Extending the LDA topic model

Nur Eser, Marcel Namyslo, Linus Ostermayer

Abstract—In an era where information is abundantly available, the pursuit of uncovering latent patterns and themes deeply embedded within vast textual datasets has become crucial across a wide range of fields. One effective approach to easily discern the content of articles or texts is through the process of categorization based on their underlying topics.
The categorization of articles can be achieved by leveraging the power of Machine Learning and Natural Language Processing (NLP). One central aprroach herefore are topic models.
Within the realms of statistics and natural language processing, topic modeling has emerged as a valuable statistical technique for discovering the abstract ”topics”.
One commonly utilized model in this context is known as Latent Dirichlet Allocation (LDA), which effectively identifies latent topics present within these extensive collections of text articles, along with other types of data.
LDA builds a model associating each document with a specific topic and each topic with a set of words, employing Dirichlet distributions as the underlying statistical framework to predict the overall topic.
The aim of this thesis is to show, how this LDA Model can be implemented and applied and furthermore, how it can be extendend and improved.

Index Terms—Computer Society, IEEE, IEEEtran, journal, LaTeX, paper, template.

CONTENTS

1 Introduction

The internet contains an almost incomprehensible amount of information from a wide range of sources. Because of this, many services exist that help users navigate the internet and find relevant content. Users are searching for content that matches their interests and come across websites that provide personalized recommendations based on their preferences.

Topic modeling is an essential tool for uncovering the latent thematic structure within a vast collection of textual data. Latent Drichlet Allocation(LDA), which is to be explained in depth in "Theory behind LDA" part is a popular algorithm. Just as the recommendation algorithm deciphers the user's interests, LDA analyzes through a sea of words, deciphering the relationships between them and revealing the underlying topics that permeate the documents. By understanding these topics, LDA empowers to organize, categorize, and recommend content in a more meaningful and personalized way. An illustration for this can be seen in Figure 1. The importance of algorithms like LDA is immense. They enable us to discover hidden patterns and themes within text data. Whether it's analyzing customer feedback, exploring research papers, or understanding social media conversations, LDA helps us distill vast amounts of information into coherent themes, providing valuable insights and knowledge. Moreover, LDA enables to compare and contrast different documents, identifying similarities and differences based on their topical content.

In summary, LDA topic models play a crucial role in understanding and recommending content. They extract hidden themes from textual data, enabling personalized recommendations, efficient information retrieval, and insightful analysis. Just as the recommendation algorithm guides us to relevant content, LDA guides us through the vast landscape of words, unveiling the underlying topics that shape our understanding of the written word.
In this paper, we aim to implement LDA topic model on 20 newsgroup dataset and then furher extend our model by revealing the topic correlations, doing auto-topic labelling and category-topic fitting. We also aim to see the effect of inclusion and exclusion of metadata.

