



Article

A New Sentence-Based Interpretative Topic Modeling and Automatic Topic Labeling [†]

Olzhas Kozbagarov ^{1,*}, Rustam Mussabayev ^{1,*}  and Nenad Mladenovic ^{2,*} ¹ Institute of Information and Computational Technologies, Pushkin Str., 125, Almaty 050010, Kazakhstan² Khalifa University, Abu Dhabi 41009, United Arab Emirates

* Correspondence: kozbagarov@gmail.com or o.kozbagarov@iict.kz (O.K.);

rmusab@gmail.com or rustam@iict.kz (R.M.);

nenadmladenovic12@gmail.com or nenad.mladenovic@ku.ac.ae (N.M.)

[†] The work was funded by a grant of the Committee of Science of Ministry of Education and Science of the Republic of Kazakhstan.

Abstract: This article presents a new conceptual approach for the interpretative topic modeling problem. It uses sentences as basic units of analysis, instead of words or n-grams, which are commonly used in the standard approaches. The proposed approach's specifics are using sentence probability evaluations within the text corpus and clustering of sentence embeddings. The topic model estimates discrete distributions of sentence occurrences within topics and discrete distributions of topic occurrence within the text. Our approach provides the possibility of explicit interpretation of topics since sentences, unlike words, are more informative and have complete grammatical and semantic constructions inside. The method for automatic topic labeling is also provided. Contextual embeddings based on the BERT model are used to obtain corresponding sentence embeddings for their subsequent analysis. Moreover, our approach allows big data processing and shows the possibility of utilizing the combination of internal and external knowledge sources in the process of topic modeling. The internal knowledge source is represented by the text corpus itself and often it is a single knowledge source in the traditional topic modeling approaches. The external knowledge source is represented by the BERT, a machine learning model which was preliminarily trained on a huge amount of textual data and is used for generating the context-dependent sentence embeddings.

Keywords: natural language processing; topic modeling; automatic topic labeling; BERT; big data; minimum sum-of-squares clustering (MSSC); machine learning; transfer learning



Citation: Kozbagarov, O.; Mussabayev, R.; Mladenovic, N. A New Sentence-Based Interpretative Topic Modeling and Automatic Topic Labeling. *Symmetry* **2021**, *13*, 837. <https://doi.org/10.3390/sym13050837>

Academic Editor: Peng-Yeng Yin

Received: 16 April 2021

Accepted: 6 May 2021

Published: 10 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Topic modeling is a statistical analysis method to discover latent semantic structures hidden in texts [1]. Topic modeling in the natural language processing (NLP) field is one of the main methods used in various applications: information retrieval [2], classification [3], categorization [4], document annotation [5], classification of social significant information [6], propaganda identification [7], mass media evaluation [8], social network analysis [9], etc. The topic model identifies which terms compose each topic and which topics are related to each text. The topic model estimates two matrices (unobservable parameters of the model): the first matrix defines the discrete probability distribution of term occurrence within each topic, the second one—the discrete distribution of topic occurrence within each text.

The first topic model was the Probabilistic Latent Semantic Analysis model (PLSA) developed by Thomas Hoffman [10]. Further, David Blei proposed the Latent Dirichlet Allocation model (LDA) [11]. This model is a modification of PLSA. In LDA, it is proposed to use Dirichlet distributions as a priori distribution of unknown model parameters: the distribution of terms in topics and the distribution of topics in texts. Further, Bayesian inference is used to find the posterior distribution of parameters, considering the observed

text data. This model has become very popular in the literature, and many various modifications have been developed [1]. In 2014, Konstantin Vorontsov proposed a new approach to topic modeling, called Additive Regularization of topic models (ARTM) [12]. According to this approach, the choice of a regularizer that puts restrictions on the unknown values of two matrices' elements determines a specific topic model. In this case, only point estimates of unknown parameters are evaluated, without estimating their distributions through Bayesian inference or Gibbs sampling, as in the case with LDA modifications. The topic model based on the ARTM approach is currently state-of-the-art and outperforms LDA modifications [13,14]. This is achieved because the ARTM approach utilize different regularizers that provide enough decorrelation and sparsity of domain-specific topics with simultaneous smoothing of background topics.

Most topic models use terms as the basic analysis units, which can be words, n-grams or phrases. As a result, each topic is represented as a discrete distribution over a set of these basic units. This type of representation complicates the interpretation of the obtained topics since there is no grammatical connection between the terms. As a result, it is difficult to find their semantic relatedness, and the contextual information of terms is lost. In most cases, the term's context is more vividly expresses the thematic essence than the term itself. Therefore, an important task is to develop such topic modeling methods where the contextual information is used in the analysis process and fully represented in obtained topics. The methods proposed in this article are aimed at solving this problem.

Attempts have been made in topic modeling to considering the text elements' coherence, e.g., considering the word order in the sentence [15]. The sentLDA model was proposed [16], which utilizes the co-occurrence of words in sentences. But in this case, the terms are still used as the basic units of analysis.

This article presents a conceptually new perspective on topic modeling. It is proposed to use a sentence as a basic unit of analysis, to consider the unit with a complete grammatical and semantically rich linguistic structure. It allows considering the word order information in the text. When a sentence becomes a basic unit of analysis inside a given text corpus, we need to evaluate sentence probabilities to which the topic model is further applied. This article proposes the method for evaluating these probabilities. Based on this evaluation, the topic model assesses the discrete distribution of sentence occurrence within each topic. The most probable sentences are considered as features of the given topic. It provides greater interpretability of resulting topics because a person interprets a set of the most probable sentences much better than a set of loosely coupled terms.

One of the exciting and promising approaches to solving various applied problems in natural language processing is using semantic information implicitly contained in the texts to be analyzed. Operating with word meanings instead of word labels opens up new vast possibilities for reaching better qualitative results. For example, in the process of text analysis, the exact meaning of the analyzed word can be determined by its context surrounding. The more similar the contextual environment of different words is, the more probably these words have similar meanings. Having a special function capable numerically generalize in embeddings contextual information for some words in the text, it is possible to compare different words' senses by comparing their embeddings' relations. Having such a function that transforms words in the text to context-dependent embeddings, it becomes possible to operate with meanings, i.e., carry out a semantic analysis by comparing senses of different text fragments.

In 2018 Google developed the generative transformer BERT model based on neural networks [17,18]. The pre-trained BERT model generates contextual embeddings for every single word or token in the text. The BERT model's generative nature allows obtaining continuous embeddings of words or tokens depending on their contextual environment. Comparing the similarity of the obtained embeddings, it is possible to make conclusions about the semantic similarity of the corresponding words. It opens wide opportunities for introducing an intellectual component into the different tasks of natural language processing.

Using BERT, it becomes possible to embed the meaning shades contained in the text for their subsequent using in solving various applied problems. The BERT also benefits from its context-sensitive nature. We are not vectorizing the word labels as in the case of word2vec [19], but their context-dependent implementations, which can be quite different in meaning in various contexts, even with identical spellings. BERT's use allows to quite successfully solve the problem of homonymy and polysemy [20]. The use of this model provides state-of-the-art results in numerous applications [17,18]. Because this model was trained using texts from the entire Internet. Therefore, this model can be used as a knowledge base about the existing world. It is a kind of Encyclopedia or a model of our reality which we can use as the additional external knowledge source. Using BERT embeddings in the process of topic modeling can be considered a kind of transfer learning. Because using it, we utilize knowledge of the external world inside our analytical process.

A feature of BERT is that by having the possibility to comparing the obtained word embeddings, we can make conclusions about the relative semantic similarity of corresponding words. Nevertheless, the components of these embeddings are not interpretable by themselves. In the proposed approach, the BERT model is used for obtaining sentence embeddings as an averaged embedding overall its constituent words (tokens). Instead of BERT, other similar models to generate context-dependent embeddings for sentences can be used [21,22].

The problem of topic modeling can be considered as the interpretative vectorization of texts. Each text must obtain a corresponding thematic embedding when we perform a topic analysis of the text collection. Each component of this embedding must interpretably determine the relation probability of some text to a certain i -th topic. In other words, we must transform a set of non-interpretative sentence embeddings as text parts to one interpretative thematic embedding of the whole text. As a result of the proposed approach, thematic classification of the text is obtained, i.e., we must solve the classification problem, where the classes are the topics.

