Topic Modeling on News Articles using Latent Dirichlet Allocation

Mykyta Kretinin*[ID] and Giang Nguyen*†[ID]
* Faculty of Informatics and Information Technologies, STU in Bratislava, Ilkovičova 2, Bratislava 84216, Slovakia
† Institute of Informatics, Slovak Academy of Sciences, Dúbravská cesta 9, Bratislava 84507, Slovakia
Emails: xkretinin@stuba.sk, giang.nguyen@stuba.sk

2022 IEEE 26th International Conference on Intelligent Engineering Systems (INES) | 978-1-6654-9209-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/INES56734.2022.9922609

***Abstract*—Topic modeling is widely used to obtain the most visible topics from a given text corpus. In this work, a demonstration of the most discussed topic modeling is presented from articles on the Reuters news website. These articles are collected and consequently processed with a Latent Dirichlet Allocation (LDA) unsupervised learning algorithm. The main goal is to build the best model(s) that accurately produces the most discussed topics. Such a model(s) can be used in real life to instantly get information about actual news to classify documents in a given dataset and extract dominated topics with their keywords. This helps to build, for example, correlations with user preferences and recommend interesting content. There are works which use different models to evaluate texts and obtain statistics about them, such as the most popular people's opinions about some question or to obtain popular and dominating subtopics of the specific topic dataset (e.g., medicine articles). As a result of the work, we were able to create a generic LDA model, trained on Wikipedia articles. The model successfully analyzes Reuters articles and extracted their topics as keyword sets. Then, they can be used to recommend content that is interesting to the target user, for example, based on the recommended content tags.**

***Index Terms*—Topic Modeling, Latent Dirichlet Allocation, Reuters Articles, Wikipedia, Ukraine, War, Covid, NLP**

I. Introduction

Today, natural language processing (NLP) methods are widely used. They help humans process documents, extract needed information from text, analyze its content, and build graphs and diagrams with statistics. NLP methods for text processing save a lot of time and optimize the process of searching and processing information. For example, topic modeling methods can be used to predict the topics of the text, helping to recommend interesting articles and videos to users.

In medicine, NLP methods allowed us to transform raw data from unstructured clinical information on patients into a structured form [1]. There, the unstructured information about the patients took a long time for doctors to read the "free" text and search for possible symptoms here. Thus, the usage of NLP helped to solve two important problems:

- Big amount of time spent on text analysis of electronic health records by physicians on a regular basis,
- Possibility of managing and mining large volumes of clinical data on large time scales.

Topic modeling is a broad field of NLP, and its results are used in the daily life of people, in their work, and in different fields of science. It helps to analyze large volumes of text, extracting and summarizing information, building graphics, and visualizing statistics. Therefore, to be able to achieve different results and use topic modeling on different datasets properly, many different models were developed. This work mainly focuses on LDA model, which is one of the most popular and famous methods in topic modeling. The application of algorithms for text preprocessing, model creation, learning, and finally, model usage on real datasets is presented in the following parts of this work.

II. Related work

The use of NLP is very useful in the analysis of social networks. Each social network contains many chats, polls, and groups with the opinions of users on a certain topic or situation. Sometimes, it may be very useful to get summarized information in graphs and tables for faster analysis, but this process of detecting and processing these data is time-consuming. This problem was very relevant during the COVID-19 epidemic, when governments of countries and world organizations wanted to know the point of view of people about preventive measures, such as mask wear, antigen tests, and vaccination.

Another goal was described in [2]. Authors studied the possibility of using NLP in the detection of disease gene associations within large volumes with a large number of complicated associations. Therefore, they described a computational framework that discovers latent disease mechanisms by dissecting disease gene associations from more than 25 million PubMed articles. They used the LDA model and network-based analysis because of their ability to detect latent associations within text and reduce noise for large volumes of data.

A good use of the LDA model was demonstrated in [3]. Here, the created LDA model topics were used to conduct a literature review on papers from online databases such as Web of Science, Scopus or Google Scholar. Then the resulting topics were merged into clusters to get top-level topics from the former ones. It helps to understand the correlation between them, so a concept map can be made of the keywords of the models, to describe the topics of the models in the most precise way.