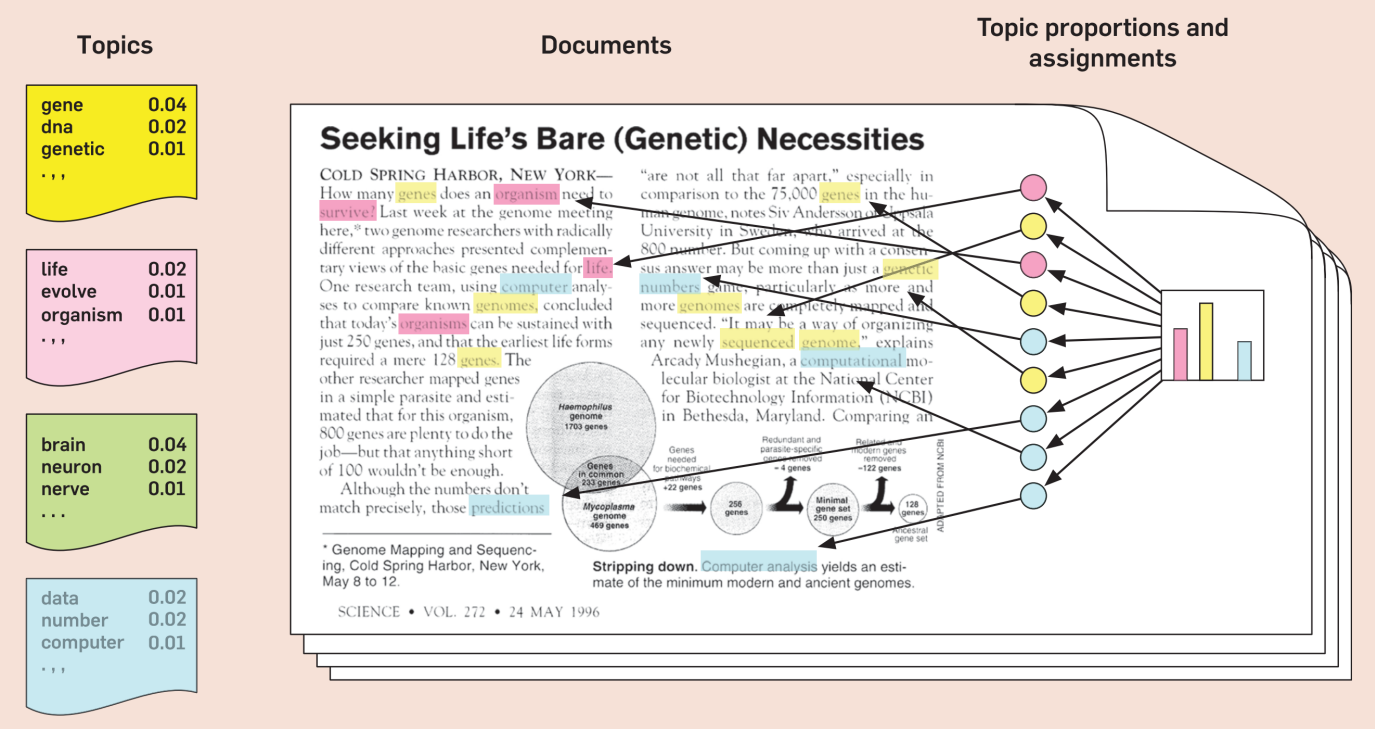


Fig. 1. LDA illustration

1.1 Theory behind LDA

1) Topic Model
There are two commonly used groups of unsupervised learning method to determine which topics are concerend within texts and documents. The first one is clustering. This model assigns documents to different clusters representing distinct topics, assuming each document belongs to only one cluster. The second one, topic modeling, on the other hand, uses a generative probabilistic model to describe the probabilities of documents belonging to each cluster, allowing for documents to contain a mixture of topics. Each topic in the model is in turn represented by a distribution of words, where the words that best describe the topic have higher probabilities.
In short: each document is a mixture of various topics and each topic is characterized by a distribution of words. The model represents documents as probabilistic distributions over topics and topics as distributions over words.
As clustering assigns always to only one cluster, it is refferd to as hard clustering, while topic modelling

is called as soft clustering.

2) Multinomial Distribution
To model the probabilities of words in documents, a commonly used representation is the multinomial distribution, which captures the relative frequency of words in a document. A common scenario used to describe the multinomial distribution is that of throwing a dice n times. Every time the dice is thrown, there is a 1/6 chance of obtaining each dice value. The probabilities can be represented in a 1 x 6 vector, where the sum of the probability values is 1.

3) Dirichlet Distribution
distributed data. It is named after the German mathematician, Peter Gustav Lejeune Dirichlet. Dirichlet processes in probability theory are "a family of stochastic processes whose realizations are probability distributions." This process is a distribution over distributions, meaning that each draw from a Dirichlet process is itself a distribution. What this implies is that a Dirichlet process is a probability distribution wherein the range of this distribution is itself a set of probability distributions
To show the relation between multinomial and dirichlet, the multinomial distribution gives the probabilities p_1 to p_K for K different events, e.g. how likely it is to roll a one, two, three, four, five or six in a roll. While in contrast, the Dirichlet distribution indicates how likely such a distribution is to occur.
In the case of topic modeling, the Dirichlet distribution is used to model the distribution of topics in a document or the distribution of words in a topic.

4) Latent Dirichlet Allocation
LDA is a generative probabilistic model that provides a statistical framework using all those statistical approaches explained before for discovering topics and their distribution across documents. It classifies or categorizes the text into a document and the words per topic, which are modeled based on the Dirichlet distributions and processes.
The LDA makes two key assumptions:

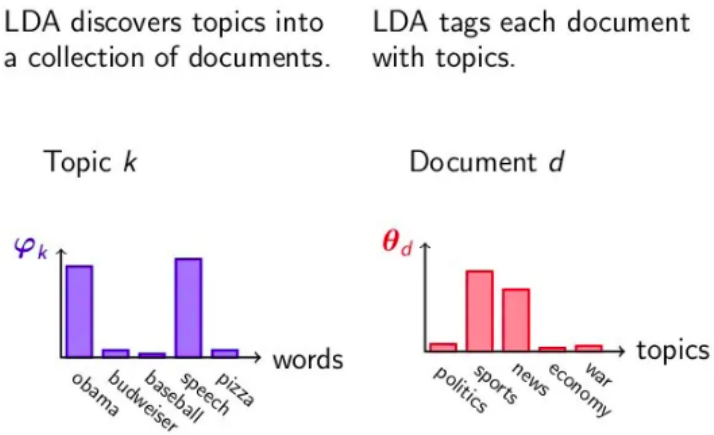
a) Documents are a mixture of topics
b) Topics are a mixture of tokens (or words)

Model Description
In literature, these three central terms are used to explain the LDA model more intuitive

- **Word**: The smallest unit of discrete data. The set of all unique words form a vocabulary.
- **Document**: A collection of words.
- **Corpus**: A collection of documents.

Additionaly, the LDA model makes these several central **assumptions**.
First, it assumes that documents are represented as bags of words, meaning the order of words is ignored, and only their frequencies or occurrences are considered.
Second, it assumes that topics are distributions over words, where each word has a certain probability of occurring in a given topic.
Third, it assumes that each document exhibits multiple topics and that the topic proportions vary across documents.
Next, the LDA model employs a Dirichlet prior to capture the distribution of topics within documents and the distribution of words within topics.

