Comparison of neural networks with generative probabilistic topic models in natural language processing based on different metrics - How to decide which topic model is superior to another?

Florian Eric Klawun

Technische Universität Berlin

14th August 2021

Institute of Software Engineering and Theoretical Computer Science

Machine Learning Group

Prof. Dr. Klaus-Robert Müller

klaus-robert.mueller@tu-berlin.de

Marchstr. 23

10587 Berlin

Acknowledgments

Firstly, I wish to thank my supervisor David Lassner for his mentoring. Without his constant support and our frequent discussions, this thesis would have never been completed despite its unusually long editing time.

I am also especially grateful to Jonas Behrendt for introducing me to the topic of data mining as well as natural language processing.

Last but not least, I am much in debt to Alyssa Larison for reading my thesis and providing useful comments and redactorial suggestions.

Abstract

Using two variants of two real-world corpora, a total of five different metrics is tested for each of five different instantiations of two topic models, the LDA and the iDeepDNe, to finally decide which topic model is superior across all metrics. The corpora used are, on the one hand, the 20 newsgroups known from historical NLP publications and a self-mined dataset, the "20subs". Different variants are created by changing the vocabulary size. On each variant, both models are tested with five different topic numbers. The results show that although the LDA is a well-performing topic model according to intrinsic metrics, the iDeepNADEe more than outperforms the LDA according to extrinsic metrics.

Keywords: perplexity, topic coherence, document retrieval, document classification, LDA, iDeepDNe, 20 newsgroups

Zusammenfassung

Anhand von zwei Varianten zweier real existierender Korpora werden insgesamt fünf verschiedene Metriken für jede der fünf verschiedenen Instanziierungen zweier Themenmodelle, der LDA und dem iDeepDNe, getestet, um schließlich zu entscheiden, welches Themenmodell über alle Metriken hinweg dem anderen überlegen ist. Die verwendeten Korpora sind zum einen die aus historischen NLP-Publikationen bekannten 20 Newsgroups und ein selbst erstellter Datensatz, die "20subs". Durch Veränderung der Vokabulargröße werden verschiedene Varianten erstellt. Für jede Variante werden beide

Modelle mit fünf verschiedenen Themenanzahlen getestet. Die Ergebnisse zeigen, dass, obwohl die LDA ein gut funktionierendes Themenmodell nach intrinsischen Metriken ist, der iDeepNADEe die LDA nach extrinsischen Metriken mehr als übertrifft.

Schlüsselwörter: perplexity, topic coherence, document retrieval, document classification, LDA, iDeepDNe, 20 newsgroups

Glossary

Abbreviation	Meaning	Page
NLP	Natural language processing	8
LSA	Latent Semantic Analysis	12
SVD	Singular Value Decomposition	13
PLSA	Probabilistic Latent Semantic Analysis	14
EM	Expectation Maximization algorithm	16
LDA	Latent Dirichlet Allocation	17
MCMC	Markov Chain Monte Carlo	20
tf-idf	Term frequency - inverse document frequency	20
20NG	20 newsgroups dataset	22,24, 31
NPMI	Normalized pointwise mutual information (coherence)	24
NMI	Normalized mutual information (coherence)	24
PMI	Pointwise mutual information (coherence)	24
iDocNADEe	Document Informed Neural Autoregressive Topic Models with Distributional Prior	25
Document NADE	Document Neural Autoregressive Topic Models	25f
SGD	Stochastic Gradient Descent	26
iDocNADE	Document Informed Neural Autoregressive Topic Models	26
DeepDocNADE	Deep Document Neural Autoregressive Topic Models	27
DocNADEe	Document Neural Autoregressive Topic Models with Distributional Prior	28
iDeepDNe	Deep Document Informed Neural Autoregressive Topic	29f

	Models with Distributional Prior	
ppl	Perplexity	24,36
c_v	Topic coherence metric with indirect cosine measure with the NPMI and the boolean sliding window.	27,38f
top n topics words	n most probable words from a topic	40
IR	Document Retrieval experiment that yields precision-scores	24,43
Precision	Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.	43
Recall	Recall is the ratio of correctly predicted positive observations to the total observations in the actual class.	45
F1	F1 scores are yielded from document classification experiments and computed as the weighted average of precision and recall.	24,45
acc	Accuracy is the ratio of correctly predicted observations to the total observations.	45

Introduction

The Internet is the largest networking project that humanity has ever undertaken in the entirety of its history. However, the flow of information and the exchange of data has taken on such huge proportions that no single individual is able to process the information that accumulates on a daily basis. To help people cope with this flood of information and to make the human processing effort manageable there is an increasing number of machine learning algorithms that filter the information or tailor it to presumed individual priorities. Most of these machine learning algorithms, which are used for example in spam detection, chatbots, virtual assistants, text translation, or text summarization, originate from the computer science branch of natural language processing (NLP). Within this field, the basic aim is to enable computers to process human language and to enable them to interact with humans by combining machine learning methods with rule-based modeling of human language.

A particular challenge here for many applications is the desire for the software to go beyond the statistical patterns and mathematical rules of human language to acquire a kind of understanding of the content the information is expressing. In particular, however, inferring more global contexts of meaning and relationship is a tricky endeavor because it is not clear how these processes occur in humans themselves. The identification of the subject matter when reading a news article or listening to a piece of information seems to happen quite intuitively and when different people read the same news article they are very likely to identify similar but different subjects as a result.

Additionally, the word "topic" itself is very vague, exemplified by the Oxford Learner's Dictionaries definition as "a subject that you talk, write or learn about" where subject means "a thing or person that is being discussed, described, or dealt with"[1]. In this sense, any attempt to exemplify the thematic context of information through words is, to some extent, subject to the subjectivity of the assessor, due to a lack of a universal and definitive way of deciding whether the thematic context of the information at hand is indeed fully covered by the proposed topics.

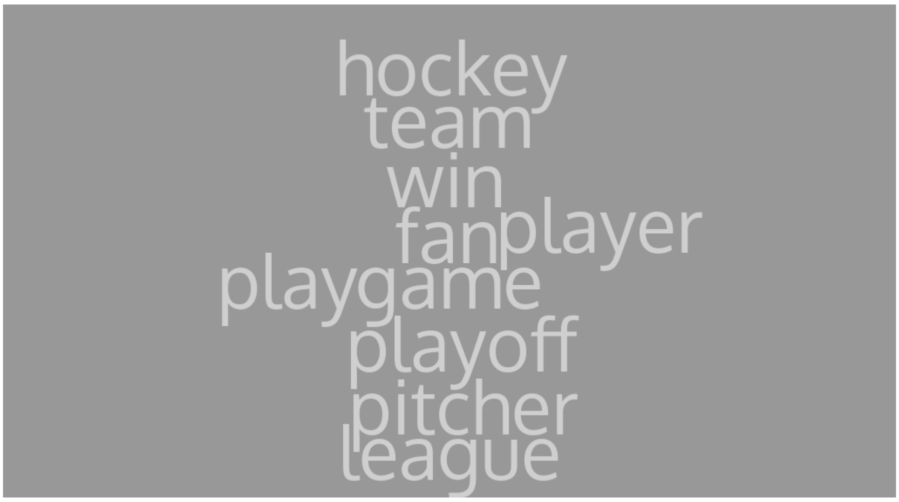


Figure 1: Word cloud of 10 terms that could be contextually or semantically linked through the topic word “hockey”, “sport” or even “noun”

Source: own creation

For example, the words from Figure 1 can be thematically grouped in multiple ways or placed in a semantically logical context. If the task were to identify from this word cloud the one word that is most representative of the thematic context of all the words in the word cloud, the choice would probably be "hockey". If, on the other hand, the task were to select from all the words in the dictionary the one that best captures the topic of this word cloud,

[1] https://www.oxfordlearnersdictionaries.com/definition/english/topic?q=topic && https://www.oxfordlearnersdictionaries.com/definition/english/subject_1?q=subject

there would be many possible answers that would solve this task in one way or another. In short, when it comes to grouping information thematically in the form of words, there is no definitive correct answer.

Transferred into the context of machine learning, however, this means that there can be no supervised algorithm that can infer topics of information. This is due to the reality in which no dataset can have unique and definitive topic labels from which the algorithm could learn. More so, any topics identified by unsupervised approaches do not necessarily have to correspond to those arising from human judgment. Last but not least, it remains difficult to compare the results of different algorithms, as without a unifying framework there is no basis to weigh the goodness, integrity, or quality of the identified topics against each other.

This thesis investigates to what extent machine learning methods can be used to identify topics in textually presented information. These approaches are summarized by Chang et al. (2009) under the keyword probabilistic topic models, which

> "*are a popular tool for the unsupervised analysis of text, providing both a predictive model of future text and a latent topic representation of the corpus. Practitioners typically assume that the latent space is semantically meaningful. It is used to check models, summarize the corpus, and guide exploration of its contents. [...] These models posit a set of latent topics, multinomial distributions over words, and assume that each document can be described as a mixture of these topics. With algorithms for fast approximate posterior inference, we can use topic models to discover both*

> *the topics and an assignment of topics to documents from a collection of documents*."[2]

The above quote, in essence, describes how NLP topic models assume that each text document, however big or small, consists of a mixture of topics which themselves are mixtures of word collections. The semantics of textual information is thus governed by hidden variables, the topics, which cannot be observed at first hand. The purpose of topic modeling is therefore to uncover these hidden variables through means of machine learning.

The paper is organized as follows. The first section describes and explains a selection of topic models which were proposed in the last thirty years. The second section contributes to this endeavor by applying two of these, LDA and iDeepDNe, on two versions of two real-world text corpora in order to generate probabilistic topic models. To assess which approach identifies better topics, the results are compared on the basis of five evaluation metrics in order to be able to draw a conclusion about which metric is most suitable for describing the quality of a topic model. By means of an experiment, the knowledge shall be gained which topic model is superior to the other.

Related work

Previous undertakings in topic modeling have mostly been based on the generative rebuilding of the underlying language model from a collection of documents. This is usually done by uncovering statistical relationships within sentences and documents, attempting to capture latent semantics in order to both group words thematically and assign text documents to one or more topics consisting of statistically related word clouds.

[2] Chang et al. (2009), p. 1.

Latent Semantic Analysis

The foundation for the probabilistic topic model was probably laid by Deerwester et al. (1990) with their work on Latent Semantic Analysis (LSA), which was intended to simplify the automatic indexing and retrieval of text documents based on their semantic structure or topic:

> *"It is designed to overcome a fundamental problem that plagues existing retrieval techniques that try to match words of queries with words of documents. The problem is that users want to retrieve on the basis of conceptual content, and individual words provide unreliable evidence about the conceptual topic or meaning of a document. There are usually many ways to express a given concept, so the literal terms in a user's query may not match those of a relevant document. In addition, most words have multiple meanings, so terms in a user's query will literally match terms in documents that are not of interest to the user."*[3]

They achieved this by treating their collection of documents as a term-document-matrix X. In this structure, each row of X represents one term from the entire collection's vocabulary and every column of X one document. Each entry in X would then be the term frequency which counts the occurrence of the specific term in the specific document. Their idea was then to assume that, in all likelihood, X would be sparse, noisy, and redundant across its many dimensions. To capture the hidden semantics governing the thematic relationships among all

[3] Deerwester et al. (1990), p. 1.

documents, they would perform a singular value decomposition (SVD) so that the semantic relationships could be interpreted through their proximity in the reduced semantic space.

The SVD of any rectangular matrix $X \in R^{txd}$ can be achieved by factoring it into the product of three other matrices; $X = T\ S\ D'$ where $T \in R^{txm}$ and $D' \in R^{mxd}$ have orthonormal, unit-length columns. $S \in R_{+}^{mxm}$ is the diagonal matrix of singular values which are by convention all positive and ordered in decreasing magnitude, t the number of rows of X, d the number of columns of X, and m the rank of X.[4] This SVD can be approximated by using a reduced model, where m is replaced by a hyperparameter k<m, i.e. keeping only the first k columns of T and S, the first k rows of D' and discarding the rest.

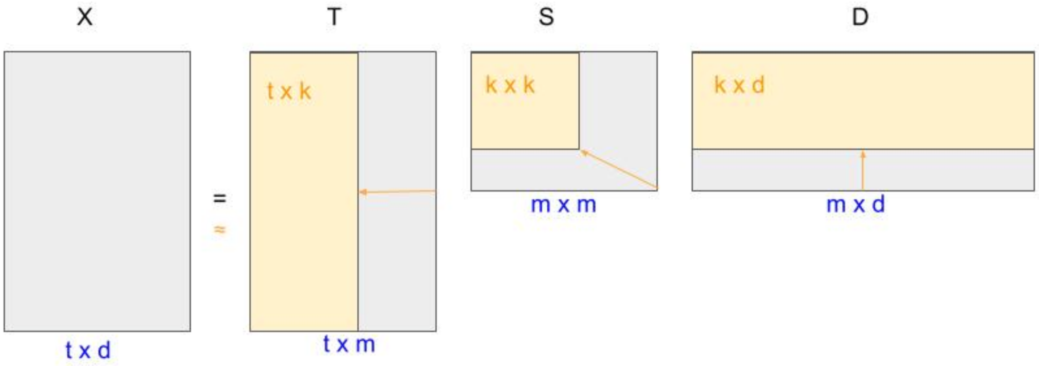


Figure 2: SVD of matrix X with dimensionality reduction using k<m

Source: based on Deerwester et al. (1990), p. 12f

This approximation yields the topic-document matrix $D' \in {}^{kxd}$ where each row represents one of k topic assignments to all documents and the term-topic matrix $T \in {}^{txk}$ where each row represents a term assignment to all k latent topics. With these vectors, Deerwester et al. (1990) were then able to compute similarities of different documents or

[4] cf. Deerwester et al. (1990), p. 11f.

search queries and hence able to improve information retrieval tasks. Shortcomings, however, were that there was no method to securely determine the best value for the hyperparameter k, i.e. the perfect amount of latent topics, or that different terms could yield different significance when determining a word. For example, the word *hockey* yields usually more information about a topic than the word *is*. Thus, a weight for the values in X would have been necessary to capture meaningful semantics. Also, by representing every word as just a single value in this semantic space, one cannot fully incorporate the meaning of polysems, words with many possible meanings. Last but not least, even though SVD is a well understood and researched method from linear algebra, the quantitative evaluation for its performance in document retrieval remains unclear and could only be evaluated through a precision and recall metric that tested its document retrieval applicability in comparison to other information retrieval methods, but does not give insights into the quality of the identified latent topics.