Automatic topic labeling is another essential aspect and subtask of topic modeling. Automatic topic labeling is an algorithmic process of generating/selecting phrases or sentences that describe a topic in the best form. This field of topic labeling is poorly studied and developed. A few articles have been published on this task since 2007 [23]. Usually, a relatively simple approach is used in practice: The N most probable terms from the topic are selected in the list form as a label for that topic. These bag-of-words labels are relatively difficult to interpret because they do not have a natural appearance. In the proposed approach for each topic, we select a whole sentence that serves as a natural label. The use of the whole sentences as topic labels significantly increases their interpretability since it is more convenient for a person to interpret an entire sentence than a loosely bound bag of keywords. Thus, using the sentence as a basic unit can help solve the automatic topic labeling problem. In this case, the formation of topics from the sentences and their subsequent labeling becomes more natural. The sentence unit for topic labeling harmoniously coincides with the topic analysis's basic unit.

The developed approach is tested on the News Aggregator data set from the UCI Machine Learning Repository [24,25].

2. Description of the Developed Topic Model

The topic model evaluates the probability of the term occurrence within given topic and the probability of topic occurrence within the given text [1]. The model expresses the estimated unobservable probabilities in terms of the probability of a word occurrence within the text as follows:

$$P(w|d) = \sum_{t \in T} P(w|t) \times P(t|d) \quad (1)$$

The probabilities of term occurrence within the text are estimated based on the term occurrence frequencies in the text collection. Unobservable probabilities estimated as model parameters based on the maximum likelihood principle:

$$\sum_{w,d} \ln P(w|d) = \sum_{w,d} \ln \sum_t P(w|t) \times P(t|d) \rightarrow \max_{P(w|t), P(t|d)} \quad (2)$$

It is necessary to find such values of the model parameters that provide the predicted probabilities close to the observed ones. It is a matrix decomposition problem: we need to find two matrices of lower dimensions, which product in the best way approximates the original frequency matrix.

The article offers a new conceptual perspective on topic modeling. It is based on the idea of using the whole sentence as a structural basic (atomic) unit of analysis. Thus, it is proposed to model the probability of sentence occurrence within text as product of probabilities of sentence occurrence within topic and topic occurrence within text. The model is defined as follows:

$$P(s|d) = \sum_{t \in T} P(s|t) \times P(t|d) \quad (3)$$

In the generative process each individual text is used for generating a topic with given probability distribution $P(t|d)$. Then the obtained topic samples are used for generating a given sentence probability distribution $P(s|t)$. Thus, each individual sentence relates only to one topic. According to the proposed approach we are modeling occurrence probability not of the sentence itself, but of its continuous embedding representation. Thus, it is assumed that thematically close sentences also have relatively close embeddings. In other words, embeddings corresponding to thematically similar sentences are in a multidimensional space in a sphere with a diameter equal to some value. Not similar sentence embeddings are located relatively far from this sphere.

The proposed algorithm is presented below in Algorithm 1.

Let us describe each step of proposed algorithm in more detailed form.

First step—Obtaining Sentence Embeddings.

- input—each sentence of a collection represented as sequence of symbols; BERT model.
- output—each sentence of a collection represented as embedding.

The model (3) defines the generative process of sentence occurrence within the text. To estimate the probability of sentence occurrence, we have to consider each sentence in some representation form. We can select its representation in original form as a sequence of symbols or as some embedding. The representation as embedding is preferable because this representation provides certain advantages. For example, two sentences that use entirely different sets of tokens can have a similar meaning, and their representation as embeddings can embed this common meaning. Different language models can be used to produce embeddings [21,22]. We chose BERT model because it showed state-of-the-art results on many NLP tasks.

Thus for each sentence from the text collection, we need to form a corresponding context-dependent sentence embedding. At first, the text is parsed into sentences. The sentences are then submitted without preprocessing to the BERT mode (Appendix A), thereby maintaining their grammatical, syntactic, and semantic information structures. Each sentence is tokenized on sequence of its constituent words (tokens). The context-dependent word/token embeddings are obtained using the BERT model. Calculating the arithmetic mean of the word/token embeddings that make up this sentence, we obtain the corresponding sentence embedding. In turn token embeddings was obtained by averaging of the last 4 layers in BERT model.

The study of other possible schemes of sentence embedding forming is the task of ongoing separate research. Although used a relatively simple BERT-based sentence vectorization scheme already provides acceptable qualitative results.

Such a context-dependent model ensures that the lexical and semantic structure of the sentence is considered. Due to its continuity and contextual dependence, the chosen model provides sense disambiguation in texts and generates more accurate embeddings. When information about the structure of sentences is preserved, we can use that useful information on subsequent analysis stages, followed by the topic modeling. This preserved information can also be useful during the process of expert interpretation of the obtained results. This information is lost when using other basic analysis units, such as words or n-grams, since texts are reduced to bags of the corresponding basic units.

Algorithm 1 The pseudocode of the developed algorithm.

Input: Texts—text collection;
 k —required number of topics;
 Pre-trained BERT model;
 d —maximum distance to centroid

1. Each Text is parsed into sentences;
 Tokenization of each sentence;
 Original Dataset = Sentence embeddings for each text sentence by BERT.
2. **If** Original Dataset is Big Data **then**
 Random permutation of Original Dataset;
 Target Dataset = Select random subset of Original Dataset;
Else Target Dataset = Original Dataset.
3. L1-normalization of Target Dataset;
 Initial Centroids = $KMeans++(TargetDataset, k)$;
 Centroids, Clusters = $KMeans(TargetDataset, InitialCentroids)$;
4. **If** Target Dataset \neq Original Dataset **then**
 Clusters = Redistribute of all sentence embeddings from the Original Dataset over the obtained Centroids by choosing the closest one.
5. n = number of Texts;
 m = number of Clusters;
 $F = n \times m$ size zero matrix;
For $i = 1, \dots, n$ **do**
 For each sentence from i -th text **do**
 j = index of cluster that sentence was assigned to
 $F[i, j] = F[i, j] + 1$
6. Likelihood Function = Minimal difference between the model probabilities ($P(t|d)$ and $P(s|t)$) and the observed probabilities F ;
 Using matrix F and applying EM-algorithm to find the unknown parameters $P(t|d)$ and $P(s|t)$ of the model (3) maximizing the Likelihood Function.

Output of Algorithm: Model parameters $P(s|t)$ and $P(t|d)$.

Second step—Target Dataset Formation.

- input—sentence embeddings of a text collection.
- output—a subset of sentence embeddings of a text collection.

We obtained sentence embeddings on the previous step. They must be represented in random order. As explained in the next step, it is necessary to group them according to their semantic similarity or topic. For that, we have to apply some clustering algorithm to form sentence clusters. But when processing real data, often represented by large text collection, the number of sentences can be quite large, and their clustering on the next step can require a relatively large amount of computing resources. Therefore, for computational efficiency, it is proposed to cluster not all sentences but a random subset from the entire dataset of sentences. The subset size must be much less than the total number of sentences but much more than the required number of clusters. Original entire dataset of sentences must be randomly permuted before random subset selecting. Let us call this subset of sentences as target dataset. The obtained random subset of the original dataset can be considered as its thinned version with fewer entities inside but saving the original dataset's

spatial shape. After thinning and saving the original dataset's spatial shape, we can expect to obtain similar clustering structures inside the reduced dataset.

If our original dataset is not so big and we have no problem with its processing, then the target dataset is equal to the original one. Otherwise, the target dataset is formed as a random sample from the original one. The sample size must be chosen as a compromise between efficiency and comprehension.

Third step—Clustering of Sentence Embeddings.

- input—a subset of sentence embeddings of a text collection; number of clusters.
- output—clusters of sentence embeddings.
- local criterion—sum of squared distances.

To apply the topic model (3), we must estimate the probability of sentence occurrence within the text. To get its empirical estimate, we have to count frequencies of a sentence occurrence within text collection. If we have used symbolic representation of a sentence, it will lead to a degenerative case when almost all sentences have a rare frequency. To overcome it, we group embeddings of similar sentences into clusters, and each cluster represents, in some sense, one meaning or theme. Thus all sentences that belongs to the same cluster are considered identical; this will provide us with more reliable estimates of sentence frequencies.