Process of the Algorithm At a high level, the Latent Dirichlet Allocation (LDA) algorithm works by iteratively estimating two main probability distributions: the topic-word distribution and the document-topic distribution. These distributions provide insights into the topics present in the corpus and the prevalence of topics in each document.



a) Topic-Word Distribution β:
The topic-word distribution represents the probability of each word given a particular topic. It is denoted by the symbol β. The algorithm aims to estimate this distribution by examining the words assigned to each topic across the corpus.

b) Document-Topic Distribution Θ:
The document-topic distribution represents the probability of each topic in a document. It is denoted by the symbol Θ. The algorithm also seeks to estimate this distribution by analyzing the topics assigned to each word in each document.

The LDA algorithm uses a generative process to model how the words in the corpus are generated:

a) Initialization:
   i) Specify the number of topics (K).
   ii) Randomly assign each word in each document to one of the K topics.
   iii) Initialize the topic-word and document-topic distributions.

b) Gibbs Sampling:
   i) Iterate through each document and each word within the document.
   ii) For each word, calculate the probability of assigning it to each topic, given the current topic-word and document-topic distributions. This probability is based on the words in the document and their assigned topics.
   iii) Sample a new topic for the word based on the calculated probabilities.

c) Update Distributions:
   i) After completing the Gibbs sampling step for all words in all documents, update the topic-word and document-topic distributions.
   ii) Update the topic-word distribution β by counting the number of times each word is assigned to each topic across the corpus.
   iii) Update the document-topic distribution Θ by counting the number of words assigned to each topic in each document.

d) Repeat:
   i) Repeat the Gibbs sampling and distribution update steps for a specified number of iterations or until convergence is achieved.
   ii) Convergence is typically determined by checking if the distributions have stabilized or if the log-likelihood of the corpus has reached a satisfactory level.

e) Output:
   i) Once the algorithm converges, the estimated topic-word β and document-topic β distributions can be examined.
   ii) The topic-word distribution provides the most probable words for each topic, and the document-topic distribution represents the prevalence of topics in each document.

To summarize, LDA iteratively estimates the topic-word and document-topic distributions by assigning topics to words and updating the distributions based on the generated data. Through this process, LDA uncovers the underlying topics in the corpus and the likelihood of topics appearing in different documents.

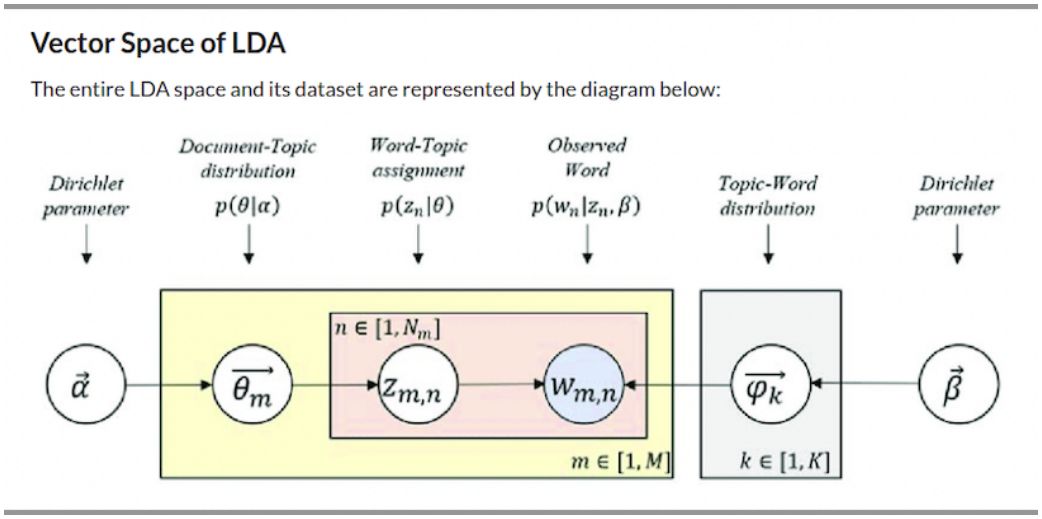


Fig. 2. Vector Space of LDA

2 Related Work

Topic modeling has been a widely studied and applied technique in the field of natural language processing and machine learning. Several approaches and variations of topic modeling have been proposed and explored in the literature. Blei et al. [6] introduced Latent Dirichlet Allocation (LDA), which is one of the most commonly used probabilistic generative models for topic modeling. They demonstrated its effectiveness in discovering latent topics from text corpora. Another popular topic modeling technique is Correlated Topic Model (CTM) proposed by Blei and Lafferty [3], which extends LDA by incorporating correlations between topics.

Various researchers have extended topic modeling techniques to address specific challenges and improve performance. For instance, Mimno et al. [7] introduced the Hierarchical Dirichlet Process (HDP) topic model, which allows the discovery of the underlying hierarchical structure of topics.

In summary, topic modeling has witnessed significant advancements and a wide range of techniques have been developed to address different aspects of topic extraction and document representation. Our study builds upon these prior works by implementing LDA and CTM models on the 20 Newsgroups dataset, providing insights into their performance and potential applications.

3 Problem Definition

The problem addressed in this study is to implement the Latent Dirichlet Allocation (LDA) topic model on the 20 Newsgroups dataset to uncover latent thematic structures within the text corpus. Additionally, we aim to explore the incorporation of additional latent structure or shared parameters within the LDA model to observe their impact on the topic modeling process.

