Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 109–113 DOI: 10.15439/2022R11 ISSN 2300-5963 ACSIS, Vol. 33

Utilize Deep learning to increase the performance of a Book recommender system using the Item-based Collaborative Filtering

Cu Nguyen Giap
Economic Information System and E-commerce
Thuongmai University
Hanoi, Vietnam
cunguyengiap@tmu.edu.vn

Le Thi Huyen Dieu
Faculty of Information Technology
FPT University -FPT Polytechnic
Hanoi, Vietnam
dieulth2@fpt.edu.com

Luong Thi Hong Lan
Faculty of Computer Science and Engineering
Thuyloi University
Hanoi, Vietnam
lanlhbk@tlu.edu.vn

Tran Thi Ngan, Tran Manh Tuan
Faculty of Computer Science and Engineering
Thuyloi University
Hanoi, Vietnam
{ngantt, tmtuan}@tlu.edu.vn

***Abstract*—Item-based Collaborative Filtering is a common and efficient approach for recommendation problems. In this study, we have investigated the power of deep learning in textual feature extraction and applied this advantage to a high-performance item-based collaborative filtering recommender system. The proposed approach has been experienced on book datasets added by texts collected from famous book review sites. The experiment proves that the proposed model has better performance thanks to the contribution of the new item profile process method based on Deep Learning.**

Index Terms—*Item-based Collaborative Filtering, Deep Learning, Textual embedded recommender system.*

I. Introduction

With the rapidly growing day-to-day data over the internet, item-based collaborative filtering (IBCF) is one of the best-known and most extensive recommendation system (RS) techniques. They provide accurate predictions when sufficient data is provided, as this technique is based on a computation of similarity among items and rating prediction using similar items.

One of the recommended application areas is the book market. Users are increasingly inclined to buy books on websites, and the role of book recommendation systems has become more important. In fact, books are a special product that differs from most of the products on e-commerce. Particularly, the book's content is often very long, and the user cannot grasp it in advance, but other information, such as title and type, etc., is not good enough to reflect the book. One of the successful RS algorithms is Item-based collaborative filtering which was introduced by Amazon and was widely applied to e-commerce [1-5]. This study wants to build an RS with book data in a more appropriate way by an increment of the IBCF algorithm.

In IBCF, several correlation-based similarity measures have been traditionally used to generate a top-k list of recommended items: Chigozirim Ajaegbu [6] combined three traditional similarity metrics, Cosine, Pearson, and Adjusted cosine, in an IBCF algorithm for Movielens RS. Pradeep Kumar Singh et al. [7] assume the similarity between users is essential to finding the similar neighbors of a target item. Monika Verma and Arpana Rawal [8] proposed an RS to predict popular books based on calculating the item-item likeness with the cosine/correlation coefficient in the attributes of the book database.

Although existing RS successfully produces decent suggestions, they still undergo some limitations, such as accuracy and scalability. In the last few years, a deep learning (DL)-based approach has been applied widely to enhance the quality of recommendations. Almaghrabi et al. [9] suggested a novel system based on DL-based augmentation for forecasting user ratings for various online databases: movies, music, and book collections. In [10-11], the authors supply a thorough review of DL-based recommendation approaches to clarify and suggest beginner researchers interested in the subject.

The RS uses an IBCF approach, for each unknown rated item i_t, it has to estimate the similarity of all items to identify the k-closest items of i_t, and then the rating of i_t is estimated from the rating of k-closest items. The common IBCF RS calculates the likeness of items based on an item profile that is normally a vector of binary, number, nominal or categorical elements.

However, in some domains, item descriptions are important to assess the similarity of products. These descriptions are textual variables, such as the book RS; assessing the book description as indispensable information is crucial. This problem carries out an important issue: extracting item textual profile features.

DL technology has been utilized in natural language processing problems for a long time, with many remarkable achievements. During the development of DL, this technology has shown a high potential application for textual feature extraction. This study was conducted to utilize the stacked denoise autoencoder and text summarize technique for textual feature extraction. And then, the extracted feature is used to improve the performance of a Book RS that uses IBCF.

To assist the proposed model, the data of the Book-crossing dataset is used to estimate the implementation of the proposed model and resemble it to the other algorithms' results. However, it must be noticed that the original Book-crossing dataset does not contain any description of the book; it has basic book information such as name, author, etc. Therefore, this dataset is added by book descriptions from famous book review websites.

