

Search and classify topics in a corpus of text using the latent dirichlet allocation model

Orlando Iparraguirre-Villanueva¹, Fernando Sierra-Liñan², Jose Luis Herrera Salazar³,
Saul Beltozar-Clemente⁴, Félix Pucuhuayla-Revatta⁵, Joselyn Zapata-Paulini⁶,
Michael Cabanillas-Carbonell⁷

¹Facultad de Ingeniería y Arquitectura, Universidad Autónoma del Perú, Lima, Perú

²Facultad de Ingeniería, Universidad Privada del Norte, Lima, Perú

³Facultad de Ingeniería, Ciencias y Administración, Universidad Autónoma de Ica, Lima, Perú

⁴Dirección de Cursos Básicos, Universidad Científica del Sur, Lima, Perú

⁵Facultad de Ingeniería, Universidad Tecnológica del Perú, Lima, Perú

⁶Escuela de Posgrado, Universidad Continental, Lima, Perú

⁷Vicerrectorado de Investigación, Universidad Privada Norbert Wiener, Lima, Perú

Article Info

Article history:

Received Sep 6, 2022

Revised Nov 13, 2022

Accepted Nov 18, 2022

Keywords:

Classify

Discovering

Latent dirichlet allocation

Text corpus

Topics

ABSTRACT

This work aims at discovering topics in a text corpus and classifying the most relevant terms for each of the discovered topics. The process was performed in four steps: first, document extraction and data processing; second, labeling and training of the data; third, labeling of the unseen data; and fourth, evaluation of the model performance. For processing, a total of 10,322 "curriculum" documents related to data science were collected from the web during 2018-2022. The latent dirichlet allocation (LDA) model was used for the analysis and structure of the subjects. After processing, 12 themes were generated, which allowed ranking the most relevant terms to identify the skills of each of the candidates. This work concludes that candidates interested in data science must have skills in the following topics: first, they must be technical, they must have mastery of structured query language, mastery of programming languages such as R, Python, java, and data management, among other tools associated with the technology.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Saul Beltozar-Clemente

Dirección de Cursos Básicos, Universidad Científica del Sur

Panamericana Sur Km 19, Villa el Salvador, Lima, Perú

Email: sbeltozar@cientifica.edu.pe

1. INTRODUCTION

Topic modeling is relatively new and is applied to explore and predict discrete data. When it comes to finding out what a large text corpus is about, it is impossible to read and summarize them manually. Latent topics are extracted from a corpus of documents using the topic modeling technique of natural language processing (NLP) [1], as well as texts with a large vocabulary, these models have proven to be very effective [2]. However, finding the correct number of latent themes in a corpus of the text remains a problem [3]. There are many methods of topic modeling, such as latent semantic analysis [4], [5] the matrix factorization does not refusal (NMF) [6] that attempt to model latent topics as probability distributions or as a set of vectors in topic space by implicitly asserting that the number of topics is known in advance.

The simplest approach to analyzing textual documents is to use a vector space model, which considers documents as vectors of word frequency [7]. A vector space model considers the terms as dimensions in a high-dimensional space so that each document is represented by a point in that space [8], the latent dirichlet

allocation (LDA) model considers topics as multinomial distributions over words, and assumes that the documents are sampled from a random mixture of these topics [8], [9]. The LDA model considers a document as a mixture of topics and is able to generate the words as long as it is assigned the latent variables, i.e., it fits the generative model to the words in the documents. This is achieved by sampling words in the documents and counting sentences in the topics as a result of running LDA on a time instance, generating weights for the next step [10].

In this work, the LDA model is used as a basis. LDA considers a supervised classification of qualitative variables in which two or more groups are known a priori, and new observations are classified into one of them according to their characteristics. Using Bayes' theorem, the probability that an observation given certain conditions, belongs to each of the classes of the qualitative variable is estimated, then the observation is assigned so that the probability is higher. This work is focused on technical specialists in technology, who seek to solve the problem of finding topics in large bags of raw text. Likewise, it aims to discover topics in a corpus of text, then classifies the most relevant terms for each of the discovered topics. We work with a set of documents extracted from the web. From this dataset, we analyze the skills of candidates interested in data science and seek to understand what groups of skill sets exist.

2. RELATED LITERATURE

LDA is a probabilistic generative statistical model that assumes that each text is a distribution of topics and each topic is a distribution of words. As a result, given a text, the model seeks to determine the proportion of one variable given the value of another variable in order to maximize the parameters of the generative model [11], [12]. LDA is a topic modeling method widely used by academics and researchers in text classification. Some related work is reviewed below.

At work [13] the authors used three different scenarios to evaluate the discovery of semantic information in various remote sensing applications, including optical and synthetic aperture radar data at different spatial resolutions. Recently, in [14] explored the structure of social discourse on aging in Korea by analyzing newspaper articles on aging. Similarly, at [15], [16] proposed a novel method combining K-Means clustering and LDA, with the aim of generating multi-document text summaries based on extractive topic modeling for news documents. Similarly, at [8] proposed a measure to identify the correct number of topics and we provide empirical evidence in favor of a better accuracy and classification of the number of topics that are naturally present in the corpus. As well as, at [17] the authors investigated how to resolve the difficulties of feature selection and categorization, in [18] used the LDA method to categorize customer reviews. Also, in [19] evaluated the performance of the LDA model, where they concluded that, it can be improved by using some auxiliary features instead of limiting it to some extent, achieving a new innovation and clustering service with better efficiency and higher accuracy through word2vec. Similarly [20], [21] proposed a model using an unsupervised approach for term extraction and integrating a regular expression constraint condition, which makes the subject more meaningful and interpretable from a limited increase in the dimensions of the vocabulary.