3.1 Data

The 20 Newsgroups dataset is a collection of newsgroup documents that cover a wide range of topics, including politics, sports, technology, and more. The 20 categories included in the dataset are: m

1) alt.atheism
2) comp.graphics
3) comp.os.ms-windows.misc
4) comp.sys.ibm.pc.hardware
5) comp.sys.mac.hardware
6) comp.windows.x
7) misc.forsale
8) rec.autos
9) rec.motorcycles
10) rec.sport.baseball
11) rec.sport.hockey
12) sci.crypt
13) sci.electronics
14) sci.med
15) sci.space
16) soc.religion.christian
17) talk.politics.guns
18) talk.politics.mideast
19) talk.politics.misc
20) talk.religion.misc

The diversity of categories provides a rich and varied dataset for exploring and analyzing different thematic aspects present in the text corpus.

The downloaded 20 Newsgroups dataset consists of a collection of 20,000 documents where 11.314 are used for training. The documents in the training set is equally distributed among 20 categories where each category makes up approximately 5% of the overall documents. Each document represents a forum post or an email from one of the 20 newsgroups. The length of the documents can vary significantly, ranging from a few sentences to several paragraphs. The data is in plain text format and follows a specific structure as follows:

1) Category Label: Each document is assigned a category label, indicating the newsgroup to which it belongs. The category labels correspond to the 20 different topics mentioned earlier.

2) Header Information: Each document starts with header information that includes metadata such as the subject, author, and date of the post or email. This header provides additional context about the document but may not be relevant for topic modeling purposes.

3) Body Text: Following the header, the actual content of the document is present in the body text. This is the main textual data that we will be working with for our analysis.

4 Proposed Solution

In this section, we present our proposed solution for topic modeling on the 20 Newsgroups dataset. We also describe the preprocessing steps we applied to the dataset. For the proposed solution, we consider three approaches to model the text corpus: Latent Dirichlet Allocation (LDA), Structural Topic Model (STM)/hierarchical Latent Dirichlet Allocation (hLDA) and Correlated Topic Model (CTM).

LDA is a widely used probabilistic topic modeling technique that assumes documents are generated from a mixture of latent topics, and each topic is represented by a probability distribution over words. LDA learns the underlying topics in a corpus and their associated word distributions, enabling the identification of the main themes or topics within the text data. It uses a generative process to assign topics to words in documents and estimate the topic-word and document-topic distributions. LDA utilizes variational Bayesian inference for estimating the latent variables and model parameters. Variational Bayesian methods approximate the true posterior distribution with a simpler distribution by optimizing a lower bound on the log-likelihood of the data.

Hierarchical LDA additionally includes weighted document-level metadata, such as author, publication and sentiment to optimize the topic document matching. Since it has no impact on the topic topic relations, it can be applied for CTM as well.

On the other hand, CTM is an extension of LDA that introduces correlations between topics. In LDA, topics are assumed to be independent of each other, whereas CTM relaxes this assumption and allows for dependencies between topics. CTM models the topic-word distribution as a logistic normal distribution, which captures the correlations between topics more effectively. CTM employs Gibbs sampling for inference. Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method that generates samples from the posterior distribution by iteratively sampling values for each variable conditioned on the values of other variables. In CTM, Gibbs sampling is used to estimate the topic assignments and model parameters, including the correlations between topics. However, Gibbs sampling showed to be computationally more expensive.

By considering both LDA and CTM, we aim to compare the performance and interpretability of the two approaches on the given text corpus. LDA provides a baseline topic modeling technique that assumes independence between topics, while CTM explores the possibility of capturing correlations between topics. This allows us to evaluate the impact of incorporating topic correlations on the modeling results and gain insights into the effectiveness of these approaches for our specific dataset. We used the same preprocessed data for better comparability

1) Preprocessing steps:
    a) Removal/Extraction of metadata:
    The 20 Newsgroups dataset contains metadata associated with each document, such as email headers and posting information. Since our focus is on topic modeling based on the textual content, we removed the metadata from the dataset to isolate the text corpus. While metadata like email addresses or server hosters may be biasing and leading to an overfitting model some might helpful creating a acccurate word topic distribution. So we're seperating the metadata from the actual textcorpus and extracting the features:
        - subject
        - keywords
        - organization

