131	Mozambique	1	Fritidshuse	1
Midtjylland	783	Løkken	129	Maputo	1	Årslev	1
Mors	630	New York	124	Handelsskole	1	Aalborg Håndbold	1
Aars	540	Natur	123	Vendsyssel Håndbold	1	Oman	1
USA	478	Erhverv	119	Biludstyr	1	Turistbranchen	1
Frederikshavn og omegn	410	Spanien	119	Brandstiftelse	1	Øl	1
Sport	408	Klitmøller	118	Nørre Dråby	1	Forlystelsespark	1
England	381	Aalestrup	118	Stae	1	Etiopien	1
London	379	Herning	116	Rebild Bakker	1	Addis Abeba	1
Hjørring og omegn	369	Stockholm	115	Kampsport	1	Lystsejlad	1
Løgstør	368	Madrid	115	Lynnedslag	1	Herfølge	1
Mariagerfjord og omegn	364	Jerslev	113	Frederikshavn White Hawks	1	Sexhikane	1
Brønderslev og omegn	360	Nørager	111	Paraguay	1	Gebyrer	1
Skagen	341	Sundhed	105	Asuncion	1	Frederikssund	1
Sæby	334	Sindal	105	Nordkraft	1	Gadeuorden	1
Vesthimmerland og omegn	327	Brovst	105	Vendsyssel Elite Badminton	1	Mjels	1
Syddanmark	301	Trafik	99	Tonga	1	Katmandu	1
Støvring	300	Blokhush	95	Efteruddannelse	1	Militærvæbner	1
Hadsund	277	Lønstrup	92	Vin	1	Fakse	1
Hanstholm	276	Uddannelse	91	Årup	1	Kulturpolitik	1
Kultur	246	Pandrup	88	Fiskeripolitik	1	Skræm	1
Rebild og omegn	245	Vrå	88	Gudumlund	1	Taiwan	1
Washington	242	Vorupør	86	Myrhøj	1	Taipei	1
Jammerbugt og omegn	229	Øster Hurup	85	Ejerslev Lyng	1	Økonomisk kriminalitet	1
Nørresundby	223	Hvalpsund	82	La Paz	1	Udviklingsbistand	1
Mariager	214	Odense	82	Byplanlægning	1	Kollerup	1
Belgien	207	Aså	80	Ajstrup	1	Askildrup	1
Bruxelles	207	Norge	79	Guatemala	1	Helsingø	1
Hirtshals	197	INDHOLDSTYPER	79	Vedsted	1	Albanien	1
Kriminalitet	192	Videnskab	78	Yemen	1	Tirana	1
Hjallerup	190	Israel	78	Skovsted	1	Jern & maskinindustrien	1

As part of the word set confirmation measure we create a Normalized Pointwise Mutual Information (NPMI) matrix of size $|W| \times |W|$, with one entry per word pair combination in W .

$$\text{NPMI}(w_i, w_j) = \frac{\log \frac{p(w_i, w_j) + \epsilon}{p(w_i) \cdot p(w_j)}}{-\log(p(w_i, w_j) + \epsilon)} \quad (4)$$

where ϵ is a low number (10^{-12}) used to avoid $\log(0)$. The NPMI matrix describes how much each word in the topic co-occurs with the other words. Each value is between -1 and 1 , with -1 meaning that the words never occur together and 1 meaning

Table XIII: Statistics over documents associated with metadata values, before and after preprocessing.

Metadata	Min	Max	Mean	Median	Std.
Author	1	9893	612.6	191	1219.7
Author (preprocessed)	15	9893	751.7	323	1311.9
Category	1	20204	2397.6	548	3812.1
Category (preprocessed)	188	20204	4090.0	2681	4227.3
Taxonomy	1	29535	123.0	6	1385.9
Taxonomy (preprocessed)	14	29535	1598.9	118.5	9298.4

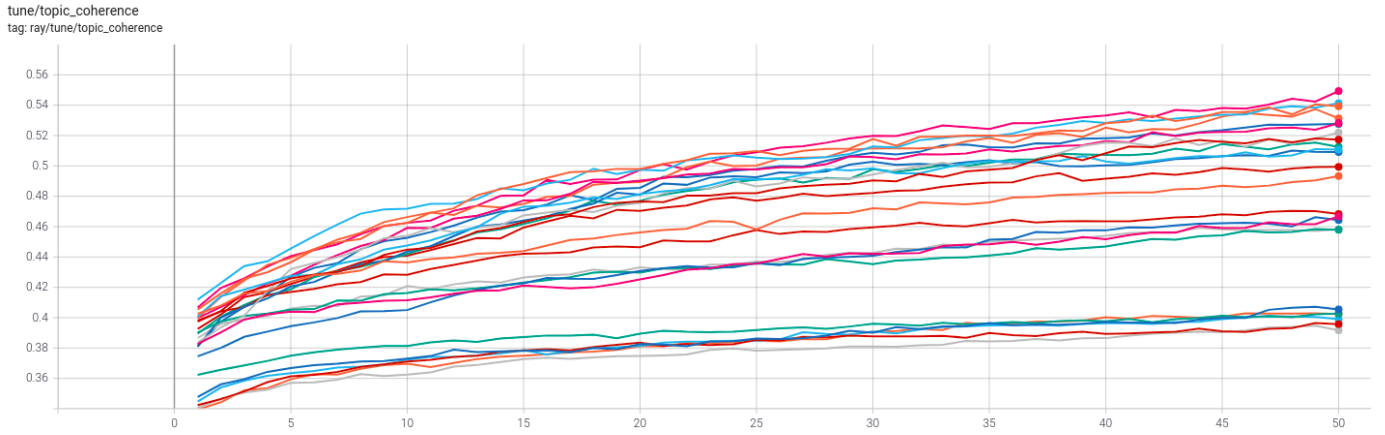


Figure 11: Grid search over the K_2 parameters in Table I.

that they only occur together.

After having calculated the NPMI matrix, we construct context vectors for both elements W' and W^* in each word-pair S_i , by summing over the rows of the NPMI matrix. This summation describes how much each top word co-occurs with the other words in W .

$$\vec{v}(W') = \left\{ \sum_{w_i \in W'} \text{NPMI}(w_i, w_j)^\gamma \right\}_{j=1, \dots, |W|} \quad (5)$$

where γ can be used to further prioritize higher values. For this paper we use $\gamma = 1$, as recommended by Syed and Spruit [16].

We now have a pair of context vectors for each word pair S_i and we want to know how different these vectors are. This is calculated using cosine similarity as a confirmation measure.

$$\phi_{S_i}(\vec{u}, \vec{w}) = \frac{\sum_{i=1}^{|W|} u_i \cdot w_i}{\|\vec{u}\|_2 \cdot \|\vec{w}\|_2} \quad (6)$$

where \vec{u} is the context vector $\vec{v}(W')$, and \vec{w} is the context vector $\vec{v}(W^*)$.

