Efficient Text Analytics: CTF-IDF and Faster IRLBA for Improved Model Performance and Reduced Dimensionality

***Abstract*—The uprising of deep learning methodology and practice in recent years has brought about a severe consequence of increasing carbon footprint due to the insatiable demands on computational resources and power. The field of text analytics also experienced a massive transformation on this trend of monopolizing methodology. In this paper, the original TF-IDF algorithm has been modified and Clement Term Frequency–Inverse Document Frequency (CTF-IDF) has been proposed for data preprocessing. This paper primarily discusses on the effectiveness of classical machine learning techniques in text analytics with CTF-IDF and faster IRLBA algorithm for dimensionality reduction. The introduction of both of these techniques in the conventional text analytics pipeline ensures a more efficient, faster, and less computationally intensive application when compared with deep learning methodology regarding carbon footprint with minor compromise in accuracy. The experimental results also exhibit a manifold of reduction in time complexity and improvement of model accuracy for the classical machine learning methods discussed further in this paper.**

***Index Terms*—Tf-Idf, Ctf-Idf, IRLBA, Dimensionality Reduction, Text Analytics, BERT, SPAM.**

I. Introduction

Since the advent of modern technology and the internet, the ubiquitous application of electronic media in academia and research, news publications, social media, government, and non-government sites has massively contributed to the upsurge of text data stored in digital appliances. To extract the essential information from this highly unstructured data, we must first employ a variety of data mining techniques to uncover potentially valuable patterns from this enormous amount of data. Text analytics is the process of retrieving unstructured data and transforming it into structured data with the application of suitable algorithms to find patterns and trends, and classify the texts into distinct groups [4]. The standard text analytics methodology can be burdensome for any small machine when dealing with big unstructured data. Contemporary text classification task requires a copious amount of text documents in each training session. Consequentially, the feature space may explode with sparse and redundant data when transformed into a document frequency matrix. This ultimately results in a heavy toll on computational power and the time required to build any machine learning and deep learning model for the prediction and classification of text data. This is proverbially known as the curse of dimensionality. On the other hand, Deep Learning methods like Bidirectional LSTM and Transformer based models like Google's BERT have shown significant improvements in the precision of text analysis but at a cost of huge computational time and resources, therefore aggravating the issue of carbon footprint.

Term Frequency–Inverse Document Frequency (TF-IDF) is considered one of the stepping stones for transforming tokenized textual data. According to a 2015 survey, TF-IDF is used by 83% of recommender systems based on textual data in digital libraries [1]. This statistical metric quantifies the significance of a word in a corpus or collection of documents [2]. The classical TF-IDF severely penalizes each word/token in the documents based on the frequency of the words among the whole corpus on a logarithmic scale which in return creates a wide range of TF-IDF values and sometimes diminishes the whole effect of various keywords [3]. We have proposed a moderate and clement approach to address this issue with CTF-IDF. The experimental results showed improvement regarding model accuracy in implementing CTF-IDF over the classical TF-IDF on text classification tasks when combined with the proposed algorithm for dimensionality reduction.

Dimensionality reduction is the method of converting high-dimensional data into a meaningful representation of lesser dimensionality [5]. This method is considered essential for transforming the features into a more compact form to increase the learning efficiency of the algorithms when the number of features exceeds significantly. Dimension reduction techniques can be utilized both with supervised and unsupervised methods. However, depending on the kind of method utilized, the properties of the dimensionality reduction technique change. For instance, dimensionality reduction techniques for unsupervised learning algorithms should work to reduce the loss of feature information. On the other hand, the goal should be to maximize class information in the case of supervised learning. There is no single strategy that works in every circumstance due to the complexity of the dimension reduction process. As a result, numerous dimension reduction techniques have been developed and proposed over the years and put to the test in various fields of study and application domains.

In this paper, we adopted "The augmented implicitly restarted Lanczos bidiagonalization algorithm" for dimensionality reduction [6]. This algorithm computes partial singular values decomposition and finds a few of the largest or smallest singular values along with the singular vectors of a sparse or dense matrix. This is a fast and memory-efficient method that serves to alleviate the problem of employing complex machine-learning algorithms and improving the overall performance of the models at the same time.