        if available, the documents have variing meta-information. Now seperated and independently treatable, the metadata get preprocessed the same way to provide a unified database.
    b) Text tokenization
    Text tokenization is the process of splitting a piece of text into smaller units, called tokens, to facilitate further analysis. We used Natural Language Toolkit (NLTK) library in Python. It is a simple and widely used tokenization function that splits a piece of text into individual words based on whitespace and punctuation. We performed additional transformations on the tokens obtained through tokenization. Firstly, we converted all tokens to lowercase to ensure case-insensitive matching and to avoid treating the same word differently based on its capitalization. Secondly, we filtered out tokens that were less than three letters long since they are often considered less meaningful in the context of language analysis. Only keywords and organizations are allowed to contain two letter tokens to keep abbreviations. We also excluded tokens that contained numbers. By removing numeric tokens, we aimed to eliminate numerical values that might not provide substantial semantic information in our analysis.
    c) Stopword removal:
    To improve the quality of the topics extracted by the LDA model, we performed stopwords removal. We leveraged the stopwords available in the nltk library, which provides a set of commonly used English stopwords. These stopwords include frequently occurring words such as ”the,” ”is,” and ”and,” which do not carry much topical information. By removing these stopwords, we aimed to reduce noise and improve the topic interpretability.
    d) Custom stopword identification:
    In addition to the default stopwords provided by nltk, we conducted a manual examination of the dataset to identify domain-specific stopwords that were not adequately captured by the default English stopwords. Initially, we trained the LDA model without the inclusion of custom stopwords. However, analyzing the results, we observed the occurrence of certain words that did not carry any significant topical meaning. These words were manually identified and deemed irrelevant for our research objectives. As a result, we decided to remove these words manually in order to enhance the quality and interpretability of the LDA topic modeling outcomes on the 20 Newsgroups dataset.
    e) Statistical Stopword removal
    Additionally in later gridsearch with multiple parameter we used another way of removing words that probable lack of topic informations. By removing words that occur only in a hand full of documents or more than a certain ratio of documents can considered as filling/commonly used word or seem to be an identifier or misspelled word. Though these words may be even topic identifying (very specific like names or scientific methods) but either way irrelevant for an statistical method and not generalized capturale.
    f) Stemming
    To treat different forms of words as similar, since there is no difference iin relevance for a topic, those forms have to be mapped to one unified form. One way to achive this

is stemming it to its base form, removing suffixes. This may result in word parts and ambiguous stemms. For example the words “chocolates”, “chocolatey” and “chocolate” are reduced to the root word “choco”.

g) Lemmatization
Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item, similar as stemming, but in contrast to stemming it brings context to the words and reduces words with similar meanings to one base form, such as plurals or verb tenses, to their common form. For example, ”running” becomes ”run” and ”better” turns to ”good”. Since lemmatization does morphological analysis of the words it’s not only a preprocessing step but can be considered as first step to word topic association. To perform well and transform words in the right way lemmatizer need information about the words type. Here we used a bayesian part of speech (PoS) tagger.

2) Training:
LDA is a parametric model, which implies that it has a fixed number of parameters. Specifically, the dimensionality of the vector representing the Dirichlet prior probability for the topic distribution in a document with k topics is k-dimensional. Determining the optimal number of topics in (LDA) poses a challenge as there is no universally accepted method. In our study, 20 Newsgroups dataset is a collection of documents classified into 20 distinct categories, and defining the k parameter as 20 is the general approach. However, we made a deliberate decision to reduce the number of topics from 20 to 15. This reduction was based on the observation that some of the categories exhibited significant overlap in terms of their content and could be considered as similar or closely related. For example, categories such as ”comp.windows.x,” ”comp.os.ms-windows.misc,” and ”comp.sys.ibm.pc.hardware” all revolve around computer systems, with a focus on different aspects such as operating systems and hardware. By treating these similar categories as a single topic, we aimed to consolidate the thematic information and achieve more coherent and distinct topics in the topic modeling process. We used the ”LatentDirichletAllocation” class from the “sklearn.decomposition” module present in the scikit-learn library.
To handle the tokenized documents they need to be vectorized. We tried two different Vectorizer, first a simple count vectorizer that counts the occurence of every token in every document and second the Tfidf Vectorizer that also weights the occurence of a token in comparision to the size of a document and measures the rarity or uniqueness of a term across the entire corpus. It helps to downweight the importance of frequently occurring terms and boost the importance of less common terms.
To obtain good parameters, especially the statistical stopword removing we applied some gridsearch exemplary on the lemmatized documents.
Training the LDA hierachical, we added to the documents token the available metadata weighted accordingly and then trained with these axtended tokens.
For the Correlated Topic Modelling, we used “CTModel” provided by “tomotopy” package. Unlike our LDA model, CTM was to be trained iteratively. Another main difference for the CTM model is that we decided the number of topics to be 20, in order to examine whether the model would catch similar categories by assigning them higher correlations. At each step we printed out the log-likelihood of the words and manually decided on the convergence of the model.

3) Evaluation:
To evaluate the LDA model we both tried manual and statistical approaches. On the one hand by looking on the first few words describing a topic we could determine if those are really fitting to each other and describing a common topic. For evaluating the topic document relation, we calculated the overall distribution over the allocated topics for all documents of each category and associated the most likely topic to the category.
In order to visualize the topic correlation information contained in our CTM model, we used a heat map. Each cell in the heat map represents the correlation value between two topics. By looking at the heat map, we can easily identify which topics are more closely related to each other and which topics have less correlation. Warmer colors like red or orange indicate a stronger positive correlation, and cooler colors like blue or green indicate a weaker or negative correlation.