Lastly, the confirmation measures are aggregated using the arithmetic mean, to achieve the coherence value of topic t .

$$C_v = \frac{1}{|S|} \sum_{i=1}^{|S|} \phi_{S_i} \quad (7)$$

C. Grid search




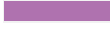

We have run a grid search based on the standard model of the LDA, which allowed us to find an approximated optimal hyperparameter configuration. We optimized for the best topic coherence measure during this grid search after 50 epochs, which was the number of epochs where the gains started to diminish. The five lowest values in Figure 11, are where the α is 0.1 and η is 0.01. This shows us that having the α high and the η low does not yield good coherence results. The next five values are where the α and η are 0.01, which also shows that a low value in both hyperparameters does not yield the best results either. There is only a minor difference between the top configurations in Figure 11, but the top two configurations are 70 and 90 topics. The best hyperparameters are $K = 90$, $\alpha = 0.01$, and $\eta = 0.1$.

D. Coloring articles

In this section, we analyze a few more articles to get a more in-depth view of how the topic distributions differ between the models. We are still looking at the top three topics for each model and taking the 200 most probable words.

As a reminder, the color scheme is shown below in Table XIV. Figure 12 shows an article about a race car driver from Aalborg, and him driving in the European Le Mans Series. The article is not that long, and therefore there are not many words marked. From the colored words in the article, the words appearing in all models are not very descriptive of what the article is about. The topic that is the most present in this article is most likely a sports topic based on the top 10 words. The top 10 words of this most present LDA topic are: ['tour', 'vandt', 'løb', 'par', 'mig', 'løbet', 'hold', 'vm', 'slag', 'nummer']. We can see based on these words that other sports are higher up in the list, such as 'tour' from Tour de France, which is likely due

Table XIV: Color scheme for each model.

Topic Model	Color
Latent Dirichlet allocation	
Author-topic model	
Category-topic model	
Taxonomy-topic model	
Word appearing in 3+ models	

dårligt løb for aalborg-kører spielberg: den aalborgensiske racerfører anders fjordbach havde sammen med teamet high class racing en gennemført skidt tredje afdeling af european le mans series på red bull ring i østrig. her blev det til en beskedent ottendeplads i lmp2-klassen. - det var selvfølgelig ærgerligt kun at blive nummer otte, men det er vel ikke en skam at have en dårlig weekend, siger anders fjordbach, der kører sammen med dennis andersen. forkert dækvalg, et mindre sammenstød, en generator, der stød af og andre små problemer var årsagen. det eneste positive var, at teamet fortsat er på tredjepladsen sammenlagt.clajen

Figure 12: An article about a race car driver from Aalborg, and him driving in the European Le Mans Series.

to there being few Le Mans articles in the dataset.

Figure 13 is about politics in Aalborg, specifically about charter ships docking in Aalborg and how the municipality is trying to solve this problem. Figure 13 is a bit longer than the one in Figure 12, which in turn colors more words. We can see that the majority of these words appear in every model, where a lot of non-descriptive words occur, such as 'finde' (to find), 'giver' (give), and 'ting' (stuff). Opposite to the original article analyzed in the paper, word combinations of the category-Topic model and LDA model are more present within this article. From the top words within the category-topic model, many generic words appear, which might be why there is a higher trend in word appearances. The author-topic model is not showing up with many unique words, but the author (Pernille K. Damsgaard) has written 922 articles, which might indicate that she usually does not write about this topic.

E. Stemming the dataset

In this project, we have done minimal preprocessing of the dataset. Stop words have not been specifically removed, but most of these are naturally removed since we filter all words that appear in more than 10% of the documents. We do minimal preprocessing because in a previous project a more aggressive preprocessing step turned out to hurt the performance of the LDA model. This made us completely avoid trying to include a stemming process, even though this was only a smaller part of

rådmand: der skal findes en løsning aalborg: rådmand hans henrik henriksen (s) er indstillet på, at der fra kommunens side bidrages til en løsning, der sikrer krydstogtgæsterne sikker adgang på tværs af slotspladsen. - lad os prøve at se, om vi ikke ved fælles hjælp kan finde en god løsning, der fungerer, på det her, siger by- og landskabsrådmanden med henvisning til dagsordenen for det kommende møde med deltagelse af visitaalborg, kommunen ved trafik og veje, politiet samt aalborg havn, der også har en interesse i udvikling af krydstogtforretningen. forud for mødet har han ikke noget bud på en løsning, og han gentager en tidligere afvisning af et traditionelt fodgængerfelt. - folk med viden på det felt har forklaret mig, at der faktisk sker flere ulykker i et fodgængerfelt, fordi nogle bilister overser striberne, hvorved det giver gående en falsk tryghed. men lad os nu slette tavlen og se på det med friske øjne, siger rådmanden, der er enig med visitaalborgs lars bech i, at der skal findes en tilfreds stillende løsning. - når vi markedsfører os i forhold til krydstogsturismen, og ved fra visitaalborg, at det ikke er ligetil at få rederierne til at vælge aalborg, skal der også være en service, der virkelig fungerer, og der tæller de små ting som adgang til byen også. så det her skal vi finde en løsning på. det ville ikke være til at bære, hvis der skete en ulykke, siger rådmanden, der mener, at det på tværs vil være muligt at finde de nødvendige penge til et eventuelt nyanlæg, siger rådmanden, der kan forestille sig, at en form for lysregulering kommer til at indgå i løsningen. - det vil ikke være tilstrækkeligt bare at male zebrastriber på vejen, siger han.

Figure 13: An article about politics in Aalborg, specifically about charter ships docking in Aalborg and how the municipality is trying to solve this problem.

Table XV: An example of topics that are similar between the non-stemmed and stemmed LDA model. Each line of words in the table is a topic. Here the topics are all about crime and the police.

Model	Topics
Non-stemmed	dræbt, mennesker, personer, politiet, mindst, angreb, oplyser, afp, reuters, byen stjalet, indbrud, klokken, nordjyllands, thisted, politi, politiet, mandag, villa, oplyser arige, politiet, mand, arig, politi, sagen, oplyser, indbrud, retten, stjalet politiet, mand, arig, politi, arige, bil, nordjyllands, thisted, bilen, klokken arig, bil, mand, politiet, kørte, hobro, bilen, thisted, klokken, arige arige, politiet, mand, arig, politi, retten, sagen, fængsel, ham, oplyser
Stemmed	stjal, indbrud, politi, tyv, klok, nordjylland, bil, thisted, tyveri, villa politi, nordjylland, brand, bil, mand, indbrud, stjal, klok, hus, beredskab sag, dom, blokhus, advokat, ham, fængsel, sagen, dreng, politi, mand politi, mand, kvind, sag, sigt, bil, mænd, nordjylland, oplys, anhold politi, mand, bil, person, oplys, kørt, kvind, sigt, nordjylland, dræbt

this previous preprocessing step. Though, after further consideration, it is worth looking into what effect adding just stemming to our current preprocessing, described in Section II, would have on the standard LDA model.