Probabilistic Latent Semantic Analysis

Eleven years after Deerwester et al put forth their ideas on LSA, Hofmann (2001) proposed an extension and modification of the method to fix some of their shortcomings:

> *"Probabilistic Latent Semantics Analysis (PLSA) stems from a statistical view of LSA. In contrast to standard LSA, PLSA defines a proper generative data model. This has several advantages: On the most general level it implies that standard techniques from statistics can be applied for model fitting, model selection and complexity control. For example, one can assess the quality of a PLSA model by measuring its predictive performance, e.g., with the help of*

> *cross-validation. More specifically, PLSA associates a latent context variable with each word occurrence, which explicitly accounts for polysemy."*[5]

His general idea, which was based on the *aspect model* proposed by Hofmann, Puzicha & Jordan (1999), is to assume that the collection of documents is $D = \{d_1, d_2,...,d_N\}$ where each d_i consists of words from the vocabulary $W = \{w_1,w_2,...,w_M\}$ and that the co-occurrences of the words within each document are governed by the latent topic variable $z_k \in \{z_1,...z_K\}$. Instead of attempting to approximate the latent topic space through dimensionality reduction like LSA, Hofmann (2001) introduced a probabilistic model which could generate the observable data. He defined the following probabilities:

- $P(d_i)$: probability that a word occurence will be observed in a particular document d_i
- $P(w_j \mid z_k)$: class-conditional probability of a specific word conditioned on the unobserved topic variable z_k
- $P(z_k \mid d_i)$: document-specific probability distribution over the latent variable space.[6]

Starting with a document, its words could then be generated through the model of Figure 3 which yields (d_i , w_j) as an observation pair.

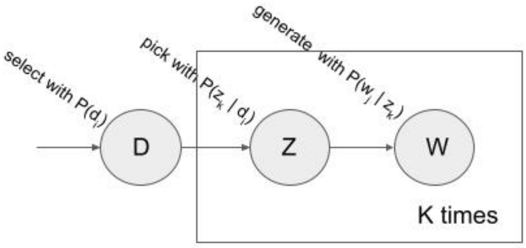


Figure 3: asymmetric generative process for PLSA

Source: based on Hofmann (2001), p. 5

[5] Hofmann (2001), p.2.

[6] cf. Hofmann (2001), p.4.

Essentially this process can be expressed as a joint probability model where

$$P(d_i, w_j) \; = \; P(d_i)\, P(w_j|d_i) \; with \; P(w_j|d_i) \; = \; \sum_{k=1}^{K} P(w_j|z_k)\, P(z_k|d_i).$$ [7]

To compute this generative scheme, he used the Expectation Maximization (EM) algorithm as a maximum likelihood estimation for the latent variables, in which the parameters would be iteratively estimated through computing the posterior probability of the latent topics. In a nutshell, EM involves iteratively applying two steps until either a convergence or early stopping condition is met. The first step (E) uses Bayes' formula to compute the posterior probability based on the current state of the parameters d_i and w_j

$$P(z_k|\, d_i, w_j) \; = \frac{P(w_j|z_k)\, P(z_k|d_i)}{\sum_{l=1}^{K} P(w_j|z_l)\, P(z_l|d_i)}$$

and the second step (M) updates the parameters to a new state according to the log-likelihood of the newly computed posterior probabilities

$$P(w_j|z_k) \; = \frac{\sum_{i=1}^{N} n(d_i, w_j)\, P(z_k|d_i, w_j)}{\sum_{m=1}^{M} \sum_{i=1}^{N} n(d_i, w_m)\, P(z_k|d_i, w_m)} \quad \text{and} \quad P(z_k|d_i) \; = \frac{\sum_{j=1}^{M} n(d_i|w_j)\, P(z_k|d_i, w_j)}{n(d_i)}$$

where $n(d_i,w_j)$ denotes the number of occurrences of w_j in d_i.[8]

By adding this probabilistic approach to topics and words on top of the general idea of LSA, Hofmann (2001) was able to introduce an unsupervised machine learning algorithm with a solid statistical background that would be superior to LSA evaluated in terms of perplexity as well as automated document indexing.[9] Yet, PLSA also suffered from shortcomings, most notably from being keen to overfit the data as the amount of parameters

[7] cf. Hofmann (2001), p.4.

[8] cf. ibid., p.6.

[9] cf. ibid, p.19.

grows linearly with the number of documents as well as lacking a method of assigning probabilities to unseen documents as $P(d_i)$ is just given and cannot be modeled according to another parameter.[10]

Latent Dirichlet Allocation

The first “real” probabilistic generative topic model was most likely introduced by Blei, Ng and Jordan (2003) with the Latent Dirichlet Allocation (LDA) that would solve the remaining challenges of the PLSA. The notation remains very similar as a corpus remains to be a collection of M documents vectors consisting of N words that are represented as mixtures over the K latent topics that are characterized by a distribution over the terms from the entire vocabulary of the corpus. The novelty of their approach lies in the usage of Dirichlet priors to sample the probability distributions for on one hand the topic distribution of a particular document and on the other hand the word distribution of a particular topic. More formally, the topics $\varphi_{1:K}$ each are a distribution over the fixed vocabulary sampled from a Dirichlet distribution with the parameter β, the specific topic distribution θ_d for document d_i is sampled from a Dirichlet distribution with the parameter α and the actual topic assignment for the *n*th word in the *d*th document is denoted as $z_{d,n.}$. The generative process of LDA can then be summarized as in Figure 4 or formally expressed as the following joint distribution:

$$P(\varphi_{1:K}, \theta_{1:M}, z_{1:M}, w_{1:M}) = \prod_{i=1}^{K} p(\varphi_i) \prod_{m=1}^{M} P(\theta_m) (\prod_{n=1}^{N} P(z_{m,n}|\theta_m) P(w_{m,n}|\varphi_{1:K}, z_{m,n}))$$

[11]

[10] cf. Blei, Ng and Jordan (2003), p. 2.

[11] cf. Blei (2012). p. 4.

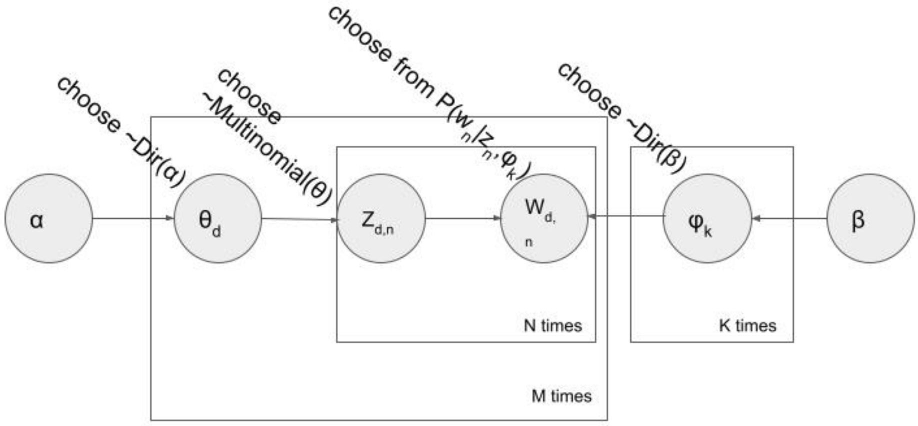


Figure 4: generative process for LDA

Source: based on Blei (2012), p. 5

As Blei, Ng and Jordan (2003) have shown, this joint distribution can be rephrased by integration over θ, summing over z, and finally taking the product of the marginal probabilities of a single document:

$$P(D|\alpha,\beta) = \prod_{d=1}^{M} \int P(\theta_d|\alpha)(\prod_{n=1}^{N_d} \sum_{z_{dn}} P(z_{dn}|\theta_d)\, P(w_{dn}|z_{dn},\beta))\, d\theta$$

This yields a generative model for the corpus that is only dependent on the hyperparameter α and β.[12] Ideally, one would apply this model to a corpus to first compute the latent multinomial variables as the topics of that corpus and second to assign new and unseen documents a mixture of the topics they contain depending on their semantic similarity. However, computing the conditional topic distribution from the training corpus is intractable in practice as this posterior

$$P(\varphi_{1:K}, \theta_{1:M}, z_{1:M}|w_{1:M}, \alpha, \beta) = \frac{P(\varphi_{1:K}, \theta_{1:M}, z_{1:M}, w_{1:M}|\alpha,\beta)}{P(w_{1:M}|\alpha,\beta)}$$

[12] cf. Blei, Ng and Jordan (2003), p. 5.

contains the marginal probability of the corpus which can only be calculated through summing all joint distributions over all the latent multinomial variables, i.e. all assignments of each word to every topic, that could possibly be instantiated.[13] Since the amount of possible assignments grows exponentially with vocabulary and topic size, this calculation consumes big amounts of resources exponentially fast. Hence, Blei, Ng and Jordan (2003) and other researchers over the course of the last two decades have proposed a plethora of approximation algorithms that could be considered to avoid this resource-heavy computation. As Blei (2012) wrote:

> *"Topic modeling algorithms form an approximation of [the posterior] by adapting an alternative distribution over the latent topic structure to be close to the true posterior. Topic modeling algorithms generally fall into two categories—sampling-based algorithms and variational algorithms. Sampling-based algorithms attempt to collect samples from the posterior to approximate it with an empirical distribution. [...] Variational methods are a deterministic alternative to sampling-based algorithms.[...] Rather than approximating the posterior with samples, variational methods posit a parameterized family of distributions over the hidden structure and then find the member of that family that is closest to the posterior."*[14]

Similar to PLSA, the EM algorithm could be used as a special case of variational inference where one assumes that the variational distributions are point estimations. In the E-step θ and φ would be used to compute the topic assignment of each word, and in the M-step θ and φ

[13] cf. Blei (2012), p. 5.
[14] ibid., p 5f.

would be updated according to the optimization of the log-likelihood based on the topic assignment of each word.

Another way of approximating this posterior is through collapsed Gibbs sampling which is a Markov Chain Monte Carlo (MCMC) algorithm that constructs a Markov chain that after a number of iterations should converge to a good approximation of $P(z_i \mid \mathbf{z_{-i}}, \alpha, \beta)$ where $\mathbf{z_{-i}}$ are all topic assignments without z_i. Without going into too much detail, this would be done by setting up two counting variables $n_{d,k}$ for the number of words assigned to topic k in document d as well as $n_{k,w}$ for the number of times word w is assigned to a topic k. Gibbs sampling then involves randomly initializing **z** and looping through the corpus for a number of iterations. Every iteration involves sampling one topic for each word in the corpus, updating the counting variables accordingly, and using them to compute the distributions of θ_d and φ_k. With these, the discrete distribution of each topic assignment $p(z=k| \cdot)$ could be calculated and afterward sampled from to update **z** for the next iteration.[15]

Blei, Ng and Jordan (2003) reported that the LDA with the variational EM procedure outperforms PLSA in terms of perplexity, document classification, and collaborative filtering when updating the raw term frequency value from the PLSA matrix to a *term frequency-inverse document frequency (tf-idf)* $w_{i,j} = tf_{i,j} \, log(\frac{N}{df_j})$ where $tf_{i,j}$ denotes the occurrences of a term w_j in a document d_i, N the total number of documents and df_j the number of documents that contain the word w_j. Tf-idf incorporates that terms that frequently appear in one document but not across the entire corpus gain extra significance and weight in comparison to those that appear frequently throughout all documents and the entire corpus.

[15] cf. Darling (2011), p. 4ff.

The current state of research

Since LDA was first proposed, the research has made extraordinary leaps fuelled by both grander theoretical insights into the mathematical processing of human language, as well as more potent computer hardware in handling large amounts of data and general advances in machine learning and neural networks. Fundamental improvements in topic modeling include, for example, better ways of vectorizing texts. For simplicity and computationability, many topic models relied on unigram text processing in which there is an assumption that each word is independent of its previous or following word. N-Gram models, however, try to ease this unrealistic assumption by processing n-word-sequences as their terms.[16] Another advance is the continued development of word embedding methods. Since topic models perform calculations on vectors, one must first transform the n-gram strings to numerical vectors. Whereas LSA and PLSA used the term frequency and LDA the tf-idf, novel approaches have gone beyond individual corpus statistics and instead tried to vectorize in a linguistically or semantically meaningful way.[17]

These developments as well as the continued research in novel machine learning algorithms have led to an increasing number of diversified topic models. Note that, due to the limited scope of this paper, it is intractable to describe all advances since the first proposal of LDA. Currently, however, there are at least three completely different topic models which could claim to be state-of-the-art - or at least could claim to be superior to LDA in some ways. The hierarchical Stochastic Blockmodel by Gerlach, Peixoto and Altmann (2018) treats corpora as bipartite networks of documents and vocabulary terms and then utilizes

[16] cf. Jurafsky and Martin (2020), Chapter 3, p. 24.
[17] cf. Mandelbaum and Shalev (2016), p.2.

community-detection to identify topic structures.[18] Though it is not discussed further nor is the approach from Sia, Dalmia and Mielke (2020) who used common unsupervised clustering algorithms like k-Means on top of novel pre-trained word embeddings like BERT applied to a specific corpus to cluster topics.[19] Instead, this paper will investigate the latest proposal from Gupta et al (2019b) and substantiate whether their proposed deep neural network approach is actually substantially superior to LDA. Validating which of these three state-of-the-art approaches is superior to another is left to future work.

