Natural Language Processing - Methods and Applications
Topic Models - A Survey of Architectures

Peter Karl Weinberger
Department of Informatics
Technische Universität München
peter.weinberger@tum.de

Abstract

Topic models are a crucial tool when it comes to the analysis of unlabeled natural language text data. A topic model reveals the hidden structures behind documents—so-called topics—and helps to understand what large document collections are all about. Over time, they have become increasingly complex as natural language processing has evolved. State-of-the-art topic models take more and more aspects of natural language into account, such as word order or word context. Thus, the quality of the documents' essential content can be more precisely extracted. This paper aims to provide a survey of various topic modeling methods.

1 Introduction

The Internet has evolved from a medium that was only used by a few universities to one that is used by billions of people every day. During this transformation, new technologies were developed to solve previously unknown problems. In the early days of the World Wide Web, for instance, it was simply unforeseeable that one day 2.3 billion people would use a social network called "Facebook" and that these users would generate enormous amounts of data (Roser et al., 2015). To handle massive datasets, many tools have been developed to process these so-called Big Data containing natural language texts. With the gigantic growth of available unlabeled text documents, ways were sought to understand what these documents are about without actually been read by a human, so that these natural language texts may be processed further.

Topic models are completely about to uncover the hidden topical patterns which pervade the collection of texts. Those patterns are called topics (Blei et al., 2003). After applying a topic model to a text collation it is possible to annotate the documents according to the assigned topics and then use those annotations to organize, summarize, and search the hitherto unlabeled texts. In topic modeling, each document consists of a certain number of topics derived from the text collection, and each topic is composed of a certain group of words.

This paper is focused on the survey of state-of-the-art topic modeling techniques, which are introduced in Section 2. Section 3 shows that the methods of topic modeling are not limited to natural language text data, but can also be applied on e.g. images. To conclude this survey, a discussion of the methods of topic modeling and the findings are available in Section 4.

2 Methods of Topic Modeling

In this paper, the different methods have been divided into categories to give the reader more structure. However, one should not perceive these subdivisions as absolute, but rather as an attempt to structure all these different and complex models. First, latent semantic analysis—the first topic model of its kind—is presented in Section 2.1. In Section 2.2 probabilistic topic models are introduced. More advanced methods using neural autoencoders are presented in Section 2.3, and in Section 2.4 embedded topic models are shown. In Section 2.5 a multi-lingual topic modeling approach is illustrated.

2.1 Latent Semantic Analysis

Latent semantic analysis (LSA) was first used in information retrieval—in this field latent semantic analysis is often also called latent semantic indexing (LSI)—to improve the detection of relevant documents based on the words given by a query (Deerwester et al., 1990). The goal of LSI is to find main components of documents. Those main components are in fact topics. In order to find these topics, the well-known singular value decomposition (SVD) can be computed. From linear algebra, it is known that a matrix $X \in \mathbb{R}^{m \times n}$ can be decomposed into three matrices, which is stated in equation 1.

$$X = T_0 S_0 D_0^T \tag{1}$$

In the application of deriving topics from natural language, the term-document-matrix X should be calculated before applying SVD. This matrix can be easily created via computing tf-idf. The SVD will now decompose the matrix X into T_0, S_0, and D_0^T. While T_0 is a unitary matrix of shape $m \times m$, S_0 of shape $m \times n$ contains the singular values of matrix X_0 in its' diagonal, and D_0^T is an unitary matrix of shape $n \times n$. From the theory behind SVD, it is known that we can choose such a decomposition that the singular values will be ordered by their size. So, we can approximate X, by selecting only the k largest singular values. While k is the rank of the matrix X. So $\hat{X}$ is the best approximation of matrix X according to the Frobenius norm, therefore equation 2 holds.

$$X \approx \hat{X} = TSD^T \tag{2}$$

The dimension reduction of the matrix X can be controlled by selecting a number of singular values at most k. Thus, the number of topics can be limited and only the most important topics will be selected, because the singular values are—as mentioned—sorted in descending order. In the end, LSA wants to find a low-rank approximation of the source matrix X. The matrix T represents the term-topic matrix, S contains the importance of the topics, and D^T includes the topic-document matrix. In the real world, the latter will not be calculated, because the size of this matrix would be too large if one considers the number of documents times the number of topics.