With stemming included, the number of unique words goes down to 60,651 from the previous 69,192 unique words. This means that 8541 semantically similar words have been stemmed down to their root forms. Even though much fewer unique words are in our dataset, which can have an effect on which articles are removed, the dataset only goes from 139,064 articles to 139,060, which is a removal of 4 articles. There are most likely fewer articles because when we filter words that appear in more than 10% of the documents after stemming, some new documents may end up empty.

In the non-stemmed LDA model, words that have no meaning topically seem to have a large influence. This is still the case in the stemmed model, where words like 'du' (you), 'mig' (me), and 'a' still appear in the top words of topics. To remove these words, we would have to include some more advanced preprocessing in the form of stop word removal and part-of-speech (POS) tagging. From experience, POS taggers cannot figure out the semantic meaning of words that are spelled the same, and then we would somehow have to choose which meaning is correct, possibly creating new errors in our data.

Examples of top words for similar topics between the non-stemmed and the stemmed LDA models can be seen in Table XV. It is seen here that some of the topics between the models are clearly about the same subjects. This is also generally seen for other topics throughout the two models. A positive effect of having done stemming is that the stemmed model's topics might also be slightly more understandable because words with similar meaning (e.g., 'politi' and 'politiet') do not appear. Other than this, including stemming does not impact the quality of the topics significantly.

F. Pyro model implementation

At the beginning of the project, we decided to look in a few directions for the best way to implement an LDA model, and we found that the probabilistic programming language Pyro could be used [1]. Pyro is a probabilistic programming language that is written in Python with PyTorch as a backend. This makes it ideal for making quick implementations of models, with the possibility of using tensors and sampling with PyTorch distribution methods. Pyro also has a built-in stochastic variational inference class that simplifies the training of a model. These features made it an ideal programming language to look into.

We found that Pyro has made two code examples of LDA available, one using purely distributions and the other using a neural network approach. We tested both of these examples to see if we could adapt an existing implementation to work with our dataset. Pyro's first example of an LDA model is very simple³. It is a functional model, but it has the limitation that all documents need to have the same number of words. This limitation is too restrictive for most datasets, including ours, and changing the model to handle any number of words makes the model unable to converge.

The other model example from Pyro is called ProdLDA⁴. This model uses an autoencoding approach where the model encodes the document-topic distribution θ and decodes the topic-word distribution β . This approach can handle more complex models and seems to be able to handle the inclusion of metadata. While this model is able to learn with our dataset, it restricts the possibility of changing the model since the main distributions of the model are learned with a neural network. These two examples show that Pyro can be useful when making basic topic models, but since we want to extend LDA with metadata, Pyro becomes too restrictive. To give us more opportunities to customize our models and not be restricted by Pyro's implementation, we chose to implement the models from scratch with Gibbs sampling.

G. Gibbs sampling

The Gibbs Sampling algorithm consists of two procedures: Random Initialization and Gibbs Sampling. In the following sections, we explain how these procedures have been implemented.

³Amortized Latent Dirichlet Allocation: <https://pyro.ai/examples/lda.html>

⁴ProdLDA: <https://pyro.ai/examples/prollda.html>


```

1 import numpy as np
2
3 def rand_initialize(documents: List[np.ndarray]):
4     wt_assignment = []
5     for doc in documents:
6         curr_doc = []
7         for word in doc:
8             # Construct the topic distribution
9             pz = _conditional_distribution()
10
11            # Draw a new topic and assign it
12            t = np.random.multinomial(1, pz).argmax()
13            curr_doc.append(t)
14
15            # Increase the topic counts
16            increase_count()
17        wt_assignment.append(curr_doc)
18    return wt_assignment

```

Listing 1: Random Initialization for the Gibbs sampler.

1) *Random Initialization*: Before a Gibbs sampling algorithm can run, every word needs to be assigned a random topic. To do this, we iterate over each word within each document and assign a random topic to it. This procedure is shown in Listing 1. Various parameters are left out to simplify the code listing of our initialization method. The `_conditional_distribution()` function creates a distribution of topics based on the current word. Standard LDA is based on Equation 8 [15] and the author-topic and category-topic models are based on Equation 9 and Equation 10, respectively.

$$P(z_i = k | w_i = m, \mathbf{z}_{-i}, \mathbf{w}_{-i}) \propto \underbrace{\frac{C_{dk}^{DT} + \alpha}{\sum_{k'} C_{dk'}^{DT} + T\alpha}}_{Doc-Topic} \underbrace{\frac{C_{mk}^{WT} + \eta}{\sum_{m'} C_{m'k}^{WT} + V\eta}}_{Topic-Word} \quad (8)$$

$$P(z_i = k | w_i = m, \mathbf{z}_{-i}, \mathbf{w}_{-i}, a_d) \propto \underbrace{\frac{C_{ak}^{AT} + \alpha}{\sum_{k'} C_{ak'}^{AT} + T\alpha}}_{Author-Topic} \underbrace{\frac{C_{mk}^{WT} + \eta}{\sum_{m'} C_{m'k}^{WT} + V\eta}}_{Topic-Word} \quad (9)$$

$$P(z_i = k | w_i = m, \mathbf{z}_{-i}, \mathbf{w}_{-i}, c_d) \propto \underbrace{\frac{C_{ck}^{CT} + \alpha}{\sum_{k'} C_{ck'}^{CT} + T\alpha}}_{Category-Topic} \underbrace{\frac{C_{mk}^{WT} + \eta}{\sum_{m'} C_{m'k}^{WT} + V\eta}}_{Topic-Word} \quad (10)$$

where C_{dk}^{DT} is the number of times document d uses topic k and C_{mk}^{WT} is the number of times topic k uses word m . C_{ak}^{AT} and C_{ck}^{CT} is the number of times author a and category c use topic k , respectively. \mathbf{z}_{-i} represents the topic assignments where the current instance is disregarded. α and η are the Dirichlet parameters for the document-topic and topic-word distribution, respectively. The first fraction describes how likely topic t appearing in document d is, and the second fraction describes which words are most probable in topic t . Following the code in Listing 1, as we initialize, the words get a higher probability of clustering together, since we increase the counts every time at line 16. In the author-topic and category-topic models, a metadata distribution is used instead of the document-topic distribution. This can be seen in Equation 9 and Equation 10

2) *Gibbs Sampling*: We can start investigating the Gibbs sampling method itself, where we iterate over each word in every document and draw a new topic based on the given topic distribution. As in Listing 1, the code has been simplified. The Gibbs sampling method is very similar to the random initialization method in Listing 1, but with a few additions. Now we introduce the `decrease_count()` which decreases the topic count for both words and documents, and `increase_count()` which increases them. This is done because new samples need to be calculated based on all other word topic assignments (not including the current word). The sampling is explained in Listing 2. On line 13, we get the current topic for the given word and on line 23 we assign a newly drawn topic to that word.