The main objective of this work is the practical application of topic modeling methods on real datasets such as text-containing news articles. In this work, the process and results

of the use of the LDA model are described, and tests are carried out on models trained with different hyperparameters to compare their results and choose the most successful model for future use. Moreover, in this paper the question of the possibility of using a generic model, trained on the Wikipedia article dataset [4], [5], instead of a specific model, which was trained on the Reuters article, is discussed. It is supposed that accuracy of the Wikipedia model will be lower, but it still has to be able to precisely predict a document topics and might be used on almost any text, as it contains words from a wide range of topics. All used datasets are unlabeled, so the models are trained with the unsupervised learning approach.

III. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [6], [7] is a topic modeling method, which allows users to get a probabilistic distribution of the topics in the document. Topics are represented by keywords, which are the most ”popular” words in the documents assigned to the current topic.

1) General requirements to the process: To start with topic modeling using the LDA model, it should first be trained. Therefore, a corpus is needed for the training process. A corpus is usually represented as a bag-of-words (BoW), or a list of pairs ”word:number_of_occurrences”, which do not present the orders and relations of the words, but their count in the text.

2) Data gathering: First, the data should be acquired. In the data collection process, the consideration of general topic bias should be taken seriously, especially if the majority of documents collected are from the same closely located source. If such a situation occurs, then the resulting model can easily be overfitted for some particular topic(s). This model cannot accurately analyze other topics in the deployment.

3) Corpus and Dictionary: The goal of this step is to transform the text into the form that the model can use, be trained on it, or analyze. To obtain the corpus from the text, it should first be processed. This step includes the removal of special characters, punctuation and stop words, and lemmatization [8], [9] of the remaining text. The text is then transformed into a list of words, which will later be transformed into the corpus. In addition, words with a very low number of occurrences can be removed as well, to reduce the corpus and speed up the model training process. However, it can affect the quality of the model. From the corpus we can get the ”word-id” relation, or the dictionary of the corpus, because some implementations require it to work. This is the case for the LDA model library *gensim (https://radimrehurek.com/gensim/models/ldamulticore.html)* which was used in the experiments in this paper.

A. Model training

When the corpus is ready, it can be used to train the model. As a result, the trained model will be able to use the information gained on the topics and the words assigned to them in the analysis of the unseen text.

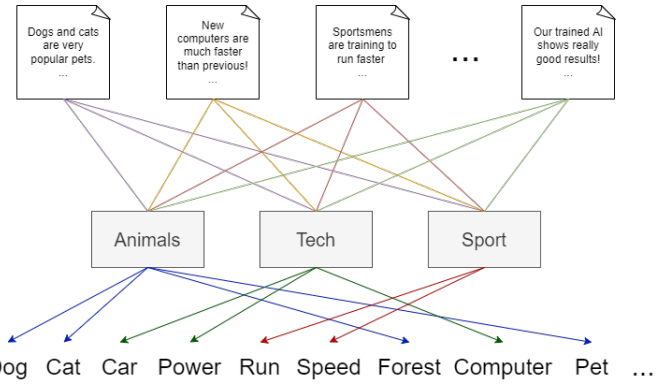

Fig. 1: Words assigned to the topic to which they most likely belong

The process of creating / training LDA models may be described in five steps, which together create the following algorithm [10]:

1) Choose number (k) of topics, which should be created by the model.
2) Distribute these k topic among the document m by assigning a topic to the words in the text. This distribution is named α.
3) Then we suppose, that for each word w in the text has been assigned wrong topic, but every other word is assigned the correct topic.
4) Assign word w a topic, basing on probability of two things:
   - What topics are actually in the analyzed document m.
   - How many times word w has been assigned to the particular topic z across all of the documents.
5) Repeat this process for each document, to get k topics with assigned words.

This educational process is iterative, which means that it has to be repeated N times to obtain a better result. Executing this algorithm only 1-2 times for text should not give us a very good result compared to a higher number of iterations.