If we are interested in making fewer incorrect predictions, we can consider decreasing the a posteriori probability limit for the decision limit. For example, [22] studied the performance of online LDA in several ways, including fitting a 100-topic model to 3.3 million Wikipedia articles in a single pass. Where it is shown that on-line LDA determines thematic models as well as those determined with a batch variational Bayes algorithm, in a short time. The same as, in [9] proposed a topic model that automatically captures topic patterns and identifies emerging topics from text streams and their changes over time. This approach allows the topic modeling framework, i.e., the LDA model, to work online to build an updated model when a new document appears incrementally.

Likewise, in [23] proposed a hybrid recommendation model using LDA-based topic filtering. This model is an extension of LDA, where words correspond to user and item characteristics and are considered suitable for dealing with startup problems, as they provide predicted ratings for new users and items through their latent dimension in [1], [24] conducted comparative experiments between LDA and latent semantic analysis (LSA), finding different points; LDA learns descriptive topics better, while LSA is better at creating a compact semantic representation of documents and words in a corpus. Finally, [25] modified the LDA algorithm to work with graph data instead of text corpora. These modifications reflect the differences between real-world graph data and text corpora. While previous studies considered many aspects of topic modeling, LDA [8] is a supervised classification method and it is the most relevant work for our approach identifies the correct number of topics offering empirical evidence in favor of accuracy and classification that are naturally present in the corpus.

3. METHOD

This section shows the development of the terminology of the LDA method, using Bayes' theorem, and the detailed implementation of the case study that seeks to discover topics in a text corpus and then automatically classify the most relevant terms for each of the discovered topics. LDA is a probabilistic generative model of supervised classification of qualitative variables in which more than one group is known a priori and new observations are classified into one of them according to their characteristics. Using Bayes' theorem, the probability that an observation, given a certain value, belongs to a qualitative variable, $P(Y=k | X = x)$, is estimated.

Finally, the observation is assigned to class k so that the predicted probability is higher. Let us now consider Bayes' theorem. Considering two events, A and B , Bayes' theorem states that the probability of B occurring is that $P(B|A)$ is equal to the probability of A and B occurring at the same time (AB) divided by the probability of occurrence A . The equation is shown below. $P(B|A)=P(AB)/P(A)$.

The ability of the LDA to correctly classify observations depends on how efficient the estimates are [2]. The closer to the true value, the closer the Bayes LDA classifier is to the true value [26]. LDA is a three-level hierarchical model, where each element is modeled on an underlying set of probabilities. Text modeling, probabilities, and estimates provide an explicit representation of a document [2], from the training of text documents using a generalization of algorithms to maximize expectations [27]. Probability is considered inadequate for Bayesian inference, since density functions are similar in form to likelihood, and from this, closed distributions are formed under-sampling in several directions [28]. However, in the case of the prior probability, the estimation is usually straightforward, the probability that any observation belongs to a given class is equal to the number of observations of that class divided by the total number of observations. Likewise, if the estimate is not so straightforward and certain assumptions are required to obtain it, it is normally distributed to estimate its value. The calculation of the LDA model estimates is shown in (1) and (2).

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_1} x_i \quad (1)$$

$$\sigma_k = \frac{1}{N-k} \sum_{k=1}^k \sum_{i:y_1} (x_i - \hat{\mu}_k)^2 \quad (2)$$

Where, $\hat{\mu}_k$ is the mean of the observations of the group k , σ_k is the weighted average of the sampling variances of the k classes with respect to total sample sizes.

The following conditions must be met for the LDA to be valid: a) Each part of the model predictor is distributed broadly across each class of response variable b) The variance of the predictor is the same for all classes of response variables. When the normality condition is not met, LDA loses accuracy, but it can still arrive at relatively good classifications.

For LDA accuracy, one has to evaluate how efficient the resulting classification is. Confusion matrices are one of the best ways to evaluate the accuracy of an LDA model. They show many true positives, true negatives and false positives. In the run, the LDA method searches for decision boundaries that are closest to the Bayes classifier, which, by definition, has the lowest total error ratio among all classifiers. Therefore, LDA tries to achieve as few misclassifications as possible, but there is no difference between false positives and false negatives.

In Figure 1, each word is associated with a latent theme; it is stated as Z . Now, this assignment of Z to a topic gives a distribution of words present in the corpus, represented by θ . Finding the best possible representation of the topic-word matrix and the document-topic matrix is the ultimate goal of LDA to determine the most optimized distribution. Since LDA assumes that documents are a set of topics and topics are a set of words, LDA works backwards from the document to identify which topics would have originated these documents and which words would have originated these topics. Since we now understand how LDA works, let's look at the steps for implementing LDA: a) data processing, b) data tagging and training, c) tagging of unseen data, and d) evaluation of the model's performance.

3.1. Data processing

For the case study, 10322 thousand of resumes were collected from the web. In addition, personal data were masked from the dataset, taking into account the degree of sensitivity this represents. From this dataset, we will analyze the skills of candidates interested in data science and the groups of sets formed from the skills. Each document in the corpus should be represented as a list of words in the processed data. Figure 2 presents the flow chart that follows the data processing.

