Improving Topic Modeling Performance through N-gram Removal

Conference Paper · December 2021

DOI: 10.1145/3486622.3493952

CITATION
1

READS
97

4 authors, including:

Andrea De Mauro
University of Rome Tor Vergata
27 PUBLICATIONS **2,009** CITATIONS
SEE PROFILE

Adham Kahlawi
University of Florence
14 PUBLICATIONS **28** CITATIONS
SEE PROFILE

Valentina Poggioni
Università degli Studi di Perugia
74 PUBLICATIONS **468** CITATIONS
SEE PROFILE

Improving Topic Modeling Performance through N-gram Removal

Mohamad Almgerbi
mohamadalimohamad.almgerbi@unifi.it
University of Florence,
Florence, Italy

Andrea De Mauro
andrea.de.mauro@uniroma2.it
University of Rome Tor Vergata
Rome, Italy

Adham Kahlawi
adham.kahlawi@unifi.it
University of Florence,
Florence, Italy

Valentina Poggioni
valentina.poggioni@unipg.it
University of Perugia
Perugia, Italy

ABSTRACT

In recent years, topic modeling has been increasingly adopted for finding conceptual patterns in large corpora of digital documents to organize them accordingly. In order to enhance the performance of topic modeling algorithms, such as Latent Dirichlet Allocation (LDA), multiple preprocessing steps have been proposed. In this paper, we introduce N-gram Removal, a novel preprocessing procedure based on the systematic elimination of a dynamic number of repeated words in text documents. We have evaluated the effects of the utilization of N-gram Removal through four different performance metrics: we concluded that its application is effective at improving the performance of LDA and enhances the human interpretation of topics models.

KEYWORDS

Topic Modeling, LDA, Coherence, Perplexity, Data Preprocessing, Big data

ACM Reference Format:
Mohamad Almgerbi, Andrea De Mauro, Adham Kahlawi, and Valentina Poggioni. 2021. Improving Topic Modeling Performance through N-gram Removal. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI-IAT '21), December 14–17, 2021, ESSENDON, VIC, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3486622.3493952

1 INTRODUCTION

Spurred by the broad advancement of digital technologies in our society, an ever-increasing quantity of data is produced every day. This also applies to natural language textual data, which is continuously generated by individuals and incrementally stored in various forms. From social networks posts to literary and scientific productions, from the e-learning platforms to digital libraries, the amount of text documents that are daily produced grows continuously [10]. Analyzing and gathering insights from such a vast amount of data is challenging and requires significant effort. Once textual data is collected, it is natural to leverage data mining processes and techniques to gather useful and previously unknown information. Among the various machine learning and natural language processing algorithms used to analyze the massive amount of textual data available online, topic modeling techniques have gained popularity in recent years [22]. Topic models provide a convenient way to analyze large amounts of unclassified text to infer conceptual patterns and relationships among documents [14]. Topic models are now leveraged to fuel several applications, including opinion analysis [24], text information retrieval [40], image retrieval [38], social network analysis [5] as a tool to gauge political sentiments [28], medical big-data mining process [34], and business analytics [25]. One of the most frequently adopted topic modeling algorithms is Latent Dirichlet Allocation (LDA) [15]. The aim of LDA is to identify topics based on the words contained in documents. The preprocessing stage is critical to obtain a better quality of topics for the documents. It transforms text into a more digestible form so that machine learning algorithms can perform better [36]. In natural language processing, useless words are referred to as stop words. Stop words are available in abundance in any human language. By removing these words, we remove the low-level information items from a text so as to give more focus to the relevant information [29]. Additionally, removal of stop words definitely reduces the dataset size and, thus, the training time due to the fewer number of tokens involved in the training [17]. However, apart from stop words, there might be many other repeated words and phrases in a corpus that adversely affect the quality of topic models, especially when they have a high frequency in the dataset. The fundamental problem with repeated phrases in a distributional semantic model is the over-representation of specific word co-occurrences to a model. The repeated phrases will leave less representational power for the remaining text. As a consequence, the combination of repeated text snippets will likely yield less coherent topics. In an attempt to create more effective topic models, this paper proposes an efficient strategy for preprocessing documents corpora before applying topic modeling. We introduce N-gram Removal, a novel preprocessing procedure based on the systematic elimination of those repeated words and phrases that may have a negative impact on the results of the LDA model.

WI-IAT '21, December 14–17, 2021, ESSENDON, VIC, Australia

ACM ISBN 978-1-4503-9115-3/21/12…$15.00
https://doi.org/10.1145/3486622.3493952

2 RELATED WORKS

Several optimization approaches have been proposed for improving LDA performance and for finding the optimal number of topics.

2.1 Improving the Performance of Topic Modeling

Authors in [32] quantified the Effects of Text Duplication on Semantic Models. The study proved that the presence of duplicated strings, either documents or repeated text within documents, has an impact on semantic models because repeated texts adversely affect the parameters learned by distributional semantic models. In [31] authors analyzed the consequence of removing stop words for topic modeling in terms of model fit, coherence, and utility. They conclude that the effects of such removal during training are limited, and that the removal of unwanted terms after the training step should be sufficient. In [11] authors conducted a comparative analysis of the effect of stop words removal on Sentiment Classification. Traditional Sentiment Classifier showed an improvement in accuracy when stop words were removed. In [6], the authors proposed the Parsimonious Language Model to remove non-significant words/terms from the corpus to produce more coherent words.