2.2 Probabilistic Topic Models

For probabilistic topic models a document is a mixture of topics with a certain probability. Therefore, each topic consists of a mixture of words, also with a certain probability. These topic models assume that a unknown generator created the observed documents with the discussed properties and aim to rebuild this very generator.

2.2.1 Probabilistic Latent Semantic Analysis

The probabilistic latent semantic analysis (PLSA) is an advancement of the latent semantic analysis, which is based on the SVD. However, PLSA uses a mixture approach to decompose a term-document-matrix—usually a tf-idf-matrix. The algorithm is most related to non-negative matrix factorization (NMF) (Ding et al., 2006). With NMF a matrix can be approximated by only two smaller matrices. Another major difference in understanding the principles of PLSA is, that PLSA is based on a statistical latent variable model—also called the aspect model.

The principles of PLSA are well stated (Hofmann, 1999). We assume this model is latent, generative, and created our documents with a particular probability. Furthermore, it is assumed that there is also a probability for every word in every document originating from a latent topic.

$$\begin{aligned} P(d, w) &= P(d)P(w|d) \text{ where,} \\ P(w|d) &= \sum_{z \in Z} P(w|z)P(z|d) \end{aligned} \tag{3}$$

The dependencies are expressed in equation 3, while a document occurs with probability $P(d)$, a latent class z—also called topic—appears with probability $P(z|d)$, and a word will be generated with probability $P(w|z)$. This principle is very remarkable, hence it will be later the foundation of more

advanced methods like the latent Dirichlet allocation. PLSA is trained via maximizing the log-likelihood shown in equation 4 applying the Expectation-Maximization (EM) algorithm, where $n(d, w)$ is the term frequency.

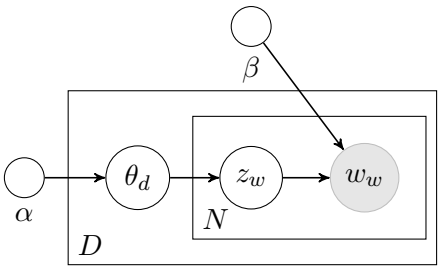
$$\begin{aligned} \mathcal{L} &= \sum_{d \in D} \sum_{w \in W} n(d, w) \log P(d, w) \text{ with,} \\ P(d, w) &= \sum_{z \in Z} P(z) P(w|z) P(d|z) \end{aligned} \tag{4}$$

PLSA has shown great improvements to the application of topic models to real-world data and can outperform more advanced techniques (Lee et al., 2010). However, PLSA's major disadvantage is that in practice, documents aren't encouraged to focus on a limited number of topics. It is more likely that the result of PLSA has a very large dispersion in terms of the distribution of topic-document probabilities. So, documents may have small weights for many different topics.

2.2.2 Latent Dirichlet Allocation

The latent Dirichlet allocation (LDA) enhances PLSA, so that the previously mentioned disadvantage of PLSA vanishes (Blei et al., 2003). While LSA and PLSA were introduced for information retrieval tasks and were only later used for topic analysis, LDA was designed for exactly this purpose.

It is assumed that all words originate from different topic distributions and that all documents were generated with a certain probability from a mixture of topic distributions. Due to the inherent Dirichlet distribution, it must be mentioned that the topics are assumed to be independent of each other. In practice, this assumption will not be true hence topics are often correlated (Li et al., 2012). This matter will be discussed in Section 2.2.3. But the assumption of independent topics can also be useful in terms of supply a straight forward implementation using the EM algorithm in combination with Bayesian parameter estimation.