For the initial development of the methodology, a SPAM dataset was used that consists of 5000 text data collected primarily from phones as text messages [citation 41]. The dataset was classified into two basic categories: Spam and Ham (not Spam). After the development of the methodology on this dataset, a comparative analysis was done on another similar SMS Phishing dataset (citation 7) in terms of model accuracy and run time. For the analysis, classical machine learning techniques like Decision Tree and Support Vector Machine have been used to evaluate the robustness and efficiency of the proposed methodology in contrast with the traditional methodology in text analytics and deep learning model like Transformer (BERT).

The contributions of this paper are,

- Introducing a modified data processing algorithm CTF-IDF to reduce the penalty received by each term in the corpus.
- Incorporation of a faster and memory-efficient "Augmented implicitly restarted Lanczos bidiagonalization" algorithm for dimensionality reduction.
- The combined effect of both of these methods in the traditional text analytics pipeline expedited the computational time and reduced the requirement for computational resources.

The proposed method is efficient on addressing the issue of the rising carbon footprint due to the advent of Deep Learning methodology while still increasing the robustness of the trained models.

II. Background Study

A. *Previous Works*

Xia Hu ET el elaborately discussed the traditional methodology for text analytics consisting of three key components: Text Preprocessing, Text Representation, and Knowledge Discovery. [19]

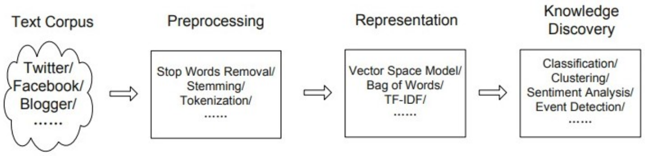
Text Corpus
Preprocessing
Representation
Knowledge Discovery
Twitter/ Facebook/ Blogger/ ......
Stop Words Removal/ Stemming/ Tokenization/ ......
Vector Space Model/ Bag of Words/ TF-IDF/ ......
Classification/ Clustering/ Sentiment Analysis/ Event Detection/ ......

Fig. 1

Our work mainly focuses on the representation stage with a new TF-IDF and a faster method for dimensionality reduction of the vector space.

TF-IDF (Term Frequency-Inverse Document Frequency) is traditionally used as a statistical method for evaluating the importance of a word in a document in relation to a corpus of documents. TF-IDF has been widely used in text classification, such as spam filtering and sentiment analysis. Research has shown that TF-IDF often outperforms other feature selection methods, such as a bag of words and n-grams [8]. TF-IDF has also been used for keyword extraction, as it assigns high weight to important terms in a document and can identify the most relevant words for summarizing a document [9]. In high-dimensional text data, TF-IDF is useful for dimensionality reduction, as it reduces the number of features while retaining important information [10]. In Recent research, TF-IDF has been combined with word embedding methods to improve the performance of text classification tasks [12]. Hybrid Approaches like merging TF-IDF with other methods, such as word2vec, to improve its performance in text analysis [11]. This paper introduces a modified TF-IDF for data representation.

On the other hand, dimensionality reduction is an essential technique in text analytics for reducing the high dimensionality of textual data while retaining its most informative features. Principal Component Analysis (PCA) is a commonly used dimensionality reduction technique that involves projecting data onto a lower-dimensional space while retaining as much variance as possible. In text analytics, PCA has been used for tasks such as sentiment analysis, document classification, and topic modeling [13] [14] [15]. Latent Dirichlet Allocation (LDA) is a generative probabilistic model that discovers latent topics in a corpus of text. LDA has been used for tasks such as topic modeling, document classification, and information retrieval. [16] [17] [18]. Non-negative Matrix Factorization (NMF) is a matrix decomposition technique that factorizes a matrix into two non-negative matrices, which can be interpreted as representing latent topics and word distributions. NMF is suitable for tasks such as topic modeling, document clustering, and sentiment analysis [20] [21]. Singular Value Decomposition (SVD) is a matrix factorization technique used to decompose a matrix into its constituent parts. It is applied mostly in document classification, topic modeling, and information retrieval. [22]. For visualization of high-dimensional word embedding and document clustering t-distributed Stochastic Neighbor Embedding (t-SNE) is preferred [23] [24] [25]. Word2Vec is frequently used for tasks such as text classification, sentiment analysis, and information retrieval [27] [26]. Another modern tool FastText has been applied in text classification, named entity recognition, and sentiment analysis. [28] [29]. GloVe is a technique that learns word embedding by factorizing a matrix of word co-occurrence statistics that has been implemented to find word similarity and text classification [30].