Thus at this step, sentence embeddings collected in the target dataset are clustered into a given number of clusters equal to the required number of topics. The purpose of clustering is to use sentence embeddings to assign semantically/thematically similar sentences to a single cluster. We can use various suitable clustering algorithms for solving this task. The minimum sum of squares clustering (MSSC) algorithms (Appendix B) are the most suitable due to their efficiency, simplicity, and good applicability for big data clustering. The most popular representative of such types of algorithms is the k -means algorithm which was used for clustering. The k -means algorithm has many advantages. First of all, it is one of the efficient clustering algorithms. Second, it has an objective function—sum-of-squared distances (SSD) based on which we can estimate the quality of obtained clusters. Third, it has only one parameter k —required number of clusters. This parameter is equivalent to a parameter that determines the number of topics in the topic modeling, so there is a natural connection from topic modeling to k -means. In perspective, we can incorporate in k -means more advanced metaheuristics as J-means [26] that provides unlike standard k -means convergence to the global optimum of SSD. Thus MSSC algorithms class and, in particular, its most famous representative k -means, have the following features: high efficiency, simplicity, the minimum number of parameters, no need for preliminary calculation of the distance matrix, possibility to reach a global optimum of SSD using advanced metaheuristics [27], the possibility of big data processing [28]. All this makes k -means the best choice for clustering the context-sensitive embeddings when solving various NLP tasks.

Before clustering, all sentence embeddings were normalized using $L1$ -norm. The initialization of centroids was carried out by the k -means++ method. In the clustering process, we use the standard Euclidean distance for calculating pairwise distances between sentence embeddings. As a result of clustering for each obtained cluster, we have a corresponding centroid in sentence embedding space.

The value of SSD can be used as the local criterion on the given step. The difference of SSD values at the first k -means iteration and last one is the indicator of cluster quality improvement.

Fourth step—Redistributing of All Sentences Over the Obtained Centroids (only if the reduced dataset was used as the target dataset).

- input—sentence embeddings of a text collection; clusters; clusters' centroids.
- output—updated clusters.

The initial clusters were formed from randomly selected subset of sentences and the most part of sentences were not considered in clustering process. We should take all

sentences into account: otherwise, a lot of information about an actual cluster structure will be lost. After discovering the clustering structures inside the reduced dataset, we can obtain the full dataset clustering results by redistributing the sentence embeddings from the full dataset between the reduced clusters obtained at the third step. As the result we will have more representative clusters.

To form more representative clusters, the following is proposed: let us call initial clusters as X , and updated representative clusters as Y . Clusters Y are initially equal to clusters X . Further, for each sentence we need to find the nearest cluster in X by calculating the distance to the cluster's centroid. If the distance between the sentence embedding and the centroid of the cluster less than some chosen value d , then this sentence is added to the corresponding cluster of Y . This process is performed for every sentence in full dataset. The d value selection effects on size of clusters and level of similarity between sentences allocated to one cluster. The sentences that was not allocated to some clusters are outliers and not considered further.

Fifth step—Calculation of Distribution Matrix of Text Sentences by Clusters.

- input—updated clusters.
- output—matrix F of sentence probabilities.

Topic model (3) is applied to the probability of sentence occurrence within the text. The probabilities are estimated empirically based on their frequencies. If two sentences belong to the same cluster, we consider them as identical.

So, at this step, we are calculating the matrix F of the distribution of text sentences by previously obtained clusters. The matrix F is initially an $n \times m$ size zero matrix, where n —is the number of texts in our collection, and m —is the number of clusters obtained after the previous step of sentence embedding.

For each i th text sentence, we are incrementing the value by one if the considered sentence was previously assigned to the j th cluster. The i th row of matrix F is vector of i th text distribution over sentence clusters. Next, rows of matrix F are normalized.

Sixth step—Applying EM-algorithm to Find the Unknown Parameters of the Model.

- input— F matrix.
- output—estimates of parameters: probability of sentences occurrence within a topic and probability of topics occurrence within a text.
- local criterion—maximum likelihood value.

The expectation-maximization (EM) algorithm (Appendix C) is a standard algorithm used in topic modeling. So using matrix F , we apply the expectation-maximization (EM) algorithm to find the model's unknown parameters that maximize the likelihood function. We need to find such model parameters that provide the minimal difference between the model probabilities and the observed probabilities (the matrix F). Obtained model parameters are the estimations of sentences-in-topics and topics-in-texts occurrence probabilities (Equation (3)). After finishing the specified number of EM -algorithm iterations, the estimates of the model parameters are calculated. The calculated probabilities allow us to determine the most probable clusters for a given topic.

The local criterion—maximum likelihood value is an indicator of convergence of EM algorithm to a local minimum.

Seventh step (optional)—Cluster Labeling.

For each cluster we can select one medoid as the representative sentence whose embedding is closest to the given cluster's centroid. Let us assign this representative sentence as the label of the cluster. As a result, we have a mapping of all clusters to their labels.

3. Dataset and Quality Criterion Estimation

The well known News Aggregator dataset [24,25] hosted on the UCI Machine Learning Repository was used for experimental verification of the developed approach. The dataset has 422,937 links to news publications. This dataset consists of news publications

from 4 categories: business, science and technology, health, entertainment. All news publications within each category are labeled by media events which are called “stories” in the dataset. Each publication is assigned only to one category and to one media event. For the experimental purpose, three subsets of 733, 3338, 9585 publications were selected. All selected publications are related to 4, 22, 82 stories (media events) in the business category. By the fact each media event can be considered as a separate topic, i.e., we have a dataset labeled by topics. Having such labeling in the dataset, we can use it as the reference for the quality estimation of our topic modeling approach.

We are considering the media events in our dataset as regular topics. In our dataset, we know which publications are related to which topics (or media events). Using this information, it is possible to identify the most topical/characteristic words for each reference topic in the dataset. For that, analyzing the publications which belong to the given topic, we calculate for each word the ratio of its frequency in given topic publications to the average frequency in publications of other topics. The top N words with such a ratio are classified as the most characteristic words for the given topic. The example of such type of words selecting for one of the 22 reference topics where N equals to 10 is given in Table 1.

Table 1. The list of top ten most topical words for one of the reference topics in the dataset.

Astra, astrazeneca, pfizer, undervalued, soriot, az, azn, pfe, glaxosmithkline, wyeth.

This topic is related to the company acquisition in pharmacology industry. The above list mainly includes drugs and the names of top pharmaceutical companies.

For estimating our topic model’s quality, the number of most topical words will be counted in each sentence cluster that corresponds to the most probable clusters of each topic. In the best case, all the most topical words of a particular reference topic should present in the separate sentence cluster of the obtained topic model. We need to identify one model sentence cluster with the highest number of reference topical words inside for each set of reference topical words. Such a number is calculated for each reference topic, and its average among all topics is used as the first quality estimation. Every cluster might have some amount of noise sentences inside. For eliminating their influence on quality estimation, only the words having relatively high frequencies in the sentence clusters are considered in the estimation process.

$$\text{Criterion 1} = \frac{1}{|Topics|} \sum_{t \in Topics} \max_{c \in Clusters} \left\{ \sum_{word \in t} [word \in c] \right\} \quad (4)$$

The additional quality criterion can be calculated as follows. According to obtained topic model for each text, we have the vector of its distribution over topics. For every text, the other nearest neighbor text is identified using cosine similarity between their topic vectors. If two nearest neighbor texts are related to the same media event according to the dataset’s reference labeling, it is a correct classification. The misclassification rate will be used as a second quality criterion.

$$\text{Criterion 2} = \frac{1}{|D|} \sum_{d \in D} [label_d = label_{\text{nearest neighbour of } d}] \quad (5)$$

4. Experimental Results

In the first step, for each sentence from each dataset, the corresponding context-dependent embedding was calculated using the BERT model. As a result for the dataset with 22 topics, sentence embeddings were obtained in total of 71,097. The reduced target datasets were formed by selection by one random sentence pro text. That is 3338 sentences with corresponding embeddings were randomly selected. According to the third step of the algorithm, the clustering was applied to the target datasets embeddings. Initial centroids were initialized by k -means++ method using different centroid numbers in the range from

100 to 4000 with step 100. Experimentally was determined that the best quality results were reached using the number of clusters (initial topics) equal to 1000. The value of threshold for a sentence to be included in a cluster (fourth step) was equal 0.1. Table 2 shows an example of four clusters from dataset 2 formed after the clustering step (third and fourth steps of the algorithm).

Table 2. An example of four clusters from dataset 2 formed after the clustering (the third step of the algorithm).

