The Latent Dirichlet Allocation (LDA) generative model for automating process of rendering judicial decisions

Petr Pylov*, Roman Maitak1, and Andrey Protodyakonov1

1T.F. Gorbachev Kuzbass State Technical University, 650000, Kemerovo, 28 Vesennya st., Russian Federation

Abstract. The Latent Dirichlet Allocation (LDA) generative model is widely used in statistical analysis and machine learning due to its ability to model the probabilities of multidimensional categorical data, such as the frequencies of different categories or the probability distribution across multiple categories. This article explores the potential application of the LDA model for the practical task of topic separation in documents related to judicial proceedings.

1 Introduction

The overwhelming majority of practical applications of the Latent Dirichlet Allocation (LDA) model are based on "artificial" data samples. This is due to the fact that mathematical sampling criteria accurately capture all the characteristics of the data required for the aforementioned method.

It is worth noting that cases where the data samples are tailored to a specific machine learning model undermine the algorithm itself in applied artificial intelligence. In nature and technical systems, there is no dataset that can be unequivocally transformed to fit a particular machine learning algorithm. On the contrary, machine learning models are developed to fit specific datasets. Therefore, analysts spend most of their working time analyzing the mathematical characteristics of the obtained dataset in the relevant domain. For this reason, applying the Latent Dirichlet Allocation model requires researchers to develop a data preprocessing module that serves as an intermediary between the domain-specific data and the machine learning model.

In the domain of rendering judicial decisions, the process of generating an automated response to a legal claim is particularly important. From the perspective of applied artificial intelligence, this task falls within the realm of natural language processing (NLP) and can be addressed based on an approach to analyzing textual information, which is regarded as a combination of topics, terms, and documents.

Let us consider how text can be represented as a vector. Text is a sequence of words arranged in a specific order (according to lexical and other rules).

* Corresponding author: pylovpa@kuzstu.ru
If we can represent words in the text as vectors, the text itself becomes analogous to a time series (a sequence structured according to certain laws and rules). Thus, the possibility of vectorization arises—reducing the dimensionality of the text to the dimensionality of unit vectors (Figure 1).

**Fig. 1. Overall Scheme of Text Vectorization**

Figure 1 illustrates the overall structure of text information vectorization, divided into constituent blocks of operations.

The first block enables the representation of a sentence as a collection of words formatted in a rectangular structure of dimensions n×k, where n-represents the width of the hypothetical rectangle, and k-represents its length.

The second operational block performs convolution (convolutional layer) based on the rules of multiple filters and an object map [3].

The third operational layer serves as an aggregating layer (referred to as 'pooling' in English) [5]. The aggregating layer does not have trainable weight coefficients; its task is to aggregate values it 'observes' within a given neighborhood (Figure 2).

**Fig. 2. Functionality of the Aggregating Layer**

For example, in Figure 2, the aggregating layer applies an aggregating function F to each cell (it can be a max pooling function, average pooling function, or any other function that performs simple mathematical operations).

The size of the cell for convolution is set to 2x2 pixels, but it depends solely on the data researcher. Convolution allows reducing the dimensionality of the cell (the aggregating function F 'folds' the 2x2 cell into a 1x1 pixel). In the first case, the max pooling function is chosen, so the maximum value of the original 2x2 cell remains as the sole value in the 1x1
cell. In the second case, the aggregating function $F$ is an average function, so the average value of the four values becomes the single value of the pixel.

Finally, the fourth operational block (layer) in Figure 1 represents a unit vector of instances. The unit vector characterizes the dimensionality of the layer (width equals one), and its approximate value composition consists of a dimensionality of $\geq 10^7$, as this vector reflects all the information of the textual data snippet.

Having acquainted ourselves with the basic architecture of a generalized neural network for text analysis (Figure 1), let's consider the fundamental hypothesis on which natural language processing is based [3].

Text analysis by applied artificial intelligence models began to develop based on the distributional hypothesis, which explores types of semantic word similarity.

The most logical question that arose at the beginning of machine text analysis was understanding the meaning of words using mathematical models.

In the middle of the last century, linguists determined that the meaning of a word can be defined as the collection of all possible contexts in which it appears in language. The context encompasses all possible sentences that can exist with that word. Thus, semantically related words appear in similar contexts - a word can be characterized by the 'company of words' in which it appears.

These principles define the essence of the distributional hypothesis: if words can be described by their contexts, then it is necessary to formalize the concept of 'context'.

Furthermore, for further formalization, two additional linguistic concepts need to be defined:

- **Syntagmatic proximity of words** (the combination of words in the same context, for example: building - construction; crane - water; function - point) is schematically represented in Figure 3.