5 RESULTS

5.1 LDA

We're now first comparing the results of of the basic LDA model with different stages of preprocessed data

Topic 0	jesus	would	christ	people	time
Topic 1	file	internet	anonymous	bill	information
Topic 2	people	would	think	know	like
Topic 3	writes	article	would	could	think
Topic 4	drive	disk	scsi	anyone	know
Topic 5	chip	encryption	keys	clipper	number
Topic 6	year	team	game	good	games
Topic 7	period	writes	power	keith	play
Topic 8	writes	article	iuic	scsi	would
Topic 9	space	nasa	research	health	center
Topic 10	jews	article	israel	writes	israeli
Topic 11	file	entry	output	program	jpeg
Topic 12	would	people	writes	article	could
Topic 13	year	health	disease	medical	time
Topic 14	launch	space	scsi	engine	satellite

Fig. 3. LDA Model results before preprocessing

Topic 0	book	point	time	first	find	word	bank
Topic 1	drive	chip	file	disk	system	encryption	output
Topic 2	jesus	church	christian	bible	christ	john	faith
Topic 3	please	mail	send	list	offer	price	email
Topic 4	year	good	time	right	well	people	much
Topic 5	game	team	play	season	hockey	player	year
Topic 6	space	information	system	data	nasa	internet	public
Topic 7	file	window	image	program	version	application	display
Topic 8	server	comp	graphics	object	also	problem	work
Topic 9	people	thing	even	believe	question	many	mean
Topic 10	armenian	people	israel	turkish	israeli	government	arab
Topic 11	card	problem	driver	work	anyone	time	monitor
Topic 12	president	state	people	government	right	stephanopoulos	time
Topic 13	year	health	disease	medical	time	drug	patient
Topic 14	launch	space	scsi	engine	satellite	year	power

Fig. 4. LDA Model results after data processing

```
alt.atheism: Topic #11 (god, believe, jesus, christian, say, one, bible, church, religion, true)
comp.graphics: Topic #13 (file, program, window, image, version, available, include, server, code, application)
comp.os.ms-windows.misc: Topic #8 (drive, card, disk, system, problem, work, bit, driver, scsi, one)
comp.sys.ibm.pc.hardware: Topic #8 (drive, card, disk, system, problem, work, bit, driver, scsi, one)
comp.sys.mac.hardware: Topic #8 (drive, card, disk, system, problem, work, bit, driver, scsi, one)
comp.windows.x: Topic #13 (file, program, window, image, version, available, include, server, code, application)
misc.forsale: Topic #5 (new, price, sell, offer, sale, include, pay, ship, condition, clinton)
rec.autos: Topic #2 (one, time, well, see, say, take, thing, look, come, back)
rec.motorcycles: Topic #2 (one, time, well, see, say, take, thing, look, come, back)
rec.sport.baseball: Topic #9 (game, team, play, year, win, player, season, league, hockey, first)
rec.sport.hockey: Topic #9 (game, team, play, year, win, player, season, league, hockey, first)
sci.crypt: Topic #14 (key, car, chip, security, phone, company, year, one, call, week)
sci.electronics: Topic #2 (one, time, well, see, say, take, thing, look, come, back)
sci.med: Topic #2 (one, time, well, see, say, take, thing, look, come, back)
sci.space: Topic #2 (one, time, well, see, say, take, thing, look, come, back)
soc.religion.christian: Topic #11 (god, believe, jesus, christian, say, one, bible, church, religion, true)
talk.politics.guns: Topic #2 (one, time, well, see, say, take, thing, look, come, back)
talk.politics.mideast: Topic #4 (people, right, kill, child, state, one, live, war, woman, country)
talk.politics.misc: Topic #4 (people, right, kill, child, state, one, live, war, woman, country)
talk.religion.misc: Topic #11 (god, believe, jesus, christian, say, one, bible, church, religion, true)
```

Fig. 5. category-topic fitting

When matching the categories with the most likely topic, we can see that some are pretty accurate and similar categories also match to the same topics:

- comp.graphics, comp.windows.x:
  file, program, window, image, version, available, include, server, code, application
- comp.os.ms-windows.misc,
  comp.sys.ibm.pc.hardware, comp.sys.mac.hardware:
  drive, card, disk, system, problem, work, bit, driver, scsi, one
- alt.atheism, soc.religion.christian, talk.religion.misc:
  god, believe, jesus, christian, say, one, bible, church, religion, true
- talk.politics.misc, talk.politics.mideast:
  people, right, kill, child, state, one, live, war, woman, country
- misc.forsale:
  new, price, sell, offer, sale, include, pay, ship, condition, clinton
- rec.sport.hockey, rec.sport.baseball:
  game, team, play, year, win, player, season, league, hockey, first

Still depending on the exact parameter there were some "general" topics catching most of the other categories:

- rec.autos, rec.motorcycles, sci.electronics, sci.med, sci.space & talk.politics.guns
  all matched to:
  one, time, well, see, say, take, thing, look, come, back

These outcomes indicate that there coud be done more and accurate stopword removing

5.2 Vectorizer

When we look on the Tf-idf Vectorizer we see some really bad results of very generous topic-word distributions consisting mainly non-topic related words. Of course this results in associating all categories with only 4 (most generous) topics. In combination with the sometimes inaccurate word distributions of the count vectorizer it promts to investigate even more effort in more adjusted preprocessing, focusing on stopword removal and focusing on non-ambiguous words.

```
alt.atheism: Topic #0
comp.graphics: Topic #1
comp.os.ms-windows.misc: Topic #1
comp.sys.ibm.pc.hardware: Topic #1
comp.sys.mac.hardware: Topic #1
comp.windows.x: Topic #1
misc.forsale: Topic #1
rec.autos: Topic #0
rec.motorcycles: Topic #0
rec.sport.baseball: Topic #0
rec.sport.hockey: Topic #2
sci.crypt: Topic #0
sci.electronics: Topic #1
sci.med: Topic #0
sci.space: Topic #0
soc.religion.christian: Topic #0
talk.politics.guns: Topic #0
talk.politics.mideast: Topic #0
talk.politics.misc: Topic #0
talk.religion.misc: Topic #0
```