Table 1 documents a selection of recently proposed topic models. Not surprisingly, all authors claimed that their proposed variant would be a superior model in comparison to some baseline models in terms of some evaluation metrics. Yet, the wide range of evaluation metrics shows that there is no single “gold standard” when it comes to judging the performance of these unsupervised topic modeling algorithms or the quality of their generated topic structures. Almost all authors use, however, perplexity, document retrieval, and some form of topic coherence. The variety of used datasets for their experiments also suggests that there is no single benchmark against which the models could be measured, although almost all authors use the 20 newsgroups (20NG) dataset. Interestingly, however, the views of the various authors differ on the question of what is the optimal or correct number of topics for the 20NG. While some probably chose k=20 intuitively because of the name of the dataset, the range goes up to k=200.

The widest range of test datasets and evaluation metrics is clearly used by Gupta et al (2019b), most likely to substantiate the superiority of their iDeepDNe.

[18] cf. Gerlach, Peixoto and Altmann (2018).
[19] cf. Sia, Dalmia and Mielke (2020).

Authors	Paper	Year	Proposed topic model	Evaluation metrics	Datasets	Number of topics for 20NG	Compared against
Teh et al.	Hierarchical Dirichlet Processes	2005	Hierarchical Dirichlet Process (HDP)	ppl	Nematode biology abstracts, NIPS sections, Alice in Wonderland	-	LDA
Salakhutdinov and Hinton	Replicated Softmax: an Undirected Topic Model	2009	Replicated Softmax (RSM) with Annealed Importance Sampling (AIS)	ppl, IR	20NG, NIPS proceeding papers, Reuters Corpus Volume I	50, 200	LDA
Larochelle and Lauly	A Neural Autoregressive Topic Model	2012	Neural Autoregressive Distribution Estimator (Document NADE)	ppl, IR	20NG and Reuters Corpus Volume 1 version 2	50, 200	LDA(only ppl), RSM
Das, Zaheer, Dyer	Gaussian LDA for Topic Models with Word Embeddings	2015	Gaussian LDA	PMI	20NG, NIPS	50	LDA
Nguyen et al	Improving Topic Models with Latent Feature Word Representations	2015	Glove-LDA	NPMI, purity, NMI, qualitative inspection, F1	Six datasets (20NG and two smaller subsets of 20NG, TMN, TMNtitle, Twitter Messages)	20	LDA
Lauly et al	Document Neural Autoregressive Distribution Estimation	2017	Deep Document NADE (DeepDocNADE)	ppl, IR, qualitative Inspection	20NG and Reuters Corpus Volume 1 version 2	50	LDA, RSM, Document NADE
Gerlach, Peixoto and Altmann	A network approach to topic models	2018	Hierarchical Stochastic Block Model (TopSBM)	Minimum Description Length	Five real Corpora (Twitter samples, Reuters-21578, Web of Science, NYT, PlosOne) and two synthetically	-	LDA

					generated from the generative process of LDA		
Gupta et al	textTOvec: Deep Contextualized Neural Autoregressive Topic Models of Language with Distributed Compositional Prior	2019a	ctx-DocNADEe	ppl, IR, F1, NPMI, c_v, qualitative inspection	Eight real datasets (e.g. 20NG and a smaller subset of 20NG)	20, 200	Variants of DocNADE, glove-LDA (only IR and F1) Gauss-LDA(only IR and F1), LDA(only NPMI)
Gupta et al	Document Informed Neural Autoregressive Topic Models with Distributional Prior	2019b	iDocNADEe / iDeepDNe	ppl, IR, F1, NPMI, c_v, qualitative inspection	15 real datasets (e.g. 20NG and a smaller subset of 20NG)	20, 200	Variants of DocNADE
Sia, Dalmia and Mielke	Tired of Topic Models? Clusters of Pre-trained Word Embeddings Make for Fast and Good Topics too!	2020	Cluster-algorithm (k-Means or Gaussian Mixture Models) on top of word embeddings (Word2vec, ELMo, GloVe,Fasttext, Spherical, BERT)	NPMI	Reuters-21578 and 20NG	20	LDA

Table 1: selection of recently proposed topic models as well as their evaluation metrics (20NG = 20 newsgroups, ppl = perplexity, IR = document retrieval, F1 = document classification, c_v = topic coherence, NPMI = normalized pointwise mutual information, NMI = normalized mutual information, PMI = pointwise mutual information)

Source: own creation

Document Informed Neural Autoregressive Topic Models with Distributional Prior

The most recent variant proposed by Gupta et al (2019b) builds on the foundations of Lauly et al (2017) to extend their DeepDocNADE with approaches of bidirectional language models and word embedding priors. Gupta et al's (2019b) model, the Document Informed Neural Autoregressive Topic Models with Distributional Prior (iDocNADEe),

> "*is a probabilistic graphical model that learns topics over sequences of words, corresponding to a language model [...] that can be interpreted as a neural network with several parallel hidden layers [which in order t]o predict the word v_i, [...] [take] [...] the sequence of preceding words $\mathbf{v}_{<i}$ [as well as the sequence of succeeding words $\mathbf{v}_{>i}$] [as input] [...] [while] incorporat[ing] word embeddings as fixed prior [...] in order to introduce complementary information. The proposed neural architectures learn task specific word vectors in association with static embedding priors.*"[20]

Assume that $p(d_i)$ is the joint distribution of a single document $\mathbf{d}_j$= $[w_1,...,w_N]$ of all words w_v with $w_v \in \{1,...,V\}$ being the index of the *v*th word in the vocabulary of size V from the corpus D = $[d_1,...d_M]$. Given the convention of $\mathbf{d}_{<i} \in \{w_1,...,w_{i-1}\}$, g(•) being a non-linear activation function, H being the number of topics as well as the number of hidden units in the neural network, $W \in R^{HxV}$ and $U \in R^{VxH}$ being weight matrices and $\mathbf{c} \in R^H$, $\mathbf{b} \in R^V$ being bias parameter vectors, Document NADE models this joint distribution by decomposing it as a product of the conditional distributions $p(w_i|\mathbf{d}_{<i})$ with $i \in \{1,...,N\}$ and computing these autoregressive conditionals via a feed-forward neural network.

In the original Document NADE model, these autoregressive conditionals were computed with a full-softmax through

$$P(w_i = y \mid d_{<i}) = \frac{exp(b_y + U_{y,:} h_i(d_{<i}))}{\sum_{y'} exp(b_{y'} + U_{y',:} h_i(d_{<i}))}$$

[20] Gupta et al (2019b), p. 1f.

In use with

$h_i(d_{<i}) = g(c + \sum_{v<i} W_{:,w_v})$ as a position-dependant hidden layer[21]. Additionally, these conditional distributions could be used to compute the log-likelihood of any document $log\ P(d_j) = \sum_{i=1}^{N} log\ P(w_i|d_{<i})$ for inference, and, using stochastic gradient descent (SGD), the parameter **b, c, W** and **U** could be learned by minimizing the average negative log-likelihood of the training documents.[22] Finally, the hidden units from the network can be summed to get the representation of a new and unseen document afterward.[23] Figure 5 illustrates this modeling approach.

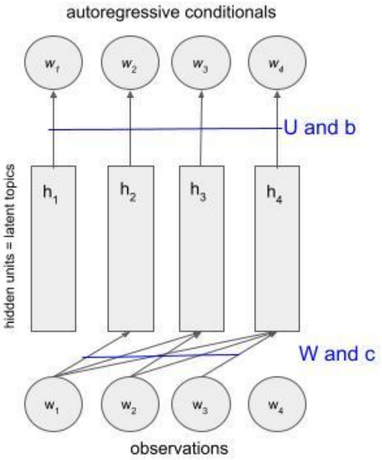


Figure 5: Illustration of Document NADE model with observations $\mathbf{w} = [w_1,w_2,w_3,w_4]$ from a document d_j using four hidden units $\mathbf{h} = [h_1,h_2,h_3,h_4]$ to compute four autoregressive conditionals $p(w_i|\mathbf{d}_{<i})$ with $i \in \{1,2,3,4\}$, the shared parameters $\mathbf{W} \in R^{4x4}$ and $\mathbf{U} \in R^{4x4}$ as well as the bias vectors $\mathbf{c} \in R^4$ and $\mathbf{b} \in R^4$

Source: based on Larochelle and Lauly (2012), p. 2

[21] cf. Larochelle and Lauly (2012), p. 4.

[22] cf. ibid., p. 3.

[23] cf. ibid., p. 5.

As Lauly et al. (2017) suggested, this model can be extended to a deep, multiple hidden layer neural network where the first hidden layer is computed as in the single hidden layer architecture and every subsequent hidden layer as

$$h_i^{(n)}(d_{<i}) = g(c^{(n)} + W^{(n)} \cdot h^{(n-1)}(d_{<i}))$$

with n = 2,...,Ń where Ń is the total number of hidden layers in the DeepDocNADE.[24] The autoregressive conditional for any word would then be computed from the last hidden layer in the known way using either a tree softmax or full softmax with the U as an output parameter and b as a bias vector $P(w_i = y \,|\, d_{<i}) = softmax(U \cdot h^{(Ń)}(d_{<i}) + b)$.[25]

The first addition to this model by Gupta et al. (2019b), to incorporate bi-directional language modeling, is illustrated by Figure 6. To account for bi-directional contextual information (**d**$_{>i}$ and **d**$_{<i}$), they introduced parallel backward layers

$$\hat{h}_i^{(1)}(d_{>i}) = g(\hat{c} + \sum_{v>i} W_{:,w_v})$$

where $\hat{c} \in$ R^{H} would be a new bias vector for the backward layer passes. They also introduced a second autoregressive conditional

$$P(w_i = y \,|\, d_{>i}) = softmax(U \cdot \hat{h}_i^{(Ń)}(d_{>i}) + \hat{b})$$

where $\hat{b} \in$ R^{V} would be a new bias vector for the backward layer passes. The updated log-likelihood function which could be used for inference and consequently to compute the loss function for SGD would then be

[24] cf. Lauly et al. (2017), p. 9.
[25] cf. ibid., p 10.

$log\, P(d_j) = \frac{1}{2} \sum_{i=1}^{N} log\, P(w_i \mid d_{<i}) + log\, P(w_i \mid d_{>i})$. An illustration for this addition for an one hidden layer architecture, iDocNADE, is shown in Figure 6.

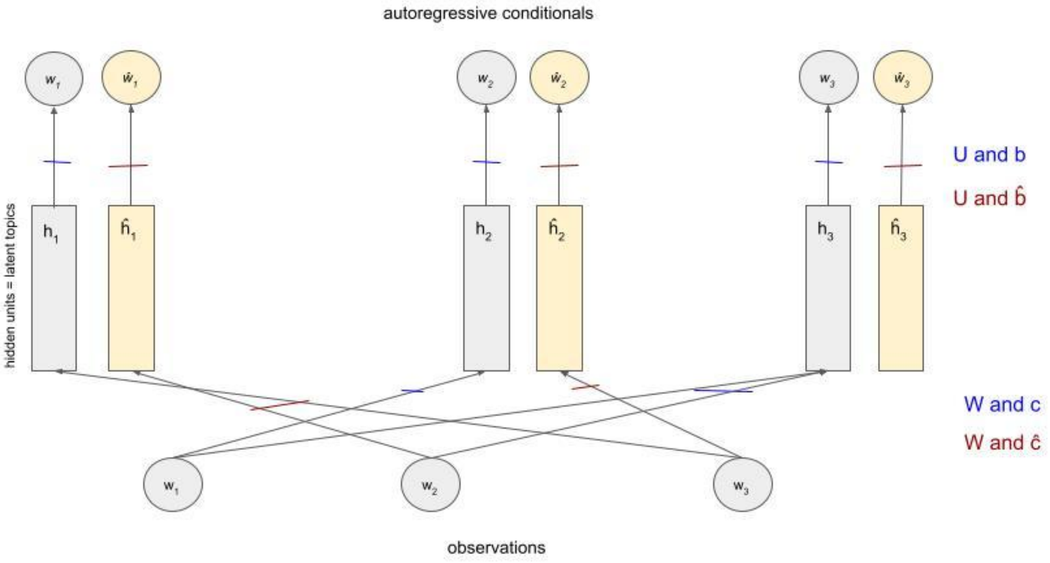


Figure 6: Illustration of iDocNADE model with observations $\mathbf{w} = [w_1,w_2,w_3]$ from a document d_j using hidden units $\mathbf{h} = [h_1,h_2,h_3]$ to compute in total six autoregressive conditionals $p(w_i|\mathbf{d}_{<i})$ and $p(w_i|\mathbf{d}_{>i})$ with i ∈ {1,2,3}, the shared parameters $\mathbf{W} \in R^{3x3}$ and $\mathbf{U} \in R^{3x3}$ as well as the bias vectors $\mathbf{c},\hat{\mathbf{c}} \in R^3$ and $\mathbf{b},\hat{\mathbf{b}} \in R^3$

Source: based on Gupta et al.(2019b), p. 2

Lastly, Gupta et al. (2019b) proposed to further enhance these models by introducing pre-trained word embedding aggregation at each autoregressive step at the first hidden layer. Given a pre-trained word embedding matrix E ∈ R^{HxV}, the position-dependant hidden layers for each word w_i would be computed through

$$\hat{h}_i^{(1)}(d_{>i}) = g(\hat{c} + \sum_{v>i} W_{:,w_v} + \lambda \sum_{v>i} E_{:,w_v})$$

and respectively

$h_i^{(1)}(d_{<i}) = g(c + \sum_{v<i} W_{:,w_v} + \lambda \sum_{v<i} E_{:,w_v})$ where λ denotes a mixture coefficient parameter[26]. Figure 7 illustrates this addition for a single hidden layer architecture, DocNADEe.