Dimension reduction methods have been proven to be crucial for many text analytics tasks, and the choice of method depends on the specific task and the characteristics of the data. PCA, LDA, NMF, SVD, t-SNE, Word2Vec, FastText, and GloVe are some of the popular dimension reduction methods used in text analytics. In this paper, we experimented with a faster singular value decomposition method for dimensionality reduction.

III. Proposed Method

The aim of this experiment is to provide an improved and efficient methodology in text analytics with classical machine learning algorithms. The basic framework consists of data preprocessing, feature extraction with Ctf-idf, projection of a Ctf-idf document vector into the SVD semantic space with IRLBA, and classification stages. The following subsection will provide a brief explanation of the success measure as well.

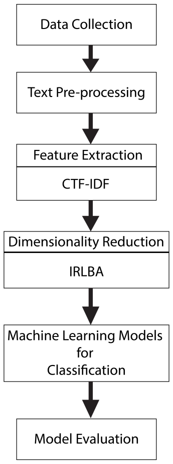

Fig. 2: Proposed method of the research.

A. *Dataset Details*

A smaller and more compact SPAM dataset has been used for the development of the method, and the SMS Phishing dataset has been used as a benchmark dataset to check the robustness of the method.

*1) **Spam Dataset**:* The SMS Spam Collection consists of labeled SMS messages gathered for studying mobile phone spam [41]. It is publicly available for research purposes. The dataset contains 5,574 authentic English text messages that are not encoded. These messages are categorized as either legitimate (ham) or spam. According to Collins Dictionary, spam messages are unsolicited electronic mail or messages sent simultaneously to a number of email addresses or mobile phones.

*2) **SMS Phishing Dataset**:* The dataset consists of categorized text messages gathered for SMS Phishing investigation. It comprises 5,971 text messages labeled as either ”Legitimate” (Ham), ”Spam,” or ”Smishing”. The collection encompasses 489 spam messages, 638 smishing messages, and 4,844 ham messages. For this research, the text documents labeled as ’Smishing’ and ’Spam’ are grouped together as ’Spam’. This dataset contains raw message content that can be preprocessed for extracting further attributes to be followed by classification using Machine Learning techniques or can be used as labeled data in Deep Learning. Additionally, the dataset includes extracted attributes from malicious messages that aid in classifying messages as either malicious or legitimate.

B. *Preprocessing techniques*

The text documents will undergo pre-processing which involves various tasks such as tokenization, removal of stop words, conversion to lowercase, and stemming. Tokenization refers to dividing a text into tokens such as words or phrases. Stop words are words that appear frequently in texts, such as conjunctions or prepositions, regardless of the topic. Lowercase conversion involves changing all uppercase letters to lowercase letters before the classification stage. Stemming is the process of obtaining the root or stem of derived words, and the commonly used stemming process for English is Porter Stem, which was introduced by N. Milić-Frayling. [31].

C. *Feature Extraction*

*1) **Document Frequency matrix**:* Tokenization of the corpus is followed by the creation of a document frequency matrix to represent the connection between terms and documents. In this matrix, each row corresponds to a document, while each column corresponds to a term, and the value entered represents the frequency of the term’s occurrence within that particular document.

*2) **N-gram modeling**:* n-gram models are now widely applied in many computational fields including text analytics. There are many variants of n-grams depending on the sequential order. In order to reduce the load of quantitative analysis and sparsely distributed data, only the unigram model has been chosen to follow through the experiment.