After making a failed bid for AstraZeneca in January, pharmaceutical heavyweight Pfizer is again pursuing a deal for its British rival that would rank among the largest in industry history.
Business secretary says he is committed to maintaining UK's position in pharma industry, but investors welcome bid Battle lines are being drawn over what would be the biggest foreign takeover of a British company, after the pharmaceutical firm AstraZeneca rejected a £60bn approach from its US rival Pfizer.
Pfizer's interest prompted a warning on jobs from Vince Cable but was welcomed by investors, who sent stock in Britain's largest drug maker up by 14%.
In addition, US drugs giant Pfizer was rumored to have tabled a \$100 bn bid for UK-based AstraZeneca, which prompted the FTSE 100-listed company to announce plans to spin off non-core assets.
Yesterday, the UK's benchmark index closed 0.22 percent higher at 6700.16 points with the advance largely due to AstraZeneca (LON: AZN) whose shares soared after Pfizer confirmed its interest in the FTSE 100 company.
U.S. drug maker Pfizer Inc. approached Britain's AstraZeneca Plc two days ago to reignite a potential \$100 billion takeover and was rebuffed, raising investor expectations it will have to increase its offer to close the deal.
AstraZeneca shares were up 11.7 percent at \$76.69 in New York on news of the latest offer, which would be the biggest foreign acquisition of a British company and one of the largest pharmaceutical deals.
Anglo-Swedish drugs giant is spurning the advances of Pfizer after the US drugs giant made a £59 bn takeover offer.
Pfizer Inc. was turned down twice by fellow drug maker AstraZeneca PLC, but the maker of Viagra and Lipitor said Monday that its proposed \$100 billion acquisition makes sense for shareholders of both companies, and it's considering its next steps.
The Pfizer offer led to a transatlantic stampede for pharma shares yesterday, driving a 50 percent spike in the amount of AstraZeneca shares changing hands in London, helping it close 13 per cent.
Bandhan, a microfinance entity, and IDFC are likely to get bank licenses after the Election Commission on Tuesday gave the green signal to RBI to announce new entrants in the sector, ending weeks of uncertainty over the crucial reform measure.
The Reserve Bank of India had sought the commission's approval to issue new bank licenses to ensure the process would not clash with the code of conduct ahead of elections, which prevents decisions that may be deemed as political from being taken by government officials or regulators.
The RBI and the government had initially set a deadline of issuing new licenses by the end of March.
A senior EC official told TOI that the RBI was competent to take its own decisions and the commission agreed that the ongoing polls need not delay its functions as a banking regulator.
Reserve Bank of India governor Raghuram Rajan left policy rates unchanged and signaled that he was waiting to see whether the post-election Budget would take the path of fiscal prudence before deciding on interest rates.
The repo rate, or the interest that banks pay when they borrow money from the RBI to meet their short-term fund requirements, has been left unchanged at 8 percent.
The Reserve Bank of India (RBI) held interest rates today while shifting its liquidity provision to longer-term repurchase operations (repos) as it continues to transform its monetary policy framework.
Liquidity conditions have tightened in March, partly on account of year-end window dressing by banks, though an extraordinary infusion of liquidity by the Reserve Bank has mitigated the tightness.
About 1200 Parisian drivers were blocking the Charles de Gaulle and Orly airports this morning and preventing private car services from picking up passengers, said Nadine Annet, vice president at the FNAT taxi association in France.

Table 2. Cont.

Taxi drivers brought parts of London, Paris and other European cities to a standstill on Wednesday as they protested against new private cab apps such as Uber which have shaken up the industry.
Uber app has caused chaos in London as London's taxi drivers come out in protest.
The capital ground to halt today as London's black taxi drivers took to the streets in protest over Uber.
During a 24-h protest in Madrid, cab drivers surrounded a car suspected of being a private taxi.
Transport in major European cities has been disrupted by strikes affecting taxis and rail services.
There was not a taxi to be found on the streets of Madrid on Wednesday morning after the city's cab drivers began a 24-h strike to protest against online carpooling companies that match individual drivers and passengers.
Taxi drivers sowed traffic chaos in Europe's top cities on Wednesday by mounting one of the biggest ever protests against Uber, a US car service which allows people to summon rides at the touch of a button.
The protest includes a reported 30,000 cab and limo drivers, from London to Paris to Madrid, who are miffed by the same gripes as their American counterparts—namely, that Uber is swiping their business without abiding by any of their rules.
In Spain, the Barcelona and Madrid cab unions represent nearly all the cabs in those cities, and they too have scheduled a protest, this protest will be 24 h and will be tougher on the citizens than the other protests.
The commuters in London, Paris, Madrid, and Berlin faced tough times today as taxi drivers in these cities decided to block the streets of the city to protest against the ride-sharing service Uber.
Bloomberg reported that over 30,000 taxi and limo drivers participating in the protest drive, creating huge trouble for the commuters as it led to massive traffic jams in tourist centers and shopping districts across Europe.
Seth Rollins grabbed the briefcase after Kane pulled Dean Ambrose off the ladder, just as he was about to win, and hit him with a tombstone, for good measure.
Kingston with a dropkick on Swagger followed by boom drop on him as Swagger was laying on top of a ladder.
Rollins went after Ambrose while everybody was out of the ring, which led to Ambrose hitting a double underhook suplex that sent Rollins into a ladder.
Rollins hit both of them with a ladder, but then RVD hit Rollins with a dropkick.
RVD hit Rolling Thunder on Rollins while he was laying on top of a ladder.
Swagger set up the ladder in the corner of the ring on RVD and wanted a superplex, but RVD fought out.
Swagger sent Zigler into a ladder that crashed into Ambrose.
Then Swagger slammed a ladder onto Kingston followed by a Swagger Bomb that crushed Kingston.
Kingston gave Rollins a back body drop that sent him crashing into the ladder that was bridged from the ropes.
With Cesaro near the top of the ladder, Orton yanked him off and hit a RKO off the ladder.

The first topic in Table 2 is related to the company acquisition in pharmacology industry. The second topic is related to policy of Reserve Bank of India. The third topic is about taxi drivers protest against Uber in Europe. The fourth topic is related to event conducted by World Wrestling Entertainment. Analysis of the obtained clusters shows that most of the sentences inside clusters are thematically close. Only a small percent of sentences assigned to the cluster have a weak coupling to its topic.

At the next step, for each cluster, the centroid is calculated, and one sentence closest to it is found. Such selected sentences will be the labels of the clusters. Label selection examples for the previous four clusters are shown in Table 3.

Table 3. An examples of label selection for four clusters from Table 2.

U.S. drugmaker Pfizer Inc. approached Britain’s AstraZeneca Plc two days ago to reignite a potential \$100 billion takeover and was rebuffed, raising investor expectations it will have to increase its offer to close the deal.

Governor Raghuram Rajan says Reserve Bank of India (RBI) should not be in the business of bailing out banks by infusing cash to make up for year-end distortions and the current policy rate has been appropriately set, the central bank chief said post the policy review on Tuesday.

The commuters in London, Berlin, Paris and Madrid faced a day of traffic chaos on Wednesday as taxi drivers mounted one of the biggest protests against the threat of Uber, a U.S. car service which allows people to summon rides at the touch of a button.

Sheamus goes up top but Reigns nails him with a Superman Punch.

The matrix F was calculated. The dimension of F is 3338×1000 , and the EM algorithm was applied to it. Based on the obtained estimates of the sentence occurrence probabilities within a topic, the most probable ones are determined.

Quality Criteria Values. The evaluation of topic modeling quality is performed using obtained model parameters. The estimates of previously defined quality criteria for dataset with 22 topics of topic modeling results are shown in Table 4. The value of the first criterion indicates the quality of the distribution of sentences in topics. In contrast, the value of the second criterion indicates the quality of the distribution of topics in texts. We ran 100 times steps from 3 to 6 of the algorithm and calculated the average and standard deviation of criteria values. When calculating the first criterion, rare words (occurred less than three times in sentence cluster) were considered as outliers.

Table 4. The quality criteria values of dataset 2.

22 Topics	Average Value	Standard Deviation	Max Value
Criterion 1	31.89%	0.02	35.90%
Criterion 2	66.28%	0.02	70.91%

The topic model assigned probabilities of sentence occurrence within the topic. For example, the two most probable clusters with sentences of the first topic were assigned probabilities of 0.14 and 0.12. They are presented in Table 5. These clusters are related to the World Wrestling Entertainment group’s event. Table 6 presents the two most probable clusters of sentences for the second topic. These clusters are related to the Affordable Care Act, well known as Obamacare.

If we run topic modeling based on words, we get the following results on the given labeled dataset: the first criterion equals to 14.95%, and the second is 80.12%. The model assigned probabilities to words. The ten most probable terms of four topics are presented in Table 7.