©PTI 2022

This article is organized into four details. The first part briefly presents the introduction and motivation of the problem and our main contributions. The second part describes the typical approach of item-based collaborative filtering and the proposed increment of this algorithm. The next part introduces the way to collect relevant data and our experiments. Finally, we are concerned about the significant results of the proposed approach and future works.

II. RECOMMENDER SYSTEM AND PROPOSAL SOLUTION BASED ON DEEP LEARNING TECHNIQUES

A. Item-based Collaborative Filtering

Sarwar introduced IBCF in [12]; the IBCF model is depicted in Figure 1.

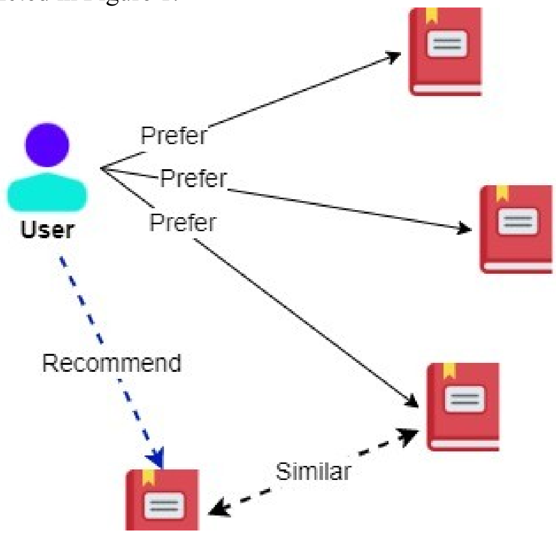

Fig. 1. Item-based collaborative filtering

In the IBCF, for each unknown rated item i_t, it has to estimate the similarity of this item with other rated items by the same user, and then a set of k-closest items of i_t is identified from the rated items. The IBCF predicts the rating of i_t by aggregating the rating of the k-closest items. The common Item-based collaborative filtering RS calculates the similarity of items based on an item profile that is normally a vector of binary, number, nominal or categorical elements. The common similarity estimators use Pearson similarity, Cosine similarity, and their adjustments.

Pearson similarity:

$$Pearson(i,j)=\frac{\sum_{u\in T_i\cap T_j}\left(r_{u,i}-\overline{r_i}\right)\left(r_{u,j}-\overline{r_j}\right)}{\sqrt{\left(\sum_{u\in T_i\cap T_j}\left(r_{u,i}-\overline{r_i}\right)^2\right)\left(\sum_{u\in T_i\cap T_j}\left(r_{u,j}-\overline{r_j}\right)^2\right)}} \quad (1)$$

Cosine similarity:

$$\mathrm{Cosine}(i,j)=\frac{\sum_{u\in T_i\cap T_j} r_{u,i}\times r_{u,j}}{\sqrt{\left(\sum_{u\in T_i\cap T_j} r_{u,i}^2\right)\left(\sum_{u\in T_i\cap T_j} r_{u,j}^2\right)}} \quad (2)$$

The similarity of the items can be estimated by the rating of other users $\left(T_i \cap T_j\right)$ given on these items only, however, the item profile can supply more information to estimate better. This study concerns this problem and uses Deep learning technology to identify better similar items for a target item.

B. Proposal model

The purpose of the proposal is to utilize Deep learning techniques to extract features of the textual variables in item attributes. These features will be used in the item similarity estimator of the IBCF algorithm. Remarkably, the original data of books, including book titles are used to search and seek book reviews from public websites that contain interesting information about the book. It is added to the similarity estimator naturally.