2.2 Finding the Optimal Number of Topics

In [30], the author established a relation between the optimal number of topics and text size. The study concluded that choosing a combination of an excessively small text chunks with a large number of topics could generate minimal topics that are specific and redundant, while choosing an excessively small number of topics based on larger text chunks will give rise to topics that are too broad and heterogeneous. Consequently, either way, they will be difficult to interpret. Kobayashi et al. [18] used perplexity to predict the optimal number of the topic in LDA. Zhao et al. [43] proposed the Rate of Perplexity Change (RPC) to predict the optimal number of topics and proved that RPC works better than perplexity in this task. Another study, [39], proposed a non-iterative method based on perplexity to find an appropriate number of topics and, thus, optimize LDA. In [12] the authors propose two new methods named Normalized Absolute Coherence (NAC) and Normalized Absolute Perplexity (NAP) for predicting the optimal number of topics. The authors run highly standard ML experiments to measure and compare the reliability of existing methods (perplexity, coherence, RPC) and proposed NAC and NAP in searching for an optimal number of topics in LDA. The study successfully proves and suggests that NAC and NAP work better than existing methods. This investigation also suggests that perplexity, coherence, and RPC are sometimes distracting and confusing to estimate the optimal number of topics.

2.3 Evaluation of Topic Model

In [7], authors used coherence to measure the performance of LDA. They also demonstrated that a larger number of topics in LDA help producing more coherent topics. Lau et al. [23] proposed two levels for automatically evaluating the performance of topic modeling: the topic level and the model level. Yarnguy et al. [41] introduced the ACO-LDA method for tuning LDA parameters to improve its performance. Experiments were conducted on various datasets fetched from the UCI and evaluated by using perplexity scores.

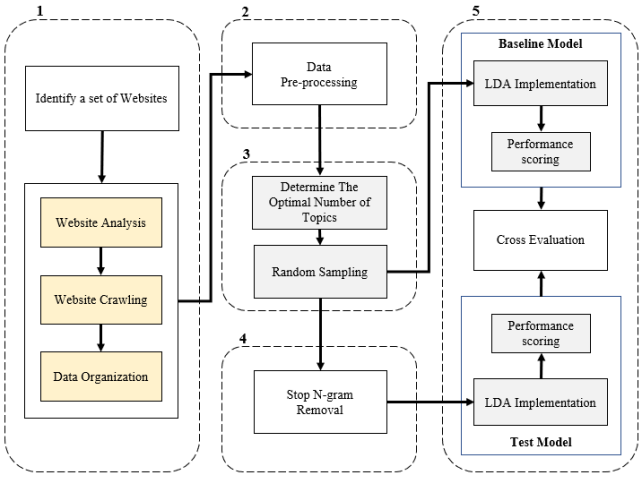

Figure 1: Overall methodology of proposed framework

3 METHODOLOGY

In this section, we will review the 5-step methodology adopted in this study, which is depicted in Figure 1. Initially, we identified a set of job classifieds websites and extracted a large number of job advertisements using Web Scraping techniques. Secondly, we applied several preprocessing steps to clean the textual data extracted from the Web (job descriptions). Third, we built several LDA models in order to determine the optimal number of topics with different numbers of topics K, with K ranging from 5 to 30. Then, we used extensive performance evaluation with the help of a number of performance metrics to evaluate the goodness of LDA models. Afterwards, we identified the duplicate sentences that may have had a negative impact on the results of the LDA and removed them from the text documents. Finally, we applied a topic modeling algorithm again on the description of the job advertisements after removing the duplicate sentences. Lastly, we evaluated the model's performance in order to determine the effect of removing duplicate sentences on the quality topic models.

3.1 Data Collection

The objective of this step is to identify a set of websites containing data that can be used in our study as the data source, and then scrape it using a custom web scraper software.

3.1.1 Identify a set of Websites. In order to extract a large number of data points, we identified multiple websites that contain job advertisements among the most prominent job-seeking platforms available on the World Wide Web. We selected websites by considering several criteria, namely: the number of posts, the geographic scope, and compatibility with web scraping. According to these selection criteria, Simply Hired and CareerBuilder online employment sites were selected as they met at best all the criteria.

3.1.2 Web scraping. Web scraping, also known as web extraction, is a technique used to extract data from the World Wide Web and save it to a file system or database for later retrieval or analysis [42]. Web Scraping consists of three main phases, shown in Figure 2,

namely: website analysis, website crawling, and data organization [21].