Parallel Gibbs Sampling: We have also implemented a Gibbs sampler, which works in parallel by splitting up the dataset into p parts, where p is the number of processes. This is to create $\frac{1}{p}$ amount of progress for each process and then combine them. Each process gets a specific split of the dataset and the available words. This is done to avoid race conditions on increasing and decreasing counts in the Gibbs sampler. However, normally the implementation of this algorithm is run on the GPU, where we implemented it for CPU where IO was very slow. Because of the slow combination, due to IO, it did not give us any speed up, so the implementation was not used for this project.

```

1 import numpy as np
2
3 def gibbs_sampling(documents: List[np.ndarray],
4   doc_topic_dist: np.ndarray,
5   doc_topic_count: np.ndarray,
6   topic_word_dist: np.ndarray,
7   topic_word_count: np.ndarray,
8   wt_assignment: List[List[int]]):
9
10  for d_index, doc in documents:
11    for w_index, word in enumerate(doc):
12      # Get the topic of the current word
13      topic = wt_assignment[d_index][w_index]
14
15      # Decrease the topic count
16      decrease_count()
17
18      # Sample a new topic
19      pz = _conditional_distribution()
20      topic = np.random.multinomial(1, pz).argmax()
21
22      # Assign topic to the current word
23      wt_assignment[d_index][w_index] = topic
24
25      # And increase the topic count
26      increase_count()

```

Listing 2: The Gibbs Sampling Method.

H. Pachinko implementation

We have implemented a Pachinko Allocation Model (PAM) algorithm able to support any DAG structure, where each layer only has edges to all the nodes in the next layer, as with the 'Four-Level PAM' presented by Li and McCallum [9].

We use Gibbs sampling for performing inference. For each word, a chain of topics is sampled by calculating the probability of all combinations of topics and making a weighted sample. The probability of each topic combination is calculated using the joint probability of the topics, as presented in Equation 11. This equation is for the 'Five-Level PAM' that we use in the paper.

$$P(Z_{w2} = t_a, Z_{w3} = t_b, Z_{w4} = t_c | \mathbf{D}, z_{-w}, \alpha, \eta) \propto \underbrace{\frac{n_{1a}^d + \alpha_{1a}}{n_1^d + \sum_{a'} \alpha_{1a'}}}_{\text{Root} \rightarrow \text{Tax.1}} \times \underbrace{\frac{n_{ab}^d + \alpha_{ab}}{n_a^d + \sum_{b'} \alpha_{ab'}}}_{\text{Tax.1} \rightarrow \text{Tax.2}} \times \underbrace{\frac{n_{bc}^d + \alpha_{bc}}{n_b^d + \sum_{c'} \alpha_{bc'}}}_{\text{Tax.2} \rightarrow \text{Topics}} \times \underbrace{\frac{n_{cw} + \eta_w}{n_c + \sum_m \eta_m}}_{\text{Topics} \rightarrow \text{Words}} \quad (11)$$

As in Li and McCallum [9], Z_{w2} , Z_{w3} , and Z_{w4} are topic assignments for the three middle layers of topics in our Five-Level PAM. The root topic is not part of this equation since all words are part of it, so the probability does not need to be calculated. Z_{-w} is the word topic assignment, for all other words except the one that is being updated. n_x^d is the number of times topic t_x occurs in document d according to Z_{-w} . The n_{xy}^d describes how many times topic t_y is sampled from its parent t_x within document d according to Z_{-w} . n_x is the number of times topic t_x occurs in the corpus according to Z_{-w} , and n_{xw} is the number of times a word w is in t_x according to Z_{-w} .

However, the PAM framework can support any number of layers using this structure. In order to do this, we must generalize the process of Gibbs sampling for pachinko using 'level' DAG structures. Firstly, before the Gibbs sampling begins a one-time random initialization is made, where each word in each document is randomly assigned to a chain of topics (one for each layer). In our PAM, some topic layers represent taxonomy layers, since some documents in the dataset already have a topic entry. These documents are assigned the topics corresponding to their taxonomy entries, and the rest of the taxonomy chain is then randomly generated if it is not already complete.

The Gibbs sampling for PAM consists of the following steps for each word in each document:

- A) Decrease count
- B) Calculate layer combinations
- C) Multiply layer combinations
- D) Weighted sample
- E) Increase count

Following is an overview of each of the steps. Firstly, the current word is removed from the counts of how many words are assigned to each topic. After the count has been decreased, we calculate for each combination of successive layers, the probability of each possible topic combination. This process is explained further in Section H1.

Each of these calculations is combined to calculate the final probability of each possible topic combination. This process is explained in Section H2. One topic combination is then sampled, using a weighted sampling based on the probabilities of all topic combinations. Finally, once a new topic combination has been chosen, the counts of how many words are assigned to each topic are increased accordingly.

In the next section, details about calculating layer combinations and multiplying layer combinations are explained, since these are the main differences between how LDA and PAM use Gibbs sampling.

1) *Calculate layer combinations:* This is done based on observations in Equation 11. The equation consist of several fractions equal to the number of layers - 1, with each fraction representing the relationship between two layers. The last fraction is a little different as it takes word topic assignments for the whole corpus into account, unlike the other fractions which only look at the word topic assignments for the current document.

In order to run efficiently, we calculate all topic combinations at the same time, rather than calculating a specific one as outlined in Equation 11. To do this, we operate on vectors and matrices rather than single values. So for the fraction $\frac{n_{ab}^d + \alpha_{ab}}{n_a^d + \sum_{b'} \alpha_{ab'}}$ from Equation 11, n_{ab}^d is a matrix which indicates the number of words in document d that has been assigned to each combination of topics in layer a and layer b , with one row for each topic in layer a and one column for each topic in layer b . Similarly, n_a^d is a vector showing the number of words in document d assigned to each topic in layer a , rather than a single topic. If some taxonomy entries for the document are already known, the matrices and vectors are sliced to only include the relevant unknown topics.

2) *Multiplying layer combinations:* Once all the two-layer combinations have been calculated, they have to be combined to find the probability of all topic combinations. To do so, the layer combinations are multiplied across the dimensions they share. So for an $A \times B$ matrix and a $B \times C$ matrix, values that share the same B entry are multiplied together to form a three-dimensional $A \times B \times C$ array. Importantly, the shared dimension is kept, unlike with matrix multiplication. By keeping all dimensions, the final array has one entry for all possible topic combinations.

Table XVI: Top 10 words for each lowest level topic in the results of our taxonomy-topic model.