In Fig. 2 the relations of the variables can be seen as follows: In the picture 2 we can see the relations of variables, where:

- α is the per-document topic density,
- β is the per-topic word density,
- θ is the topic distribution for document *m*,
- η is the word distribution for specific topic,
- ***z*** are topics of the document *m*,
- ***w*** is the specific word

α and β are vectors of real numbers that are usually the same for all topics/words, respectively.

θ and η are matrices, where $\theta(i, j)$ represents the probability that the i-th document contains the j-th topic and $\eta(i, j)$ represents the probability that the i-th topic contains the j-th word [11].

Among all these variables, only w is grayed out because it is the only observable variable in the system, while the

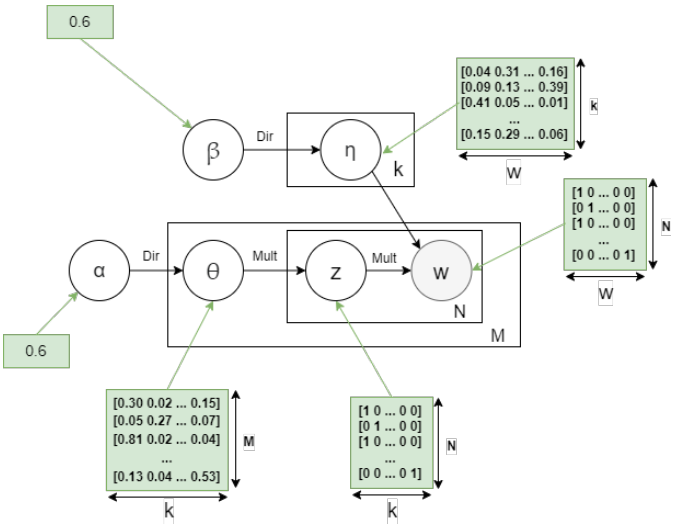

Fig. 2: Plate notation representing the LDA model with variables and their possible values

others are hidden from us. In the algorithm above, our goal in steps 1-4 is to assume a word for a topic, so we can suggest the word w as the final result of the process. From the beginning, the topics z are unknown and need to be filled with the words w. Similarly, θ and η are also unknown and will be calculated using the words w and the topics z. These variables are affected by predefined model hyperparameters α (per-document topic density) and β (per-topic word density) accordingly. The higher α, the documents consist of more topics (θ), and the higher β, the topics consist of most of the corpus words (η).

IV. Experimenting with LDA model

Our goal is to obtain a precise model that can accurately predict the topic of the text. Only then will it be able to produce reliable statistics on the most popular and discussed topics among the documents in the scraped article set.

Therefore, a strong "basement" should be built, which means that the trained model should show good metrics. In this way, its words will be distributed in the right way, and the prediction of the unseen text will be more accurate. This will be achieved in the hyperparameter tuning process.

A. Hyperparameter tuning

To obtain a good model, the grid search is used over two hyperparameters of the model, *number of topics* and α (also known as the topic density per document). Totally 12 models were trained on a 100MB Wikipedia abstract dump dataset (*enwiki-latest-pages-articles-multistream24.xml-p56564554p57025655.bz2*, Feb-2022). Each model has a different number of topics and a value of α. Number of topics was in the range of [4,14] with a step of 2 (4, 6, ..., 14 topics) and α was distributed symmetrically / asymmetrically between the model topics. In a symmetric distribution, α is the same for each topic:

$$\alpha = \frac{1}{number\_of\_topics} \tag{1}$$

For an asymmetric setting, the value of α depends on the index of the topic:

$$\alpha = \frac{1}{topic\_index + \sqrt{(number\_of\_topics)}} \tag{2}$$

After all models were trained, their metrics were compared to obtain the most suitable model for further use. The results are presented in Table I