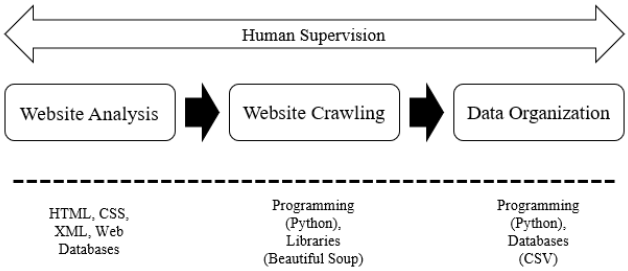

Figure 2: Web Scraping phase.

In this study, we created a custom web scraper software using Python [37]. The software was able to extract titles and descriptions of every job post available on the selected websites. We used six keywords that collectively covered the field of Big Data Analytics, namely: "big data", "data science", "business intelligence", "data mining", "machine learning" and "data analytics". Using the web scraper, we were able to extract 14,495 online job advertisements. After we deleted the duplicate job advertisements, we built a dataset containing 9,067 job ads.

3.2 Data Preprocessing

Usually, the raw data extracted from the internet is unstructured and requires to be preprocessed before being utilized for modeling [33]. Data preprocessing, such as normalization, feature extraction, and dimension reduction, is necessary to accomplish a better classification of the textual data. The aim of preprocessing is to find the most informative set of features to improve the performance of the classifier [1]. In any Machine Learning procedures, preprocessing is the step in which the data gets transformed, or encoded, to bring it to such a state that the machine can easily parse it. In other words, the features of the data, once preprocessed, can be easily interpreted by the algorithm. The preprocessing stage is critical in data analysis and can heavily influence the Topic Modelling results. In this study, the preprocessing phase consisted of several sequential steps. Initially, we started with the tokenization of the text to split the document into tokens. Then, stemming was carried out and every word was converted to its root form, so to harmonize different forms of the same word onto the same entity. Then, we performed lemmatization on the documents, which is converting the word to its lemma. Afterwards, punctuation, weblinks, HTML tags, and meaningless characters were removed, as in [9]. Individual stop words were then removed from the text documents [16]. Finally, we created the dictionary and corpus needed for Topic Modelling.

3.3 Determine the Optimal Number of Topics

The aim of The Latent Dirichlet Allocation is to identify topics based on the words it contains. In many topic modeling algorithms including LDA, finding the optimal value for the number of topics is not trivial. The number of topics has a great influence on the results of the clustering process but the evaluation of the result is subjective, difficult to be interpreted, and time-consuming. An overly small number of topics would make LDA unable to identify meaningful topics, while a number of topics which is too high would lead to an excessively complex model, difficult to be interpreted and validated [13]. It is often required to repeat the application of the topic modeling algorithm several times with different numbers of topics and evaluate the best number according to a set criteria. In this study, we developed a procedure in Python to determine the optimal number of topics by evaluating the performance of LDA algorithm using four different metrics, namely coherence, log-likelyhood, perplexity, and human interpretability. First, we leveraged LDA to generate several topic models with a varying number of topics, k ranging from 5 to 30. We then identified the value of k that produced the highest coherence value. As shown in Fig. 3(a), we obtained the maximum coherence score when the number of topics was 9. The second step was to find the highest Log-likelihood value within the same range of value for k: also in this case, the highest Log-likelihood value was obtained with a number of topics equal to 9.

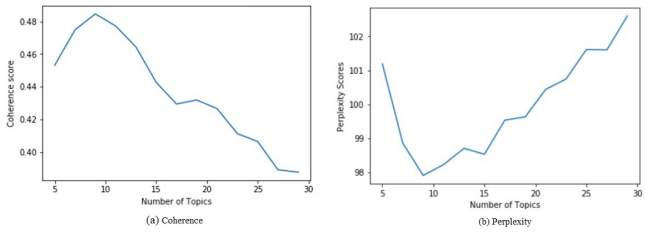

Figure 3: Coherence and Perplexity scores for different number of topics.

Similarly, as shown in Fig. 3(b) the value of perplexity was also minimized with $k = 9$ topics, confirming what we found with the other metrics. Lastly, we evaluated the interpretability of the models by means of human judgement. The authors were presented with multiple topic models, with different values of k and consensually agreed that the most meaningful model was with $k = 9$. By having all four metrics recommending the same value for k, we set 9 as the number of topics and proceeded with the rest of the steps.

3.4 Identify and Remove Duplicate Sentences

The objective of this step was to identify and remove duplicate sentences that may have a negative impact on the results of the LDA algorithm. Initially, the description of each job was divided into a group of text strings using the Python nltk.tokenize package. Tokenization is the process of splitting a textual string into a list of shorter substrings [8]. By applying this function to the corpus of job ads descriptions, we observed that many substrings were repeated in job posts, jeopardizing the ability of the topic modelling algorithm to infer the sets of words related to meaningful concepts. We created a list of the substrings that were repeated with a frequency higher than 10 over the dataset.This was determined after preliminary tests. Fig. 4 shows some examples of the repeated substrings in text documents.