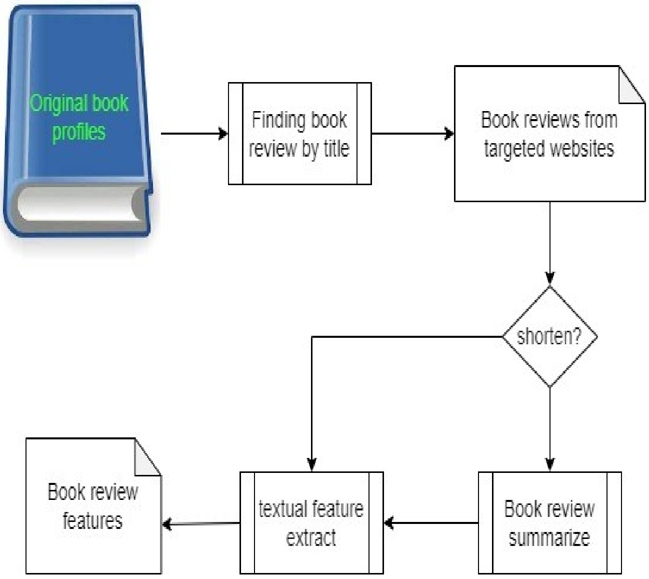

Fig. 2. The process of book review feature extraction

In fact, a book review contains a lot of exciting information that readers are concerned about and need when choosing a suitable book. Many websites publish book reviews, but in this study, we are concerned about the reviews presented in two critical aspects: literary reviews and commercial reviews. Therefore, several book-sale websites and some literary analysis websites will be targeted.

The proposed Textual embedding Item-based collaborative Filtering RS (TE_IbCF RS) consists of two phases. The detail of the proposed model is described as follows:

Phase 1: predict unknown item rating

a. Estimate item similarity to the rated item by users rating, extracted textual features, and selected original item attributes using Cosine similarity.

b. Select the k-nearest item set.

c. Predict unknown item rating by the following formula [13]:

$$r_{u,i}=\overline{r_i}+\frac{\sum_{j\in N_i}(r_{u,j}-\overline{r_j})*sim(i,j)}{\sum_{j\in N_i} sim(i,j)} \quad (3)$$

Where $\overline{r_i}$ is the average rating of item i.

Phase 2: Recommendation by the top-N algorithm.

Notice that: in Phase 1, given these similarity measures are not enough to evaluate the appropriateness of the recommendations. The data is the most influential factor in finding the best suggestions; you should provide actual input data to get the accurate output. For book items in the knowledge base, we use the textual summary to describe the textual details, which usually gives the book's main topics.

C. Deep learning for textual embedding RS

To predict the interest of buyers and recommend books to them accordingly, two kinds of DL are applied for textual feature extraction.

- Text summarization technique
- Stacked denoising autoencoders

The text summarization technique is applied to shorten the review of the book if the user selects this option. The Stacked denoising autoencoders are used to extract textual features from book reviews. It has the function of dimensional reduction also.

Text summarization techniques:

The text summarization technique is used for massive applications that deal with a vast amount of strings and long text [14]. In the designed Book recommender system, the book description contains noise and irrelevant information due to the bias of the writer. Therefore a text summarization technique based on the Term frequency-inverse Document frequency (TF-IDF) has been applied to create a shortened version of the book description. This algorithm includes the following steps:

Step 1: word segmentation.

Step 2: remove stop-words

Step 3: word score estimation

Step 4: Sentence value calculation and filtering (using a user-defined threshold).

For example, a book description is "*Readers beware. The brilliant, breathtaking conclusion to J.K. Rowling's spellbinding series is not for the faint of heart--such revelations, battles, and betrayals await in Harry Potter and the Deathly Hallows that no fan will make it to the end unscathed. Luckily, Rowling has prepped loyal readers for the end of her series by doling out increasingly dark and dangerous tales of magic and mystery, shot through with lessons about honor and contempt, love and loss, and right and wrong. Fear not, you will find no spoilers in our review--to tell the plot would ruin the journey and Harry Potter and the Deathly Hallows is an odyssey the likes of which Rowling's fans have not yet seen and are not likely to forget. But we would be remiss if we did not offer one small suggestion before you embark on your final adventure with Harry--bring plenty of tissues*."

The above description is summarized as "*The brilliant, breathtaking conclusion to J.K. Rowling's spellbinding series is not for the faint of heart--such revelations, battles, and betrayals await in Harry Potter and the Deathly Hallows that no fan will make it to the end unscathed. Fear not, you will find no spoilers in our review--to tell the plot would ruin the journey and Harry Potter and the Deathly Hallows is an odyssey the likes of which Rowling's fans have not yet seen and are not likely to forget.*"

The advantage of the text summarization technique is that it maintains the major meaning of the original review but is present in a much shorter form. Therefore, the vector representation form of a book review will be shorter also, and it causes cheaper time costs for other processes followed.