The topics presented as bag-of-words are less interpretative than topics based on sentences. When we consider a topic as a bag-of-words, we usually have to guess what the topic is, and to make the task easier, we have to connect in mind these words by ourselves. Based on sentences, we can interpret topics in a more detailed way. We do not have to associate words because easy interpretable topics are now presented as coherent syntactic and semantic constructions.

Table 5. Two most probable clusters of sentences of the topic 1. Subset of the first 9 sentences is selected.

Cena climbed up and grabbed the WWE & World Title to win the match after 27 min.

Rusev takes the opportunity to crush his foe and locks in The Accolade for the victory.

Orton goes to belt town, but this time Roman Reigns is back to break the climb and the two rivals scrap it out on the ladder.

Cena takes out Kane and Orton, and wins!

Paige jumps in for another flurry but AJ counters with a roll up for the win.

Jey comes right behind him with a second splash for the win.

They make it back in and Paige applies the submission before turning it into a 2 count.

Kane stands guard and holds the ladder as Rollins climbs up to grab the briefcase for the win.

They raise Rollins' arms and the briefcase in victory.

Kingston knocked Swagger out of the ring with a ladder shot.

Ambrose and Swagger were sent out of the ring.

Ambrose ran back into the ring as Rollins was about to win.

Rusev and Lana entered the ring.

Triple H and Stephanie McMahon made their way down to ringside.

He threw Cesaro and Del Rio out of the ring.

Cena took Del Rio out of the ring.

Orton sent Sheamus out of the ring.

Cena gave Kane the Attitude Adjustment.

Table 6. Two most probable clusters of sentences of the topic 2. Subset of the first 8 sentences is selected.

Before the health-care law, one insurance company held at least half the individual market in 30 states, according to the Kaiser report.

Only 56 percent polled said they plan to purchase health insurance.

As of February 24, the average premium for an individual health plan selected through eHealth without a subsidy was \$274 per month, a 39 percent increase over the average individual premium for pre-Obamacare coverage.

In 2010, when the Affordable Care Act was passed, a single insurer had more than half the individual insurance market in 30 states.

More than 5 million people have enrolled in private health insurance under Obamacare, according to the administration.

Nationally, the number of uninsured people in 2012 was estimated at about 47 million.

Young people are vital to the success of President Barack Obama's signature healthcare law.

Regarding people with preexisting conditions, for instance, they point to the high risk insurance pools, which would be managed and subsidized by states.

They will have to spend thousands of dollars more before their benefits actually take effect.

The individual penalty increases each year through 2016.

Now, he will pay \$22 a month for his health insurance.

For instance, a sprained ankle can cost a person \$220, while charges for a broken arm average nearly \$7700.

If you remain uninsured, you may be liable for a penalty of up to 1 percent of your income.

If you wanted to keep roughly the same premium, you often had to double your out-of-pocket costs.

Premiums were often 25 percent of a person's income.

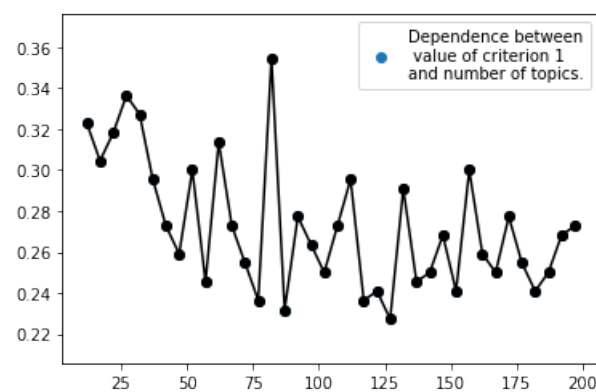
Just over half of uninsured people said they'd started to pay, compared with nearly 9 in 10 of those signing up on exchanges who said they were simply switching from one health plan to another.

Table 7. Four topics of dataset 2 presented as bag of words.

Topic 1	Topic 2	Topic 3	Topic 4
uber	wwe	pfizer	rbi
drivers	match	astrazeneca	inflation
taxi	ladder	uk	banks
london	cena	offer	india
black	him	read	rajan
app	vs	takeover	cent
cab	rollins	bid	monetary
protest	orton	drugs	liquidity
driver	reigns	inc	january
cabs	title	bc	repo

The first criterion determines the quality of obtained topics. The second criterion determines the quality of text embeddings. The first criterion take precedence over the second because the primary purpose of the topic modeling application is to obtain latent topics in texts. In case it is required to get only text embeddings (exception is interpretative text embeddings), there are many other methods. The developed approach shows better results than the standard one in quality of obtained topics, but we have to keep in mind that the two topic models were based on distinct analysis units. As stated above, the developed approach is a new conceptual perspective as sentences are basic units of analysis instead of words. It is an entirely new perspective in topic modeling, and the issue of how to compare topic models based on different bases on analysis requires more elaboration.

The topic model has a parameter—number of topics. The value of parameter has to be initialized before running EM algorithm. For dataset 2 we know in advance that the dataset has 22 topics. To test if the algorithm is robust, we want to investigate how criteria values will change if we didn't know exact number of topics. The Figures 1 and 2 shows that increase in number of topics improves the value of criterion 2 and deteriorates the value of criterion 1 only on few percents.

**Figure 1.** The value of criterion 1 for different values of parameter: number of topics in EM algorithm.

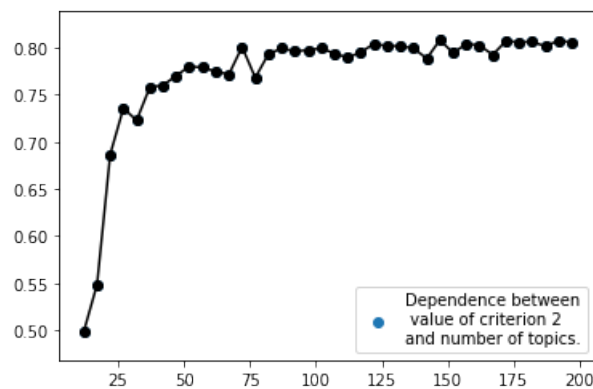


Figure 2. The value of criterion 2 for different values of parameter: number of topics in EM algorithm..

In addition the experiment was conducted for dataset 1 and 3. The dataset 1 has 4 topics, the dataset 3 has 82 topics. The results for the datasets are presented in Tables 8 and 9.

Table 8. The quality criteria values of dataset 1.

4 Topics	Average Value	Standard Deviation	Max Value
Criterion 1	73.55%	0.10	80.00%
Criterion 2	94.12%	0.09	99.18%

The dataset 1 has 733 texts. There were 15,925 sentences. The dimension of matrix F was 733×100 . If we run topic modeling based on words, we get the following results on the given labeled dataset: the first criterion equals to 10.00%, and the second one is 99.59%.

Table 9. The quality criteria values of dataset 3.

82 Topics	Average Value	Standard Deviation	Max Value
Criterion 1	20.28%	0.01	22.31%
Criterion 2	55.40%	0.01	57.45%

The dataset 3 has 9585 texts. There were 192,352 sentences. The dimension of matrix F was 9585×3000 . If we run topic modeling based on words, we get the following results on the given labeled dataset: the first criterion equals to 12.31%, and the second one is 65.56%.

The experimental results prove that the proposed approach can be applied successfully in practice to obtain more interpretive topics.

Applicability for big data. The applicability of the developed approach to big data depends only on the third clustering step. If a proper clustering algorithm is selected to handle a massive dataset, the approach can also be applied to a large data set. For example, the clustering algorithm presented in [28] can be successfully applied to big data. Other steps of the developed approach can be easily extended to big data.

5. Conclusions

The new conceptual approach to the topic modeling problem is presented. Unlike usual techniques, the whole sentence as the basic (atomic) unit of analysis is used. Each sentence was represented by corresponding context-dependent embedding, which was obtained using the BERT model. The main specifics of the proposed approach is assessing the probabilities of the sentence occurrence within texts, based on which the topic model evaluates the probabilities of the sentence occurrences within topics and topics within texts. The developed approach was verified on the News Aggregator data set. The performed numerical experiments indicate that the proposed model is adequate and that the formed topics have a high-level of interpretability.