We noticed that such repeated snippets of text were often providing general information about the company or the organization offering the job. In order to avoid any negative impact on the results

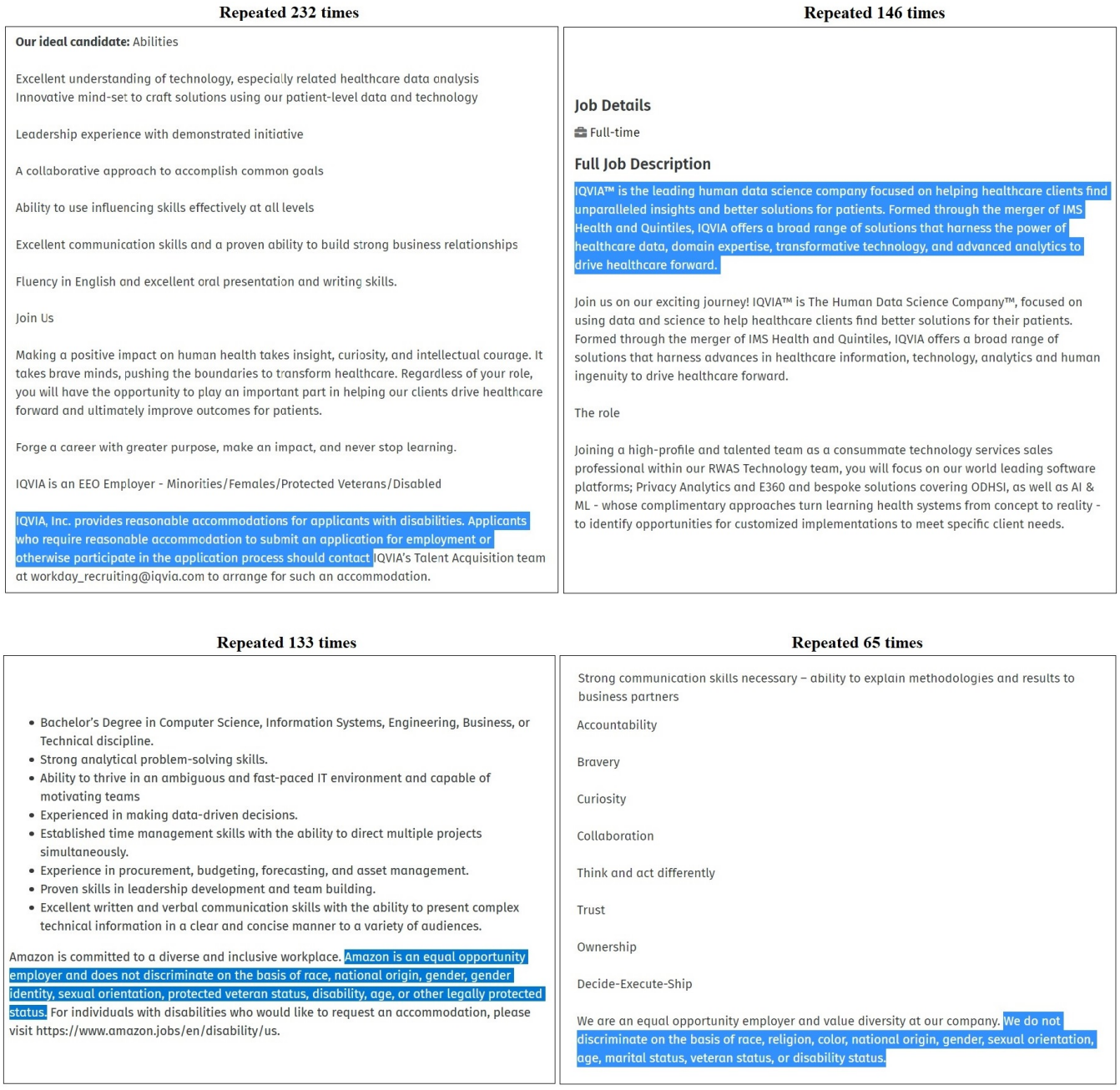

Figure 4: Examples of repeated phrases (marked in blue) within the corpus of documents.

of the LDA, these duplicate substrings were removed from the text documents. To ensure that all duplicated sentences, irrespectively from their length, were removed from the dataset, we developed an original extension of the standard stop-words preprocessing step, which we call N-gram Removal. By using N-gram Removal, the text is divided into a group of sentences with a predetermined number of words for each substring. In our case, after repeated trials we found that the number of words less than 20 may cause the removal of some words that should not be removed, and a number higher than 35 words did not specify any repeated substrings, so the substring length was determined to range between 20 and 35. We first found the 20-word long duplicated substrings that repeated more than 10 times in the corpus, and created a list of such sentences (stop N-grams). Then we progressively increased the window of words in each stop sentence (from 20 to 35 words), creating larger N-grams until we found all long repetitions in the corpus and removed them all from the original text accordingly. 3,330 substrings that were repeated with a frequency higher than 10 over the dataset

were removed from the text documents. it is obvious that in order to apply this method to another dataset a preliminary test phase devoted to parameter tuning has to be implemented.