Topic	Top 10 words
1	nordjylland, læger, patienter, region, læge, regionen, praktiserende, sygehus, patienterne, behandling
2	turister, ferie, lokale, gæster, øen, skagen, strand, byen, steder, ligger
3	hvordan, unge, fokus, mennesker, skabe, verden, vigtigt, made, handler, hinanden
4	gamle, maske, ganske, altsa, først, faktisk, forsvaret, næsten, mest, set
5	direktør, fly, lufthavn, københavn, selskabet, passagerer, thomas, søren, sas, aarhus
6	mal, halvleg, minutter, kampen, serie, kamp, mors, fc, morsø, thisted
7	hobro, lørdag, kaffe, jul, mariager, gamle, klokken, december, børn, mulighed
8	venstre, valg, valget, partiet, partier, parti, stemmer, mette, politik, regering
9	procent, viser, tal, antallet, milliarder, pct, seneste, penge, millioner, indland
10	arige, mand, arig, retten, politiet, ham, mænd, fængsel, manden, sagen
11	virksomheder, nordjylland, nordjyske, virksomhederne, arbejdspladser, samarbejde, arbejdskraft, udvikling, job, vækst
12	hjørring, teater, vendsyssel, forestillingen, kl, løkken, publikum, lørdag, klokken, festivalen
13	offentlige, penge, bedre, mener, kommunerne, regeringen, ansatte, kommuner, brug, arbejde
14	mig, henrik, christensen, ham, hans, andersen, jensen, arbejde, rigtig, synes
15	arets, prisen, bedste, pris, meter, vandt, dansk, thisted, guld, nordjyske
16	salg, gamle, peter, sælge, solgt, firmaet, købe, ejer, niels, sat
17	brand, branden, beredskab, gik, kvinder, huset, nordjyllands, ilden, matte, ild
18	foredrag, bøger, bibliotek, bogen, bog, biblioteket, forfatter, foredraget, historie, dk
19	mig, maske, du, folk, synes, ting, faktisk, nogen, altid, tror
20	syrien, dræbt, tyrkiet, fn, angreb, al, mennesker, israel, stat, islamisk
21	unge, uddannelse, studerende, gymnasium, elever, uddannelser, universitet, procent, uddannelsen, nordjylland
22	havn, havnen, hanstholm, fisk, skagen, hirtshals, meter, vandet, skibe, skibet
23	nielsen, løb, nummer, vm, jakobsen, formel, dm, banen, skelund, løbet
24	grader, vejr, varme, sommer, regn, vejret, dage, vand, landet, uge
25	aab, jacob, kasper, rasmus, jakob, pedersen, andersen, friis, minut, allan
26	tyske, døde, hans, børn, skriver, personer, politiet, mennesker, tyskland, død
27	klubben, hold, medlemmer, unge, sport, fodbold, u, klub, cup, træning

28 børn, skole, børnene, forældre, skolen, elever, forældrene, unge, barn, voksne
 29 vm, league, klubben, spiller, fodbold, spillere, kampe, manchester, em, champions
 30 sagen, fængsel, retten, sag, arige, dom, dømt, dommen, ars, idømt
 31 arbejde, arbejdsmarkedet, pension, job, grønland, ret, f, personer, nedslidte, folk
 32 dette, politikere, maske, mig, vel, disse, mennesker, hvorfor, nogen, land
 33 frederikshavn, millioner, svenske, sverige, norge, milliarder, norske, procent, dollar, solgt
 34 hobro, hadsund, morgen, mariagerfjord, bio, sker, rebild, øster, arr, skørping
 35 gamle, museum, naturen, området, natur, skov, ligger, lille, projektet, historiske
 36 formand, jensen, nielsen, jens, erik, jørgen, sørensen, bestyrelsen, pedersen, sagde
 37 køre, trafik, trafikken, kører, biler, vejen, vej, egholm, motorvej, forbindelse
 38 lars, f, arbejde, dansk, formand, medlemmer, ansatte, rasmussen, leder, mig
 39 kommunen, sagen, mener, kommunens, sag, hjørring, kystsikring, teknik, området, sager
 40 hans, film, ham, filmen, anders, tv, erne, fylder, skuespiller, liv
 41 universitet, professor, forskere, forskning, forskerne, viser, verden, institut, procent, aarhus
 42 mig, min, mit, ham, aldrig, gik, lille, maske, mine, altid
 43 fc, point, aab, kampe, kamp, hold, brøndby, holdet, vendsyssel, mal
 44 aars, vesthimmerland, brønderslev, løgstør, vesthimmerlands, farsø, børn, frivillige, dronninglund, kors
 45 dyr, naturen, natur, ulve, fugle, ulven, arter, dyrene, vilde, ulv
 46 bank, banken, millioner, nordjyske, banks, penge, ebh, jyske, kunder, bankens
 47 kirke, kirken, sognepræst, præst, søndag, koret, gudstjeneste, aften, kor, organist
 48 sagen, skriver, fejl, oplysninger, sag, politiet, reglerne, mener, indland, kontrol
 49 eu, europa, lande, europæiske, tyskland, kommissionen, parlamentet, tyske, bruxelles, polen
 50 usa, trump, amerikanske, new, præsident, york, donald, washington, skriver, hus
 51 jammerbugt, brønderslev, projektet, aabybro, nyt, plads, klar, kvadratmeter, byen, brovst
 52 usa, præsident, trump, kina, rusland, amerikanske, russiske, iran, reuters, sagde
 53 du, facebook, digitale, medier, it, data, dk, bruger, via, nettet
 54 borgmester, kommuner, v, sagde, byrådet, nordjyske, mogens, borgmesteren, arne, per
 55 biler, bil, model, bilen, modeller, a, vw, e, ford, toyota
 56 mad, vin, øl, restaurant, spise, smag, kød, maden, hund, spiser
 57 vandt, runde, slag, par, vm, turneringen, wozniacki, nummer, open, sæt
 58 kr, mio, morsø, kommunen, penge, mors, millioner, rebild, budget, tilskud
 59 musik, koncert, sange, spiller, koncerter, band, koncerten, festival, musikken, publikum
 60 virksomheden, millioner, a, direktør, procent, medarbejdere, selskabet, overskud, ansatte, virksomhed
 61 minut, hobro, mal, kamp, mikkell, vendsyssel, kampen, frederikshavn, pirates, kampe
 62 støvring, tog, skagen, rebild, dsb, nordjyske, trafik, køre, nordjylland, kører
 63 arig, mand, bil, politiet, kørte, politi, bilen, arige, skete, klokken
 64 sygdom, behandling, mennesker, patienter, medicin, parørende, sygdommen, syge, sundhed, psykisk
 65 boliger, området, kommunen, lokalplan, projektet, byggeri, møller, vindmøller, teknik, område
 66 tv, the, film, of, dr, filmen, serien, fem, a, hver
 67 energi, el, grøn, grønne, procent, co, omstilling, strøm, varme, kr
 68 butikken, butikker, butik, sæby, kunder, kunderne, nykøbing, varer, købmand, mors
 69 elever, unge, skole, eleverne, skolen, skoler, klasse, børn, folkeskolen, lærere
 70 unge, politiet, kvinder, antallet, mænd, borgere, politi, vold, personer, udsatte
 71 plads, byen, p, hotel, by, gamle, byens, pladser, området, hus
 72 politiet, politi, indbrud, mand, stjalet, nordjyllands, klokken, arig, oplyser, thisted
 73 du, jorden, haven, planter, vand, ned, træer, blomster, sma, jord
 74 ældre, borgere, kommunen, millioner, penge, nordjylland, plejehjem, borgerne, kommunens, budget
 75 du, din, dig, dit, altsa, dine, maske, nemlig, bruge, hvordan
 76 tour, etape, løbet, kilometer, michael, fuglsang, nord, france, hold, spar
 77 regeringen, penge, bedre, samfund, mennesker, dette, disse, børn, dansk, sikre
 78 kunst, udstillingen, udstilling, værker, kunstnere, museum, malerier, kunstner, billeder, kunsten
 79 mig, min, hendes, hende, rigtig, arige, altid, arbejde, mor, mine
 80 km, kr, hk, t, m, bilen, motor, a, bil, l
 81 stjalet, indbrud, mandag, madsen, klokken, kl, politiet, tirsdag, onsdag, thisted
 82 dansk, regeringen, løkke, v, venstre, lars, folkeparti, df, rasmussen, mener
 83 ord, bogen, bog, liv, verden, skrevet, hendes, du, historie, skriver