*3) **Modified TF-IDF (CTF-IDF)**:* TF-IDF, short for Term Frequency-Inverse Document Frequency, combines two distinct measurements, TF and IDF, to analyze multiple documents. When dealing with multiple documents, TF-IDF is employed, leveraging the notion that uncommon and infrequent words provide greater insights into the content of a document compared to frequently occurring words across all documents. A modified CTF-IDF algorithm is proposed for this experiment. The modified algorithm assigns greater IDF values for the rarer terms in the whole corpus are calculated through the inverse hyperbolic sine function. The most infrequent terms in the corpus convey the most significance in classifying the document whereas the common terms should have very little significance in determining the nature of the document. In that case, CTF-IDF is more prominent in leveraging the rarity of any term and assigning much higher IDF value attached to it. CTF-IDF is also less inclement on penalizing the most frequent words so the CTF-IDF value of any word never diminishes. Not losing any information from the corpus is necessary on the application of matrix decomposition in the later stage for dimensionality reduction. The mathematical details are given below,

$$tf(t,d) = \frac{f_d(t)}{max_{w\epsilon d} f_d(w)} \tag{1}$$

$$idf(t,D) = arcsinh(\frac{|D|}{|d\epsilon D : t\epsilon d|}) \tag{2}$$

$$fidf(t, d, D) = tf(t, d) * idf(t, D) \quad (3)$$

$$tfidf'(t, d, D) = \frac{idf(t, D)}{|D|} + tfidf(t, d, D) \quad (4)$$

Here,

- $f_d(t)$ = Frequency of term **t** in document **d**
- D = Corpus of documents

*4) **Dimensionality Reduction with IRLBA**:* Another fundamental aspect of this experiment is to reduce the training time of each learning algorithm as much as possible while preserving accuracy. The proposed method of augmented implicitly restarted Lanczos bidiagonalization algorithm (IRLBA) is an extension of the Lanczos bidiagonalization algorithm that finds an estimated number of largest or the smallest singular values and corresponding singular vectors of a sparse or dense matrix using the mechanism of Baglama and Reichel (citation 34). It is a fast and memory-efficient method for truncated singular value decomposition and principal components analysis [35].

In this study, the transformation through IRLBA was iterated over many numbers of right singular vectors and an optimum 300 most significant right singular vectors have been chosen based on the descending order of singular values. The number of iterations determines the number of desired singular values to compute. The mathematical formulation is provided below. Given an input matrix A of size m x n, where m ¿= n, this algorithm iteratively constructs two matrices B and C, both bidiagonal, such that B is similar to A.

- **Initialization**
  - Choose a starting vector v_1 of size m x 1 with unit norm: $||v_1||$ = 1
  - Set $\beta_0 = 0$ and $v_0 = 0$
- **Iteration**
  For k = 1 to p (where p is the desired number of singular values):
  - Compute w_k = A * v_k - β_{k-1} * v_{k-1}.
  - $\alpha_k = ||w_k||$.
  - Normalize w_k: v_{k+1} = $\frac{w_k}{alpha_k}$.
  - Compute z_k = $A^T * v_{k+1} - \alpha_k * v_k$.
  - $\beta_k = ||z_k||$.
  - Normalize $z_k : u_{k+1} = \frac{z_k}{\beta_k}$.
- **Implicit Restart**
  - Compute the bidiagonalization of B and C for the first p iterations using the Lanczos bidiagonalization algorithm.
- **Augmentation**
  - Compute the singular value decomposition of the bidiagonal matrix C of size p x p: C = U * S * V^T.
- **Implicit Restart (continued)**
  - Set B = U^T * B * V, which updates B to be more similar to A.
- Repeat Steps 2-5 until convergence or desired accuracy is achieved.

At the end of the algorithm, B and C will be similar, and the singular values of A can be computed from the singular values of C.

D. Classification Methods

For simplicity, Support Vector Machine and Decision Tree Classifiers have been chosen as the classical machine learning algorithms for classification in this experiment. On the other hand, the deep learning-based Transformer model BERT has been employed for comparative purposes with the proposed methodology.

SVM is a learning algorithm designed for solving two-group classification problems, as originally introduced by [36]. In this case, SVM is employed to categorize texts into positive or negative classes. SVM is particularly effective for text classification due to its ability to handle a large number of features with few examples when the problems can be linearly separated, as stated in [37].

Decision tree classifiers are widely recognized as one of the most popular and prominent approaches for representing classifiers in data classification. Decision Trees often replicate human cognitive processes when making decisions, thereby making them easily comprehensible and interpretable. A 10-fold cross-validation method has been employed during the training session.