3.5 LDA Implementation and Performance Evaluation

One of the most common approaches to topic modeling is the Latent Dirichlet Allocation (LDA) [15]. It models a fixed number of topics that are selected as a parameter based on the Dirichlet distribution for words and documents. The result is a flat, soft probabilistic clustering of terms by topics and documents by topics [2]. As mentioned earlier, the efficacy of LDA and the resulting human interpretability largely depends on the value of the number of topics, k, which requires prior knowledge about the contents of the dataset [20]. In this study, once we determined the possible optimal number of topics using different performance metrics, we applied LDA algorithm several times to two different versions of the document corpus: the first one being the original corpus we had before removing stop N-grams, and the second one being the cleaned corpus, with stop N-grams removed. Topic models learn topics represented as sets of important words automatically from unlabeled documents in an unsupervised way. This is an attractive method to bring structure to otherwise unstructured text data, but topics are not guaranteed to be interpretable. Therefore, numerous metrics have been proposed to distinguish between good and bad models. In this study, we adopted an extensive performance evaluation approach with the help of four performance metrics to evaluate the performance of LDA model before and after removing Stop N-grams. The four metrics we utilized were coherence, log-likelihood, perplexity, and human interpretability. Our approach to comparing the performance of the LDA algorithm consists of several important steps. The first step was to repeat the implementation of the LDA algorithm 100 times on different random sample of the same two corpora (with and without stop N-grams). Each sample included 80% of the documents drawn without repetition from the original corpus. Each time, after applying LDA, we computed the coherence, log-likelihood, and perplexity scores. Lastly, we evaluated the models by a human interpretability perspective, by creating a questionnaire consisting of ten queries. Each query contained two alternatives resulting from each of the corpora (with and without the application of N-gram Removal), and the alternatives were shown by means of table. Then, we presented the blind questionnaire to students and experts in the field of data science asking to score the most meaningful alternative for each topic model.

4 RESULTS AND DISCUSSION

By applying the LDA algorithm 100 times on the corpus of job ads descriptions in both cases (before and after the removal of Stop N-grams), using a number of topics K = 9 (the optimal value of number of topics in this study), we obtained different LDA models in both cases. We run an extensive performance evaluation with the help of four performance metrics to assess the effect of removing stop N-grams on the quality of the LDA models, namely coherence, log-likelihood, perplexity, and human interpretability. In this section, we present our findings according to each performance criteria.

4.1 Coherence

Topic Coherence measures score a single topic by measuring the degree of semantic similarity between high scoring words in the topic. These measurements help distinguish between topics that are semantically interpretable topics and topics that are just artifacts of statistical inference [35]. It is assumed that the higher the value of coherence, the higher probability of getting higher accuracy from that model [3]. Fig.5(a) shows that coherence score increases when stop N-grams are removed from the corpus of documents. Furthermore, we run a t-test to verify the statistical significance of the score increase. The t-test result confirmed the null hypothesis rejection, which assumed that N-gram Removal did not create a meaningful difference in the results of the analysis across the two samples. Therefore, removing stop N-grams from text documents proved to improve the quality of topics.

4.2 Log-likelihood

Log-likelihood measures the probability of the observed data, given the model how well a model fits the observed data. The higher the log-likelihood, the better the model for the given data [19]. Looking at the performance evaluation results in the 100 tests and comparing the log-likelihood scores in the two cases (before and after the removal of Stop N-grams), we found that that there was an increase in the log-likelihood scores, indicating an improvement in the quality of topics. Fig.5(a) shows the box plot of Log-likelihood improvement using Stop N-grams removal. Finally, the previous conclusion was also confirmed by the t-test, which rejected the same null hypothesis introduced above.

4.3 Perplexity

Perplexity as well is one of the intrinsic evaluation metrics often used to evaluate and compare the results of the LDA topics inference. It captures how surprised a model is of seeing new data it has not seen before, and is measured as the normalized log-likelihood of a held-out test set. It is assumed that the lower the value of perplexity, the higher will be the accuracy [27]. As in Fig.5(b), we found that perplexity was significantly lower when the Stop N-grams were removed from the corpus of the documents. As a result of the t-test, which rejected the null hypothesis, we concluded that there is a meaningful reduction of perplexity by applying Stop N-gram Removal.

4.4 Human Interpretability

Although there are many automated metrics used to evaluate the topic modeling, they are not considered sufficient to create high-quality topic models that are humanly interpretable. Incorporating human knowledge in unsupervised learning is a promising approach to creating high-quality topic models [4]. In this study, we created a questionnaire consisting of ten queries. Each query contains two alternatives (before and after the removal of Stop N-grams), the alternatives are described through tables. The rows of each table include the most significant words used to describe a job competency. The questionnaire was presented to numerous computer science students at the University of Perugia, in addition to experts in the field of data science. We then asked the survey participants to score the results of this procedure by choosing among