![Fig. 3. Schematic Representation of Syntagmatic Word Similarity](image)

- **Paradigmatic proximity of words** (interchangeability of words in the same context, for example: building - house; crane - faucet; function - mapping) is schematically represented in Figure 4.

![Fig. 4. Schematic Representation of Paradigmatic Word Similarity](image)

From Figures 3-4 and their descriptions, it can be inferred that syntagmatic proximity is an observable characteristic, meaning that words can be encountered together (within the same sentence), while the paradigmatic representation of words is an unobservable characteristic. It characterizes the interchangeability of words and is closely related in meaning to the concept of synonyms.
2 Materials and Methods

Let us formalize the distributional hypothesis. To do this, we will define the initial conditions:

(1)

Let us note that in the system of conditions (1), two alternative vector representations are introduced for each word:

- When the word is in the context as a predictor, it is used to predict the word \( w_i \);
- The word \( w_i \) itself is the predicted word, and another vector \( u_w \) is assigned to it.

Having fully formalized the problem with the system (1), the final step is to define the maximum likelihood criterion (2).

(2)

The maximum likelihood criterion is defined as the sum of logarithms of the probability model over all word positions in the text. The dimensionality of the text is determined by the \( U \times V \) matrix, where \( U, V \in \mathbb{R}^{W \times d} \), with \( W \) representing the size of the vocabulary and \( d \) representing the dimensionality of the word vector representations.

3 Results and discussion

As a basis for consideration, the Latent Dirichlet Allocation (LDA) model was examined. This model includes a regularizer that is non-zero, distinguishing it from the Probabilistic Latent Semantic Analysis (PLSA) model [1].

The formula for the regularizer in the Latent Dirichlet Allocation model is similar to cross-entropy or the maximum likelihood principle:

(3)
$$Dir(\varphi | \beta) = \frac{\Gamma(\beta_0)}{\prod_w \Gamma(\beta_w)} \prod_w \varphi_{wt}^{\beta_{wt} - 1}, \quad \varphi_{wt} > 0, \quad \beta_0 = \sum_w \beta_w, \quad \beta_t > 0$$ (4)

$$Dir(\theta_d | \alpha) = \frac{\Gamma(\alpha_0)}{\prod_t \Gamma(\alpha_t)} \prod_t \theta_{td}^{\alpha_{td} - 1}, \quad \theta_{td} > 0, \quad \alpha_0 = \sum_t \alpha_t, \quad \alpha_t > 0$$ (5)

Furthermore, the Dirichlet distribution can be either too sparse or highly concentrated. For example, a sparse distribution is concentrated near the edges and vertices of the n-dimensional simplex (Figure 5).

Fig. 5. Representation of a specific case of sparse Dirichlet distribution

Most commonly [3], in the case of a sparse distribution, vectors are generated in such a way that the majority of coordinates in the vectors are close to zero but never exactly zero (according to the constraint conditions).

The greater the sum of all parameters in the Dirichlet distribution, the more concentrated the distribution plot will be (Figure 6).

Fig. 6. Representation of a specific case of concentrated symmetric Dirichlet distribution
In legal case management tasks, this approach allows for the differentiation of document topics and automated separation of court cases based on their distinct outcomes [4, 5].

Adding information to the machine learning model that the columns of the matrix are generated from a Dirichlet distribution can be done based on the principle of maximum posterior probability. This involves obtaining the joint likelihood of the data and the model by multiplying the likelihood of the sample by the prior Dirichlet distributions:

\[
\ln \prod_{d \in D} \prod_{w \in d} p(w, d | \Phi, \Theta)^{n_{dw}} \prod_{t \in T} \text{Dir}(\Theta_t | \beta) \prod_{d \in D} \text{Dir}(\Theta_d | \alpha) \rightarrow \max \Phi, \Theta \quad (6)
\]

After taking the logarithm of the joint likelihood (6), we obtain the regularizer (3) with the following frequency estimates:

\[
\phi_{wt} = \text{norm}_w(n_{wt} + \beta_w - 1), \quad \theta_{td} = \text{norm}_t(n_{td} + \alpha_t - 1) \quad (7)
\]

4 Conclusion

Based on the biased frequency estimates of the Dirichlet Latent Allocation regularizer, it seems feasible to implement a module for topic differentiation in legal case management, as the task of automated decision-making based on a lawsuit falls within the realm of natural language processing.

From an applied perspective, a sparse Dirichlet distribution indicates the presence of an individual characteristic in the document text, which is likely to classify the document under a different topic (i.e., based on the text characteristic, a decision will be made to assign the document to a different target class, resulting in a different court decision).

This important feature allows for the utilization of the Dirichlet Latent Allocation model as a submodule in a second legal opinion automation system for making judicial decisions during case management processes.

References

3. R.Wiblin, Positively shaping the development of artificial intelligence (2017)