TABLE I: Results of hyperparameter tuning - 12 models

Hyperparameter tuning results				
α	num_topics	perplexity	C_V	C_{UMass}
Symmetric	4	-9.304	0.550	-1.253
	6	-9.423	0.579	-1.401
	8	-9.819	0.572	-1.987
	10	-10.556	**0.608**	-1.735
	12	-11.855	0.574	-2.483
	14	-13.199	0.577	-2.499
Asymmetric	4	**-9.291**	0.541	**-1.245**
	6	-9.396	0.550	-1.517
	8	-9.806	0.548	-2.067
	10	-10.580	0.579	-1.943
	12	-11.901	0.536	-2.436
	14	-13.153	0.575	-2.147

Three metrics are used for model evaluation and comparison: $perplexity$ in (3), C_{UMass} (p(*rare word* | *common word*)) and C_V ($\log(PMI)$, for PMI in (4)) coherence scores. In short, the coherence score indicates the degree of semantic similarity between words on the topic, while the perplexity indicates how well a probability model predicts a sample or how much it is confused with the analyzed content. The best score applied to all of these metrics is the highest. Based on the results, the model with 6 topics and a symmetric α distribution is chosen because it obtains the closest results to the best in all metrics, while the other models have good results (close to the best ones) only in one or two particular metrics.

Normalized perplexity score:

$$\ln(P(W) = \ln(P(w_1, w_2, ..., w_N)) \tag{3}$$

where

$\ln(P)$ is a normalized perplexity function;
W is a full sentence/text;
w_i is an i-th word of the sentence/text W;
N is a number of words w in the text W;
$P(w_1, w_2, ..., w_N)$ is a probability that the model assigns to text W;

Pointwise Mutual Information (PMI):

$$score = log(\frac{p(w_i, w_j)}{p(w_i)p(w_j)}) \tag{4}$$

where

$p(w)$ is a probability that the word w will be seen in a random document,

$p(w_i, w_j)$ is a probability of seeing both w_i and w_j words in the same document.

Models are not trained on the dataset of Reuters articles because they have a high potential to be unbalanced. This may lead to overfitting the model for certain topics and inaccurate results in realization. On the contrary, Wikipedia articles are not grouped by topic, so the Wikipedia dumps used are considered balanced enough to produce good training results.

B. Topic modeling on the articles datasets

The trained model can finally be used to analyze crawled articles to get brief information about their content. The most important information we can obtain from model analysis is the most popular topic among the given data set. Fig. 3a and Fig. 3b present document counts from the dataset, which were assigned to the particular topic of the model. Here, in Fig. 3a, the most discussed topics were about politics (topic 0, 298 assigned documents) and about people's life and health (topic 2, 642 documents, due to the words "covid", "people", "climate"). At the same time, in February (Fig. 3b) the war in Ukraine was a very discussed topic, since multiple topics (0th and 4th) are related to the war, while the "covid" topic (2nd) was still relevant.

This information can be used to help people understand the topics most discussed today and to compare the topics discussed in different periods. These data may be used, for example, by other news websites to get the actual most popular topics or to understand trends over a particular period of time and to publish appropriate content that will be interesting to readers.

Using *Intertopic Distance Map*, the similarity of the model topics can be visually displayed by comparing the distance between them. Examples of such visualization are shown in Fig. 4, where the topic "1" is highlighted with its 30 most relevant terms. The top terms are calculated with the usage of $relevance$ (λ), with values in the range [0,1]. With λ = 1 the most relevant terms of the topic will generally be displayed, even if they occur somewhere else, when λ = 0 will be displayed terms that belong only to this topic.

In Fig. 4, the top terms are the same as those of the topic "2" in Fig. 3a, as they display information about the same topic of the model. However, these top keywords do not appear only in the text of this topic, so we may be interested in those that were not assigned to any other topic. This case is shown in Figure 5, where the top 30 terms of the topic are assigned only to this topic and do not appear anywhere else. This may give us a more accurate explanation about the topics of the model, as in most cases these words have a closer meaning to each other and appear in documents with similar content only.