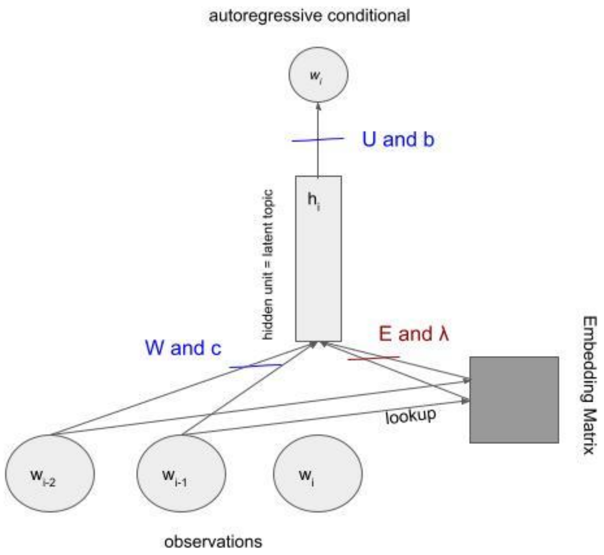


Figure 7: Illustration of DocNADEe model with observations **w** from a document d_j using i hidden units **h** to compute one autoregressive conditionals $p(w_i|\mathbf{d}_{<i})$ with the shared parameters **W** and **U** as well as the bias vectors **c** and **b** as well as the embedding matrix E and its mixture coefficient λ

Source: based on Gupta et al.(2019b), p. 2

Gupta et al. (2019b) compared different variations of these models against each other on 15 different datasets across the evaluation metrics perplexity, topic coherence, document retrieval, document classification as well as qualitative inspection. According to their findings, the best performing model is the iDeepDNEe, a DeepDocNADE with bi-directional language modeling as well as embedding priors.[27] Figure 8 illustrates this state-of-the-art topic model.

[26] cf. Gupta et al. (2019b), p.3.

[27] cf. Gupta et al. (2019b), p.5ff.

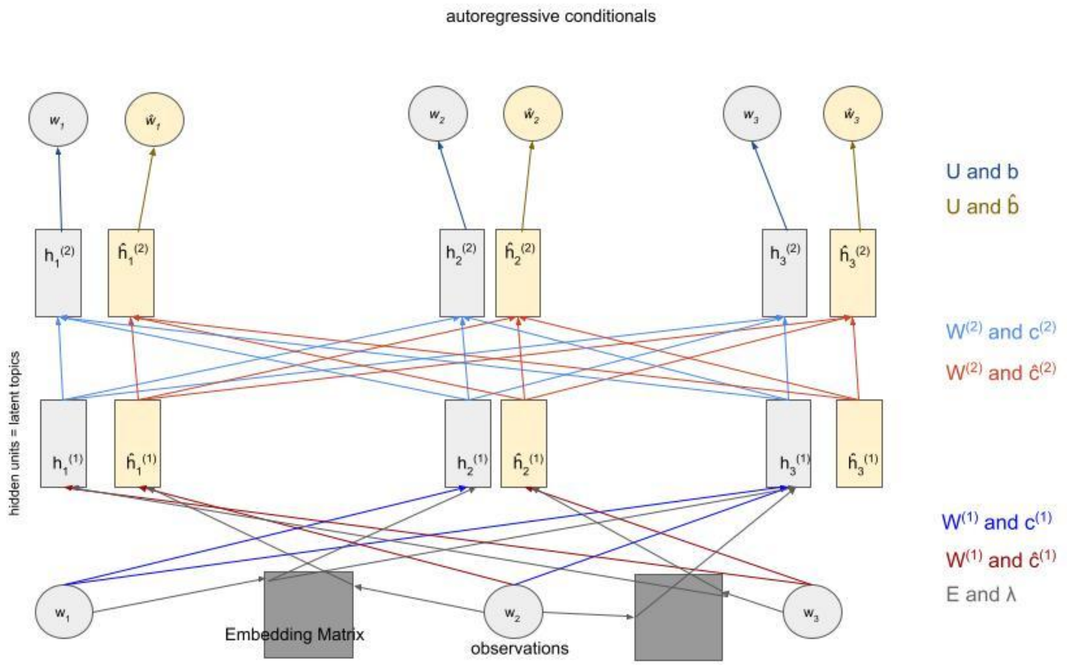


Figure 8: Illustration of iDeepDNe model with observations $\mathbf{w} = [w_1,w_2,w_3]$ from a document d_j using two layers of hidden units $\mathbf{h} = [h_1^{(n)},h_2^{(n)},h_3^{(n)}]$, $n \in \{1,2\}$ to compute in total six autoregressive conditionals $p(w_i|\mathbf{d}_{<i})$ and $p(w_i|\mathbf{d}_{>i})$ with $i \in \{1,2,3\}$, the shared parameters $\mathbf{W} \in R^{3x3}$ and $\mathbf{U} \in R^{3x3}$ as well as the bias vectors $\mathbf{c},\hat{\mathbf{c}} \in R^3$ and $\mathbf{b},\hat{\mathbf{b}} \in R^3$ and the embedding matrix $E \in R^3$ and its mixture coefficient λ. Note that shared parameters are denoted by arrows having the same color and that the two embedding matrices are the same one and only depicted twice due to a clearer arrangement.

Source: based on Figures 5-7

Experimental evaluation

Since the iDeepDNe has never actually been compared against LDA, but instead only indirectly as shown in Table 1, it will be this work's contribution to verify iDeepDNe's superiority over LDA. The experiment will evaluate the performance of the iDeepDNe with the best performing hyperparameter choice as suggested by Gupta et al. (2019) on four datasets with five different metrics and compare it against the results from LDA with hyperparameters optimized from cross-validation. The four different datasets consist of two unique corpora, each in a long and short version. Every model is applied five times to each dataset with different choices for the number of topics. The guiding hypothesis is that iDeepDNe will outperform LDA in all metrics across all datasets and topic number configurations. All code implementations and results are available on https://github.com/flo-kla/bachelorarbeit.

Models

LDA was implemented using gensim (https://github.com/RaRe-Technologies/gensim) where variational Bayes was used for inference. The three hyperparameters, the number of topics K, the Dirichlet hyperparameter alpha for document-topic density, and the Dirichlet hyperparameter beta for word-topic density, were optimized using single-fold cross-validation with c_v topic coherence as the maximization function.

iDeepDNe was implemented using Gupta et al.'s (2019b) repository (https://github.com/pgcool/iDocNADEe). The hyperparameters were set according to the best performance as reported by Gupta et al. (2019b). For example, as model architecture the double hidden layer neural network was chosen, the learning rate was set to 0.001, the activation function to sigmoid, and λ for the embedding mixture to 1.0. For the pre-trained embedding prior, GloVe embeddings were used (https://github.com/stanfordnlp/GloVe). Early stopping with a patience of 25 iterations on the validation set is used to avoid overfitting and to select the model.

Datasets

The two unique corpora are the 20 Newsgroups dataset (20NG), used by many other papers from Table 1, and a self-created dataset called 20subs. The 20NG are used precisely because many historical publications have relied on them, and therefore it is hoped that the results of this work have great comparability. However, a second data set is needed to further validate the findings obtained from the experiments with the 20NG. The 20subs as a self-generated dataset which on the one hand has sufficiently similar or matching structural parameters to the 20NG to be comparable, but on the other hand contains controlled deviations to test whether the putative superiority of a model on one dataset is also transferable to another. Prominent distinguishing features are the length of the individual documents, the linguistic-cultural background as well as the period of origin. Parameters that are the same or similar are the language, the total number of documents, the type of preprocessing, the superordinate topic fields as well as the publication medium.

The 20NG are almost 20,000 documents, forum posts and comments, partitioned nearly evenly across 20 different newsgroups from internet forums from the beginning and mid of the 1990s. They were downloaded from http://qwone.com/~jason/20Newsgroups/. The 20subs are also almost 20,000 documents, forum posts and comments, from May 2015 partitioned evenly across 20 different subreddits from the modern internet forum Reddit. The entirety of all Reddit posts and comments from May 2015 was downloaded from https://www.kaggle.com/reddit/reddit-comments-may-2015 as an .sqlite-database and, subsequently, the first 1000 posts and comments from the 20 relevant subreddits were chosen to be included in the 20subs. In the vast majority of documents, the subreddit origin is identical to a newsgroup name from the 20NG, which should ensure close thematic proximity to the first dataset. Table 2 shows a comparison of all origin subreddits and all newsgroups.

20subs	20NG
r/hockey	rec.sport.hockey
r/politics	talk.politics.misc
r/atheism	alt.atheism
r/baseball	rec.sport.baseball
r/Bitcoin	sci.crypt
r/Christianity	soc.religion.christian
r/cars	rec.autos
r/motorcycles	rec.motorcycles
r/guns	talk.politics.guns
r/space	sci.space
r/DebateReligion	talk.religion.misc
r/worldpolitics	talk.politics.mideast
r/GameSale	misc.forsale
r/hardware	comp.sys.ibm.pc.hardware
r/medicalschool	sci.med
r/Windows10	comp.windows.x
r/mac	comp.sys.mac.hardware
r/graphic_design	comp.graphics
r/AskElectronics	sci.electronics
r/windows	comp.os.ms-windows.misc

Table 2: names of the subreddits and newsgroup where all the documents in the corpora originate from

Source: own creation

It is interesting to observe that some researchers have assumed that k=20 is the ground truth topic number for the 20NG. Since the various documents ultimately come from 20 different newsgroups, this thought may not be entirely absurd, but if one reads through the names of the newsgroup, then it seems as if other “ground truth” topic numbers would also be possible since many newsgroups are thematically close to each other. Another interpretation would be, for example, to choose k=7, since the forums (rec, sci, talk, misc, comp, alt, soc) that are superordinate to the twenty newsgroups could also indicate semantic connections on the macro-level of the corpus (i.e. topics). In the opinion of this work, it is premature to deduce the ground truth number of topics on the basis of the name, which is why several possible k are also tested in the experiment.

As a varying parameter, the size of the vocabulary should also be controlled. Both datasets get a version with scarce vocabulary and a version with rich vocabulary. To better examine the influence of vocabulary size, there is one version each of the 20NG and the 20subs that have approximately the same vocabulary size and one version each that deviates to one extreme or the other. Table 3 lists the most important dataset statistics.

Dataset	Vocab size	#docs	Mean doc length	Std. of doc length
20NG (long)	**4119**	**18846**	**75,56**	**162,06**
~training		9051	78,15	175,81
~validation		2263	76,01	156,09
~test		7532	72,31	145,74
20NG (short)	**1469**	**18846**	**61,15**	**131,51**
~training		9051	63,1	142,42
~validation		2263	61,25	128,32
~test		7532	58,78	118,1
20subs (long)	**13288**	**19800**	**13,29**	**22,33**
~training		9440	13,62	23,07
~validation		2360	13,72	22,86
~test		8000	12,77	21,26
20subs	**4000**	**19800**	**11,98**	**19,822**

(short)				
~training		9440	12,26	20,36
~validation		2360	12,38	20,37
~test		8000	11,52	19

Table 3: comparison of the dataset statistics with vocabulary size in words, number of documents, average document length in words, and standard deviation of document length in words.

Source: own creation

Preprocessing

Due to the way in which the text documents are preprocessed having an enormous influence on the quality of the resulting topic modeling, and the notion that in most other papers the preprocessing is not entirely transparent or understandable - potentially leading to deviations in some results or benchmarks - this work includes a fixed set of preprocessing steps that are applied equally to all four datasets. The preprocessing for this work includes:

- lowercasing all strings
- removing all numbers and all punctuation
- removing all symbols and emojis
- reducing all words to their word root (lemmatization) using the spaCy repository (https://github.com/explosion/spaCy)
- removing 305 stopwords, i.e. words which do not add meaning to a sentence, with the spaCy default stopwords list
- keeping only the x most used terms in the vocabulary (x = individual vocabulary size)

Besides those preprocessing actions, one could have also done stemming, normalization, part-of-speech-tagging, or building n-grams. This, however, as well as investigating the effect of different preprocessing combinations is left to future work.

Metrics and results

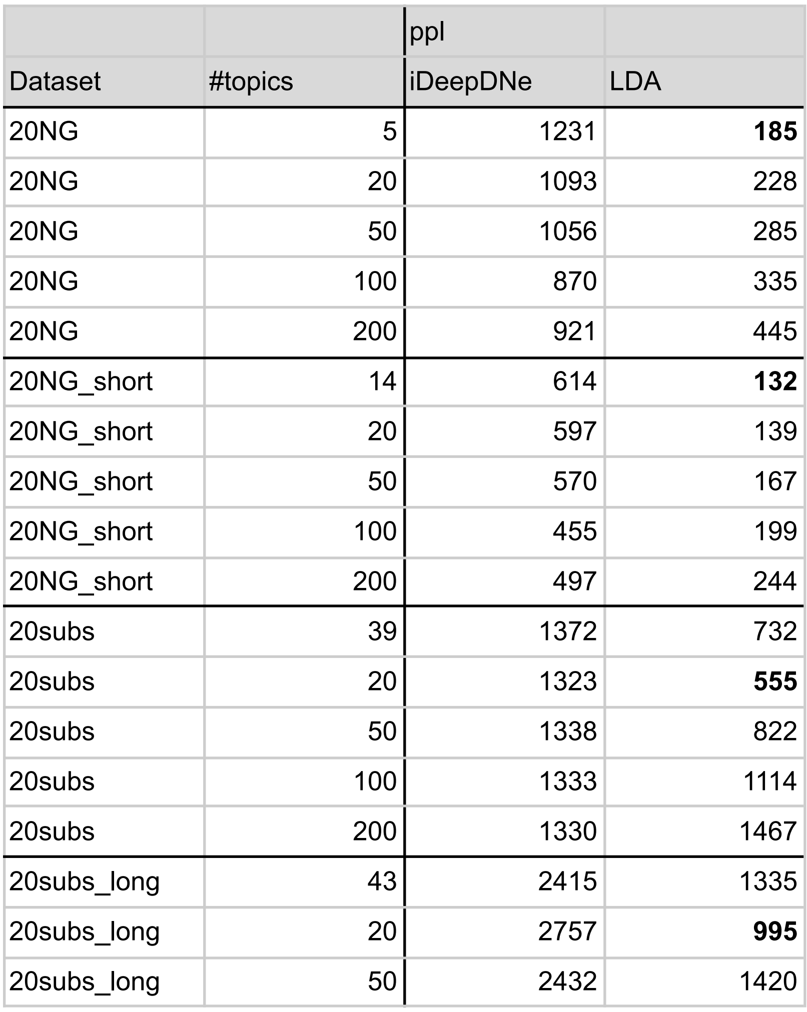
Since there is neither a linguistically clear definition of what constitutes a topic nor a dataset with a unique “gold standard” of topic labels against which an NLP topic model could be measured, it remains impossible from a human perspective to clearly and perfectly evaluate the performance of a topic model. For this reason, various metrics have emerged in NLP to approximate this performance evaluation and have proven useful for research for a variety of reasons including their comparability. This work evaluates the performance of the topic models based on perplexity, c_v topic coherence, document retrieval, document classification as well as qualitative inspection. Note, however, that further evaluation metrics like minimum description length, other computations for topic coherence, intruder detection, maximum entropy, purity, or Kullback–Leibler divergence exist. Incorporating these into experiments and benchmarks will be left to future work.