1: **for** document d_d in corpus D **do**
2: Choose $\theta_d \sim$ Dirichlet(α)
3: **for** position w in d_d **do**
4: Choose a topic $z_w \sim$ Multinomial(θ_d)
5: Choose a word w_w from $p(w_w|z_w, \beta)$, a multinomial distribution over words conditioned on the topic and the prior β.
6: **end for**
7: **end for**

Figure 1: The plate diagram on the left illustrates the basic principles of LDA. On the right hand side, the essential algorithm used in LDA is displayed.

The simplified algorithm shown in Figure 1 describes the main matter of LDA in a very concise way. To be more specific, the word probabilities are present in the matrix β of shape number of topics times number of words in the vocabulary where $\beta_{ij} = p(w^j = 1|z^i = 1)$. For high significance, LDA assigns words within a document to as few topics as possible, while for each topic it assigns as few terms as possible with a high probability to create more specific terms per topic. If both goals are balanced, groups of frequently used terms appear automatically and the algorithm converges.

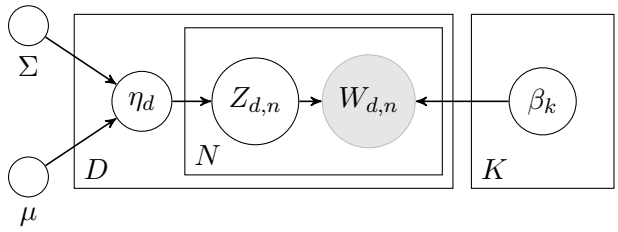
To influence the result of LDA essentially three parameters α, β, and number of topics N can be adjusted. To briefly explain these parameters, one can assume symmetrical Dirichlet distributions. Now a low α puts more emphasis on the fact that each document consists of only a few dominant topics, while a high value return many more relatively dominant topics. Similarly, a low β puts more weight on the fact that each topic consists of only a few dominant words and vice versa. In the final implementation e.g. Gensim (Řehůřek and Sojka, 2010), one can only modify the α parameter. Typical the a-priori belief for each topics' probability will be learned through training or set to $\frac{1}{N}$.

Because of the highly interpretable results after applying LDA, and with the ability to refining the results of LDA (Fujino, 2014), it is probably the most popular topic modeling method right now as

mentioned through out the scientific community (e.g. (Larochelle and Lauly, 2012), (Liu et al., 2016), or (Xie et al., 2019)). Due to its popularity, LDA is widely used, including applications in human resource management, such as the fully automated identification of competencies from CVs (Schiller, 2019). Thereby LDA extracts the topics of CVs, which can be e.g. the educational background of applicants.

2.2.3 Correlated Topic Models

A limitation of LDA is to assume independence between topics. However in practice, it is much more likely that topics are correlated. For example scientific papers about natural language processing will be more probably also about topic models or text similarity than about psychological disorders. With correlated topic models (CTM) (Blei and Lafferty, 2006), this disadvantage of LDA no longer remains.



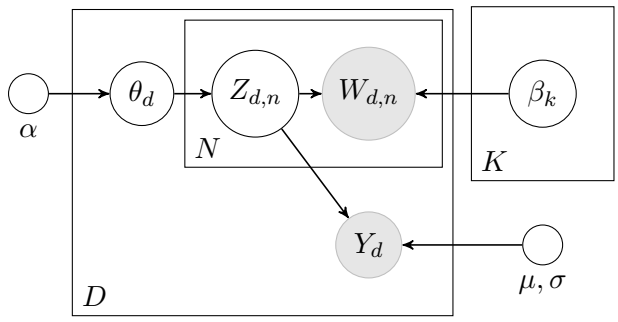
for document d_d in corpus D **do**
  Choose a $\eta \mid \{\mu, \Sigma\} \sim N(\mu, \Sigma)$
  for each N words in w_n **do**
    Choose a topic assignment $Z_n \mid \eta$ from Multinomial$(f(\eta))$
    Choose a word $W_n \mid \{z_n, \beta_{1:K}\}$ from Multinomial(β_{z_n})
  end for
end for

Figure 2: The plate diagram on the left illustrates the fundamental principles of CTM. On the right, the essential algorithm used in CTM is shown.