Topic	Documents_assigned	Topic_Keywords
0.0	298.0	minister, government, party, state, president, election, united, country, leader, people
1.0	0.0	team, win, championship, final, game, world, player, round, season, time
2.0	642.0	country, year, trust, china, new, covid, read, climate, state, people
3.0	35.0	year, release, film, new, first, monday, work, award, show, art
4.0	23.0	force, league, kill, attack, war, year, army, season, club, military
5.0	2.0	road, building, house, south, church, river, station, city, area, mount

(a) October articles (11.05.2021)

Topic	Documents_assigned	Topic_Keywords
0.0	392.0	ukraine, state, feb, minister, president, government, russia, party, united, country
1.0	3.0	team, game, olympic, win, championship, final, world, round, player, season
2.0	535.0	year, trust, china, covid, read, state, people, report, new, plan
3.0	33.0	year, film, release, new, monday, work, award, first, show, time
4.0	36.0	troop, military, force, myanmar, attack, kill, year, war, army, kyiv
5.0	1.0	road, house, church, building, park, south, station, mount, area, village

(b) February articles (26.02.2022)

Fig. 3: Per-topic documents count

C. Generic versus specific LDA model

Sometimes, it may be essential to train a model on data that are related to the future data we want to analyze. For example, if a model is used to analyze medical articles, it should also be