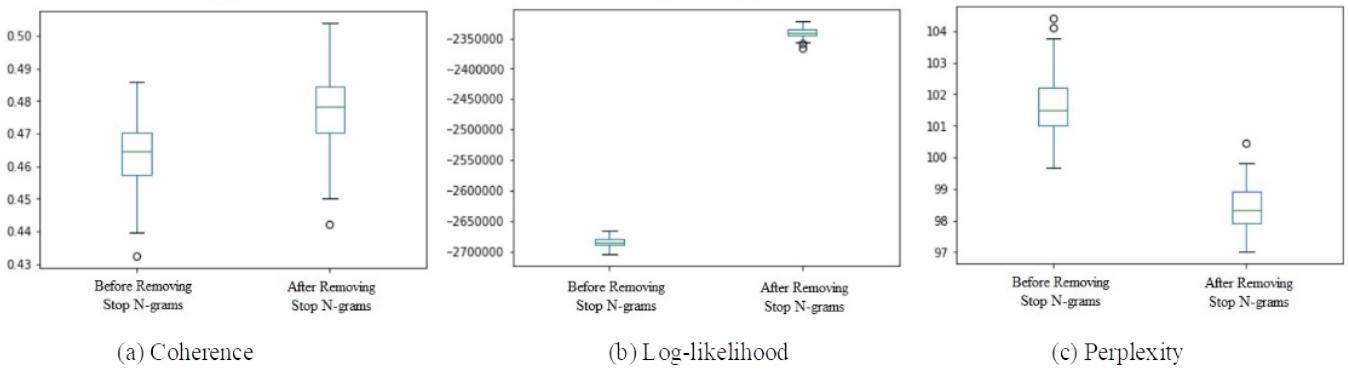

Figure 5: Improvements using stop N-gram Removal.

the two alternatives provided, the one that they believe makes the most sense.

Topic Number	Top words before removing stop N-grams	Top words after removing stop N-grams
1	healthcare, strategic, health, sale, clinic, patient, client, team, social, customer	marketing, digital, product, customer, team, sale, benefit, manager, strategic, competitive
2	develop, report, design, function, technology, team, program, product, solution, integration	develop, software, html, program, java, design, code, web, CSS, JavaScript
3	communication, aws, research, solution, develop, customer, technique, script, organize, method	team, solution, manage, technology, lead, leadership, administration, strategy, organize, expertise
4	security, manager, benefit, customer, application, system, team, model, insurance, technology	system, cycle, process, develop, solution, problem, security, technology, cyber, cybersecurity
5	manager, project, process, client, team, plan, develop, lead, assistant, leadership	manager, project, solve, plan, program, process, assistant, leadership, risk, organization
6	data, SQL, database, big, warehouse, ETL, model, design, Hadoop, process	data, SQL, database, big, warehouse, ETL, Hadoop, spark, script, structure
7	data, statistics, science, analytics, model, analysis, scientist, program, mathematics, develop	data, analytics, statistics, model, analyze, decision, regression, predict, sas, dataset
8	business, data, analytics, intelligence, team, customer, solution, product, develop, market	business, intelligence, analytics, strategy, finance, tableau, SQL, dashboard, process, decision
9	learning, machine, algorithm, team, product, science, deep, model, customer, scientist	learning, science, machine, model, python, TensorFlow, algorithm, problem, deep, NLP

Figure 6: An example of the LDA models used in the questionnaire.

Fig. 6 shows an example of the LDA models that we presented to the participants, where each table contained the top ten words for nine topics for each of the two alternatives (before and after the removal of Stop N-grams). As shown in Fig. 7 most of the survey respondents chose the alternative in which the stop N-grams were removed, which means that the removal of stop N-grams brought to more interpretable topic models.

In the next two subsections, we present two additional benefits that we found in relation of N-gram Removal.

4.5 Finding the Optimal Number of Topics

We found that an additional benefit of applying N-gram Removal was to improve and simplify the process of finding the optimal number of topics. As mentioned earlier, selecting the best number of topics (on which success of LDA depends on) can be challenging,

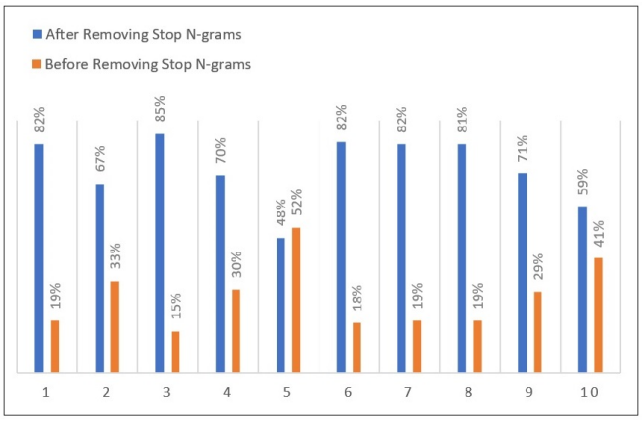

Figure 7: Survey results.

especially if there is no prior knowledge about the data. In this study, we found that removing the non-meaningul subsentences had a positive effect on the quality of the models, making it easier to determine the optimal number of topics. Fig. 8 shows a comparison of the chart used to determine the optimal number of topics by coherence scores in both cases (before and after the removal of stop N-grams). The figure visually suggests that the choice of the maximum value of coherence is easier after applying N-gram Removal.

the optimal number of topics .jpg

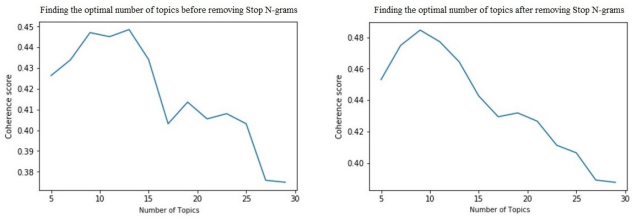

Figure 8: Finding the optimal number of topics.