CTM introduces another two hyperparameters μ and Σ. Both hyperparameters are estimated via applying a variational EM algorithm. Furthermore, while LDA highly depends on the Dirichlet distribution, CTM changes the base distribution from which hyperparameters are derived from the previous mentioned Dirichlet to a logistic transformation of a multivariate Gaussian. This logistic normal distribution used to model the latent topic portions of a document can represent correlations between topics which cannot be captured by a single Dirichlet. All these changes enables the model to cover related topics.

2.2.4 Supervised Topic Models

LDA as well as all presented topic modeling methods learns the latent structures of texts unsupervised. One drawback of these approaches is that using an unsupervised model as a classifier of labeled data might not be feasible. Although it is indeed possible to create a topic model that examines, for example, film ratings, but it is not necessarily true that the model will also find topics that correspond to a number of stars in this setting. It is much more likely that such a model learns the different genres and models them as topics, since the texts are mostly about the film and not about the actual number of stars. For such applications a supervised extension of LDA was introduced called sLDA (Mcauliffe and Blei, 2008).



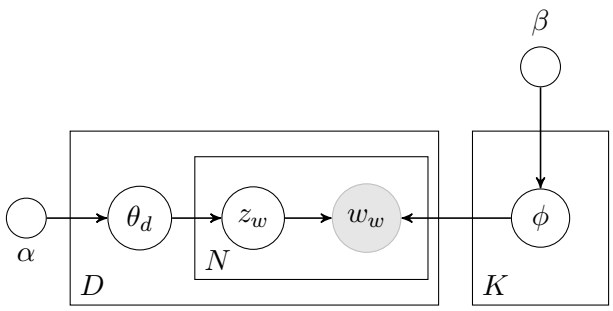
for document d_d in corpus D **do**
  Choose $\theta \mid \alpha \sim \text{Dirichlet}(\alpha)$
  for position w in d_d **do**
    Choose a topic assignment $z_n \mid \theta$ from Multinomial (θ)
    Choose a word $w_n \mid z_n, \beta_{1:K}$ from Multinomial (β_{z_n})
  end for
  Choose a response variable $y \mid z_{1:N}, \eta, \sigma^2$ from $N(\eta^T \overline{z}, \sigma^2)$
end for

Figure 3: The plate diagram on the left illustrates the basic principles of sLDA. On the right side, the key algorithm of sLDA is presented.

As shown in Figure 3, the main difference between LDA and sLDA is another parameter called the response variable y. This response variable is not limited to be simply text categories. It can also represent a certain movie rating, or with other words: y is not limited to be categorical, thus it can be assumed that $y \in \mathbb{R}$. All parameters $\beta_{1:K}$, η, and σ^2 will be learned via the same variational EM-algorithm as used in LDA (Blei et al., 2003). The topics are still obtained in the same way as for regular LDA, with probability calculations being used to construct N topics. However, sLDA goes one step further: sLDA also correlates the response variable y internally with the topics to use the derived topic vectors for unseen documents to predict how the response variable should look like.

2.2.5 Topic Modeling for Sentences

LDA is used to uncover the hidden structures of documents, while ignoring the essential structure of sentences within all those documents. To overcome this issue sentenceLDA was introduced (Balikas et al., 2016). This method samples topics for each coherent text span instead of documents. Where a coherent text span can be a paragraph, sentence, or phrase.



for document d_d in corpus D **do**
  Choose $\theta_m \sim$ Dirichlet(α)
  Choose $S_d \sim$ Poisson(ξ)
  for sentence $s \in [1, S_d]$ **do**
    Choose $W_s \sim$ Poisson(ξ_d)
    Choose $z_{d,s} \sim$ Multinomial(θ_m)
    for word $w \in [1, W_{d,s}]$ in s **do**
      Choose $w \sim$ Multinomial$(\phi_{z_{d,s}})$
    end for
  end for
end for

Figure 4: On the left, a plate diagram illustrates the basic principles of sentenceLDA. On the right hand side, the essential algorithm used in sentenceLDA is displayed.