For processing, the first step is the preparation of the data set in a text file, followed by data extraction, loading, and transformation, for which the Python programming language is used, supported by the stringr library, a preliminary cleaning is performed, and then the text file is converted into csv format. Table 1 shows the plain text file.

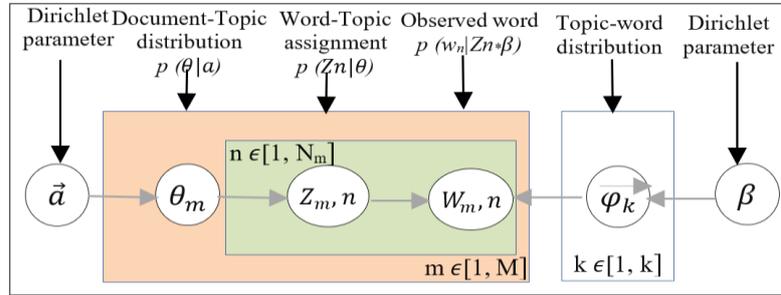


Figure 1. LDA space and its data set



Figure 2. Data processing flowchart

Table 1. Raw data prior to processing

Id	Specialist	Skills
1	Data Scientist Intern	['TECHNICAL SKILLS\xa0', , '\xa0', , '• R • Tableau • Machine Learning\xa0', , '• D3.js • SQL, PostgreSQL, pgadmin 4 • JavaScript\xa0', , '• Python • HTML/CSS • Statistics, Probability']
2	Junior Data Scientist	['TECHNICAL SKILLS\xa0', , 'Languages Java, C, C++, Python, R, Scala, SQL\xa0', , 'Web Services SOAP, REST\xa0', , 'Web Technologies HTML, CSS, JavaScript, PHP\xa0', , 'Database DB2, MySQL\xa0', , 'Software Android Studio, Eclipse, IntelliJ IDEA, NetBeans, GIT']
3	Data Scientist	['TECHNICAL SKILLS\xa0', , '• Proficient: Java, Haskell, Python, R, SQL, C, C++, Matlab, Excel, VBA\xa0', , '• Familiar with: JMP, Tableau, ArcGIS, Weka, TensorFlow']

Table 1 presents the text file data before processing. Subsequently, the Sklearn and Gensim packages and libraries are applied, the latter capable of running multiple cores. Therefore, it has a better performance. It is at this stage that the maximum number of columns is set. With the enable_notebook function (), the automatic D3 display of the prepared model data is enabled. Similarly, with the function clean_up_spacy () The cleaning process is started for each document, among other functions. As shown in Figure 3.

```

1 pd.set_option(arg,400)
3 pyLDAvis.enable_notebook ()
4 def clean_up_spacy(text):
5     text_out = set ()
6     doc= nlp(text)
7     for token in doc:
8         if len (token)<count and token.is_punct is False:
9             if token.text!= "":
10                 text_out.add(token.text)
11 text_out = list(text_out)
12 return text_out
    
```

Figure 3. Cleaning code and tokenization of the LDA model

Next, we proceed with the loading of the pre-trained LDA model. To do so, we use the functions: `load()`, `dictionary.load()`, `mmCorpus()`, which allow us to load the document, the dictionary of the corpus, and thus create the document-topic matrix. This matrix is used for automatic document labeling. Something strange happens when we use LDA with the Gensim function. It happens that with Gensim when it shows the most coherent topics, it also shows the word representation and the coherence score, but does not map the topic ID. To overcome this problem and correctly map the coherence score with the correct topic ID, we use the following cell: `show_topics(num_topics, formatted, num_words)`, `top_topics(doc_term_matrix, dictionary, topn)`, `DataFrame([], columns=['Topic','words'])`. Finally, we load and display the points of coherence with the function `to_csv('CoherenceScore.csv')`, as shown in Table 2.

Table 2. Relevant words for each topic

Topic	Words with Relevance
0 Topic1	{xml, technical, html, java, uml, sql, pl, windows, oracle, agile}
1 Topic2	{ms, Python, r, technical, c, data, java, sql, oracle, windows}
2 Topic3	{project, core, computer, analytics, analysis, r, team, data, areasof, c++}
3 Topic4	{project, skills, linkedin, powershell, salesforce, unix, technical, linux}
4 Topic5	{skills, computer, Python, r, excel, technical, matlab, data, sql, windows}
5 Topic6	{means, key, excel, technical, k, access, teradata, sql, sql_server, oracle}
6 Topic7	{spark, hadoop, Python, r, tableau, technical, pandas, scikit, sql, numpy}
7 Topic8	{s., d. core, software, m., skill, j., r., taleo}
8 Topic9	{spark, Python, r, tableau, technical, hive, pig, java, hadoop, sql}
9 Topic10	{python, css, mysql, c, html, java, javascript, sql, c++, php}
10 Topic11	{relevant, research, illustrator, french, data, spanish, native, english, indesign}
11 Topic12	{sas, Python, r, excel, tableau, technical, matlab, java, sql, c++}

The text cleaning code is shown in Figure 3, and the word list of each document is also generated. Table 2 shows the most relevant words in the model, thus converting the corpus into a document-term matrix for LDA and then defining the core of the LDA model. It is important to specify why the number of topics was set to 12. The most complicated part of LDA modeling is the selection of the number of topics. In addition, the most commonly used way is to measure the degree of semantic similarity between the high-scoring words in the topic. We used Figure 4 as a reference to determine the ideal number of topics, where it can be seen that the best number is between 11 and 12.