Fig. 6. category-topic fitting with TF-IdF Vectorizer

5.3 hLDA

Applying weighted metadata to the documents leads to different, shifted results.
The computer top category ist now correctly divided into graphics, hardware related and windows related:

- comp.graphics:
  (list, program, mail, image, information, available, include, graphic, send, file)
- comp.os.ms-windows.misc, comp.windows.x:
  (window, help, need, file, want, info, please, look, change, windows)
- comp.sys.ibm.pc.hardware, comp.sys.mac.hardware:

(drive, card, problem, driver, system, mac, disk, video, color, bit)

And also the religion and politics related categories are brought together:

- talk.religion.misc, soc.religion.christian:
  (god, christian, jesus, bible, church, atheist, believe, hell, religion, life)
- talk.politics.mideast, talk.politics.misc:
  (armenian, people, government, state, israel, president, child, kill, clinton, israeli)

But other categories on the other hand got worse and mixed up:

- rec.autos, rec.motorcycles, rec.sport.baseball, sci.electronics:
  (time, car, one, year, look, back, take, well, come, go)
- sci.med, alt.atheism, talk.politics.guns:
  (one, people, say, thing, see, well, many, way, even, something)

Or stayed unaltered:

- rec.sport.hockey:
  (team, scsi, win, nhl, hockey, play, game, ide, year, season)
- misc.forsale:
  (sale, new, game, player, political, baseball, price, league, offer, sell)
- sci.crypt:
  (key, chip, clipper, encryption, security, secret, algorithm, house, escrow, white)
- sci.space:
  (question, space, faq, science, answer, system, part, nasa, launch, resource)

5.4 Correlated Topic Model

topic 0	game	team	player	play	season
topic 1	many	number	people	also	thing
topic 2	jesus	first	also	font	message
topic 3	russian	study	child	health	general
topic 4	april	period	science	division	field
topic 5	armenian	file	image	book	turkish
topic 6	space	national	list	technology	launch
topic 7	encryption	security	public	control	house
topic 8	word	many	state	must	example
topic 9	right	people	state	government	israel
topic 10	drive	system	data	disk	chip
topic 11	file	window	program	application	version
topic 12	people	world	group	government	member
topic 13	people	christian	believe	evidence	bible
topic 14	question	take	time	case	many
topic 15	year	first	last	second	time
topic 16	problem	work	system	help	using
topic 17	anyone	need	please	used	want
topic 18	thing	much	well	really	something
topic 19	good	also	come	sure	give

Fig. 7. CTM results

We further extended the model by applying auto-topic labelling. Auto-topic labelling module was also provided by tomotopy, but the choice of parameters were crucial. After evaluating, many parameter combinations, we reached meaningful labelling for some topics. Here are the labels we extracted for chosen topics. The topic enumeration in this part differs from the CTM model results represented in previous part. We represent here the results for randomly chosen 3 topics.

Labels for topic 11:	words:	Likelihood of the word to the topic:
persecution	christian	0.024953113868832588
enemy	israel	0.016869815066456795
holy	church	0.01543207373470068
nation	bible	0.015208425000309944
buried	faith	0.01290803961455822
religious	israeli	0.012716340832412243
	christ	0.0117258969694376
	religion	0.01003255695104599
	woman	0.01003255695104599
	attack	0.009617209434509277

Fig. 8. Auto-Labeling results for topic 11, along with the words of the topic

Labels for topic 3:	words:	Likelihood of the word to the topic:
international	armenian	0.04371361434459686
republic	turkish	0.01884584128856659
march	anonymous	0.013233180157840252
document	service	0.013151243329048157
cambridge	national	0.012086066417396069
region	greek	0.011799288913607597
	university	0.011717352084815502
	information	0.011020890437066555
	russian	0.010201523080468178
	turkey	0.009382156655192375

Fig. 9. Auto-Labeling results for topic 3, along with the words of the topic

Labels for topic 8:	words:	Likelihood of the word to the topic:
representative	people	0.024152597412467003
february	state	0.015195026993751526
congressional	government	0.01433244626969099
united	right	0.010318126529455185
united state	general	0.00962142739444971
meeting	example	0.009555074386298656
	world	0.0087588457390666
	history	0.008261202834546566
	public	0.008062145672738552
	firearm	0.007929440587759018