84 løbet, rebuild, blokhus, løb, deltagerne, deltagere, turen, kl, kilometer, tur
85 thisted, thy, mors, hanstholm, klitmøller, vorupør, hurup, lokale, nationalpark, morsø
86 handbold, mors, thy, hold, kamp, sæson, kampe, kampen, point, holdet
87 prins, fylder, henrik, hans, larsen, tv, kim, københavn, senere, født
88 eu, britiske, brexit, storbritannien, aftale, london, may, premierminister, johnson, theresa
89 frivillige, foreningen, løgstør, aktiviteter, lokale, foreninger, medlemmer, forening, formand, lørdag
90 landbrug, landbruget, landmænd, vand, miljø, affald, bedre, natur, vandløb, fødevarer

I. Category and author PAM

Given the good results of the taxonomy-topic model, we decided to test PAM with the two other metadata types: Author and Category. These metadata are not layered and can therefore not utilize the layered nature of PAM in the same way. Instead, a Four-Level PAM is used, with a root layer, a metadata layer where authors and categories are locked into topics using the technique outlined in Section V-B, a layer with 90 'blank' topics, and a word layer.

As can be seen from the results in Table XVII, these models achieve better results than the previous author and category models and the LDA model. However, the taxonomy-topic model is still better overall. The same conclusion is reached after manual inspection of the topics of these models.

For comparison we also ran a Four-Level PAM without any metadata, using 100 and 90 as the sizes of the two middle layers. This ended up providing very good results, slightly better than the taxonomy-topic model. The elapsed time of the Four-Level PAM was ~ 128 hours for 50 epochs, roughly the same as our Five-Level taxonomy-topic model, which had an elapsed time of 132 hours for 50 epochs. The slower speed compared to the size of the DAG structure is due to this model being unable to skip observed values when sampling since it does not incorporate any metadata. This also shows that while Category PAM and Author PAM get better results than their LDA counterparts, it is better to run PAM without modifications. This could be due to our models being unable to make good use of the extra information provided by the metadata. It might also point towards these two particular metadata types not being particularly useful in this specific project. The category metadata is generally very vague and some categories have seemingly no connection between documents. As discussed earlier, the author metadata might not be as useful within journalism as authors don't write about the same topics as with scientific papers. And while the PAM without metadata does achieve better topic coherence than the taxonomy-topic model, they are close enough in both topic coherence and manual inspection of the quality of topics that no conclusions can be drawn.

J. The author-category model

We have created a combination model, as an extension of our metadata models, to see whether using multiple metadata types at the same time to draw topics, would affect the performance of the topic model. The idea is that this model combination should give insight into what a model learns when multiple metadata influence the topics chosen. The model we have created is the Author-Category combination model. As the name suggests, this model includes an author-topic distribution and a category-topic distribution, and the plate notation can be seen in Figure 14.

To combine the author and category metadata, we use the notation described in Rosen-Zvi et al. [15] and in Section G.

$$P(z_i = k | w_i = m, \mathbf{z}_{-i}, \mathbf{w}_{-i}) \propto \underbrace{\frac{C_{ak}^{AT} + \alpha}{\sum_{k'} C_{ak'}^{AT} + T\alpha}}_{\text{Author-Topic}} \underbrace{\frac{C_{ck}^{CT} + \alpha}{\sum_{k'} C_{ck'}^{CT} + T\alpha}}_{\text{Category-Topic}} \underbrace{\frac{C_{mk}^{WT} + \eta}{\sum_{m'} C_{m'k}^{WT} + V\eta}}_{\text{Topic-Word}} \quad (12)$$

where C_{ak}^{AT} and C_{ck}^{CT} is the number of times author a and category c use topic k , respectively. The intuition behind this is to multiply the three distributions together to get a combined distribution to draw a topic from. But before drawing a topic, we normalize it based on the sum of the distribution.

$$dist = \frac{x}{\sum_1^K x} \quad (13)$$

Table XVII: Topic coherence of author PAM, category PAM, and PAM without metadata (marked with bold) compared to previous models.

Topic Model	Topic Coherence
Latent Dirichlet allocation	0.520
Author-topic model	0.335
Category-topic model	0.370
Taxonomy-topic model	0.660
Author PAM	0.598
Category PAM	0.585
Pachinko Allocation Model	0.670

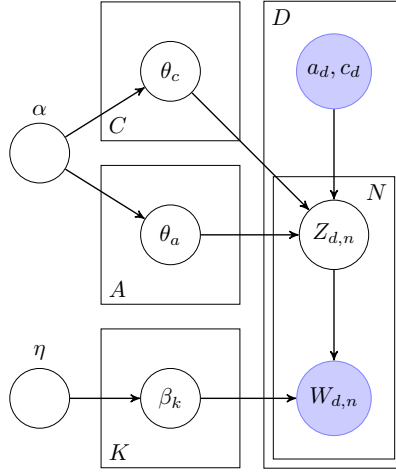


Figure 14: Plate notation for the author-category model.

Author-category model analysis: While modeling single metadata fields is the main focus of this paper, looking at the results of models combining multiple metadata may also bring new observations. For this purpose, we are examining the results of the Author-Category model. In Table XIX the metric results for this model are shown, and in Table XVIII, random samples of topics from the LDA, author-topic, category-topic, and author-category model can be seen. For the author-category model, the top words in the topics are mostly semantically incoherent, although the author-topic and category-topic models also have topics that are difficult to understand. It seems, to some degree, that the topics are a mix of the top words of the two combined metadata topic distributions, which makes sense since we draw word topics from the multiplication of these. While the topics may be less understandable, having a topic distribution for authors and categories gives more opportunities for, e.g., applying these in article recommendation, compared to only having one topic distribution.