Generalization: perplexity

Perplexity (ppl) is one of the most commonly used intrinsic evaluation metrics as it is relatively trivial to calculate and intuitive to understand. As the topic model effectively approximates or replicates latent posterior probability distributions, perplexity can be understood to measure how well or accurately the learned distribution predicts a test sample or how "perplexed" a model is in the face of previously unseen data. The goal of the perplexity metric is density estimation in terms of a high likelihood on a held-out test set. Formally, the perplexity is calculated through

$$ppl(D_{test}) = exp(-\frac{\sum_{d=1}^{M} log\,P(d_d)}{\sum_{d=1}^{M} N_d})$$

where P(d_d) for all d ∈ {1,...,M} is the inferred likelihood calculated through the aforementioned computations of the two topic models and N_d the individual number of words of the document d_d.[28] Observe that since iDeepDNe learns by optimizing the average negative log-likelihood of P(d_i) using SGD, it should yield competitive results. Generally speaking, a lower perplexity score indicates a better performance.



		ppl	
Dataset	#topics	iDeepDNe	LDA
20NG	5	1231	**185**
20NG	20	1093	228
20NG	50	1056	285
20NG	100	870	335
20NG	200	921	445
20NG_short	14	614	**132**
20NG_short	20	597	139
20NG_short	50	570	167
20NG_short	100	455	199
20NG_short	200	497	244
20subs	39	1372	732
20subs	20	1323	**555**
20subs	50	1338	822
20subs	100	1333	1114
20subs	200	1330	1467
20subs_long	43	2415	1335
20subs_long	20	2757	**995**
20subs_long	50	2432	1420

[28] cf. Blei, Ng and Jordan (2003), p. 16.

20subs_long	100	2339	1947
20subs_long	200	2300	2923

Table 4: ppls values for all datasets and models. **Bold** values indicate the best value in every dataset.

Source: own creation

Table 4 shows two interesting observable facts. On the one hand, LDA achieves better results in every combination of data set and topic number, on the other hand, the perplexity of iDeepDNe seems to decrease monotonically with the number of topics and the perplexity of LDA seems to increase monotonically with the number of topics. The best results in each data set are consistently obtained by the LDA on the overall lowest number of topics. The ppl values of both models are consistently lower and therefore better on the shorter vocabulary variant of both corpora; this suggests that ppl is dependent on vocabulary size or document length.

Accordingly, LDA is clearly the superior topic model in terms of generalizability for new documents. However, this opens up questions about the extent to which perplexity actually quantifies what it purports to quantify and whether the inferences that can be drawn from this quantification are actually argumentatively legitimate by this metric. These questions, however, cannot be answered within the scope of this thesis but would merit future attention.

Interpretability: topic coherence

As Lau, Newman and Baldwin (2014) have shown, topic coherence is an intrinsic evaluation metric that is positively correlated with human judgment and can be used to estimate the degree of semantic similarity between topics.[29] Ultimately, topic coherence

[29] cf. Lau, Newman, Baldwin (2014). p . 8.

should provide information about the extent to which the learned topics are purely statistical artifacts or actually interpretable topics. Röder, Both and Hinneburg (2015) argued that

> *“[t]he best performing coherence measure [...] is a new combination found by systematic study of the configuration space of coherence measures. This measure (CV) combines the indirect cosine measure with the NPMI and the boolean sliding window.“*[30]

As they argue, coherence aims to measure how well the words of a given word set semantically support each other. The computation of topic coherence, therefore, consists of four individual steps: first, the desired set of words for the coherence computation is segmented into subsets and pairs; second, the computation of the semantical support of all given pairs and subsets through a confirmation measure; third, the computation of word probabilities; lastly the aggregation of scalar values into the final measure.

The word segmentation for c_v is given by

$$S^{one}_{set} = \{(W', W^{*}) \mid W' = \{w_i\};\ w_i \in W;\ W^{*} = W\}$$

where every word of the desired set is compared to the vocabulary using context vectors.[31]

The probability estimation for c_v is the *Boolean sliding window* where word counts are determined by the usage of a sliding window. The method is called “Boolean” because the number of occurrences of words as well as the distance between occurrences is not considered. Instead, the sliding window scans the document one word per step to create a new virtual document every time. The probability of a word is estimated by dividing the number of virtual documents, in which the word appears, through the total number of

[30] Röder, Both and Hinneburg (2015), p.8.

[31] cf. ibid., p.4

documents.[32] The confirmation measure of c_v for two words or word subsets (W',W*) is given by

$$cv_{\widehat{cos(NPMI)}}(W', W^{*}) = \frac{\sum_{i=1}^{|W|} v_{i}(W') \cdot v_{i}(W^{*})}{||v(W')||_{2} \cdot ||v(W^{*})||_{2}}$$

where the context vectors v(W') and respectively v(W*) are given by

$$v(W') = \{ \sum_{w_{i} \in W'} NPMI(w_{i}, w_{j}) \}_{j=1,...,|W|}$$

and finally the NPMI of two words (w,v) by

$$NPMI(w, v) = \frac{log \frac{P(w,v) + \varepsilon}{P(w) \cdot P(v)}}{-\,log\,(\,P(w,v) + \varepsilon\,)}\ .$$[33]

Lastly, these scalar values are aggregated using the arithmetic mean for c_v.[34] C_v ranges from [0,1] where higher values indicate a stronger topic coherence.

To implement the c_v topic coherence, the gensim repository is used. It is computed using the top 10 and top 20 topic words of every generated topic, i.e. the ten or twenty words with the highest probability of belonging to the respective topic, and then averaged across all topics. The size of the boolean sliding window is set to 110 as suggested by Röder, Both and Hinnerburg (2015).[35]

		average c_v for the top 10 words of every topic		c_v ranges for the top 10 words all topics	
Dataset	#topics	iDeepDN(e)	LDA	iDeepDN(e)	LDA
20NG	5	0,4736	0,4024	[0.39,0.61]	[0.22,0.41]

[32] cf. ibid., p. 4

[33] cf. Röder, Both and Hinneburg (2015), p. 5

[34] cf. ibid., p. 5

[35] cf. ibid., p. 7

20NG	20	**0,4828**	0,4554	[0.42,0.64]	[0.16,0.44]
20NG	50	0,3919	0,41	[0.20,0.53]	[0.15,0.45]
20NG	100	0,3454	0,358	[0.14,0.68]	[0.14,0.47]
20NG	200	0,2987	0,482	[0.15,0.60]	[0.14,0.58]
20NG_short	14	0,3773	0,3249	[0.21,0.52]	[0.16,0.45]
20NG_short	20	0,3512	0,3496	[0.24,0.54]	[0.14,0.48]
20NG_short	50	0,3023	0,3053	[0.18,0.51]	[0.12,0.48]
20NG_short	100	0,3109	0,3922	[0.12,0.62]	[0.14,0.49]
20NG_short	200	0,2772	**0,4622**	[0.17,0.67]	[0.13,0.66]
20subs	39	0,5001	0,4359	[0.39,0.63]	[0.42,0.44]
20subs	20	**0,6152**	0,4186	[0.51,0.76]	[0.40,0.43]
20subs	50	0,4207	0,4312	[0.27,0.54]	[0.34,0.52]
20subs	100	0,4801	0,4882	[0.37,0.65]	[0.37,0.56]
20subs	200	0,4797	0,4172	[0.31,0.61]	[0.34,0.54]
20subs_long	43	**0,6374**	0,4728	[0.53,0.70]	[0.38,0.42]
20subs_long	20	0,5421	0,3847	[0.54,0.54]	[0.42,0.47]
20subs_long	50	0,501	0,432	[0.26,0.64]	[0.33,0.52]
20subs_long	100	0,4866	0,4111	[0.33,0.65]	[0.34,0.54]
20subs_long	200	0,5281	0,3847	[0.33,0.66]	[0.34,0.65]
		average c_v for the top 20 words of every topic		c_v ranges for the top 20 words all topics	
Dataset	#topics	iDeepDN(e)	LDA	iDeepDN(e)	LDA
20NG	5	0,4874	0,3875	[0.41,0.54]	[0.22,0.41]
20NG	20	**0,5881**	0,3192	[0.43,0.73]	[0.16,0.44]
20NG	50	0,4678	0,4026	[0.27,0.66]	[0.15,0.45]
20NG	100	0,3687	0,2961	[0.19,0.66]	[0.14,0.47]
20NG	200	0,3356	0,4606	[0.13,0.54]	[0.14,0.58]
20NG_short	14	0,4187	0,2432	[0.24,0.58]	[0.16,0.45]
20NG_short	20	0,4061	0,3648	[0.23,0.55]	[0.14,0.48]
20NG_short	50	0,3551	0,2531	[0.13,0.54]	[0.12,0.48]
20NG_short	100	0,3432	0,3766	[0.17,0.59]	[0.14,0.49]
20NG_short	200	0,2805	**0,4471**	[0.18,0.51]	[0.13,0.66]
20subs	39	**0,6689**	0,4376	[0.60,0.72]	[0.42,0.44]
20subs	20	0,588	0,4282	[0.55,0.61]	[0.40,0.44]

20subs	50	0,5534	0,4362	[0.31,0.65]	[0.34,0.55]
20subs	100	0,5858	0,5011	[0.47,0.75]	[0.37,0.57]
20subs	200	0,5972	0,5359	[0.47,0.71]	[0.34,0.65]
20subs_long	43	**0,7396**	0,5189	[0.66,0.80]	[0.38,0.49]
20subs_long	20	0,6658	0,4968	[0.65,0.69]	[0.41,0.47]
20subs_long	50	0,584	0,4979	[0.49,0.73]	[0.33,0.52]
20subs_long	100	0,5949	0,4732	[0.41,0.72]	[0.34,0.53]
20subs_long	200	0,6474	0,4712	[0.56,0.74]	[0.34,0.65]

Table 5: c_v values and individual topic coherence ranges for all datasets and models computed on either the top 10 topic words or the top 20 topic words. **Bold** values indicate the best value in every dataset.

Source: own creation

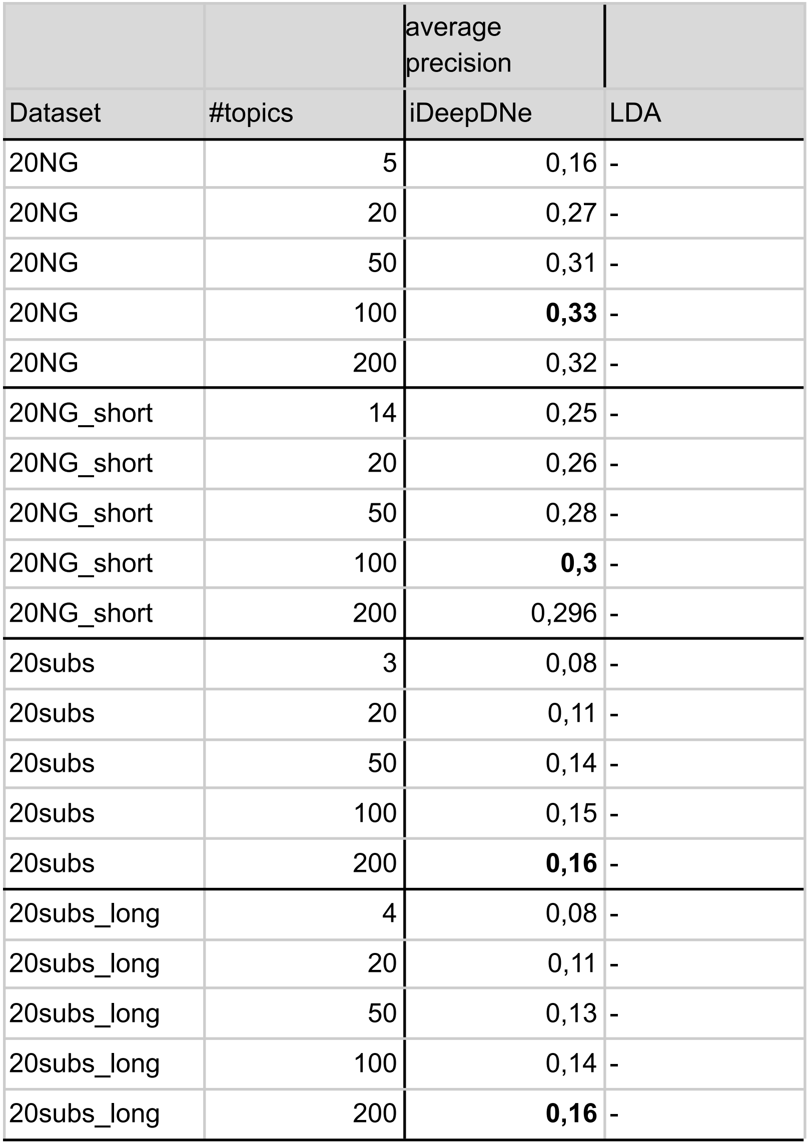
Table 5 also shows interesting anomalies. On the one hand, LDA achieves comparably good and in some cases even better results than iDeepDNe on many combinations of dataset and topic number, but iDeepDNe still achieves better results for each data set almost consistently and the best results for all datasets except for the 20NG_short. Interestingly, the best value for the 20subs is achieved on the topic number optimized for LDA. By fine-tuning the optimal topic number, probably some results of the comparisons in Table 1 could have also been improved. This also suggests that the ground truth number for both the 20NG and the 20subs is not k=20, and indeed it is misguided to derive this ground truth number by name. The generated topics of the iDeepDNe usually have a larger variation of the individual topic coherence than the LDA, whose individual topic coherences are in most cases much closer to each other. A wide range of individual topic coherences could indicate that the specific topic model represents some latent topics extremely accurately, but also generates topics that contribute very little to the actual latent semantic contexts of the corpus. On the other hand, narrow individual topic coherence could signify a big overlap of the topic mixtures. The c_v values from both models are consistently higher and thus better on the longer vocabulary variant of the two corpora, which suggests that c_v somewhat depends on vocabulary size or document length.