Bidirectional Encoder Representations from Transformers (BERT) is a highly sought-after new language representation model designed to pre-train deep bidirectional representations from the unlabeled text by joint conditioning on both left and right contexts in all layers [39]. In this experiment, Multilingual BERT (mBERT) is used and it is flexible in providing sentence representations for 104 languages [40]. It is recommended that no data preprocessing is required for modeling in BERT.

E. Performance Parameters

Four effective measures that have been used in this study are based on confusion matrix output, which are Sensitivity, Specificity, Balanced Accuracy, and Training time.

- Sensitivity or Recall (True Positive Rate)=TP / (TP + FN)
- Positive Predictive Value or Precision =TP / (TP + FP)
- F-Measure = 2*(Precision*Recall) / (Precision + Recall)
- Training Time= Amount of time required to train the Learning Algorithm

The usefulness of these metrics is ubiquitous in text classification for comparative analysis among numerous learning algorithms. The F-measure serves as a middle ground between recall and precision, representing a balance between the two. Its value is significant only when both recall and precision are at high levels. When α (a parameter) equals 0, the F-measure is equivalent to recall, while α = 1 makes it equivalent to precision. The F-measure ranges from 0 to 1, with 0 indicating that no relevant documents were retrieved and 1 indicating that all retrieved documents are relevant and all relevant documents were retrieved.

F. *Execution Environments*

All the modeling with classical learning algorithms and analyses were carried out in R (v. 4.0.3) on an old computer equipped with a Core 2 Duo processor (Operating at 3.00 GHz base) and 8 GB of RAM. The modeling with Transformers (mBERT) was executed in a computer with an Intel Core i5-7500 processor (Base clock 3.40 GHz), 32 GB of RAM, and NVIDIA GeForce GTX 1050Ti graphics card with 4 GB DDR5 RAM and 768 CUDA cores. Text preprocessing, Decision Trees, SVMs, IRLBA, and Transformers were respectively implemented using the quanteda, caret, e1071, irlba, TensorFlow, reticulate, and keras packages in R.

IV. Results and Discussions

For each trained model, a couple of comparisons were made in terms of model accuracy and training time. Each dataset was split into two parts, one for training and the other for testing: SPAM dataset with a ratio of 70:30: 70% for training and 30% testing. The SMS Phishing dataset was also partitioned with a ratio of 70:30. Table I, 2, 3, and 4 summarizes the performance metrics for the decision tree and Support Vector Machine Model for the SPAM and SMS Phishing dataset respectively. Table 5 accumulates the results after training both of the SPAM and SMS Phishing data with the Transformers model (mBERT).
In the first phase for the SPAM dataset, all the raw text data went through the preprocessing stages followed by feature extraction through traditional tf-idf and modified tf-idf (Ctf-idf) respectively. Decision tree models were then built upon the tf-idf and Ctf-idf transformed data for classification. From the table I, it can be seen that there are no discernible changes in the performance metrics for both cases in the decision tree models. The time required to train each of the models was 13 minutes and 12 minutes respectively.

Models	Precision	Recall	F1-score	Training Time
TF-IDF	0.9627	0.9687	0.9657	13 min
CTF-IDF	0.9627	0.9687	0.9657	12 min
TF-IDF(IRLBA)	0.9869	0.9420	0.964	17 sec
CTF-IDF(IRLBA)	0.9889	0.9508	0.97	16 sec

TABLE I: SPAM Data 10-fold Cross-Validation Performance Metrics for Decision Tree Model

In the second phase for the SPAM dataset, the tf-idf and Ctf-idf feature extracted models were transformed by IRLBA. After the transformation, it required only 17 seconds to train a decision tree model on tf-idf and 16 seconds on Ctf-idf transformed corpus (Table 1). The computational time was significantly reduced by the application of IRLBA. Table 1 also shows the combined effect of Ctf-idf and IRLBA transformation helps to increase the F1-score from 96.57% to 97% in contrast with the tf-idf and IRLBA transformed data where it decreases slightly.