The developed approach is the fundamentally new statement of topic modeling task because before only n-grams were units that topic models dealt with. The high-level nature of a sentence as a basic unit, its integrity, self-sufficiency combined with the power of contextual embeddings provides more interpretable results than the traditional topic modeling approaches. In addition, since information about syntactic and semantic coherence between words in texts is not lost during modeling, possible positive effects on applications that are based on topic modeling can emerge. So we might expect that precision and effectiveness of methods in applications based on topic modeling will be improved. That is, on the one hand, the approach provides much more interpretability, on the other hand, different approaches that are based on topic modeling might show improvements in results because texts coherence is preserved when the developed topic model is applied.

Moreover, the obtained topics naturally are suitable for automatic topic labeling because sentences generally more unambiguously can determine the topics' scope and sense. Humans better perceive sentences as topic titles than typically used lists of words.

The developed approach is an example of transfer learning in action. Extensive development and wide spread of language models based on neural networks provided improvements of NLP tasks. Transfer learning is used in different applications. It turns out topic modeling is not an exception.

Moreover, the approach indicates practical importance of big data processing methods and of more advanced optimization methods like Variable Neighborhood search to reach possible improvements in effectiveness and results.

The developed algorithm uses BERT model. To train BERT model there should be available large amount of textual data. So the algorithm cannot be applied to low resource languages because of absence of large textual data. In addition, the process of validation of algorithm depends on datasets labeled by topic. Nowadays there are a lot of textual collections but few are labeled by topic. So it is not possible to conduct thorough analysis on different datasets to have more verification power when testing topic models. New datasets labeled by topic can stimulate appearance of more robust conclusions on topic models quality.

Further, it is planned to explore the possibility of improving the developed model. For example, to analyze how selecting a layer in a BERT model affects external criteria, whether how the preprocessing process affects the result. It is also required to study the possibility of using various approaches to context embedding aggregation to utilize grammatical and syntactic information. The next interesting question is how the choice of the clustering algorithm affects the results. Moreover, abstractive summarization algorithms' application to obtained sentence clusters can be considered for the automatic generation of topic labels.

Author Contributions: Conceptualization, O.K., R.M. and N.M.; methodology, O.K., R.M. and N.M.; software, O.K.; validation, O.K.; formal analysis, O.K. and R.M.; investigation, O.K. and R.M.; resources, O.K.; data curation, O.K.; writing—original draft preparation, O.K. and R.M.; writing—review and editing, N.M.; visualization, O.K.; supervision, R.M. and N.M.; project administration, R.M.; funding acquisition, R.M. All authors have read and agreed to the published version of the manuscript.

Funding: The work was funded by the Committee of Science of Ministry of Education and Science of the Republic of Kazakhstan under the grant AP09259324.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: News Aggregator dataset [24,25] hosted on the UCI Machine Learning Repository was used to conduct the experiment.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. BERT

Bidirectional Encoder Representations from Transformers (BERT) is a Transformer-based machine learning technique for natural language processing (NLP) [17,18]. BERT was developed in 2018 by Google. The original English-language BERT has two models: (1) the BERT BASE: 12 Encoders with 12 bidirectional self-attention heads, and (2) the BERT LARGE: 24 Encoders with 24 bidirectional self-attention heads. Both models are pre-trained from unlabeled data extracted from the BooksCorpus with 800 M words and English Wikipedia with 2500 M word.

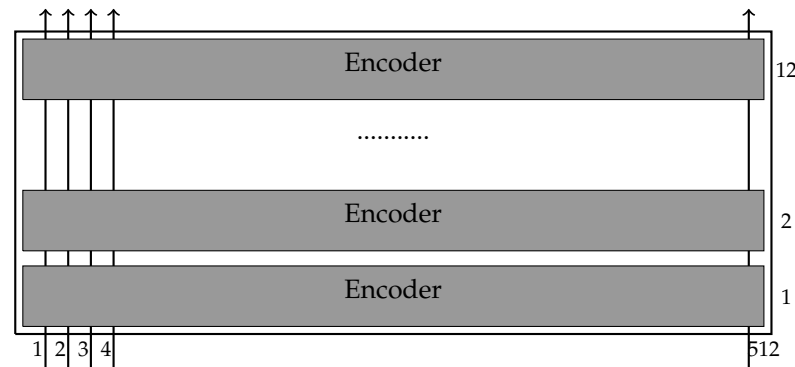


Figure A1. The BERT base architecture.

BERT is based on Transformer's attention mechanism but includes only one of its mechanism—an encoder. The attention mechanism provides learning of contextual relations between words (or tokens) in a text. BERT is stack of bidirectional Transformer encoder layers which consist of multiple self-attention heads. For every input token in a sequence, each head computes key, value and query vectors, used to create a weighted representation. The outputs of all heads in the same layer are combined and run through a fully-connected layer. Each layer is wrapped with a skip connection and followed by layer normalization.

The conventional workflow for BERT consists of two stages: pre-training and fine-tuning. BERT uses two training strategies:

Masked language modeling (MLM). Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. The prediction of the output words requires:

- Addition of a classification layer on top of the encoder output.
- Calculation of the probability of each word in the vocabulary using softmax.

The BERT loss function takes into consideration only the prediction of the masked values and ignores the prediction of the non-masked words.

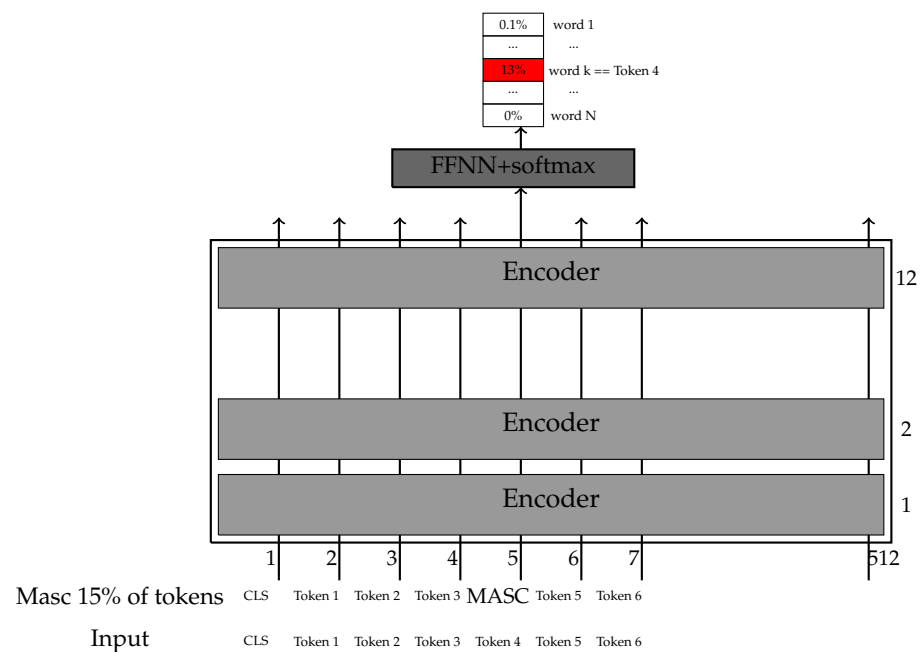


Figure A2. The training of BERT model using Masked language modeling strategy. Use the output of the masked word's position to predict the masked word.

Next Sentence Prediction (NSP). The model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original text. During training, 50% of the inputs are a pair in which the second sentence is the subsequent sentence, while in the other 50% a random sentence from the text collection is chosen as the second sentence.

The input is preprocessed in the following way before entering BERT:

- A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence.
- A sentence embedding indicating sentence A or sentence B is added to each token.
- A positional embedding is added to each token to indicate its position in the sequence.

To predict if the second sentence is indeed connected to the first, the following steps are performed:

- The entire input sequence goes through the model.
- The output of the [CLS] token is transformed into a 2×1 shaped vector, using a simple classification layer.
- Calculation of the probability using softmax.

When training the BERT model, Masked LM and Next Sentence Prediction are trained together, with the goal of minimizing the combined loss function of the two strategies.

In fine-tuning for downstream applications, one or more fully-connected layers are typically added on top of the final encoder layer.