4.6 Time Complexity of the Algorithm

By removing sentences that have multiple repetitions, we remove the low-level information tokens from our text. The removal of Stop N-grams definitely reduces the dataset size and, thus, reduces the training time due to the fewer number of tokens involved in the training. We assessed the performance of LDA also in terms of time complexity [26]. We found that the average execution of the LDA algorithm over 100 runs before removing Stop N-grams was 562.6 seconds, with a standard deviation being 1.1. Then, when Stop N-grams were removed, the average execution of the LDA algorithm for the 100 runs reduced to 475.3 seconds with a standard deviation being 1.3, which means that the removal of Stop N-grams helped to significantly reduce the execution time of the LDA algorithm.

5 CONCLUSIONS

In this paper, we introduced N-gram Removal, an unsupervised procedure to remove repeated snippets of texts of variable length from documents corpora. We investigated the effects of applying N-gram Removal as a preprocessing step for LDA topic modeling and evaluated the incremental performance using a dataset of about 9,000 web-scraped jobs ads. We have found that the application of N-gram Removal drove a statistically meaningful performance uplift in terms Coherence, Log-likelihood, Perplexity, and Human understanding. We also observed two additional benefits in using the proposed processing, namely: a 21% decrease of run-time, due to the simplification of the corpus, and an enhanced simplicity in performing hyperparameters optimization when applying LDA. Future research shall further investigate the role of Stop N-gram Removal procedure in other supervised and unsupervised text mining procedures and prove its general value in different documents corpora.

REFERENCES

[1] Suad A Alasadi and Wesam S Bhaya. 2017. Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences* 12, 16 (2017), 4102–4107.

[2] Mohamad Almgerbi, Andrea De Mauro, Adham Kahlawi, and Valentina Poggioni. 2021. A Systematic Review of Data Analytics Job Requirements and Online-Courses. *Journal of Computer Information Systems* (2021), 1–13.

[3] Hesam Amoualian, Wei Lu, Eric Gaussier, Georgios Balikas, Massih R Amini, and Marianne Clausel. 2017. Topical coherence in lda-based models through induced segmentation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1799–1809.

[4] Sudeep Bhatia, Russell Richie, and Wanling Zou. 2019. Distributed semantic representations for modeling human judgment. *Current Opinion in Behavioral Sciences* 29 (2019), 31–36.

[5] Biraj Dahal, Sathish AP Kumar, and Zhenlong Li. 2019. Topic modeling and sentiment analysis of global climate change tweets. *Social Network Analysis and Mining* 9, 1 (2019), 1–20.

[6] Jitesh Kumar Dewangan, Aakanksha Sharaff, and Sudhakar Pandey. 2020. Improving topic coherence using parsimonious language model and latent semantic indexing. In *ICDSMLA 2019*. Springer, 823–830.

[7] Anjie Fang, Craig Macdonald, Iadh Ounis, and Philip Habel. 2016. Examining the coherence of the top ranked tweet topics. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 825–828.

[8] João Ferreira, Hugo Gonçalo Oliveira, and Ricardo Rodrigues. 2019. Improving NLTK for processing Portuguese. In *8th Symposium on Languages, Applications and Technologies (SLATE 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[9] Salvador García, Sergio Ramírez-Gallego, Julián Luengo, José Manuel Benítez, and Francisco Herrera. 2016. Big data preprocessing: methods and prospects. *Big Data Analytics* 1, 1 (2016), 1–22.

[10] Matthew Gentzkow, Bryan Kelly, and Matt Taddy. 2019. Text as data. *Journal of Economic Literature* 57, 3 (2019), 535–74.

[11] Kranti Vithal Ghag and Ketan Shah. 2015. Comparative analysis of effect of stopwords removal on sentiment classification. In *2015 international conference on computer, communication and control (IC4)*. IEEE, 1–6.

[12] Mahedi Hasan, Anichur Rahman, Md Razaul Karim, Md Saikat Islam Khan, and Md Jahidul Islam. 2021. Normalized Approach to Find Optimal Number of Topics in Latent Dirichlet Allocation (LDA). In *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*. Springer, 341–354.

[13] Vera Ignatenko, Sergej Koltcov, Steffen Staab, and Zeyd Boukhers. 2019. Fractal approach for determining the optimal number of topics in the field of topic modeling.. In *Journal of Physics: Conference Series*, Vol. 1163. IOP Publishing, 012025.

[14] Karoliina Isoaho, Daria Gritsenko, and Eetu Mäkelä. 2021. Topic modeling and text analysis for qualitative policy research. *Policy Studies Journal* 49, 1 (2021), 300–324.