Models	Precision	Recall	F1-score	Training Time
TF-IDF	0.9807	0.9827	0.9817	35 sec
CTF-IDF	0.9793	0.9832	0.9812	37 sec
TF-IDF(IRLBA)	0.975	0.9912	0.983	5 sec
CTF-IDF(IRLBA)	0.9783	0.9965	0.9873	5 sec

TABLE II: SPAM Data Performance Metrics for SVM

All the procedures were the same for Support Vector Machine as well. It can be seen from table 2 that F1-Scores are hovering over 98% for preliminary tf-idf and Ctf-idf transformed data with 35 and 37 seconds of training time for each model respectively. After the projection of tf-idf and Ctf-idf transformed data in the semantic space through IRLBA, the overall performance of the models increased. It can be seen from Table 2 that the F-Scores improved in both cases but the Ctf-idf-IRLBA transformed data performed better with an F-Score of 98.73%. Not only that, the computational time was also vastly reduced to 5 seconds only for training each of the models in SVM.
For SPAM dataset, the Transformers (mBERT) models raised the training and the validation accuracy slightly above 99% but with the expense of a huge computational power (Table 5). It required NVIDIA graphics card to run all the CUDA cores simultaneously for up to 1:30 hours on the minimum level to train the model.

Models	Precision	Recall	F1-score	Training Time
TF-IDF	0.9767	0.9184	0.9467	10.33 min
CTF-IDF	0.9767	0.9184	0.9467	9.67 min
TF-IDF(IRLBA)	0.9413	0.9601	0.9506	22.09 sec
CTF-IDF(IRLBA)	0.9520	0.9683	0.9601	19.46 sec

TABLE III: SMS Phishing Data 10-fold cross-validation Performance Metrics for Decision Tree Model

The robustness of the methodology is tested on the SMS Phishing dataset. Table 3 shows, without the application of IRLBA, the Ctf-idf performs no better than the tf-idf transformed data with an F1-score of 94.67% in decision tree models. But, the training time for the Ctf-idf model was slightly better than the tf-idf model taking 9.67 minutes. After the IRLBA transformation of the tf-idf and Ctf-idf models, the F-1 scores increased above 95% for both models but again Ctf-idf-IRLBA performed better. Table 3 also shows the tremendous reduction in training time requiring approximately 20 seconds which took almost 10 minutes previously.

Models	Precision	Recall	F1-score	Training Time
TF-IDF	0.9832	0.9642	0.9736	14.312 sec
CTF-IDF	0.9854	0.9738	0.9796	12.9 sec
TF-IDF(IRLBA)	0.9703	0.9913	0.9807	9.82 sec
CTF-IDF(IRLBA)	0.9737	0.9945	0.9840	9.8 sec

TABLE IV: SMS Phishing Data Performance Metrics for SVM

For Support Vector Machine, Table 4 shows the methodology exhibits the same characteristic reduction in training time but the difference is not that significant. Again, the Ctf-idf transformed model worked better on SMS Phishing data than the tf-idf model before transforming the models through IRLBA. After the application of IRLBA it can be seen from Table 4 that the F1-Scores again improved up to 98% for both of the models and still Ctf-idf-IRLBA maintained better performance.

Data	No. of Epoch	Training Accuracy	Validation Accuracy	Training Time
SPAM Data	7	0.9913	0.9904	1:30 hr
SMS Phishing Data	15	0.9976	0.9888	1:43 hr

TABLE V: Transformer(mBERT) Models Performance Metrics for SPAM and IMDB Data

In the case of the Transformers (mBERT) model, table 5 shows the model performs a little better than the SVM with a validation accuracy of 98.88% for the SMS Phishing dataset. As expected the transformer model took a considerable amount of training time over 1 hour and 35 minutes with the help of a NVIDIA graphics card at the backend with all the CUDA cores running simultaneously.

It is quite evident from the experiment that Ctf-idf transformation of the dataset with the combination of IRLBA algorithm for dimensionality reduction is significantly faster in training any classical machine learning model and at the same time it improves the performance of the model after shrinking and transforming the feature space from thousands of columns to a handful of informative columns. The transformers (BERT) models are great in producing state-of-the-art accuracy but with a huge cost of computational power and producing a higher carbon footprint.

V. Discussion and Conclusions

In this era of deep learning, we believe the classical machine learning techniques still have much to offer for the greater good of humanity. The proposed methodology in this paper aims at increasing the efficiency of classical machine learning algorithms with producing very little carbon footprint. In the future mathematical techniques for measuring the optimum number of right singular vectors using faster matrix decomposition method carrying most of the vital information from sparse matrices can be developed to transform the heterogeneous data more swiftly and efficiently.