K. The author-doc and category-doc models

Looking at the results in Table II, the author-topic and category-topic models do not get very high scores in topic coherence. In an attempt to improve these results, we combine the original LDA model with these models, adding a topic distribution for each document and combining it with the existing topic distributions, as with the author-category model, which is explained in Section J. Therefore, we create two extra models: the author-doc and category-doc model. We run them using the same hyperparameters as all the other models and they get similar results to the standard LDA model.

The function for choosing a topic is shown for both models in Equation 14 and Equation 15.

$$P(z_i = k | w_i = m, \mathbf{z}_{-i}, \mathbf{w}_{-i}) \propto \underbrace{\frac{C_{ak}^{AT} + \alpha}{\sum_{k'} C_{ak'}^{AT} + T\alpha}}_{\text{Author-Topic}} \underbrace{\frac{C_{dk}^{DT} + \alpha}{\sum_{k'} C_{dk'}^{DT} + T\alpha}}_{\text{Doc-Topic}} \underbrace{\frac{C_{mk}^{WT} + \eta}{\sum_{m'} C_{m'k}^{WT} + V\eta}}_{\text{Topic-Word}} \quad (14)$$

where C_{ak}^{AT} and C_{dk}^{DT} is the number of times author a and document d use topic k , respectively.

$$P(z_i = k | w_i = m, \mathbf{z}_{-i}, \mathbf{w}_{-i}) \propto \underbrace{\frac{C_{ck}^{CT} + \alpha}{\sum_{k'} C_{ck'}^{CT} + T\alpha}}_{\text{Category-Topic}} \underbrace{\frac{C_{dk}^{DT} + \alpha}{\sum_{k'} C_{dk'}^{DT} + T\alpha}}_{\text{Doc-Topic}} \underbrace{\frac{C_{mk}^{WT} + \eta}{\sum_{m'} C_{m'k}^{WT} + V\eta}}_{\text{Topic-Word}} \quad (15)$$

where C_{ck}^{CT} and C_{dk}^{DT} is the number of times category c and document d use topic k , respectively.

When looking at the topics that the two models produce, and comparing them, we can see some subtle differences, which might indicate the influence of the metadata. We have taken 3 topic pairs, which seem to be about the same topics, and compare them between the models. The first topic pair in Table XX is about tennis since some of the words are talking about Caroline Wozniacki, who is a professional Danish tennis player, and "turneringen" (the tournament). Specifically, "caroline" is not in the top 10 of the category-doc model which might indicate that the authors have written about "caroline wozniacki" before in other contexts. The word "slag" (hit) is also used within tennis, which might indicate that the category metadata helps bring sports words higher up in the ranks. Otherwise, there are no significant differences in the words they use.

Looking at the second pair of topics, which is about the EU and Great Britain, we can see that they are very similar. The 8th word for the author-topic model is "sagde" (said), which is not a very informative word regarding the topic, but an author might use these kinds of words many times during an article. Other than that, the topics are very similar when looking at the 10 most probable words.

Table XVIII: A sample of random topics' top 10 words, for the LDA, author-topic, category-topic, and author-category model. Each section in this table presents 15 random topics, where each topic is randomly picked from the model on the left and each line represents a topic.

Model	Topics
LDA	<p>millioner, direktør, sæby, seneste, tv, mener, stadig, landbrug, fokus, bedre stjalet, indbrud, klokken, nordjyllands, thisted, politi, politiet, mandag, villa, oplyser</p> <p>området, boliger, natur, naturen, ligger, du, vand, dyr, a, skov</p> <p>du, hans, ham, arige, mig, sagde, stedet, folk, liv, min</p> <p>millioner, procent, tv, selskabet, milliarder, skriver, største, aars, direktør, københavn</p> <p>thy, thisted, mors, unge, nielsen, arbejde, jensen, a, frederikshavn, folk</p> <p>området, kommunen, boliger, kr, projektet, byen, a, millioner, by, nyt</p> <p>arige, politiet, mand, arig, politi, retten, sagen, fængsel, ham, oplyser</p> <p>børn, du, børnene, hvordan, min, mor, forældre, livet, mennesker, skole</p> <p>biler, km, bilen, kr, hk, bil, t, a, vw, motor</p> <p>hobro, hadsund, mariager, morgen, mariagerfjord, bio, the, sker, kl, filmteatret</p> <p>mig, min, hans, liv, altid, du, mennesker, maske, verden, lille</p> <p>millioner, bank, sagen, nordjyske, penge, dansk, sag, skat, lars, sagde</p> <p>ebh, bank, finn, finansiel, kunst, lørdag, indbrud, vendsyssel, nordjylland, banks</p> <p>km, kr, hk, t, bilen, thisted, bil, biler, a, m</p>
Author-topic	<p>du, formand, tale, fem, kr, betyder, dermed, mal, arets, ligger</p> <p>jens, ford, vif, london, januar, team, problemerne, eh, vendsyssel, bla</p> <p>seneste, bjørne, gruppen, vendsyssel, erik, abent, middelboe, lavendel, nationalpark, motorvejen</p> <p>foie, karstensen, elin, bonderup, fredrik, derhjemme, hector, hee, kjøller, lillian</p> <p>skriver, hobro, sine, kommuner, dk, jammerbugt, set, min, mig, bedste</p> <p>jobi, tilværelse, crowdfunding, klippekortet, knude, nyby, thea, bpa, regi, judo</p> <p>du, sine, formand, seneste, jensen, hvert, nyt, hvordan, finde, kommunen</p> <p>gaido, fordele, albert, smed, forslag, fie, tørken, krævede, egon, tingene</p> <p>rundt, netop, gange, mig, gik, kr, større, landet, universitet, livet</p> <p>du, dansk, thisted, mig, procent, eu, ned, arbejde, hans, mener</p> <p>mig, millioner, skriver, ham, kommunen, hver, nordjylland, unge, sine, mand</p> <p>set, glas, odense, vesthimmerland, leth, markedet, trump, ni, regionerne, prins</p> <p>bælum, juul, udlændingenævnet, fruevej, vaarst, svitlana, jernstøberi, bislev, bannere, lo</p> <p>carl, resultat, poul, krabbe, p, ansat, begynde, holger, ledelse, g</p> <p>ned, procent, arige, eu, made, ham, først, større, mennesker, lyder</p>
Category-topic	<p>foregar, passer, imidlertid, yde, mængder, parlamentet, boris, henvendelser, white, berg</p> <p>yderste, sæsonen, lykkes, jernbaner, salgsprisen, efterladt, kakeeto, aab, frygt, rigtigt</p> <p>du, børn, mig, hans, unge, procent, mener, politiet, hvordan, thisted</p> <p>mener, langt, seneste, ting, mors, give, egen, hurtigt, seks, nej</p> <p>du, thisted, dansk, unge, mig, børn, a, hans, procent, arbejde</p> <p>jasmin, chelsea, rahbek, norden, partnerskabet, malstregen, eva, modernisering, festligt, byggefirmaet</p> <p>nordjyske, hjørring, giver, forhold, hobro, heller, mors, rundt, række, mulighed</p> <p>etiske, træningstilbud, lei, statuen, raab, torsdagscafe, aula, pattedyr, berømmelsen, ydet</p> <p>omtumlet, sydvendt, gla, dine, golde, trilogi, guidning, jungersen, areal, konservatorer</p> <p>nordjyske, sagde, plads, made, dette, fredag, området, heller, fald, byen</p> <p>sine, hver, skabe, juni, lars, tyskland, vendsyssel, michael, interesse, din</p> <p>min, jorden, dit, udfordring, thomas, datter, konkurrence, situation, museum, drive</p> <p>the, løbet, stjalet, regering, gaet, tredje, sikkert, byens, området, turen</p> <p>du, min, gode, gamle, ad, henrik, eu, finde, sat, hobro</p> <p>bedre, thy, haft, ham, hver, gik, synes, lars, millioner, eksempel</p>
Author-category	<p>ebh, klare, glas, kvaliteten, finansiel, rebecca, karrieren, storgaard, tørre, børnehave</p> <p>du, unge, sagde, samtidig, procent, bedste, hans, brønderslev, hvordan, kommende</p> <p>socialdemokratiske, placeres, ellemann, laustsen, fischer, san, regnskabsar, ordentligt, vejgaard, symptomer</p> <p>grønland, norden, morris, kenneth, logan, taliban, niki, robinson, tonnies, quinoa</p> <p>ganske, største, rød, tyrkiet, tilfældet, dybt, bo, f, fodbold, utrolig</p> <p>tror, klar, disse, handler, a, pedersen, holder, mors, borgmester, november</p> <p>lola, børnetallet, anklagemyndighed, lagring, madbar, daimler, fortænke, celtic, videnskabsfolk, lokalitet</p> <p>reducerede, forældrepar, utrygt, opgøres, pmi, judy, fusk, claude, matine, forsikrings</p> <p>egebjerg, overbelægning, gundersen, floden, karenbauer, involvere, imerco, udviklingsområder, hh, skygger</p> <p>drive, elektrificeres, bryggeriforeningen, fortjenstmedalje, brændstofpriser, sports, afmærket, ball, fanebærer, legal</p> <p>velfærdsstat, fortæl, nævneværdigt, solskin, adsbøl, børnesoldater, uvelkomne, reaktioner, daniel, messing</p> <p>benn, hadet, knortegas, højskolens, højreradikale, vietnamesere, sner, florence, partiprogram, puk</p> <p>du, børn, hans, unge, thisted, hvordan, arige, procent, mand, sagde</p> <p>frederiksen, skabt, tæt, halv, sociale, jammerbugt, jørgensen, norge, danskere, ligesom</p> <p>skarperer, venskab, landvind, china, motionsform, spørges, drøner, tværfaglige, islændinge, bevæget</p>