The main difference between LDA and sentenceLDA is that—as shown in Figure 4—they apply LDA sentence wise with minor changes to the origin LDA.

2.2.6 Prediction Focused Topic Models via Feature Selection

The ability of supervised learning is essential for classification tasks. However, the majority of topic models are unsupervised methods due by the fact that topic models are used to uncover hidden structures of texts—commonly unknown. Therefore, all recently described models are unsupervised except sLDA. A drawback of sLDA is that the resulting topics are likely to consist of irrelevant words regarding to the aimed classification class. The existence of features that are irrelevant for a supervised classification task hampers optimization and also decreases the topic coherence of the returned topics. Therefore, prediction-focused supervised LDA (pf-sLDA) was developed (Ren et al., 2019).

Topic modeling via pf-sLDA includes simultaneously learning which features are irrelevant while training and fits a supervised topic model only considering relevant features. In other words: pf-sLDA decides while training from which distribution a generated word originates either from a distribution that maps relevant or irrelevant words. In fact, this turns pf-sLDA into a dual-channel topic model. Thereby, the target variable y depends only on the first channel. The second channel thus serves as an outlet for words that are irrelevant for the prediction of the target. From which channel a word originates is determined by a corresponding Bernoulli switch ξ, which has prior probability p.

With this simple extension, pf-sLDA is able to select only relevant features of a particular topic. The relevant features are derived from a multinomial distribution with the parameter β, and the irrelevant features are drawn from a multinomial distribution with the parameter π. Words are considered irrelevant if their presence in the relevant topics with parameters β would hinder the prediction of the target variable y. Simultaneously, a word will be only considered as irrelevant if this change outweighs the cost of considering it irrelevant given the probability p.

2.3 Topic Modeling using Neural Autoencoders

In this section, topic modeling architectures are introduced, which are taking advantage of powerful deep neural networks. Deep neural networks are well-known for their ability to learn high complex data structures and thus they are useful tools for extracting topical patterns in natural language.

2.3.1 Document Neural Autoregressive Distribution Estimator

Document Neural Autoregressive Distribution Estimator (DocNADE) is an unsupervised autoregressive artificial neural network, which learns the hidden structures of documents (Larochelle and Lauly, 2012). Neural Autoregressive Distribution Estimation (NADE) is used within the model, which is a specified architecture of neural autoencoders. In fact, a hierarchy of binary logistic regressions is implemented to calculate a probability for each word in the vocabulary. Therefore, a binary tree of logistic regressions calculates the probability of an observed word.

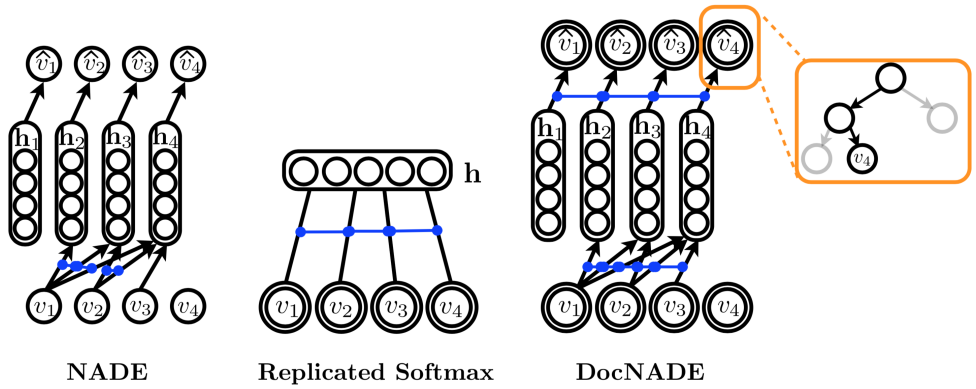


Figure 5: The blue colored lines indicate shared weights. On the left the origin architecture of the NADE model is displayed. Here $\hat{v}_i$ is short hand for $p(v_i|v_{<i})$. The center shows the architecture of the replicated Softmax used in DocNADE, and on the right the architecture of DocNADE itself is illustrated.