References

[1] Beel, J., Gipp, B., Langer, S. et al. Research-paper recommender systems: a literature survey. Int J Digit Libr 17, 305–338 (2016). https://doi.org/10.1007/s00799-015-0156-0

[2] Rajaraman, A., &amp; Ullman, J. D. (n.d.). Data Mining (Chapter 1) - mining of massive datasets. Cambridge Core. https://www.cambridge.org/core/books/abs/mining-of-massive-datasets/data-mining/E5BFF4C1DD5A1FB946D616D619B373C2

[3] L. Cheng, Y. Yang, K. Zhao, and Z. Gao, "Research and Improvement of TF-IDF Algorithm Based on Information Theory," Aug. 2018, doi: https://doi.org/10.1007/978-3-030-14680-1_67.

[4] J. E. McLaughlin, K. Lyons, C. Lupton-Smith, and K. Fuller, "An introduction to text analytics for educators," Currents in Pharmacy Teaching and Learning, vol. 14, no. 10, pp. 1319–1325, Oct. 2022, doi: https://doi.org/10.1016/j.cptl.2022.09.005.

[5] L. van der Maaten, E. Postma, and H. Herik, 'Dimensionality Reduction: A Comparative Review', Journal of Machine Learning Research - JMLR, vol. 10, 01 2007.

[6] Baglama, James, and Lothar Reichel. "Augmented implicitly restarted Lanczos bidiagonalization methods." SIAM Journal on Scientific Computing 27.1 (2005): 19-42.

[7] mishra, sandhya; Soni, Devpriya (2022), "SMS PHISHING DATASET FOR MACHINE LEARNING AND PATTERN RECOGNITION", Mendeley Data, V1, doi: 10.17632/f45bkkt8pr.1.

[8] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, 'The Impact of Features Extraction on the Sentiment Analysis', Procedia Computer Science, vol. 152, pp. 341–348, 01 2019.

[9] Erra, U., Senatore, S., Minnella, F., & Caggianese, G. (2015). Approximate TF–IDF based on topic extraction from massive message stream using the GPU. Information Sciences, 292, 143-161. https://doi.org/10.1016/j.ins.2014.08.062

[10] Dhar, A., Dash, N.S., Roy, K. (2018). Application of TF-IDF Feature for Categorizing Documents of Online Bangla Web Text Corpus. In: Bhateja, V., Coello Coello, C., Satapathy, S., Pattnaik, P. (eds) Intelligent Engineering Informatics. Advances in Intelligent Systems and Computing, vol 695. Springer, Singapore. https://doi.org/10.1007/978-981-10-7566-7_6

[11] C. -z. Liu, Y. -x. Sheng, Z. -q. Wei and Y. -Q. Yang, "Research of Text Classification Based on Improved TF-IDF Algorithm," 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), Lanzhou, China, 2018, pp. 218-222, doi: 10.1109/IRCE.2018.8492945.

[12] C. De Boom, S. Van Canneyt, T. Demeester, and B. Dhoedt, 'Representation learning for very short texts using weighted word embedding aggregation', Pattern Recognition Letters, vol. 80, pp. 150–156, 2016.

[13] G. Zu, W. Ohyama, T. Wakabayashi, and F. Kimura, 'Accuracy Improvement of Automatic Text Classification Based on Feature Transformation', in Proceedings of the 2003 ACM Symposium on Document Engineering, Grenoble, France, 2003, pp. 118–120.

[14] X. Han, G. Zu, Wataru Ohyama, T. Wakabayashi, and F. Kimura, "Accuracy Improvement of Automatic Text Classification Based on Feature Transformation and Multi-classifier Combination," pp. 463–468, Nov. 2004, doi: https://doi.org/10.1007/978-3-540-30483-8_57.

[15] M. Zareapoor, "Information Engineering and Electronic Business," Information Engineering and Electronic Business, vol. 2, pp. 60–65, 2015, doi: https://doi.org/10.5815/ijieeb.2015.02.08.