[15] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. 2019. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications* 78, 11 (2019), 15169–15211.

[16] Jashanjot Kaur and Preetpal Kaur Buttar. 2018. STOPWORDS REMOVAL AND ITS ALGORITHMS BASED ON DIFFERENT METHODS. *International Journal of Advanced Research in Computer Science* 10, 5 (2018).

[17] Jashanjot Kaur and Preetpal Kaur Buttar. 2018. A systematic review on stopword removal algorithms. *International Journal on Future Revolution in Computer Science & Communication Engineering* 4, 4 (2018), 207–210.

[18] Hayato Kobayashi. 2014. Perplexity on reduced corpora. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 797–806.

[19] Denis Kochedykov, Murat Apishev, Lev Golitsyn, and Konstantin Vorontsov. 2017. Fast and modular regularized topic modelling. In *2017 21st Conference of Open Innovations Association (FRUCT)*. IEEE, 182–193.

[20] Fedor Krasnov and Anastasiia Sen. 2019. The number of topics optimization: Clustering approach. *Machine Learning and Knowledge Extraction* 1, 1 (2019), 416–426.

[21] Vlad Krotov and Leiser Silva. 2018. Legality and ethics of web scraping. (2018).

[22] Akshay Kulkarni and Adarsha Shivananda. 2019. Deep learning for NLP. In *Natural language processing recipes*. Springer, 185–227.

[23] Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. 530–539.

[24] Baojun Ma, Hua Yuan, Yan Wan, Yu Qian, Nan Zhang, and Qiongwei Ye. 2016. Public opinion analysis based on probabilistic topic modeling and deep learning. (2016).

[25] Nicolas Pröllochs and Stefan Feuerriegel. 2020. Business analytics for strategic management: Identifying and assessing corporate challenges via topic modeling. *Information & Management* 57, 1 (2020), 103070.

[26] Jipeng Qiang, Ping Chen, Tong Wang, and Xindong Wu. 2017. Topic modeling over short texts by incorporating word embeddings. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 363–374.

[27] Santosh Kumar Ray, Amir Ahmad, and Ch Aswani Kumar. 2019. Review and implementation of topic modeling in Hindi. *Applied Artificial Intelligence* 33, 11 (2019), 979–1007.

[28] Debabrata Sarddar, Raktim Kumar Dey, Rajesh Bose, and Sandip Roy. 2020. Topic Modeling as a Tool to Gauge Political Sentiments from Twitter Feeds. *International Journal of Natural Computing Research (IJNCR)* 9, 2 (2020), 14–35.

[29] Serhad Sarica and Jianxi Luo. 2020. Stopwords in technical language processing. *arXiv preprint arXiv:2006.02633* (2020).

[30] Stefano Sbalchiero and Maciej Eder. 2020. Topic modeling, long texts and the best number of topics. Some Problems and solutions. *Quality & Quantity* 54, 4 (2020).

[31] Alexandra Schofield, Måns Magnusson, and David Mimno. 2017. Pulling out the stops: Rethinking stopword removal for topic models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. 432–436.

[32] Alexandra Schofield, Laure Thompson, and David Mimno. 2017. Quantifying the effects of text duplication on semantic models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2737–2747.

[33] Suvarn Sharma and Amit Bhagat. 2016. Data preprocessing algorithm for web structure mining. In *2016 Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS)*. IEEE, 94–98.

[34] Chang-Woo Song, Hoill Jung, and Kyungyong Chung. 2019. Development of a medical big-data mining process using topic modeling. *Cluster Computing* 22, 1 (2019), 1949–1958.

[35] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. 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*. 952–961.

[36] Symeon Symeonidis, Dimitrios Effrosynidis, and Avi Arampatzis. 2018. A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. *Expert Systems with Applications* 110 (2018), 298–310.
[37] David Mathew Thomas and Sandeep Mathur. 2019. Data analysis by web scraping using python. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 450–454.
[38] Nguyen Anh Tu, Dong-Luong Dinh, Mostofa Kamal Rasel, and Young-Koo Lee. 2016. Topic modeling and improvement of image representation for large-scale image retrieval. *Information Sciences* 366 (2016), 99–120.
[39] Hongbin Wang, Jianxiong Wang, Yafei Zhang, Meng Wang, and Cunli Mao. 2019. Optimization of Topic Recognition Model for News Texts Based on LDA. *J. Digit. Inf. Manag.* 17, 5 (2019), 257.
[40] Xing Wei and W Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 178–185.
[41] Thanakorn Yarnguy and Wanida Kanarkard. 2018. Tuning Latent Dirichlet Allocation parameters using ant colony optimization. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 10, 1-9 (2018), 21–24.
[42] Bo Zhao. 2020. Encyclopedia of Big Data. *Encyclopedia of Big Data* (2020), 3–5.
[43] Weizhong Zhao, James J Chen, Roger Perkins, Zhichao Liu, Weigong Ge, Yijun Ding, and Wen Zou. 2015. A heuristic approach to determine an appropriate number of topics in topic modeling. In *BMC bioinformatics*, Vol. 16. Springer, 1–10.