Figure 5, which is extracted from the original paper, shows the methods used in DocNADE. Especially the detail, that each conditional $p(v_i|v_{<i})$ is in fact decomposed into a tree of binary logistic regressions is well presented. Although, this neural topic model is highly advanced it ignores the word order.

2.3.2 K-Competitive Autoencoder for Text

The K-Competitive Autoencoder for Text (KATE) is a neural autoencoder approach (Chen and Zaki, 2017). Experiments with the 20 Newsgroups dataset showed that KATE outperforms other topic models like LDA or DocNADE. KATE uses competitive learning to let the neurons of the autoencoder compete against each other for the right to react to a certain subset of the input data. Because of this strategy, the autoencoder includes neurons with high specialization. The neurons are more about to learn individual patterns within the input data. In more detail, they ensure this behavior by selecting the k neurons with the highest activation value as the winner. Applying this autoencoder to textual data, it is possible to retrieve semantically meaningful representation of words, documents, and topics.

2.3.3 Autoencoding Variational Inference for Topic Models

Autoencoding Variational Inference for Topic Models (AVITM) yields a extension of LDA called ProdLDA (Srivastava and Sutton, 2017). They simply replace the word-level multinomial distribution assumption with a weighted product of experts. This means, that they replaced line 5 of the algorithm shown in Figure 1 with w_n is defined as $w_n|\beta, \theta \sim Multinomial(1, \sigma(\beta\theta))$. The function σ is indeed the AVITM blackbox inference method, which returns μ and Σ of the multinomial distribution after learning the hidden structures within the data via applying a neural autoencoder with variational inference. They claim that the most significant advantages of ProdLDA over LDA are the better topic coherence, and the computational efficiency. Latter is due the fact, that on unseen data AVITM does only require to pass through the forward pass of a neural network.

2.4 Embedded Topic Models

Embedded topic models aim to project topics and words of documents into a linear space while respecting the context of the words. Therefore, embedded topic models have the advantage that they take the word order into account and their results are interpretable by humans.

2.4.1 TopicVec

TopicVec uses LDA and the generative word embedding approach Positive-Semidefinite Vectors (PSDVec) in order to create a generative topic embedding model (Li et al., 2016). Unlike word2vec, PSDVec is a matrix factorization-based method. The topic embeddings of TopicVec have the same dimensionality as the word embedding space created by PSDVec. The goal of TopicVec is to approximate the latent semantic centroids of the word embeddings. Thus, in this paper, topics can bee seen as a N-dimensional hyperball. The algorithm learns topic and word embeddings independently while applying variational inference.

2.4.2 lda2vec

In contrast to TopicVec, lda2vec uses word2vec word embeddings, but they both use LDA to model the latent structures of documents (Moody, 2016). This model uses the skip-gram architecture of word2vec to embed the words into vector space by predicting the surrounding context words of a pivot word. The concrete word2vec model is expanded to simultaneously learn word, document and topic vectors. In lda2vec the pivot word vector and a document vector are added to generate a context vector. This context vector is then used to predict context words. As in TopicVec, the topic vectors are learned in word space, which allows easy interpretation. With the ability to create context vectors, lda2vec is able to yield topics not over just documents, but also regions.

2.4.3 ctx-DocNADE

DocNADE is a bag-of-words based method, which ignores the context of words. As mentioned, this is a major drawback of this model because understanding natural language depends highly on the context. The proposed ctx-DocNADE model addresses exactly that issue (Gupta et al., 2018). This approach builds on the DocNADE model and extends it by adding a long short-term memory language model (LSTM-LM) to DocNADE. A LSTM-LM is most commonly a pre-trained language model, which is able to predict the most probable word given the previous word. In the end, LSTM-LMs are the core components of recent autocompletion systems like the Personalized Language Model for Query Auto-Completion (Jaech and Ostendorf, 2018). The identified latent topic vectors of DocNADE are added to hidden vectors derived from the LSTM-LM. After combining the approaches of DocNADE and a LSTM-LM, ctx-DocNADE is able to identify word occurrences in collocation patterns and also across documents.