[16] D. Blei, B. Edu, A. Ng, M. Jordan, and J. Edu, "Latent Dirichlet Allocation," Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003, Available: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?ref=https://githubhelp.com

[17] D. Blei, A. Ng, and M. Jordan, 'Latent Dirichlet Allocation', in Advances in Neural Information Processing Systems, 2001, vol. 14.

[18] Jelodar, H., Wang, Y., Yuan, C. et al. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. Multimed Tools Appl 78, 15169–15211 (2019). https://doi.org/10.1007/s11042-018-6894-4

[19] JY. Cao, S. Liu, P. Zhao, and H. Zhu, "Rp-net: A pointnet++ 3d face recognition algorithm integrating rops local descriptor," IEEE Access, 2022.

[20] [1]"Non-negative Matrix Factorization, A New Tool for Feature Extraction: Theory and Applications." Accessed: May 23, 2023. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf & doi=f488014381ac79b2c4dd8921abb734b117218c7a

[21] Lee, D., Seung, H. Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999). https://doi.org/10.1038/44565

[22] L. Wang and Y. Wan, “Sentiment Classification of Documents Based on Latent Semantic Analysis,” pp. 356–361, Jun. 2011, doi: https://doi.org/10.1007/978-3-642-21802-6_57.

[23] S. Liu et al., ”Visual Exploration of Semantic Relationships in Neural Word Embeddings,” in IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 553-562, Jan. 2018, doi: 10.1109/TVCG.2017.2745141.

[24] R. Bamler and S. Mandt, ‘Dynamic Word Embeddings’, in Proceedings of the 34th International Conference on Machine Learning, 06–11 Aug 2017, vol. 70, pp. 380–389.

[25] L. Com and G. Hinton, “Visualizing Data using t-SNE Laurens van der Maaten,” Journal of Machine Learning Research, vol. 9, pp. 2579–2605, 2008, Available: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf?fbcl

[26] L. Ma and Y. Zhang, ”Using Word2Vec to process big text data,” 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 2015, pp. 2895-2897, doi: 10.1109/BigData.2015.7364114.

[27] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. ArXiv. /abs/1301.3781

[28] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). FastText.zip: Compressing text classification models. ArXiv. /abs/1612.03651

[29] I. Santos, N. Nedjah and L. de Macedo Mourelle, ”Sentiment analysis using convolutional neural network with fastText embeddings,” 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Arequipa, Peru, 2017, pp. 1-5, doi: 10.1109/LA-CCI.2017.8285683.

[30] J. Pennington, R. Socher, and C. Manning, “GloVe: Global Vectors for Word Representation,” Association for Computational Linguistics, 2014. Available: https://aclanthology.org/D14-1162.pdf

[31] M. F. Porter, “Readings in information retrieval,” K. Sparck Jones and P. Willett, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, ch. An Algorithm for Suffix Stripping, pp. 313–316.

[32] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, ‘Learning Word Vectors for Sentiment Analysis’, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 142–150.

[33] Y. Zhao, ‘Chapter 10 - Text Mining’, in R and Data Mining, Y. Zhao, Ed. Academic Press, 2013, pp. 105–122.

[34] Baglama, James, and Lothar Reichel. ”Augmented implicitly restarted Lanczos bidiagonalization methods.” SIAM Journal on Scientific Computing 27.1 (2005): 19-42.

[35] “Package ‘irlba’ Type Package Title Fast Truncated Singular Value Decomposition and Principal Components Analysis for Large Dense and Sparse Matrices,” 2022. Accessed: May 23, 2023. [Online]. Available: https://cran.r-project.org/web/packages/irlba/irlba.pdf

[36] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[37] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in Machine Learning: ECML-98, ser.Lecture Notes in Computer Science, C. Nédellec and C. Rouveirol, Eds.Springer Berlin Heidelberg, 1998, vol. 1398, pp. 137–142.

[38] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

[39] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv. /abs/1810.04805

[40] Libovický, J., Rosa, R., & Fraser, A. (2019). How Language-Neutral is Multilingual BERT? ArXiv. /abs/1911.03310

[41] Tiago A. Almeida, José María G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of SMS spam filtering: new collection and results. In Proceedings of the 11th ACM symposium on Document engineering (DocEng ’11). Association for Computing Machinery, New York, NY, USA, 259–262. https://doi.org/10.1145/2034691.2034742