Applicability: Document retrieval (precision)

The document retrieval is an external evaluation metric that has also been regularly used by other authors to demonstrate the performance of their proposed models. Generally speaking, it tests the quality of the document representations learned by the respective models. The experiment uses the method of treating the document representations from the training and validation set as a database for retrieval and the document representations from the test set as a query set. The cosine similarity between the vector representations of the query set and the database could then be used to retrieve a certain fraction of the closest documents in the database and finally, by comparing the "label" of the query documents and the fetched database documents, precision curves can be computed as the ratio of correctly predicted positive observations to the total predicted positive observations.[36] The label in this case means the name of the original newsgroup or subreddit. While it does not necessarily have to be the case that the learned vector representations correspond positively with the original newsgroup or subreddit name, especially since these newsgroup or subreddit groupings are not the ultimate ground truth number of topics for the two corpora, it is still a useful heuristic to test how well the topic models incorporated the corpora's origins. For example, a forum post that initially came from the newsgroup or the subreddit "atheism" should, intuitively, not result in retrieving documents from the newsgroup or the sub

[36] cf. Lauly et al. (2017), p. 18, and Gupta et al. (2019b), p. 6.

"hockey." All in all, 14 different retrieval fractions are used to compute the averaged precision.



		average precision	
Dataset	#topics	iDeepDNe	LDA
20NG	5	0,16	-
20NG	20	0,27	-
20NG	50	0,31	-
20NG	100	**0,33**	-
20NG	200	0,32	-
20NG_short	14	0,25	-
20NG_short	20	0,26	-
20NG_short	50	0,28	-
20NG_short	100	**0,3**	-
20NG_short	200	0,296	-
20subs	3	0,08	-
20subs	20	0,11	-
20subs	50	0,14	-
20subs	100	0,15	-
20subs	200	**0,16**	-
20subs_long	4	0,08	-
20subs_long	20	0,11	-
20subs_long	50	0,13	-
20subs_long	100	0,14	-
20subs_long	200	**0,16**	-

Table 6: average precision scores for all datasets and models. **Bold** values indicate the best score in every dataset.

Source: own creation

Unfortunately, the precision values from the experiments with the LDA could not be computed because the implementation crashed each time when retrieving the closest n documents due to insufficient memory. The computations on google colab machines (https://colab.research.google.com/) with both 35.25 GB RAM machines and TPU backend and with 51.01 GB RAM machines and GPU backend also crashed due to insufficient memory. Optimization of the experiment with regard to memory consumption using other frameworks, programming languages, or a more efficient implementation is beyond the current capabilities of the author.

What can be seen from the iDeepDNe results, however, is that the model behaves slightly differently on both data sets. The precision values for the 20subs are strictly monotonically increasing with the number of topics and over both variants of the 20subs are almost identical for each number of topics. The change in vocabulary size does not seem to affect the document retrieval results in any way. However, the retrieval values for the 20NG are always nearly twice as high as those for the 20subs of the same number of topics. The exact precision curves are shown in the supplementary materials.

Applicability: Document classification

Similar to document retrieval, document classification can be used as an external evaluation metric to inspect the quality of the learned word vector representations. While the documented representations of the training and validation sets in combination with their origin label are still used as the database, instead of retrieval, they are now used to train a logistic regression classifier with L2 regularization. This classifier is then used to predict the label of the document representations from the test set. Afterward, accuracy scores can be computed as the ratio of correctly predicted observations to the total observations as well as F1 scores as the weighted average of precision and recall, where recall is the ratio of correctly predicted positive observations to the total observations in the actual class.[37] The logistic regression is implemented using the sklearn repository (https://github.com/scikit-learn/scikit-learn) and 13 different reciprocals of regularization strength. The reported values are averages and a higher score generally indicates a better performance.

		average accuracy		average F1	
Dataset	#topics	iDeepDNe	LDA	iDeepDNe	LDA
20NG	5	0,32	0,05	0,25	0,024
20NG	20	0,51	0,057	0,48	0,034
20NG	50	0,59	0,053	0,57	0,023
20NG	100	**0,63**	0,053	**0,62**	0,016
20NG	200	0,61	0,053	0,59	0,01
20NG_short	14	0,47	0,058	0,43	0,037
20NG_short	20	0,5	0,055	0,47	0,034
20NG_short	50	0,54	0,056	0,52	0,024
20NG_short	100	**0,59**	0,055	**0,57**	0,017
20NG_short	200	0,58	0,054	0,56	0,01
20subs	39	0,24	0,061	0,22	0,036
20subs	20	0,22	0,06	0,19	0,036
20subs	50	0,3	0,056	0,28	0,034
20subs	100	0,331	0,058	0,32	0,026
20subs	200	**0,335**	0,058	**0,33**	0,019
20subs_long	43	0,22	0,053	0,2	0,039
20subs_long	20	0,22	0,058	0,2	0,035
20subs_long	50	0,29	0,053	0,27	0,037
20subs_long	100	0,31	0,056	0,3	0,041
20subs_long	200	**0,35**	0,054	**0,34**	0,034

Table 7: F1 and accuracy values for all datasets and models. **Bold** values indicate the best value in every dataset.

Source: own creation

[37] cf. Gupta et al. (2019b), p. 7.

Table 7 shows that with respect to this experiment, the iDeepDNe achieves an order of magnitude better results. It almost appears as if the vector representations of the LDA are almost unusable for this task of document classification. The F1 scores and accuracy values from the iDeepDNE are almost consistently higher and thus better on the longer vocabulary variant of the two corpora, which suggests that both metrics might depend on vocabulary size or document length. The classification scores tend to increase with the number of topics for the iDeepDNe and tend to decrease for the LDA. Furthermore, the results on the 20NG variants are almost twice as high as the results on the 20subs variants in both metrics. In terms of document classification of this type, the iDeepDNe is clearly the superior topic model.

Human sense: Topic inspection

Although it is not necessarily a mark of quality that a machine produces topics that are represented by a meaningful grouping of words according to human judgment, it is at least satisfying to see that the word groups, or topic distributions over words, are not wholly "chaotic." In the end, since topic modeling in NLP is not interesting for its own sake but rather should function as a tool for concrete applications, it is essential to check if the learned topics can be understood by humans and if good results in the other metrics are automatically correlated with more "meaningful" word groups.

The supplementary materials, therefore, contain both the topics with the highest and the lowest coherence of all trained models. If one looks closely at them one will notice that iDeepDNe generally generates topic word clouds, which might also belong together thematically according to human judgement. However, none of the highest coherent topics is truly convincing because it is obvious that the modeled topics are the mixtures of several different subjects whereas humans would probably generate singular subject topics. Surprisingly, the topics generated by the LDA are not really coherent according to human judgment, since many of them contain additional fill words or terms which are commonly used on Reddit that were not caught and filtered by preprocessing, as well as other artifacts from preprocessing such as single letters or word monstrosities. This suggests that LDA's topic modeling is significantly more prone to noise and glitches, while iDeepDNe can generate quite robust topics under the same conditions. The "correct" preprocessing therefore plays a decisive role in the quality of the topics to be generated, especially with LDA. The fact that there is a big overlap of words in LDA’s highest as well as the lowest coherent topic suggests that many topics overlap indeed to a strong degree.

Discussion

By intrinsic measures, such as ppl and c_v, the LDA is a well-performing topic model that can even keep up with the state-of-the-art iDeepDNe in parts, if not perform better on these metrics. By extrinsic measures such as an experimental applicability test of the topic model or human inspection of the generated topics, LDA is inferior to iDeepDNe in every way. Although the initial hypothesis that the iDeepDNe is the superior model in every situation has not been confirmed, it has become apparent that it is deservedly state-of-the-art. In order to verify whether iDeepDNe is currently the unsurpassable topic model, it would have to be compared against other state-of-the-art models. In particular, different forms of preprocessing and their influences on the results would have to be considered as well as different decision metrics.

The research question of how to determine which topic model is superior to another has not become easier after this experiment and must therefore remain open. In any case, the experiment has shown that the common intrinsic decision metrics, such as ppl and c_v, are not necessarily optimal for preferring one model over the other. Although the LDA has proven to be better than the iDeepDNe in terms of ppl and only slightly worse in terms of c_v topic coherence, it is clear from the applications of the iDeepDNe that it is the better topic model in this situation.

Conclusion

Since the assessment and evaluation of machine-generated language models or topics are already on uncertain ground due to the linguistic sponginess of what distinguishes a "good" and from a "bad" topic structure, it is challenging to understand which evaluation metric could be applied to quantify this distinction. Maybe some assumptions of topic modeling need to be questioned from the ground up and maybe before answering the question. which topic model is superior to another, first another question needs to be fully answered: what constitutes an excellent topic model?

This work has given an extensive overview of the latest developments in NLP topic modeling as well as an overview of some evaluation metrics which are and were commonly used by researchers to prove the superiority of their proposed topic model. With the insights from self-conducted experiments, it has become clear that objectively proving the superiority of an unsupervised NLP topic modeling algorithm is no trivial endeavor. As has been shown in the appropriate places, there are still many more impulses and ideas to deal with this topic further, since by far not all questions have been clarified and even more have been raised by this work than have been answered.

Literature references

Blei (2012). *Probabilistic Topic Models.*

Communications of the ACM: Volume 55, Issue 4 (77-84), p. 4,5.

Blei, Ng and Jordan (2003). *Latent Dirichlet Allocation.*

Journal of Machine Learning Research: 3 (993-1022), p. 2,5,16.

Chang et al. (2009). *Reading Tea Leaves: How Humans Interpret Topic Models.*

NIPS'09: Proceedings of the 22nd International Conference on Neural Information Processing Systems (288-296), p. 1.

Das, Zaheer and Dyer (2015). *Gaussian LDA for Topic Models with Word Embeddings.*

ACL Anthology: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).

Darling (2011). *A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling.*

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (642-647), p. 4ff.

Deerwester et al. (1990). *Indexing by Latent Semantic Analysis*.

JASIST: Journal of the association for information science and technology, Volume 41, Issue 6 (391-468), p. 1, 12f.

Gerlach, Peixoto and Altmann (2018). *A network approach to topic models.*

Science Advances: Volume 4, Issue 7.

Gupta et al. (2019a). *textTOvec: deep contextualized Neural Autoregressive Topic Models of Language with distributed compositional prior.*

ICLR2019.

Gupta et al. (2019b). *Document Informed Neural Autoregressive Topic Models with Distributional Prior.*

AAAI: Proceedings of the AAAI Conference on Artificial Intelligence 33 (6505-6512), p. 1-7.

Hofmann (2001). *Unsupervised Learning by Probabilistic Latent Semantic Analysis.*

Kluwer Academic Publishers: Machine Learning, 42 (177-196), p. 2,4,5.

Jurafsky and Martin (2020). *Speech and Language Processing: International Edition.*

Publisher: Prentice Hall, p. 24 from chapter 3.

Larochelle and Lauly (2012). *A Neural Autoregressive Topic Model.*

NIPS'12: Proceedings of the 25nd International Conference on Neural Information Processing Systems, p. 2,3,4,5.

Lauly et al. (2017). *Document Neural Autoregressive Distribution Estimation.*

Journal of Machine Learning Research: Volume 18 (1-24), p. 9,10,18.

Lau, Newman and Baldwin (2014). *Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence.*

ACL Anthology: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (530-539), p. 8.

Mandelbaum and Shalev (2016). *Word Embeddings and Their Use In Sentence Classification Tasks.*

arXiv: 1610.08229v1, p. 2.

Ngyuen et al. (2015). *Improving Topic Models with Latent Feature Word Representations.* MIT Press: Transactions of the Association for Computational Linguistics 3 (299-313).

Röder, Both and Hinneburg (2015). *Exploring the Space of Topic Coherence Measures.* WSDM'15: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (399-408), p. 4,5,7,8.

Salakhutdinov and Hinton (2009). *Replicated Softmax: an Undirected Topic Model.* NIPS'09: Proceedings of the 22nd International Conference on Neural Information Processing Systems.

Sia, Dalmia and Mielke (2020). *Tired of Topic Models? Clusters of Pretrained Word Embeddings*

Make for Fast and Good Topics too!

EMNLP2020: Association for Computational Linguistics.

Teh et al. (2005). *Hierarchical Dirichlet Processes.*

Journal of the American Statistical Association: Volume 101 (1566-1581).