In the original paper, they also propose ctx-DocNADe besides ctx-DocNADE. The ctx-DocNADe models' LSTM-LM layer is pre-trained and initialized with a certain embedding matrix and corresponding weights, while the LSTM-LM layer of ctx-DocNADE is randomly initialized. Especially the proposed ctx-DocNADe model outperforms other generative topic models like LDA or DocNADE in terms of topic coherence, perplexity, and applicability verified on 15 data sets.

2.4.4 Topic Modeling in Embedding Spaces

A downside of generative topic model approaches like LDA is that they fail on very large and heavy-tailed word distributions. The proposed embedded topic model (ETM) aims to eliminate this problem by combining LDA with word embeddings (Dieng et al., 2019). Like the TopicVec model presented in Section 2.4.1, ETM embeds both words, and topics into the same embedding space. The model assumes, that words of a document are generated from a categorical distribution. The parameters of this distribution are derived from the inner product of the word embedding and the embedding of the corresponding topic. While the paper states, that this method supplies interpretable topics even from large vocabularies, it is not able to extract topics from parts of the documents. The upside of ETM is the robustness against stop words.

2.5 Multilingual Topic Models

A major downside of all presented approaches is that they do not address the problem of different languages within a corpora of documents.

2.5.1 Multilingual Anchoring

Multilingual Topic Anchors (MTAnchor) faces this very issue by introducing an interactive anchor-based topic modeling algorithm (Yuan et al., 2018). An anchor word is a word that appears with high probability in one topic, but with low probability in all other topics. With this interactive approach a user can choose anchor words for each of the different languages, which appear in the corpora. The different anchor words per language should of course have the same meaning for a topic in the different languages. This requires a person who is able to interpret the different meanings of a word in each of these languages, which can be a very complex task. Nevertheless, MTAnchor allows to design comprehensive multi-language topic models, while a topic is also present in all other given languages.

3 Outlook

The ability of topic models to reproduce the hidden patterns of data collections can also be applied to non-text data in general, transforming the algorithms into powerful tools for various data types.

3.1 Discriminative Topic Modeling

LDA is a generative model specially developed for texts. To apply topic models to other data, it is possible to give up the generative approach in order to replace it with a discriminative method (Korshunova et al., 2019). This approach is called logistic LDA, which can be applied for e.g. images or other non-text data. To illustrate the power of logistic LDA, the authors of the paper applied the algorithm on a Pinterest image data set where the "documents" are boards and "words" are images. Then the topic model created topics, each of which is composed only of images related with e.g. dogs, fashion or architecture.

Logistic LDA is both a supervised and unsupervised learning model. Like LDA, the model is able to learn the hidden structure of the input data unsupervised. At the same time, logistic LDA inherit a supervised extension if class labels are present, which makes it very flexible for several applications.

4 Discussion

Topic models have evolved to a variety of different methods, each with its advantages and drawbacks. Therefore, the selection of a well-suited architecture for a specific task can be very hard. Another striking detail which does not simplify the selection of models is that all stated models are presented in different papers of different years, so as a conclusion they all use different topic coherence measurements to evaluate their models. This essentially is the crux of the matter. To be clear, this is generally the case in academia, but the majority of topic modeling approaches is unsupervised, thus evaluation is critical. The author doesn't want to suggest that scientists choose the right measurement methods just to make their model look better, but the danger is definitely there. Most use cases involve the understanding of unlabeled text data, that means that no classification test set is available. Other evaluation approaches like word or topic intrusion require staff to evaluate the models, although recent research replaces the suggestive factor (Bhatia et al., 2018). Those evaluation methods are good to get a feeling of how well a method seems to be to a human, but are not objective at all. This paper did not cover evaluation methods, but it should be stressed that the evaluation of topic modeling methods is very important in order to choose a model that best fits the data. However, many applications, such as the discussed one in Section 2.2.2, calculate only one topic coherence measurement to select the best model, which can be misleading.