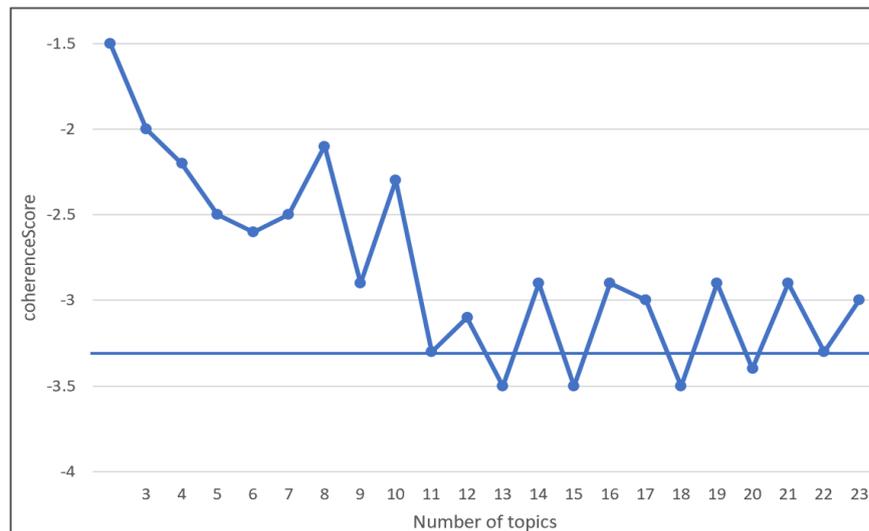


Figure 4. Topic coherence

To choose the number of clusters, we applied the thematic coherence approach. With this approach, we quantify the coherence of a topic by measuring the degree of semantic similarity between the top-rated words. This technique helps to differentiate between themes that are interpretable by humans and those that are artifacts of statistical inference. To calculate the coherence of a topic model, in (3) is used.

$$coherence = \sum_{i < j} score(w_i, w_j) \tag{3}$$

Through (3), pairwise scores are calculated for each of the words selected above. The average consistency score per topic is taken for all topics in the model to arrive at an optimal score for the model, as shown in Figure 4.

3.2. Data tagging and training

For data training in the LDA model, it is essential to specify the number of relevant topics. Each word is assigned to a topic, based on the probability, it is assigned a score according to the dataset it possibly belongs to. Again, this word is assigned to another topic, and its probabilistic score is calculated; after this iterative process, the list of words of each topic with a probability of belonging to a particular topic is obtained. For example [29] argues that these words tend to coexist in the same context, and words with high frequency have more significant positions in each topic. In the LDA model, a set of documents share the same topics, but the proportions are different for each document. To demonstrate data labeling and training, as shown in Table 3, we used the functions as: `get_document_topics(doc_term_matrix, topic_probability)`, this function returns the distribution of topics for the document arc as a list, in turn ignoring topics that are below minimum probability `dataframe(list(topic))`. Also, the `doc2topic` network was used, the function of this network is to model the input co-occurrences of core and context words. It takes the word ID and the document ID as input, fed through two separate embedding layers of the same dimensionality. Each dimension represents a topic and is modeled between a word and its document ID [30]. This network trains by negative sampling, i.e., for any document, both real concurrent words and random words are fed to the network.

Table 3. Labeling and data training

Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic9	Topic10	Topic11	Topic12	Automated topic_id
0	0.00641	0.00641	0.00641	0.00641	0.00641	0.736178	0.00641	0.00641	0.199718	0.00641	0.006411	Topic7
1	0.003968	0.003968	0.003968	0.003968	0.003968	0.003968	0.003968	0.003969	0.956348	0.003968	0.003968	Topic10
2	0.005208	0.005208	0.005208	0.005208	0.005208	0.005209	0.005208	0.005208	0.005209	0.005208	0.942708	Topic12
3	0.002604	0.042106	0.002604	0.002604	0.002604	0.002604	0.002604	0.200676	0.245217	0.002604	0.491168	Topic12
4	0.004902	0.412774	0.004902	0.004902	0.004902	0.004902	0.004902	0.004902	0.004902	0.205122	0.337986	Topic2

3.3. Tagging of unseen data

In the data labeling process, the words have to be ordered with respect to their probability score and, for this, the perplexity technique is used to determine the probability when evaluating the models. Perplexity tries to measure how surprised this model is when it receives a new set of data. This measures the normalized log likelihood of the test set, the lower the perplexity, the better the model.

$$l(D') = \frac{\sum_D \log_2 P(w_d; \theta)}{\text{Count of tokens}} \tag{4}$$

$$perplexity(D') = 2^{-l(D')} \tag{5}$$

In (4) calculates the log likelihood; the probability of observing some unseen data, given a previously learned model. This checks whether the model captures the distribution of the retained set; if not, the perplexity is very high, which tells us that the model is not efficient. In (5) the perplexity is calculated to evaluate the model. To find the probability of the unseen data, as shown in Table 4. Functions such as `dictionary.doc2bow(clean_up_spacy(text))` were used. This function counts the number of occurrences of each distinct word, in turn, converts the ID of each word to an integer and returns the result as a sparse vector. The function `topic_probability.sort_values(probability, ascending)` is responsible for sorting the data in the column in ascending or descending order. Also, other functions were used, such as: `topic_probability()`, `dataFrame()`, among others. Table 4 shows the degree of probability for each of the topics. In it we can see that topic 12, 9, and 10 have the same probability within the unseen data. The same is true for topics 2, 5, 1, 3, 6, 8, 4 and 11, respectively.