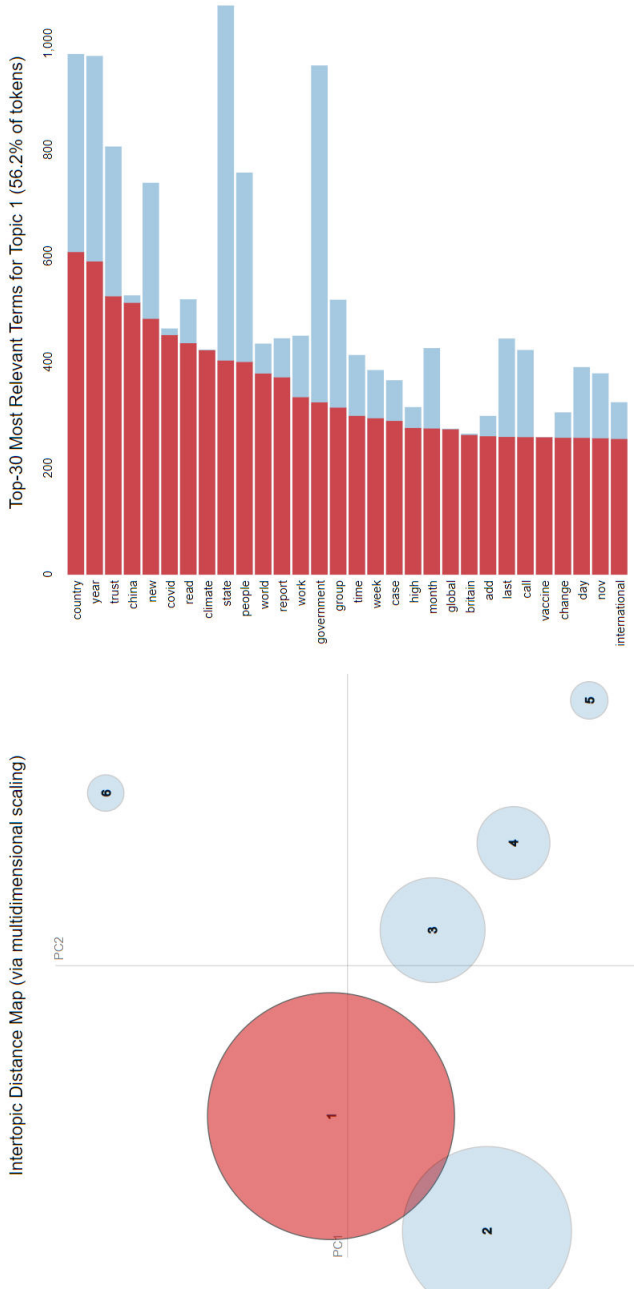

Fig. 4: Intertopic Distance Map for October articles with λ=1

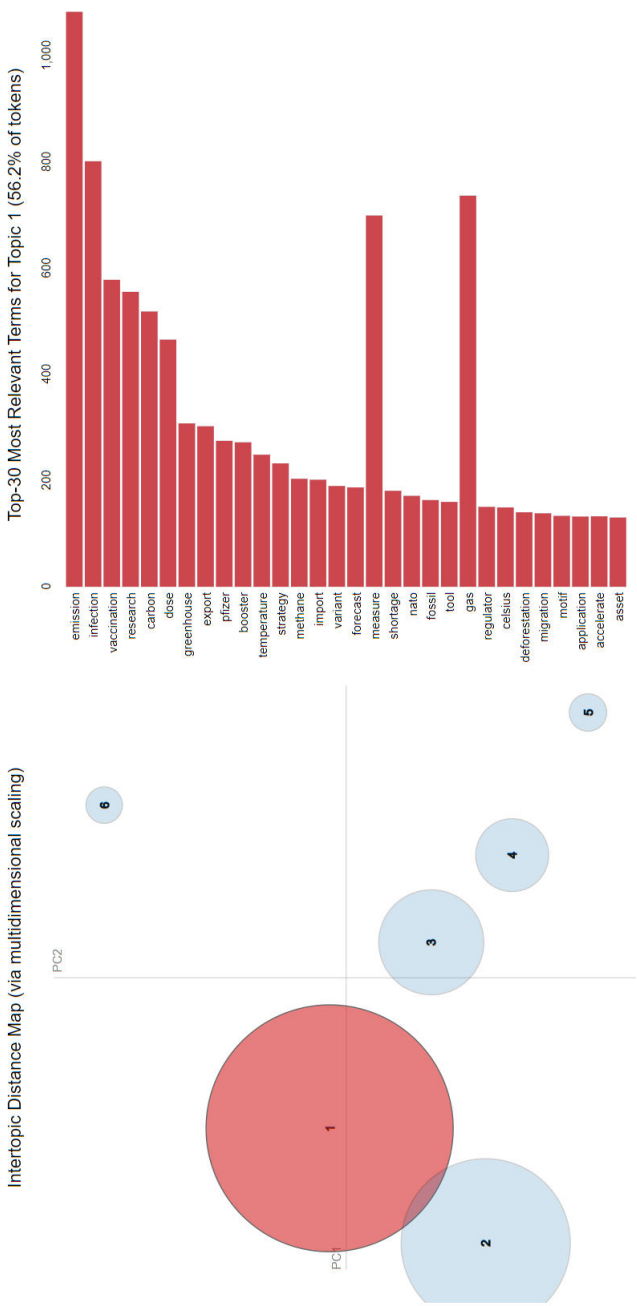

Fig. 5: Intertopic Distance Map for October articles with λ=0

trained on some medical data to be able to separate documents by medical subtopics and obtain better results.

Here, the model was trained on Wikipedia datasets instead of the Reuters news article dataset. The main reason was a possible unbalance in the Reuters dataset, as the topics discussed are very often biased, which may lead to model overfitting over some topics. Wikipedia, in contrast, is a very generalized dataset, so it may have many terms and even topics that do not occur very often. This may also lead to inaccurate results, but the models will not be biased. Moreover, the Wikipedia trained model can potentially be used on any testing dataset, as it is generic and will show similar results with almost any possible topic. To check if the Reuters model will work better on Reuters articles, two models have been trained on Reuters and Wikipedia datasets of similar size. Reuters training dataset was scrapped on 20.04.2022, so after training it was biased toward the topics of politics and war in Ukraine. These models have to analyze 1000 November articles to check the resulting topics and the distribution of the document words. From the test, the Reuters trained model got better results, as expected for a specific model. The documents in the test dataset were quite equally distributed by it, in contrast to the Wikipedia trained model. However, both models were

able to successfully predict the main topic of the documents.

Therefore, every approach in model training has its own pros and cons, but for this work, balanced but generalized Wikipedia dumps were enough to experiment with the LDA model, as they were able to accurately predict topics of documents in the testing dataset.