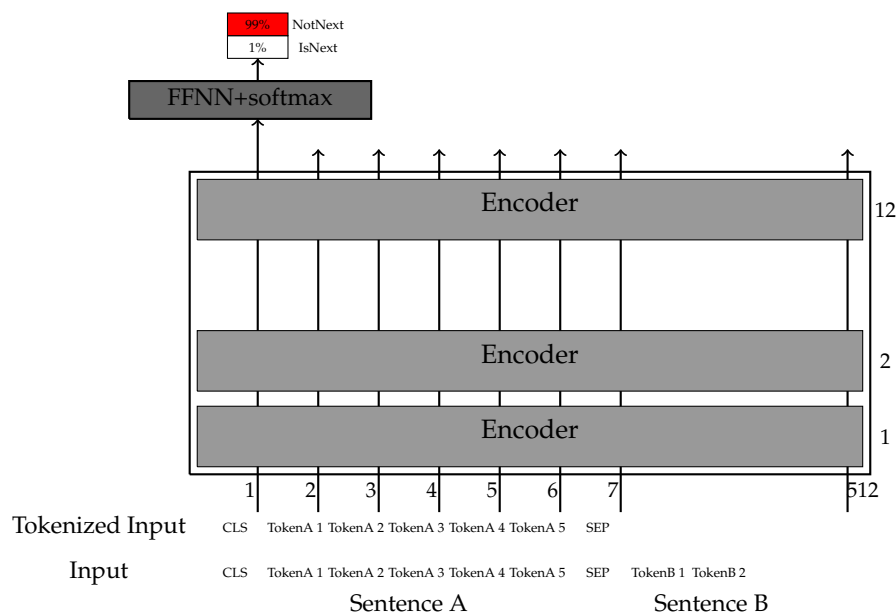


Figure A3. The training of BERT model using Next Sentence Prediction (NSP). Prediction of the probability that sentence B follows sentence A.

Appendix B. Minimum Sum of Squares Clustering (MSSC)

In a clustering problem, we are given a set $X = \{x_1, \dots, x_n\}$ of n samples, where each x_i is represented as a point in \mathbb{R}^m with coordinates (x_{i1}, \dots, x_{im}) and we seek to partition X into k disjoint clusters $C = (C_1, \dots, C_k)$, so as to minimize the sum of squares clustering (MSSC) [27]. In MSSC we aim to form the clusters and find a center $c_j \in \mathbb{R}^m$ for each cluster, in such a way that the sum of the squared Euclidean distances of each point to the center of its associated cluster is minimized.

The mathematical formulation of MSSC is the following:

$$\sum_{i=1}^n \sum_{j=1}^k b_{ij} \|x_i - c_j\|^2 \rightarrow \min$$

$$\sum_{j=1}^k b_{ij} = 1, \quad b_{ij} \in \{0, 1\}$$

$$c_j \in \mathbb{R}^m, \quad j \in 1, \dots, k$$
(A1)

For each sample and cluster, the binary variable b_{ij} takes the value 1 if sample i is assigned to cluster j and 0 otherwise. The variables $c_j \in \mathbb{R}^m$ represent the positions of the centers. In the objective, $\|\cdot\|$ represents the Euclidean norm.

For general k and m , MSSC is NP-hard. Optimal MSSC solutions are known to satisfy at least two necessary conditions:

Property A1. In any optimal MSSC solution, for each $j \in \{1, \dots, k\}$, the position of the center c_j coincides with the centroid of the points belonging to C_k :

$$c_j = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$
(A2)

Property A2. In any optimal MSSC solution, each sample x_i is associated with the closest cluster $C_{k_{\min}(i)}$ such that:

$$k_{\min}(i) = \underset{j}{\operatorname{argmin}} \|x_i - c_j\|$$
(A3)

These two properties are fundamental to understand the behavior of various MSSC algorithms. One of these is k -means algorithm.

The k -means algorithm was developed by Lloyd in 1957. The algorithm iteratively modifies an incumbent solution to satisfy first Property A1 and then Property A2, until both are satisfied simultaneously. An important part of the k -means algorithm is the choice of initial centers.

K -means operates as follows [26]. A starting partition of $X = \{x_1, \dots, x_n\}$ into clusters $C = (C_1, \dots, C_k)$ is chosen and centroids $c_j = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$ of its clusters are calculated. Then each object x_i is reallocated to its nearest centroid c_j ; if no change in reallocations occurs, the algorithm stops with a local solution. Otherwise, the centroids are recalculated and the procedure iterated.

In the most simplified implementation of the k -means algorithm, the initialization of the initial values of centroids is carried out randomly. The convergence rate and the accuracy of the k -means algorithm, in the sense of the SSD criterion minimization, can be significantly increased if, instead of random initialization of centroids, more advanced heuristics will be used [29]. One such advanced heuristic initialization is k -means++. K -means++ works as follows [30]: the first cluster center c_1 is chosen uniformly at random from the clustered entities. Each next cluster center c_j is chosen from the remaining entities with probability proportional to its squared distance from the entity's closest existing cluster center. Although initialization based on k -means++ takes more time than random initialization, correct initialization provides a faster convergence rate of the algorithm. That is, it takes fewer iterations of the k -means algorithm to reach a local minimum. Moreover, often k -means++ provides the more minimal value of SSD objective function than random initialization of centroids. So k -means with k -means++ initialization provides more efficient packing of objects into the clusters.

Appendix C. General EM-Algorithm

The problem of maximizing the marginal likelihood of a probabilistic model in which there are some observed variables X , latent variables Z and parameters Ω is the following:

$$\log p(X|\Omega) \rightarrow \max_{\Omega} \quad (A4)$$

Let $q(Z)$ be an arbitrary density function. Then:

$$\begin{aligned} \log p(X|\Omega) &= \int q(Z) \log p(X|\Omega) dZ = \int q(Z) \log \frac{P(X, Z|\Omega)}{p(Z|X, \Omega)} dZ = \\ &= \int q(Z) \log \frac{p(X, Z|\Omega)}{q(Z)} \frac{q(Z)}{p(Z|X, \Omega)} dZ = \int q(Z) \log p(X, Z|\Omega) dZ - \\ &- \int q(Z) \log q(Z) dZ + \int q(Z) \log \frac{q(Z)}{p(Z|X, \Omega)} dZ = \\ &= L(q, \Omega) + \text{KL}(q(Z)||p(Z|X, \Omega)) \end{aligned} \quad (A5)$$

The Kullback-Leibler divergence $\text{KL}(q(Z)||p(Z|X, \Omega))$ estimates distance between two distributions. The Kullback-Leibler divergence is non-negative, asymmetric and equal to zero if and only if distributions are equal.

Since $\text{KL}(q(Z)||p(Z|X, \Omega))$ is nonnegative, the term $L(q, \Omega)$ is lower bound on the value of $\log p(X|\Omega)$. Then instead of maximizing $\log p(X|\Omega)$ over Ω , it is proposed to maximize the lower bound $L(q, \Omega)$ with respect to q and Ω . Thus, the main idea behind the EM algorithm is to iteratively repeat two steps:

$$\begin{aligned} 1. L(q, \Omega) &\rightarrow \max_q \\ 2. L(q, \Omega) &\rightarrow \max_{\Omega} \end{aligned} \quad (A6)$$

At the first step maximizing $L(q, \Omega)$ over q is equivalent to minimizing $KL(q(Z)||p(Z|X, \Omega))$, because $\log p(X|\Omega)$ is independent of q . The minimum is 0 and is reached at $q(Z) = p(Z|X, \Omega)$. Therefore, if it is possible to write out analytically the distribution $p(Z|X, \Omega)$, then this distribution should be taken as q and $\log p(X|\Omega)$ will be the exact lower bound.

At the second step:

$$L(q, \Omega) = \int q(Z) \log p(X, Z|\Omega) dZ - \int q(Z) \log q(Z) dZ \rightarrow \max_{\Omega} \iff \int q(Z) \log p(X, Z|\Omega) dZ \rightarrow \max_{\Omega} \tag{A7}$$

since the second term is independent of Ω . The first term corresponds to the expected value

$$\int q(Z) \log p(X, Z|\Omega) dZ = \mathbf{E}_q \log p(X, Z|\Omega) \tag{A8}$$

Thus, the EM algorithm consists of alternating two types of steps: E-step (Expectation) corresponds to preparation for calculating the expected value, M-step (Maximization)—maximizing the expected value of logarithm of joint probability of observed variables X and latent variables Z given parameters.

$$\begin{aligned} \mathbf{E}\text{-step} : & KL(q(Z)||p(Z|X, \Omega)) \rightarrow \min_q \iff q(Z) = p(Z|X, \Omega) \\ \mathbf{M}\text{-step} : & \mathbf{E}_{q(Z)} \log p(X, Z|\Omega) \rightarrow \max_{\Omega} \end{aligned} \tag{A9}$$

The sequence of parameter values obtained during iterations of the EM-algorithm gives a non-decreasing sequence of values of $L(q, \Omega)$, which is the lower bound for $\log p(X|\Omega)$.