3.4. Evaluation of model performance

Finally, the performance of the algorithm is evaluated with the help of the confusion matrix and the prediction accuracy of the subject is found. For this purpose, the scoring method was used, which provides an evaluation criterion for the problem. The scoring parameter was also used, which allows cross-validation to be performed, based on an internal scoring strategy; then the `sklearn` metrics module was implemented to evaluate

the loss, score and usefulness of the prediction performance. Finally, we worked with the sklearn.metrics.confusion_matrix library to calculate the confusion matrix to evaluate the prediction accuracy, where we obtained as output data the following matrix [11, 0, 0, 0, 0, 0, 9, 0, 0, 0, 0, 6, 0, 0, 3, 0] and, to know the linear discriminant and the accuracy achieved by the algorithm, we used the metrics. Accuracy score library, which gave as a result [1,1] which represents 100% accuracy.

Table 4. Labeling of unseen data

	Topic	Probability
6	Topic7	0.923611
11	Topic12	0.006945
8	Topic9	0.006945
9	Topic10	0.006945
1	Topic2	0.006944
4	Topic5	0.006944
0	Topic1	0.006944
2	Topic3	0.006944
5	Topic6	0.006944
7	Topic8	0.006944
3	Topic4	0.006944
10	Topic11	0.006944

4. RESULTS AND DISCUSSION

In this section, we rank the terms according to their relevance within the topics and describe the results of the case study. 30 relevant terms were systematized by topic. For the primary estimation, we performed a test to find the topics with the optimal value in the definition of relevance for interpretation.

In theme 5, where we can find repeated words, such as the word Python, R, technical, data, SQL y Windows. That words are repeated in the different topics is not a problem. This leads us to the conclusion that the different topics {1...12} are closely related to the skills possessed by data science candidates. Figure 5 shows the terms that are most frequently repeated in the different topics, for example. The term SQL is repeated in 10 topics, followed by technical in 8 topics, the programming language R in 7 topics, as well as Python, Java, data tableau and Oracle. Currently, data science is a comparatively new field. Many data scientists come from different disciplines, statistics, engineering, even social sciences and business. This means that as companies seek to harness the power of data for their increasingly digital business, companies in all industries are looking for data scientists and vice versa. As a result, there is an increasing demand on the web for professionals with data management skills. To find the most relevant terms within the corpus, we first had to preprocess the data and then find the ideal number of clusters, as shown in Figure 4. Then, it was classified into 12 topics, as shown in Table 2.

Figure 6 shows the top 30 most relevant terms for each selected topic. By ej. In topic 12, when the relevance metric approaches one. (1), The frequency estimate of terms within topic 12 is higher than the overall frequency of terms; the opposite is true for topic 6, where the overall frequency is higher than the frequency estimate within topic 6. AlSumait *et al.* [9] where they considered different metrics to find the most relevant terms within the corpus, which is fine, however, the most practical way is the one used in the development of this work. This study focused on topic modeling through LDA. Although LDA is a widely used technique, it suffers from order effects when shuffling the order of the data, generating different themes. This limitation can sometimes cause errors and generate misleading data, such as an inaccurate description of the topic [31]. Computational specialists are working on the LDA model to find a method that generates more stable results.

The 12 themes extracted from the LDA model are summarized in Table 1 and shown graphically in Figure 6. To better understand the theme, the keywords associated with candidates interested in data science are examined, as well as the candidate's profile on the theme. For example, the labeling of data to identify the skills of each candidate interested in data science can be seen in themes 7, 10, 12, 12, 12, and 2 in Table 2. It should be noted that terms from different themes may be repeated in each theme. Topic 1 includes relevant terms such as the standard format of XML, technical, HTML, Java, UML, SQL, PL, Windows, Oracle and agile. Topic 2 also includes words such as Python, MS, R, technical C, data, Java, SQL, Oracle y Windows. Some of these words are repeated in the topic 3, for example, programming language R, C and the word data. Finally, the results indicate that the skills that data science candidates should possess should be mainly technical, mastering structured query language, mastering R, Python, Java programming languages, and data processing, among others.