Table XIX: Results from the combination models: author-category, author-doc, and category-doc (marked with bold) compared to previous models.

Topic Model	Topic Coherence	Topic Difference
Latent Dirichlet allocation	0.520	0.575
Author-topic model	0.335	0.615
Category-topic model	0.370	0.560
Author-category model	0.390	0.537
Author-doc model	0.543	0.574
Category-doc model	0.530	0.575

Table XX: Top 10 words for similar topics within the extension models author-doc and category-doc. The topics have been manually selected.

Model pairs	1	2	3	4	5	6	7	8	9	10
Author-doc	wozniacki	vandt	open	sæt	turneringen	caroline	hobro	runde	nummer	arige
Category-doc	vandt	wozniacki	nummer	runde	open	sæt	turneringen	par	slag	dansk
Author-doc	eu	brexit	britiske	storbritannien	aftale	may	parlamentet	sagde	london	premierminister
Category-doc	eu	brexit	britiske	storbritannien	parlamentet	may	aftale	europa	london	johnson
Author-doc	natur	dyr	landbrug	naturen	landmænd	skov	hektar	vand	lille	danmarks
Category-doc	naturen	natur	du	dansk	hvordan	maske	området	landbrug	penge	kystsikring

The third pair of topics is about nature and agriculture. These two topics are not as similar as the other two pairs we have looked at, but they have two different viewpoints on this topic. The author-doc model describes words concerning agriculture since two of the words used are "landbrug" (agriculture) and "landmand" (farmer). It also mentions nature, with words such as: "natur" (nature), "dyr" (animals), "skov" (forest), and "vand" (water). The category-doc model describes nature as well but is more focused on areas within nature since the words "området" (the area) and "kystsikring" (coastal protection) are used. The model might focus more on the debate within the nature topic, which could be about coastal protection.

L. Applications of our models

There are a variety of different applications that topic modeling could be used for. Blei [3] describes many different purposes for topic modeling, like exploring the history of news articles over time. The LDA model, created by Blei et al. [6], has seen many extensions over the years to try and improve the generality of the model. Normally LDA works by inferring hidden topic structure, which is based on two distributions, namely the document-topic and topic-word distributions. An extension to the LDA is the author-topic model, created by Rosen-Zvi et al. [15], which creates a relationship between the author metadata and the corpus. This idea of taking metadata into account seems to be overlooked, even though a few papers have touched upon this area, the application possibilities for these kinds of metadata integrations are endless.

News media groups, like Nordjyske, are trying to find new ways of integrating smarter and more intelligent methods for keeping their customers, and using topic modeling could help improve their processes such as searching, recommendation, grouping of articles, and information completion. We give a brief overview of these to explain how topic modeling might play a role in improving these processes.

The first example is to improve search functionality for their articles. This problem could be alleviated by using the topics created by our topic models for performing query expansions finding similar words to the ones that appear in the query, providing more context to find results from. Once initial search results have been found, topic models could also be used to find other results with similar topics. These techniques can be particularly useful as they can help find articles that do not use any of the words in a given query but are still relevant to the query, which is a property that most basic search algorithms do not possess.

Another very important problem within news agencies is the recommendation of articles. Topic models can be used particularly for content-based filtering, finding other articles with similar topic distributions, either based on a user's preferences represented by an interaction history or based on a specific article being read.

Grouping items together, or clustering, can serve many different purposes. Topic modeling provides a new way of grouping together articles based on topic similarity.

A topic model might also be used to fill in missing metadata information in an article dataset. For example, with the taxonomy-topic model, we sample new taxonomy sequences for the majority of the dataset, which does not already have a taxonomy sequence.