V. Conclusion

In this work, a possible usage of the LDA model has been described to analyze news articles, with preliminary acquisition, preprocessing, and model training on text data sets. To achieve better results and higher accuracy, here we used a hyperparameter tuning process, where we chose a model for dataset analysis. The results demonstrated that with the statistics acquired on the analyzed datasets, we can understand the dominant topics by their keywords. By this way, we can compare popular topics over a particular period of time, so it is possible, for example, to monitor the topic changing over time according to the reader's interest. From the results, we were able to state that in October there were popular topics about politics and Covid, while in February there were relevant articles about the war in Ukraine and Covid. These topics accurately reflected the real situation in the world and reaffirmed the development and change in the relevant topics. Moreover, the acquired Intertopic Distance Maps are able to separate words that occur only in the current topic from those distributed among many topics. With that, it is possible to get more information on any topic of the model. As an additional way to use acquired topics, their keywords and tags can be used to search for similar content on the Internet. Therefore, if the previous user's search results were analyzed this way, it would be possible to find similar content, which would save a lot of time and help find potentially interesting information.

Acknowledgements

This work is supported by VEGA 2/0125/20 New Methods and Approaches for Distributed Scalable Computing and the Operational Programme Integrated Infrastructure for the project: International Center of Excellence for Research on Intelligent and Secure Information and Communication Technologies and Systems – Phase II (ITMS code: 313021W404), co-funded by the European Regional Development Fund (ERDF).

References

[1] K. Kreimeyer, M. Foster, A. Pandey, *et al.*, "Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review," *Journal of biomedical informatics*, vol. 73, pp. 14–29, 2017. DOI: 10.1016/j.jbi.2017.07.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046417301685.

[2] Y. Zhang, F. Shen, M. R. Mojarad, *et al.*, "Systematic identification of latent disease-gene associations from pubmed articles," *PloS one*, vol. 13, no. 1, e0191568, 2018. DOI: 10.1371/journal.pone.0191568. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0191568.

[3] M. Weiss and S. Muegge, "Conceptualizing a new domain using topic modeling and concept mapping: A case study of managed security services for small businesses," *Technology Innovation Management Review*, vol. 9, pp. 55–64, 2019, ISSN: 1927-0321. DOI: http://doi.org/10.22215/timreview/1261. [Online]. Available: https://timreview.ca/article/1261.

[4] E. Cambria and B. White, "Jumping nlp curves: A review of natural language processing research," *IEEE Computational intelligence magazine*, vol. 9, no. 2, pp. 48–57, 2014. DOI: 10.1109/MCI.2014.2307227. [Online]. Available: https://www.gwern.net/docs/ai/2014-cambria.pdf.

[5] S. Dlugolinsky, G. Nguyen, M. Laclavik, and M. Seleng, "Character gazetteer for named entity recognition with linear matching complexity," in *Third World Congress on Information and Communication Technologies (WICT 2013)*, IEEE, 2013, pp. 361–365. DOI: 10.1109/WICT.2013.7113096.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003. [Online]. Available: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?TB_iframe=true&width=370.8&height=658.8.

[7] M. Weiss and S. Muegge, "Conceptualizing a new domain using topic modeling and concept mapping: A case study of managed security services for small businesses," *Technology Innovation Management Review*, vol. 9, no. 8, 2019. [Online]. Available: https://www.timreview.ca/article/1261.

[8] D. Khyani, B. Siddhartha, N. Niveditha, and B. Divya, "An interpretation of lemmatization and stemming in natural language processing," vol. 22, pp. 350–357, 2020.

[9] D. Maier, A. Waldherr, P. Miltner, *et al.*, "Applying lda topic modeling in communication research: Toward a valid and reliable methodology," *Communication Methods and Measures*, vol. 12, no. 2-3, pp. 93–118, 2018. DOI: 10.1080/19312458.2018.1430754.

[10] T. Doll, *Lda topic modeling: An explanation*, Accessed 02.03.2022, 2018. [Online]. Available: https://towardsdatascience.com/lda-topic-modeling-an-explanation-e184c90aadcd.

[11] T. Ganegedara, *Intuitive guide to latent dirichlet allocation*, Accessed 02.03.2022, 2018. [Online]. Available: https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158.