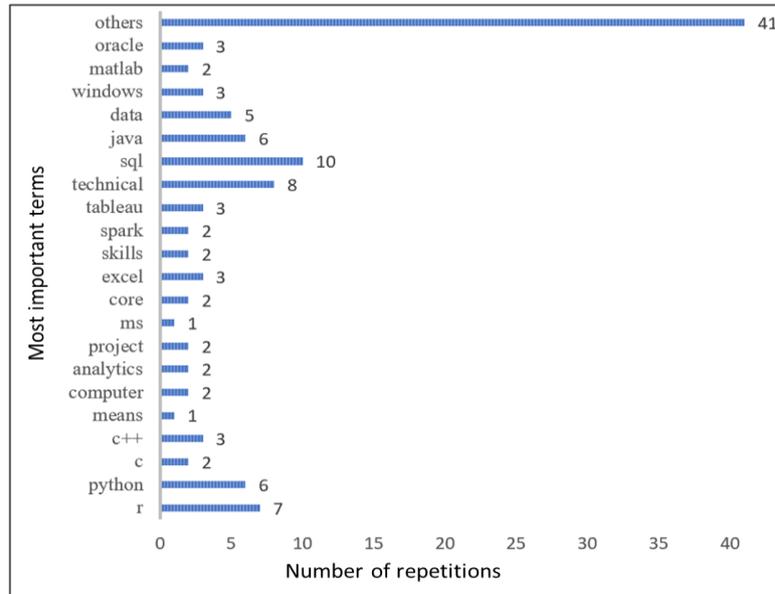


Figure 5. Number of times the words are repeated

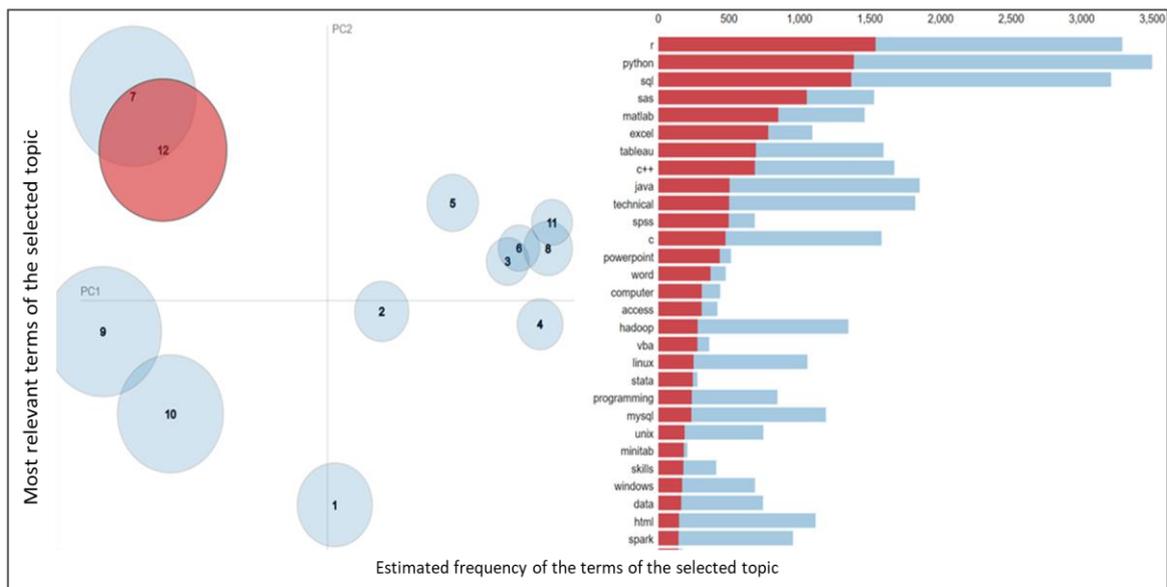


Figure 6. 30 most relevant terms for topic 12

5. CONCLUSION

LDA is a very powerful tool for working with large amounts of structured and semi-structured text documents. The data analysis process is often complex because of the need to achieve a high level of accuracy and the enormous amount of data. To find the topics, the following structure had to be followed: 1) document extraction and data processing, 2) data labeling and training, 3) labeling of unseen data, and 4) evaluation of the model's performance. It is essential to specify that, to find the number of themes, the theme coherence approach was applied, resulting in 12 themes. From this, the most relevant terms for each of the 12 topics were classified, then we proceeded with the labeling and training of the data, for which we used different libraries, including that of probability.

The present study has both theoretical and practical relevance. On the theoretical side, the study applies a machine learning approach, with the LDA model to discover topics in a text corpus, using curricular data extracted from the web. On the practical side, the study recommends the LDA model together with

machine learning be applied in cases of topic identification from large documents, thus allowing it to be adapted and improved to achieve a better level of accuracy.

In this study, from the processing to the classification of the most relevant terms, it became evident that certain terms are repeated in the different topics. For Ej, the term SQL is repeated in 10 topics, technical in 8 topics, R in 7 topics, Python in 6 topics, and so on. With this, we can conclude that the different topics are closely related to the skills possessed by data science candidates. As with all research, there are limitations. The attribute log was applied to the initial data set, however, within the data set, there may be a variety of combinations and factors that can affect the prediction results. In future work, machine learning with the LDA model can be optimized to improve detection and classification accuracy.