Index of Tables and Figures

	shared parameters $W \in R^{3x3}$ and $U \in R^{3x3}$ as well as the bias vectors $c,\hat{c} \in R^3$ and $b,\hat{b} \in R^3$		
Figure 7	Illustration of DocNADEe model with observations **w** from a document d_j using i hidden units **h** to compute one autoregressive conditionals $p(w_i\|\mathbf{d}_{<i})$ with the shared parameters **W** and **U** as well as the bias vectors **c** and **b** as well as the embedding matrix E and its mixture coefficient λ	29	based on Gupta et al.(2019b), p. 2.
Figure 8	Illustration of iDeepDNe model with observations $\mathbf{w} = [w_1,w_2,w_3]$ from a document d_j using two layers of hidden units $\mathbf{h} = [h_1^{(n)},h_2^{(n)},h_3^{(n)}]$, $n \in \{1,2\}$ to compute in total six autoregressive conditionals $p(w_i\|\mathbf{d}_{<i})$ and $p(w_i\|\mathbf{d}_{>i})$ with $i \in \{1,2,3\}$, the shared parameters $\mathbf{W} \in R^{3x3}$ and $\mathbf{U} \in R^{3x3}$ as well as the bias vectors $\mathbf{c},\hat{\mathbf{c}} \in R^3$ and $\mathbf{b},\hat{\mathbf{b}} \in R^3$ and the embedding matrix $E \in R^3$ and its mixture coefficient λ. Note that shared parameters are denoted by arrows having the same color and that the two embedding matrices are the same	30	based on Figures 5-7.

	one and only depicted twice due to a clearer arrangement.		
Table 1	selection of recently proposed topic models as well as their evaluation metrics	23f	Own creation.
Table 2	names of the subreddits and newsgroup where all the documents in the corpora originate from	33	Own creation.
Table 3	comparison of the dataset statistics with vocabulary size in words, number of documents, average document length in words and standard deviation of document length in words	34	Own creation.
Table 4	PPL values for all datasets and models. Bold values indicate the best value in every dataset.	37	Own creation.
Table 5	c_v values and individual topic coherence ranges for all datasets and models computed on either the top 10 topic words or the top 20 topic words. Bold values indicate the best value in every dataset.	40ff	Own creation.

Table 6	average precision scores for all datasets and models. Bold values indicate the best score in every dataset.	44	Own creation.
Table 7	F1 and accuracy values for all datasets and models. Bold values indicate the best value in every dataset.	46	Own creation.
Table 8	Comparison of top 20 topics words of highest and lowest coherent topic, source: own creation.	57-61	Own creation.
Table 9	Precision curves on retrieval fractions fract={0.0001, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.8, 1.0} and a moving average as trendline	62f	Own creation.

Supplementary Materials

[A] Table 8: Comparison of top 20 topics words of highest and lowest coherent topic, source: own creation.

[B] Table 9: Precision curves on retrieval fractions fract={0.0001, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.8, 1.0} amd a moving average as trendline, source: own creation.

[A]

Dataset	K	top 20 topic words generated by LDA	top 20 topic words generated by iDeepDNe
20NG	5	people, like, think, know, time, use, s, good, year, want, way, come, work, find, line, look, right, thing, need, x	homosexuality', 'soldier', 'fbi', 'marriage', 'firearm', 'bd', 'arabs', 'hockey', 'homosexual', 'nsa', 'muslim', 'armenian', 'troop', 'league', 'civilian', 'health', 'israel', 'shot', 'score', 'peter
20NG	5	q, d, f, u, t, know, p, x, s, think, m, people, e, good, like, time, work, fb, use, look	key', 'encryption', 'disk', 'chip', 'hit', 'traffic', 'dealer', 'civilian', 'email', 'civil', 'force', 'average', 'floppy', 'block', 'company', 'clipper', 'turkish', 'israeli', 'act', 'board'
20NG	20	know, think, people, time, like, good, s, d, right, use, year, want, come, way, line, find, thing, work, look, tell,	captain', 'phi', 'simms', 'vlb', 'games', 'jew', 'sb', 'z', 'rbi', 'simm', 'macintosh', 'jupiter', 'com', 'cmos', 'fpu', 'scripture', 'irq', 'statistic', 'dp', 'adam'
20NG	20	q, know, d, s, u, x, think, like, t, f, good, people, year, use, work, problem, come, want, time, e	encryption', 'chip', 'polygon', 'ftp', 'upgrade', 'clipper', 'infection', 'symptom', 'colormap', 'chastity', 'compile', 'shameful', 'sensitivity', 'software', 'accessdigexnet', 'yeast', 'industry', 'scispace', 'font', 'sphere'
20NG	50	know, think, people, good, like, use, s, right, time, want, work, look, problem, thing, year, line, come, way, try, new	bull', 'chastity', 'armenian', 'science', 'extermination', 'village', 'widget', 'handler', 'tower', 'manager', 'cryptosystem', 'shameful', 'palestinian', 'ciphertext', 'lebanese', 'cruel', 've', 'patent', 'pp', 'ai'
20NG	50	f, q, d, t, like, know, think, x, people, good, use, u, time, s, line, work, e, year, problem, m	condition', 'tire', 'motherboard', 'setting', 'excellent', 'slot', 'irq', 'cache', 'internal', 'local', 'ide', 'com', 'jumper', 'home', 'card', 'sale', 'vlb', 'regard', 'mhz', 'lately'

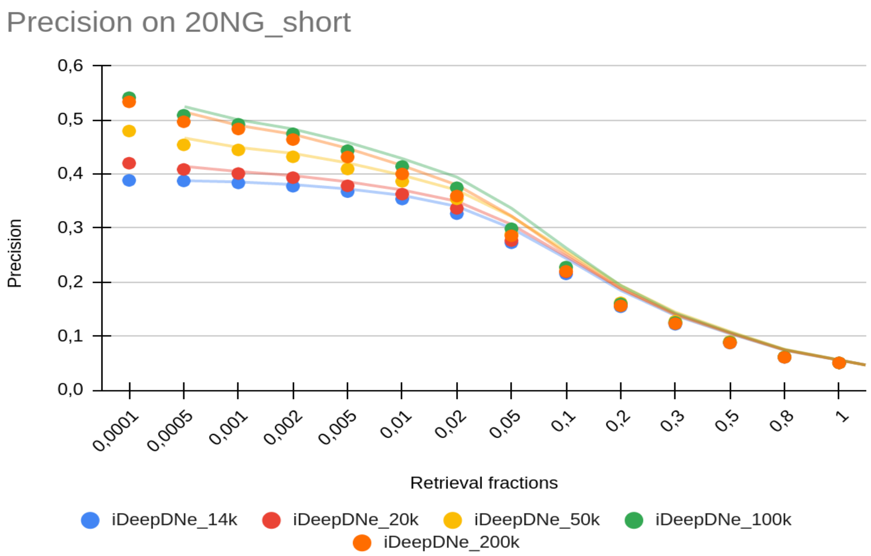
20NG	100	people, know, use, think, time, like, s, good, look, year, want, question, way, line, work, come, right, thing, believe, try	armenian', 'pp', 'turkish', 'troop', 'civilian', 'fight', 'escape', 'longer', 'finally', 'exist', 'soldier', 'proceed', 'p', 'republic', 'refugee', 'wound', 'muslim', 'extermination', 'bullet', 'serve'
20NG	100	d, know, think, s, people, like, x, time, u, good, use, p, need, q, m, file, t, want, come, system	night', 'oil', 'score', 'e', 'blue', 'gateway', 'mailing', 'trade', 'response', 'condition', 'playoff', 'motherboard', 'nntppostinghost', 'card', 'regularly', 'cache', 'nd', 'meg', 'series', 'l'
20NG	200	q, t, d, u, f, p, m, e, fb, g, know, people, v, s, c, like, think, r, x, l	bd', 'arab', 'tap', 'jon', 'sin', 'clipper', 'keith', 'sexual', 'sgi', 'kid', 'civilian', 'billion', 'troop', 'crypto', 'firearm', 'economic', 'deletion', 'follower', 'encryption', 'clh'
20NG	200	q, u, d, know, people, like, f, s, use, think, good, time, work, e, year, t, line, look, want, m	amp', 'ground', 'advice', 'cache', 'ya', 'engine', 'runner', 'left', 'remain', 'wave', 'report', 'slot', 'shift', 'flash', 'buffer', 'clutch', 'master', 'far', 'strike', 'cage'
20NG_short	14	people, good, like, think, know, s, time, work, thing, line, use, right, want, year, find, look, need, tell, d, come	homosexuality', 'soldier', 'fbi', 'marriage', 'firearm', 'bd', 'arabs', 'hockey', 'homosexual', 'nsa', 'muslim', 'armenian', 'troop', 'league', 'civilian', 'health', 'israel', 'shot', 'score', 'peter'
20NG_short	14	q, u, d, people, know, think, s, m, like, t, x, good, use, right, time, e, f, p, way, want	key', 'encryption', 'chip', 'force', 'traffic', 'act', 'civil', 'board', 'hit', 'turkish', 'clipper', 'civilian', 'property', 'average', 'escrow', 'wiretap', 'security', 'company', 'shift', 'kill'
20NG_short	20	think, know, people, like, good, time, want, year, work, s, thing, use, line, way, find, come, need, right, problem, try	firearm', 'nsa', 'armenian', 'soldier', 'russian', 'radar', 'troop', 'detector', 'rocket', 'army', 'coverage', 'nhl', 'clutch', 'penalty', 'fbi', 'launch', 'civilian', 'rob', 'atf', 'satellite'
20NG_short	20	d, like, people, think, know, s, q, use, time, x, m, good, p, t, u, work, e, need, line, way	absolute', 'atheism', 'objective', 'dog', 'bike', 'eric', 'hell', 'morality', 'disagree', 'approach', 'natural', 'motorcycle', 'fully', 'apple', 'effect', 'belief', 'spend', 'save', 'existence', 'conclusion'
20NG_short	50	think, people, like, know, time, good, use, right, s, line, want, year, thing, come, need, way, find, believe, work, look	doctor', 'detector', 'db', 'client', 'mhz', 'dog', 'circuit', 'signal', 'disease', 'scientific', 'dod', 'radar', 'host', 'xv', 'dos', 'motif', 'utility', 'pl', 'push', 'paul'
20NG_short	50	f, u, q, people, know, d, think, t, s, e, good, time, like, use, m, work, p, right, thing, system	church', 'routine', 'creation', 'image', 'nd', 'book', 'mailing', 'lose', 'forward', 'shuttle', 'organization', 'billion', 'degree', 'science', 'sin', 'seek', 'mode', 'fast', 'center', 'feel'

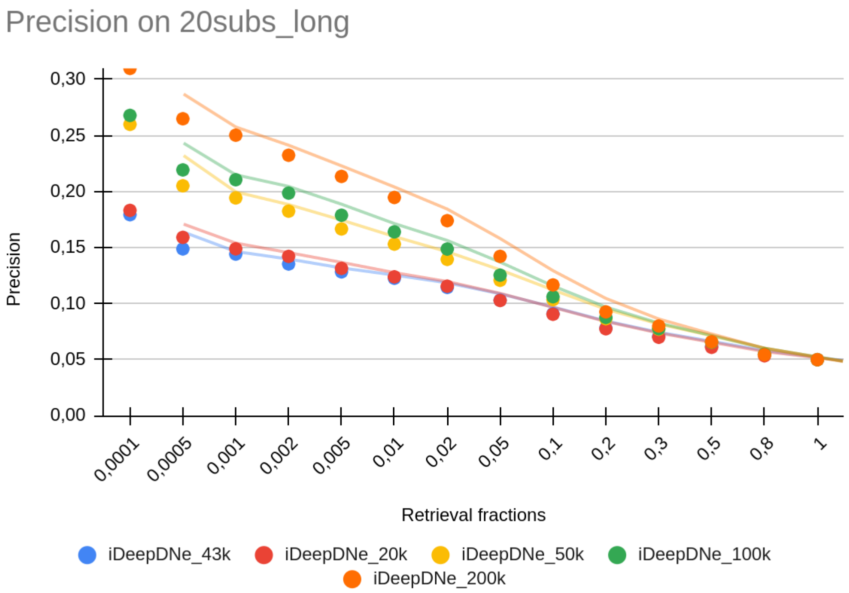
20NG_short	100	know, think, people, time, like, use, good, work, s, line, want, need, way, come, look, find, thing, right, tell, year	massacre', 'armenian', 'universe', 'page', 'muslim', 'pass', 'village', 'troop', 'mountain', 'soul', 'civilian', 'million', 'turkish', 'exist', 'genocide', 'escape', 'personal', 'way', 'road', 'arab'
20NG_short	100	q, like, f, know, s, d, people, think, t, u, time, x, use, good, work, e, thing, find, year, line	arab', 'authority', 'sky', 'spacecraft', 'israeli', 'international', 'turkish', 'launch', 'satellite', 'muslim', 'flight', 'troop', 'orbit', 'massacre', 'vehicle', 'recall', 'planet', 'village', 'nation', 'jewish'
20NG_short	200	q, t, f, d, fb, u, m, e, p, g, v, l, s, know, z, vg, like, think, r, x	escrow', 'wiretap', 'publish', 'detector', 'clutch', 'pat', 'crypto', 'shuttle', 'jack', 'massacre', 'engine', 'health', 'armenian', 'radar', 'turkish', 'battery', 'privacy', 'mountain', 'enforcement', 'solar'
20NG_short	200	x, q, s, f, d, file, t, know, like, people, think, line, use, u, good, p, time, year, want, need	condition', 'trade', 'oil', 'hp', 'dod', 'email', 'sale', 'dealer', 'nd', 'bike', 'penalty', 'fall', 'shift', 'mile', 'playoff', 'team', 'room', 'distribution', 'remote', 'st'
20subs	39	like, people, time, s, know, think, human, way, work, delete, need, gt, good, thing, want, find, mean, year, d, love	guarantee', 'karma', 'ah', 'solely', 'huh', 'wan', 'anki', 'partner', 'potential', 'reinstall', 'slave', 'na', 'passion', 'buddy', 'wd', 'lt', 'anyways', 'cylinder', 'underestimate', 'icon'
20subs	39	s, like, good, use, time, look, find, want, people, actually, year, work, way, thing, come, delete, gt, try, need, sure	form', 'accurate', 'group', 'circuit', 'water', 'special', 'perspective', 'aspect', 'order', 'reflect', 'regard', 'bfhttheoryautotldrconcept', 'permanent', 'ability', 'resistance', 'fps', 'color', 'console', 'strike', 'additionally'
20subs	20	s, like, need, know, way, think, good, use, people, work, delete, point, thing, game, right, time, feel, post, want, high	karma', 'username', 'immediate', 'partner', 'lt', 'swap', 'guarantee', 'potential', 'join', 'prophecy', 'successful', 'huh', 'behance', 'excite', 'congrats', 'tea', 'rtechsupport', 'eh', 'anki', 'reinstall'
20subs	20	like, think, know, good, gt, s, people, time, mean, find, work, way, year, need, use, new, look, maybe, circuit, thing	karma', 'username', 'immediate', 'partner', 'lt', 'swap', 'guarantee', 'behance', 'rtechsupport', 'congrats', 'excite', 'huh', 'resubmit', 'prophecy', 'potential', 'rbuildapc', 'successful', 'modshttpwwwredditcommessagecomposeto', 'join', 'tea'
20subs	50	c, people, gt, think, know, good, like, want, s, year, delete, thing, actually, ac, time, need, e, way, lot, vote	sound', 'pass', 'karma', 'potential', 'external', 'type', 'normal', 'post', 'shop', 'excited', 'range', 'stuff', 'straight', 'excellent', 'shape', 'username', 'false', 'k', 'hair', 'glock'
20subs	50	like, people, s, good, use, work, need, year, find, time, know, delete, think, want, thing, great, lot, gt, mean, look	leave', 'typical', 'welcome', 'group', 'hand', 'public', 'copy', 'bfhttheoryautotldrconcept', 'form', 'consider', 'special', 'contain', 'cross', 'product', 'admin', 'chinese', 'muslim', 'input', 'improve', 'patent'