In conclusion, state-of-the-art topic models are extremely powerful for encoding hidden patterns of text collections and are consequently an essential part of the natural language processing toolbox. Just the fact that a machine learning model can understand what unseen and unlabeled text data is about is as astonishing as it sounds like. The rapid developments in this area in recent years are remarkable. In the beginning, the bag-of-words models were used, i.e. independent words, today topic models can also capture and map the context around a word. Thus, improved models can be created, so that the algorithms can detect the meaning of large text collections even better and more accurately.

References

Georgios Balikas, Massih-Reza Amini, and Marianne Clausel. 2016. On a topic model for sentences. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 921–924.

Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. 2018. Topic intrusion for automatic topic model evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 844–849.

David Blei and John Lafferty. 2006. Correlated topic models. *Advances in neural information processing systems*, 18:147.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Yu Chen and Mohammed J Zaki. 2017. Kate: K-competitive autoencoder for text. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 85–94.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2019. Topic modeling in embedding spaces. *arXiv preprint arXiv:1907.04907*.

Chris Ding, Tao Li, and Wei Peng. 2006. Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method. In *AAAI*, volume 42, pages 137–43.

I. Fujino. 2014. Refining lda results and ranking topics in order of quantity and quality with an application to twitter streaming data. In *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pages 209–216.

Pankaj Gupta, Yatin Chaudhary, Florian Buettner, and Hinrich Schütze. 2018. Texttovec: Deep contextualized neural autoregressive topic models of language with distributed compositional prior. *arXiv preprint arXiv:1810.03947*.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57.

Aaron Jaech and Mari Ostendorf. 2018. Personalized language model for query auto-completion. *arXiv preprint arXiv:1804.09661*.

Iryna Korshunova, Hanchen Xiong, Mateusz Fedoryszak, and Lucas Theis. 2019. Discriminative topic modeling with logistic lda. In *Advances in Neural Information Processing Systems*, pages 6767–6777.

Hugo Larochelle and Stanislas Lauly. 2012. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, pages 2708–2716.

Sangno Lee, Jaeki Song, and Yongjin Kim. 2010. An empirical comparison of four text mining methods. *Journal of Computer Information Systems*, 51(1):1–10.

Lianghao Li, Xiaoming Jin, and Mingsheng Long. 2012. Topic correlation analysis for cross-domain text classification. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Shaohua Li, Tat-Seng Chua, Jun Zhu, and Chunyan Miao. 2016. Generative topic embedding: a continuous representation of documents. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 666–675.

Yang Liu, Quanxue Gao, Shuo Miao, Xinbo Gao, Feiping Nie, and Yunsong Li. 2016. A non-greedy algorithm for l1-norm lda. *IEEE Transactions on Image Processing*, 26(2):684–695.

Jon D Mcauliffe and David M Blei. 2008. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128.

Christopher E Moody. 2016. Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Jason Ren, Russell Kunes, and Finale Doshi-Velez. 2019. Prediction focused topic models via vocab selection. *arXiv preprint arXiv:1910.05495*.

Max Roser, Hannah Ritchie, and Esteban Ortiz-Ospina. 2015. Internet. *Our World in Data*. https://ourworldindata.org/internet.

Alexander Schiller. 2019. Knowledge discovery from cvs: A topic modeling procedure.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*.

Xiaolong Xie, Yun Liang, Xiuhong Li, and Wei Tan. 2019. Culda: solving large-scale lda problems on gpus. In *Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing*, pages 195–205.

Michelle Yuan, Benjamin Van Durme, and Jordan L Ying. 2018. Multilingual anchoring: Interactive topic modeling and alignment across languages. In *Advances in Neural Information Processing Systems*, pages 8653–8663.