REFERENCES

- [1] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, "Exploring topic coherence over many models and many topics," In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pp. 12–14, 2012, Accessed: Oct. 29, 2022. [Online]. Available: <http://mallet.cs.umass.edu/>
- [2] D. M. Blei, A. Y. Ng, and J. B. Edu, "Latent dirichlet allocation Michael I. Jordan," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [3] Y. Liu, F. Du, J. Sun, and Y. Jiang, "iLDA: An interactive latent dirichlet allocation model to improve topic quality," *J Inf Sci*, vol. 46, no. 1, pp. 23–40, 2020, doi: 10.1177/0165551518822455.
- [4] Y. Kalmukov, "Comparison of latent semantic analysis and vector space model for automatic identification of competent reviewers to evaluate papers," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, pp. 77–85, 2022, doi: 10.14569/IJACSA.2022.0130209.
- [5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4–5, pp. 993–1022, May 2003, doi: 10.1016/b978-0-12-411519-4.00006-9.
- [6] T. Aonishi, R. Maruyama, T. Ito, H. Miyakawa, M. Murayama, and K. Ota, "Imaging data analysis using non-negative matrix factorization," *Neurosci Res*, vol. 179, pp. 51–56, Jun. 2022, doi: 10.1016/J.NEURES.2021.12.001.
- [7] J. C. Campbell, A. Hindle, and E. Stroulia, "Latent dirichlet allocation: extracting topics from software engineering data," in *The Art and Science of Analyzing Software Data*, Elsevier Inc., 2015, pp. 139–159, doi: 10.1016/B978-0-12-411519-4.00006-9.
- [8] R. Arun, V. Suresh, C. E. V. Madhavan, and M. N. Murty, "On finding the natural number of topics with latent dirichlet allocation: Some observations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6118 LNAI, no. PART 1, pp. 391–402, 2010, doi: 10.1007/978-3-642-13657-3_43.
- [9] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking," *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 3–12, 2008, doi: 10.1109/ICDM.2008.140.
- [10] H. Jelodar *et al.*, "Latent dirichlet allocation and topic modeling: models, applications, a survey," *Multimedia Tools and Applications* 2018 78:11, vol. 78, no. 11, pp. 15169–15211, Nov. 2018, doi: 10.1007/S11042-018-6894-4.
- [11] U. Chauhan and A. Shah, "Topic modeling using latent dirichlet allocation," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, Sep. 2021, doi: 10.1145/3462478.
- [12] O. Iparraguirre-Villanueva, V. Guevara-Ponce, F. Sierra-Linan, S. Beltozar-Clemente, and M. Cabanillas-Carbonell, "Sentiment analysis of tweets using unsupervised learning techniques and the K-Means algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, 2022, doi: 10.14569/IJACSA.2022.0130669.
- [13] R. M. Asiyabi and M. Datcu, "Earth observation semantic data mining: Latent dirichlet allocation-based approach," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 15, pp. 2607–2620, 2022, doi: 10.1109/JSTARS.2022.3159277.
- [14] S. C. Lee, "Topic modeling of korean newspaper articles on aging via latent dirichlet allocation1," *Asian Journal for Public Opinion Research*, vol. 10, no. 1, pp. 4–22, 2022, doi: 10.15206/AJPOR.2022.10.1.4.
- [15] S. Twinandilla, S. Adhy, B. Surarso, and R. Kusumaningrum, "Multi-document summarization using K-Means and latent dirichlet allocation (LDA) - significance sentences," *Procedia Comput Sci*, vol. 135, pp. 663–670, 2018, doi: 10.1016/J.PROCS.2018.08.220.
- [16] M. Kondath, D. P. Suseelan, and S. M. Idicula, "Extractive summarization of Malayalam documents using latent dirichlet allocation: An experience," *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 393–406, Jan. 2022, doi: 10.1515/JISYS-2022-0027.
- [17] N. Eligüzel, C. Çetinkaya, and T. Dereli, "A novel approach for text categorization by applying hybrid genetic bat algorithm through feature extraction and feature selection methods," *Expert Syst Appl*, vol. 202, Sep. 2022, doi: 10.1016/J.ESWA.2022.117433.
- [18] A. Poushneh and R. Rajabi, "Can reviews predict reviewers' numerical ratings? The underlying mechanisms of customers' decisions to rate products using latent dirichlet allocation (LDA)," *Journal of Consumer Marketing*, vol. 39, no. 2, pp. 230–241, Mar. 2022, doi: 10.1108/JCM-09-2020-4114.
- [19] T. Ramathulasi and M. Rajasekharababu, "Augmented latent dirichlet allocation model via word embedded clusters for mashup service clustering," *Concurr Comput*, vol. 34, no. 15, Jul. 2022, doi: 10.1002/CPE.6896.
- [20] M. Venugopalan and D. Gupta, "An enhanced guided LDA model augmented with BERT based semantic strength for aspect term extraction in sentiment analysis," *Knowl Based Syst*, vol. 246, Jun. 2022, doi: 10.1016/J.KNOSYS.2022.108668.
- [21] B. Li, W. Xu, Y. Tian, and J. Chen, "A phrase topic model for large-scale corpus," *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analytics, ICCCBDA 2019*, Apr. 2019, pp. 634–639, doi: 10.1109/ICCCBDA.2019.8725681.
- [22] M. D. Hoffman, D. M. Blei, and F. Bach, "Online learning for latent dirichlet allocation," *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, 2010.
- [23] M. Kawai, H. Sato, and T. Shiohama, "Topic model-based recommender systems and their applications to cold-start problems," *Expert Syst Appl*, vol. 202, Sep. 2022, doi: 10.1016/J.ESWA.2022.117129.
- [24] T. Cvitanic, B. Lee, H. I. Song, K. Fu, and D. Rosen, "LDA v. LSA: A comparison of two computational text analysis tools for the functional categorization of patents," *In International Conference on Case-Based Reasoning*, 2016.
- [25] K. Henderson and T. Eliassi-Rad, "Applying latent dirichlet allocation to group discovery in large graphs," *Proceedings of the ACM Symposium on Applied Computing*, 2009, pp. 1456–1461, doi: 10.1145/1529282.1529607.
- [26] J. J. Deely and D. V. Lindley, "Bayes empirical bayes," *J Am Stat Assoc*, vol. 76, no. 376, pp. 833–841, 1981, doi: 10.1080/01621459.1981.10477731.