Fig. 10. Auto-Labeling results for topic 8, along with the words of the topic

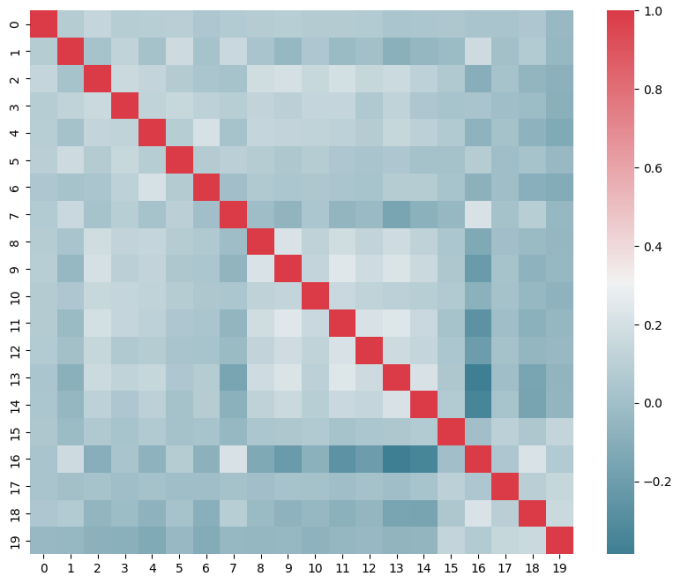


Fig. 11. Topic Correlation according to CTM

6 Discussion

6.1 Interpratation of the results for LDA model

Comparing the results of the final model [4] with the categories in our dataset, we can associate each topic with its corresponding category. For example, topic 14 is about space, topic 2 religion, topic 9 atheism, topic 14 med, and so on. The obtained results are clear and coherent.
Comparing fig[4] with fig[3] we can observe how our preprocessing steps enhanced our model.
Ecluding custom stop words results in more significant word distribution over topics. Especially the statistical stopword removal eliminated plenty of filling words and resulted in much more accurate word-topic distributions in concern of well fitting but also seperating from other topics.
Category-topic fitting successfully matched each topic with its corresponding category as can be seen in fig[5].

6.2 hLDA

The use of hierarchical LDA helped to differ between close topics and categorize based on nuances, but also mixed up topics that weren't that close or mainly contained unspecific words. The main problem here was, that the metadata varied strongly and those inconsistent results may are caused by this - categories with (good) metadata got gradually seperated while those lacking of metadata got biased and mixed up due to a smaller database. To evaluate the hierachical LDA better, further tests on a more consistand text corpus are needed.

6.3 Interpratation of the results for CTM model

CTM model although slightly different from the LDA model, as expected, yielded acceptable results as well. Referring to our heatmap fig[6] in order to draw inferences about topic correlations, we observe that warmest shade is white, suggesting that any two of our topics are not found to be highly correlated. However, we can also make use of this map to reveal the topics that are negatively correlated. Topic 16 and 13 are found to be highly uncorrelated, which upon investigation makes sense since topic 13 is about religion whereas topic 16 is related to information systems.fig[5].
Auto-topic labelling yielded meaningful labels for some topics but needs improvement as it can not fully capture meaningful labels for every topic.

7 Threads to Validity

We need to take into account a few things that could have an impact on the validity of our findings in our study. First off, because the dataset only covers a limited range of categories and is limited to a specific timespan and focuses on certain countries, citeies and organizations, it could not be completely generalizable to other text corpora and may be biased/overfitted to certain names (organizations, cites). Additionally, the preprocessing techniques we used were specific to our dataset , such as exclusion of custom stop words. Biases may be introduced and had an impact on the model's performance and interpretability depending on the selection of hyperparameters, evaluation metrics, and preprocessing techniques used. Finally, the complexity of the models may have been constrained by the computational resources at our disposal.

8 Conclusion

We implemented the basic LDA model on our tiny text corpus , 20 newsgroup dataset. We applied many preprocessing stages in line with our data. We further extended this model by incorporating topic correlations and visualizing the results. In addition, we implemented auto-topic labelling and category-topic fitting. In our research, choosing an already categorized dataset enabled us to efficiently compare our final findings with the supposedly one. We adopted various approches in extending our model and we did not only enhanced our initial topic modelling algorithm but also showed and implemented many ways for evaluating and visualizing our results.

9 Future Directions

In this paper, we have demonstrated the application of topic modeling using the 20 Newsgroups dataset which is a tiny text corpus. One area of future work is to apply our approach to larger text corpora. The dataset we use, although representative, is relatively small compared to real-world text collections. However, scaling up to a larger corpus has its own challenges considering computational resources and time. Training-times were 1 and 5 minutes for our LDA model and CTM model respectively. As the corpus gets larger, parallel processing could be implemented at either documentation level or word level to distribute the workload.

Our work mainly focused on topic modeling and how it can be extended. In that regard, we used auto-topic labeling only as a subpart in our main work. However, another avenue for future work could be the improvement of automatic topic labeling. This could be reached by incorporating external knowledge, utilizing word embeddings and considering topic coherences.

We used variational Bayes inference for our LDA model and Gibbs sampling inference for our CTM model. Thus, although we have used the same preprocessed dataset, the differences in results may be better observed if same approach was taken. However, since our regard in implementing CTM model was to observe topic coherences, differences in inference methods did not affect our objective.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
[2] David M. Blei , "Probabilistic Topic Models"
[3] David M. Blei & John D. Lafferty. "Correlated Topic Models"
[4] Blei, D., Griffiths, T., Jordan, M. "The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies."
[5] C. Bishop, D. Spiegelhalter, and J. Winn. "A variational inference engine for Bayesian networks."
[6] David M. Blei, M. Jordan. "Latent Dirichlet allocation." Journal of Machine Learning Research
[7] Jennifer Lindgren, "Evaluating Hierarchical LDA Topic Models for Article Categorization"