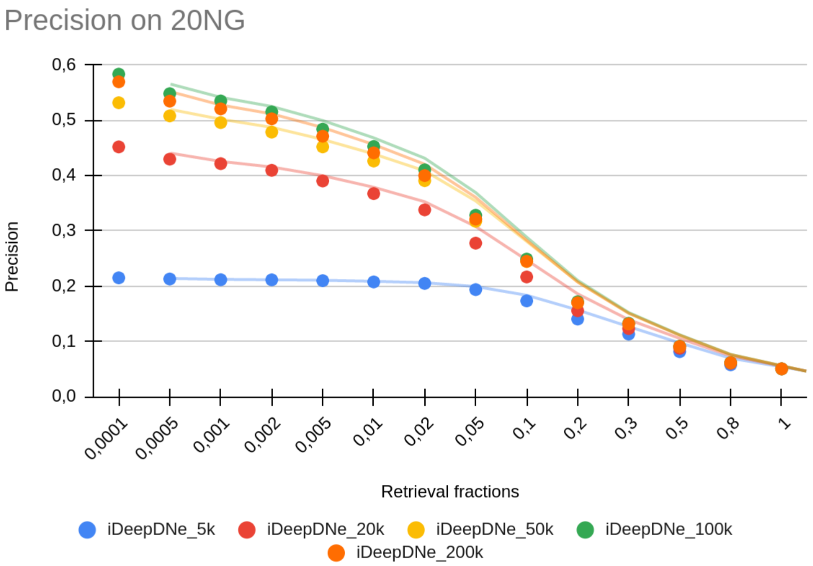
20subs	100	thing, like, s, think, people, time, need, feel, know, delete, good, yes, mean, want, work, way, happen, find, lot, year	pull', 'discussion', 'incredibly', 'typical', 'carbon', 'round', 'present', 'circuit', 'contain', 'buy', 'length', 'finger', 'paint', 'phase', 'jewish', 'rule', 'region', 'red', 'chrome', 'logically'
20subs	100	like, s, people, thing, think, know, need, argument, want, time, work, delete, year, good, run, lot, use, d, try, right	leave', 'contain', 'product', 'bfhttheoryautotldrconcept', 'keyword', 'present', 'consider', 'tldrs', 'pm', 'typical', 'fmfaqautotldrbot', 'circuit', 'welcome', 'patent', 'tldr', 'ride', 'expose', 'reply', 'color', 'constructive'
20subs	200	like, s, want, think, people, good, know, delete, right, work, come, year, use, way, thing, care, look, time, d, need	pull', 'consider', 'wear', 'picture', 'carbon', 'ride', 'bfhttheoryautotldrconcept', 'mm', 'send', 'strip', 'content', 'block', 'core', 'thought', 'circuit', 'turbo', 'suck', 'sentence', 'special', 'round'
20subs	200	like, s, thing, good, think, find, live, use, know, people, delete, god, work, need, want, time, look, right, life, way	picture', 'public', 'current', 'keyword', 'color', 'consider', 'bfhttheoryautotldrconcept', 'block', 'find', 'group', 'special', 'welcome', 'cross', 'powerful', 'typical', 'incredibly', 'software', 'ride', 'ability', 'arm'
20subs_long	43	time, think, s, like, delete, look, good, work, year, use, mean, way, thing, people, need, app, want, human, come, start	ring', 'slide', 'wear', 'wheel', 'commenter', 'animal', 'snipe', 'regard', 'laser', 'contemporary', 'additionally', 'dominate', 'length', 'reproduce', 'priest', 'generate', 'uh', 'mag', 'deep', 'curse'
20subs_long	43	like, gt, think, s, thing, time, people, use, look, year, information, point, good, know, come, karma, swap, find, provide, link	guarantee', 'lt', 'partner', 'username', 'karma', 'skip', 'immediate', 'ah', 'past', 'anyways', 'mic', 'potential', 'collector', 'beneficial', 'jesus', 'optimal', 'test', 'elementary', 'iraqi', 'depreciation'
20subs_long	20	think, s, use, good, like, know, work, time, people, battery, circuit, look, way, need, gt, delete, want, right, come, mean	karma', 'username', 'immediate', 'partner', 'lt', 'guarantee', 'swap', 'anyways', 'approve', 'donation', 'innovation', 'resubmit', 'anki', 'fornicate', 'font', 'mob', 'chill', 'wd', 'pedantic', 'starting'
20subs_long	20	s, like, people, work, way, need, thing, use, year, good, look, information, app, find, think, time, help, try, swap, know	karma', 'username', 'partner', 'immediate', 'lt', 'resubmit', 'anyways', 'guarantee', 'approve', 'ahead', 'phone', 'abuse', 'swap', 'donation', 'laptop', 'starting', 'worth', 'bitch', 'innovation', 'font'
20subs_long	50	like, think, year, work, know, time, thing, s, way, hit, good, come, drive, game, find, try, need, gt, pretty, post	slide', 'barrel', 'chinese', 'zionist', 'wheel', 'begin', 'ignore', 'church', 'bridge', 'generate', 'fan', 'mission', 'sun', 'frequency', 'curve', 'refuse', 'contain', 't', 'contemporary', 'calibrate'
20subs_long	50	like, think, thing, need, work, right, time, karma, information, know, swap, look, year, people, s, way, read, sure, come, link	tldr', 'gt', 'welcome', 'typical', 'wheel', 'bot', 'fmfaqautotldrbot', 'bfhttheoryautotldrconcept', 'write', 'animal', 'ring', 'copy', 'chinese', 'wear', 'generate', 'police', 'refuse', 'autotldr', 'send', 'tldrs'

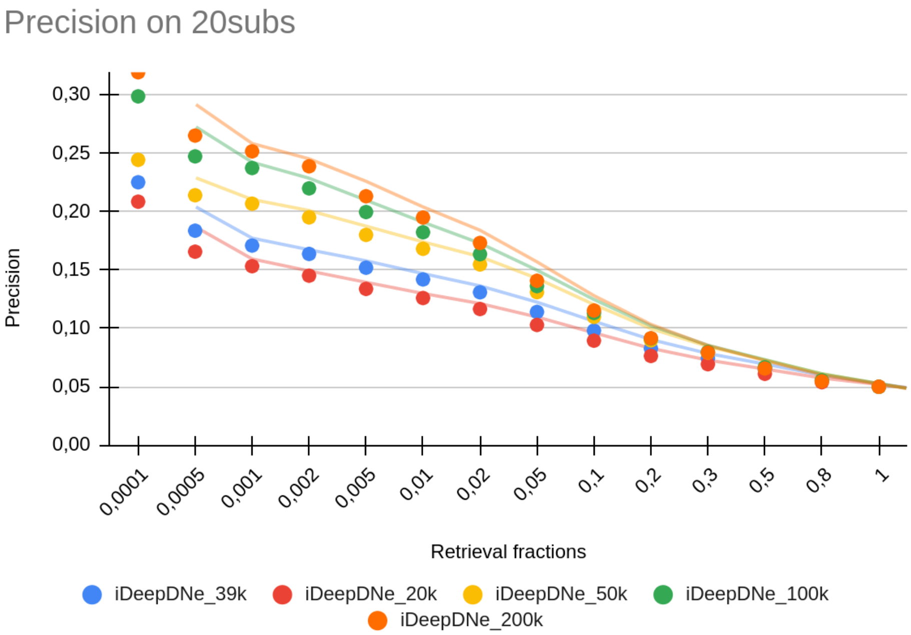
20subs_long	100	think, like, good, thing, time, s, want, mean, work, look, love, right, little, gt, way, drive, leave, link, find, game	ride', 'edge', 'correct', 'ability', 'disagree', 'wear', 'core', 'gold', 'faqhttpnpredditcomrautotldrcomment', 'wave', 'evolutionary', 'color', 'wheel', 'typical', 'anybody', 'avoid', 'content', 'gun', 'accurate', 'contain'
20subs_long	100	like, s, know, way, work, need, gt, information, people, come, help, swap, karma, try, look, find, year, thing, mean, different	wear', 'keyword', 'leave', 'contain', 'animal', 'ability', 'admin', 'patient', 'faqhttpnpredditcomrautotldrcomment', 'participate', 'gt', 'ride', 'bot', 'jewish', 'avoid', 'edge', 'effect', 'tldr', 'reduce', 'deny'
20subs_long	200	swap, information, karma, help, year, thing, guarantee, background, date, provide, immediate, trade, link, comment, join, successful, partner, username, potential, think	edge', 'stretch', 'carbon', 'ride', 'ground', 'public', 'resistor', 'wave', 'thread', 'wear', 'cap', 'regard', 'special', 'anybody', 'chinese', 'selection', 'trigger', 'patient', 'exhaust', 'trap'
20subs_long	200	think, marriage, s, know, people, right, state, union, civil, like, gay, core, work, lot, year, gt, day, actually, account, point	wear', 'previous', 'regard', 'cause', 'accurate', 'civilization', 'individual', 'jewish', 'patient', 'contain', 'ride', 'unit', 'keyword', 'fact', 'exact', 'public', 'brain', 'satellite', 'female', 'welcome'

[B]


Precision on 20NG_short
Precision
0,6
0,5
0,4
0,3
0,2
0,1
0,0
0,0001
0,0005
0,001
0,002
0,005
0,01
0,02
0,05
0,1
0,2
0,3
0,5
0,8
1
Retrieval fractions
iDeepDNe_14k
iDeepDNe_20k
iDeepDNe_50k
iDeepDNe_100k
iDeepDNe_200k


Precision on 20subs_long
Precision
0,30
0,25
0,20
0,15
0,10
0,05
0,00
0,0001
0,0005
0,001
0,002
0,005
0,01
0,02
0,05
0,1
0,2
0,3
0,5
0,8
1
Retrieval fractions
iDeepDNe_43k
iDeepDNe_20k
iDeepDNe_50k
iDeepDNe_100k
iDeepDNe_200k


Precision on 20NG
Precision
0,6
0,5
0,4
0,3
0,2
0,1
0,0
0,0001
0,0005
0,001
0,002
0,005
0,01
0,02
0,05
0,1
0,2
0,3
0,5
0,8
1
Retrieval fractions
● iDeepDNe_5k
● iDeepDNe_20k
● iDeepDNe_50k
● iDeepDNe_100k
● iDeepDNe_200k


Precision on 20subs
Precision
0,30
0,25
0,20
0,15
0,10
0,05
0,00
0,0001
0,0005
0,001
0,002
0,005
0,01
0,02
0,05
0,1
0,2
0,3
0,5
0,8
1
Retrieval fractions
● iDeepDNe_39k
● iDeepDNe_20k
● iDeepDNe_50k
● iDeepDNe_100k
● iDeepDNe_200k

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt gegenüber der Technischen Universität Berlin, dass die vorliegende, dieser Erklärung angefügte Arbeit selbstständig und nur unter Zuhilfenahme der im Literaturverzeichnis genannten Quellen und Hilfsmittel angefertigt wurde. Alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind kenntlich gemacht.

Ich reiche die Arbeit erstmals als Prüfungsleistung ein.

Ich versichere, dass diese Arbeit oder wesentliche Teile dieser Arbeit nicht bereits dem Leistungserwerb in einer anderen Lehrveranstaltung zugrunde lagen.

Titel der schriftlichen Arbeit: *Comparison of neural networks with generative probabilistic topic models in natural language processing based on different metrics - How to decide which topic model is superior to another?*

Verfasser: Name: Klawun

Vorname: Florian Eric

Matr.-Nr. 379164

Betreuende Dozenten:

Name: Lassner

Vorname: David

Name: Prof. Dr. Müller

Vorname: Klaus-Robert

Mit meiner Unterschrift bestätige ich, dass ich über fachübliche Zitierregeln unterrichtet worden bin und verstanden habe. Die im betroffenen Fachgebiet üblichen Zitiervorschriften sind eingehalten worden. Eine Überprüfung der Arbeit auf Plagiate mithilfe elektronischer Hilfsmittel darf vorgenommen werden.

Berlin, den 14. August 2021

Ort, Datum

Unterschrift