- [27] T. Hofmann, "Probabilistic latent semantic indexing," *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 50-57.
- [28] J. M. Dickey, J. M. Jiang, and J. B. Kadane, "Bayesian methods for censored categorical data," *J Am Stat Assoc*, vol. 82, no. 399, pp. 773-781, 1987, doi: 10.1080/01621459.1987.10478498.
- [29] S. N. Hidayatullah and Suyanto, "Developing an adaptive language model for Bahasa Indonesia," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 1, pp. 488-492, 2019, doi: 10.14569/IJACSA.2019.0100163.
- [30] C. Sievert and K. E. Shirley, "LDAvis: A method for visualizing and interpreting topics," *In Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pp. 63-70, 2014.
- [31] A. Agrawal, W. Fu, and T. Menzies, "What is wrong with topic modeling? And how to fix it using search-based software engineering," *Inf Softw Technol*, vol. 98, pp. 74-88, Jun. 2018, doi: 10.1016/J.INFSOF.2018.02.005.

BIOGRAPHIES OF AUTHORS



Orlando Iparraguirre-Villanueva    systems Engineer with a master's degree in Information Technology Management, Ph.D in Systems Engineering from Universidad Nacional Federico Villarreal Peru. ITIL® Foundation Certificate in IT Service, Specialization in Business Continuity Management, Scrum Fundamentals Certification (SFC). National and international speaker/panelist (Panama, Colombia, Ecuador, Venezuela, Mexico). Undergraduate and postgraduate lecturer in different universities in the country. Advisor and jury of thesis in different universities. Consultant in information technologies in public and private institutions. Coordinator, Director in different private institutions. Specialist in software development, IoT, business intelligence, open-source software, augmented reality, machine learning, text mining, and virtual environments. He can be contacted at email: iv.orlando.c@gmail.com.



Fernando Sierra-Liñan    he is a Bachelor's degree in Education, specializing in Science and Technology at USIL, a Master's degree in Edumatics and University Teaching at UTP, a Bachelor's degree in Systems Engineering and Computer Science at UTP, with a technical specialty in Computer Science and Computer Science. He is currently working as a researcher and thesis advisor in the faculty of Computer Engineering and Systems at the Universidad Privada del Norte, Lima-Peru. He has 20 years of teaching experience. His areas of interest are programming, database and data analysis. He can be contacted at email fernando.sierra@upn.edu.pe and pfsierra.D02052@gmail.com.



Jose Luis Herrera Salazar    he is a Professional in Systems Engineering with experience in planning, analysis, design and programming of computer systems and databases. Developer of equipment maintenance management systems, attendance control systems, warehouse control systems, production systems, academic systems. analysis, design and database administrator, management of human potential. strategic planning of business and social programs, strategic planning and objectives control using balanced scorecard and logical framework methodology. He has self-learning capacity and facility to develop team work and under pressure. He can be contacted at email luis.herrera@autonomadeica.edu.pe.



Saúl Beltozar-Clemente    he is a Bachelor's Degree in Mathematics and Physics, Master's Degree in University Teaching from the National University of Education UNE-Peru. Certification in Hybrid Teaching from the University of Monterrey-Mexico. Undergraduate teaching at Universidad Científica del Sur, Universidad Privada del Norte, Universidad Tecnológica del Perú. Consultant in information technologies in public and private institutions focused on education. He can be contacted at the following e-mail address: saulbelto@gmail.com.



Félix Pucuhuayla-Revatta     he is a Ph.D. in Systems Engineering from Federico Villarreal University (graduated), Doctorate in Education, Master of Science in Electronics with mention in Automation and Control from the National University of Callao (graduated), Master in Education Administration-Cesar Vallejo University. Teacher at Universidad Nacional Enrique Guzmán y Valle, Universidad Privada del Norte, Universidad Tecnológica del Perú, Universidad Alas Peruanas, Universidad de Ciencias y Humanidades; also at IESTP José Pardo, IESTP CEPEA, IESTP AOE. In the industrial field, Production Supervisor and Project Advisor in several companies. Researcher in the field of electronic engineering, mechatronics, systems and education. Advisor and teacher of thesis and thesis project. He can be contacted at email c18883@utp.edu.pe.



Joselyn Zapata-Paulini     she is a Bachelor in Systems Engineering and Computer Science from the Universidad de Ciencias y Humanidades, Master in Science with environmental management and sustainable development at the Universidad Continental, Peru. She has several international publications. Specialized in the areas of augmented reality, virtual reality, and internet of things. Author of scientific articles indexed in IEEE Xplore, Scopus and WoS. She can be contacted at email 70994337@continental.edu.pe.



Michael Cabanillas-Carbonell     he is an Engineer and Master in Systems Engineering from the National University of Callao-Peru, Ph.D candidate in Systems Engineering and Telecommunications at the Polytechnic University of Madrid. President of the chapter of the Education Society IEEE-Peru. Conference Chair of the Engineering International Research Conference IEEE Peru EIRCON. Research Professor at Norbert Wiener University, Professor at Universidad Privada del Norte, Universidad Autónoma del Perú. Advisor and Jury of Engineering Thesis in different universities in Peru. International lecturer in Spain, United Kingdom, South Africa, Romania, Argentina, Chile, China. Specialization in software development, artificial intelligence, machine learning, business intelligence, augmented reality. Reviewer IEEE Peru and author of more than 50 scientific articles indexed in IEEE Xplore and Scopus. He can be contacted at mcabanillas@ieee.org.