In topic modeling topics (T) are latent variables, words (W) occurrences in texts (D) are observable variables. The unknown parameters of the model $p(w_i|t_i)$ and $p(t_i|d_i)$ are denoted by ϕ_{wt} and θ_{td} . The marginal distribution $p(T|W, D, \Omega)$ on E step is evaluated by following:

$$\begin{aligned} p(T|W, D, \Omega) &= \prod_{i=1}^N p(t_i|w_i, d_i, \Omega), \text{ where } p(t_i|w_i, d_i, \Omega) = \\ p(t_i|w_i, d_i, \Omega) &= \frac{p(w_i|t_i, d_i, \Omega)p(t_i|d_i, \Omega)}{\sum_{t=1}^T p(w_i|t, d_i, \Omega)p(t|d_i, \Omega)} = \{p(w_i|t_i, d_i, \Omega) = p(w_i|t_i, \Omega)\} \\ &= \frac{p(w_i|t_i, \Omega)p(t_i|d_i, \Omega)}{\sum_{t=1}^T p(w_i|t, \Omega)p(t|d_i, \Omega)} = \frac{\phi_{wt}\theta_{td}}{\sum_{t=1}^T \phi_{wt}\theta_{td}} \end{aligned} \tag{A10}$$

On the M-step we consider the following optimization problem:

$$\begin{aligned} \mathbf{E}_{p(T|W, D, \Omega)} \log p(W, D, T|\Omega) &= \sum_{i=1}^N \mathbf{E}_{p(T|W, D, \Omega)} \log p(w_i, d_i, t_i|\Omega) = \\ &= \sum_{i=1}^N \mathbf{E}_{p(T|W, D, \Omega)} \log p(d_i)p(w_i|t_i, \Omega)p(t_i|d_i, \Omega) = \\ &= \sum_{i=1}^N \mathbf{E}_{p(T|W, D, \Phi, \Theta)} (\log \phi_{wt} + \log \theta_{td_i}) + const \rightarrow \max_{\phi, \theta} \end{aligned} \tag{A11}$$

The solution to the optimization problem is:

$$\begin{aligned}
 \phi_{wt} &= \frac{n_{wt}}{\sum_{w=1}^W n_{wt}} \\
 \theta_{td} &= \frac{n_{td}}{\sum_{t=1}^T n_{td}} \\
 n_{wt} &= \sum_{i=1}^N [w_i = w] p(t_i = t | w_i, d_i) \\
 n_{td} &= \sum_{i=1}^N [d_i = d] p(t_i = t | w_i, d_i)
 \end{aligned} \tag{A12}$$

Iterationally repeating E-step (A10) and M-step (A12) we estimate the unknown parameters of the model.

References

- Blei, D. Probabilistic topic models. *Commun. ACM* **2012**, *55*, 77–84. [\[CrossRef\]](#)
- Boyd-Graber, J.; Hu, Y.; Mimno, D. Applications of topic models. *Found. Trends Inf. Retr.* **2017**, *11*, 143–296. [\[CrossRef\]](#)
- Reisenbühler, M.; Reutterer, T. Topic modeling in marketing: Recent advances and research opportunities. *J. Bus. Econ.* **2019**, *89*, 327–356.
- Liu, L.; Tang, L.; Dong, W.; Yao, S.; Zhou, W. An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus* **2019**, *5*, 1608. [\[CrossRef\]](#) [\[PubMed\]](#)
- Yanina, A.; Golitsyn, L.; Vorontsov, K. Multi-objective topic modeling for exploratory search in tech news. In Proceedings of the Communications in Computer and Information Science, vol 789. AINL-6: Artificial Intelligence and Natural Language Conference, St. Petersburg, Russia, 20–23 September 2017; pp. 181–193.
- Mukhamediev, R.; Yakunin, K.; Mussabayev, R.; Buldybayev, T.; Kuchin, Y.; Murzakhmetov, S.; Yelis, M. Classification of Negative Information on Socially Significant Topics in Mass Media. *Symmetry* **2020**, *12*, 1945. [\[CrossRef\]](#)
- Yakunin, K.; Ionescu, G.; Murzakhmetov, S.; Mussabayev, R.; Filatova, O.; Mukhamediev, R. Propaganda Identification Using Topic Modeling. *Procedia Comput. Sci.* **2020**, *178*, 205–212.
- Yakunin, K.; Mukhamediev, R.; Mussabayev, R.; Buldybayev, T.; Kuchin, Y.; Murzakhmetov, S.; Yunussov, R.; Ospanova, U. Mass Media Evaluation Using Topic Modeling. *Commun. Comput. Inf. Sci.* **2020**, *1242*, 165–178.
- Cristani, M.; Tomazolli, C.; Olivieri, F. Semantic social network analysis foresees message flows. In Proceedings of the 8th International Conference on Agents and Artificial Intelligence, ICAART, Roma, Italy, 24–26 February 2016; pp. 296–303.
- Hoffmann, T. Probabilistic latent semantic analysis. In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence—UAI, Stockholm, Sweden, 30 July–1 August 1999; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1999; pp. 28–296.
- Blei, D.; Ng, A.; Jordan, M. Latent Dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
- Apishev, M.; Vorontsov, K. Learning topic models with arbitrary loss. In Proceedings of the 26th Conference of FRUCT (Finnish-Russian University Cooperation in Telecommunications) Association, Yaroslavl, Russia, 23–25 April 2020; pp. 30–37.
- Kohedykov, D.; Apishev, M.; Golitsyn, L.; Vorontsov, K. Fast and modular regularized topic modeling. In Proceedings of the 21st Conference of FRUCT (Finnish-Russian University Cooperation in Telecommunications) Association, Helsinki, Finland, 6–10 November 2017; pp. 182–193.
- Ianina, A.; Vorontsov, K. Regularized multimodal hierarchical topic model for document-by document exploratory search. In Proceedings of the 25th Conference Of FRUCT (Finnish-Russian University Cooperation in Telecommunications) Association, Helsinki, Finland, 5–8 November 2019; pp. 131–138.
- Pagliardini, M.; Gupta, P.; Jaggi, M. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv* **2017**, arXiv:1703.02507.
- Balikas, G.; Amini, M.; Clausel, M. On a topic model for sentences. In Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, 17–21 July 2016; pp. 921–924.
- Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805v1.
- Rogers, A.; Kovaleva, O.; Rumshisky, A. A primer in BERTology: What we know about how BERT works. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 842–866. [\[CrossRef\]](#)
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 3111–3119.

20. Wiedemann, G.; Remus, S.; Chawla, A.; Biemann, C. Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. In Proceedings of the Konferenz zur Verarbeitung natürlicher Sprache/Conference on Natural Language Processing (KONVENS), Erlangen, Germany, 9–11 October 2019.
21. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana, 1–6 June 2018; Volume 1 (Long Papers), pp. 2227–2237.
22. Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 328–339.
23. Bhatia, S.; Lau, J.; Baldwin, T. Automatic labeling of topics with neural embeddings. In Proceedings of the 26th COLING International Conference on Computational Linguistics, Osaka, Japan, 11–16 December 2016; pp. 953–963.
24. News Aggregator Dataset. Available online: <https://archive.ics.uci.edu/ml/datasets/News+Aggregator> (accessed on 12 April 2021).
25. Gasparetti, F. Modeling user interests from web browsing activities. *Data Min. Knowl. Discov.* **2017**, *31*, 502–547. [[CrossRef](#)]
26. Hansen, P.; Mladenović, N. J-Means: A new local search heuristic for minimum sum of squares clustering. *Pattern Recognit.* **2001**, *34*, 405–413 [[CrossRef](#)]
27. Gribel, D.; Vidal, T. HG-means: A scalable hybrid genetic algorithm for minimum sum of squares clustering. *Pattern Recognit.* **2019**, *88*, 569–583. [[CrossRef](#)]
28. Krassovitskiy, A.; Mladenovic, N.; Mussabayev, R. Decomposition/Aggregation K-means for Big Data. In *International Conference on Mathematical Optimization Theory and Operations Research (MOTOR 2020)*; Communications in Computer and Information Science (CCIS) Book Series; Springer: Cham, Switzerland, 2020; Volume 1275, pp. 409–420.
29. Franti, P.; Sieranoja, S. How much can k-means be improved by using better initialization and repeats?. *Pattern Recognit.* **2019**, *93*, 95–112. [[CrossRef](#)]
30. Arthur, D.; Vassilvitskii, S. K-means++: The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2007; pp. 1027–1035.