Stacked Denoising Autoencoders:

A stacked denoising autoencoder (SDA) is a deep learning technology that can reduce the impact of noise in the input and extract the potential feature without the requirement of predefined labels [15-17]. In this study, a specific architecture of SDA was used to extract the useful part of the textual descriptions of books in a dataset. The SDA architecture is depicted in figure 3.

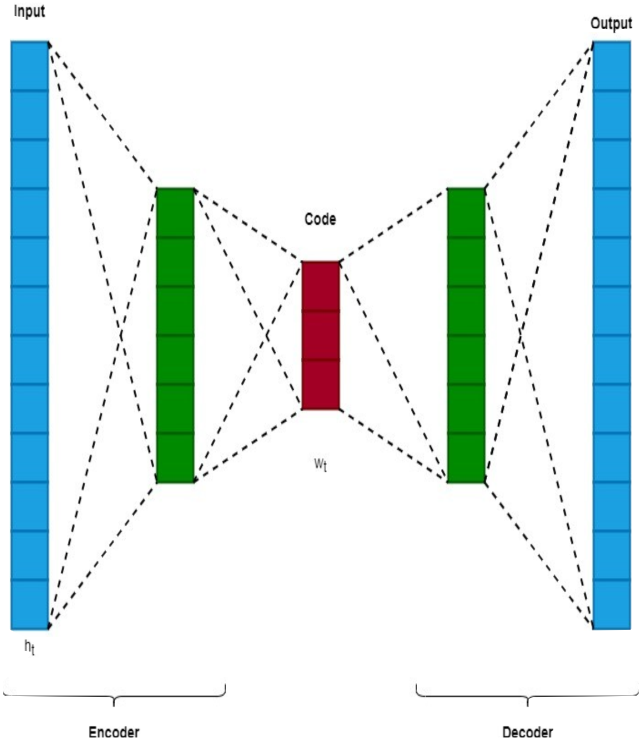

Fig. 3. Stacked denoising autoencoders

For textual feature extraction, the input layer is passed by the vector representation of text. This study uses the N-gram transformation of the original text. The code layer has a size suited to the expected reduction purpose. In this study, the standard size of the encoded feature is set at 10% of the input string. The number of neurons in the input and output layers is equal and depends on the book description's length.

For implementation, the book review is transformed by a 2-gram (or bi-gram) model that is suitable for the short and middle texts [18-19]. In short, texts are split into two-word sequences, and the vector representations of texts are conducted from the frequency of split strings. The detail of the bi-gram model was presented in the study of Shinsuke et al. [20].

Besides, when creating a vector representation of a text by bi-gram, the study has to apply a word segmentation algorithm and a stop-word removal algorithm. Because all book reviews are presented in English, this study uses the standard stop-word library integrated into the Sklearn library.

III. EXPERIMENTS

In this section, we estimate our proposed model for book RS scenarios. The practical outcomes indicate proof of meaningful advancement over competitive baselines.

A. Data construction

In order to experience the proposed model of RS, a well-known dataset of books, Book-crossing, is used. However, the public Book-crossing dataset does not contain any description of the book; therefore, it must be added by book reviews that are collected from two famous sites: amazon.com and goodread.com. The former has commercial reviews of books, and the latter supplies literary studies.

Within the time constraint, the testing dataset is a subset of Book-crossing data, in which 1000 books were selected randomly, and the dataset contains a relative of 2x1000 book reviews. Their reviews present the more interesting information about the book than the original data.

TABLE I. ATTRIBUTES OF BOOK IN THE DATA SET

Attribute	Type
book_id	Original attribute
best_book_id	Original attribute
work_id	Original attribute
books_count	Original attribute
authors	Original attribute
original_publication_year	Original attribute
original_title	Original attribute
title	Original attribute
language_code	Original attribute
average_rating	Original attribute
ratings_count	Original attribute
work_ratings_count	Original attribute
work_text_reviews_count	Original attribute
Amazon	Collected from amazon.com
rating_Amazon	Collected from amazon.com
Goodread	Collected from goodread.com
rating_Goodread	Collected from goodread.com

In fact, the book profile's quality is thickened and more useful for readers with the above extra reviews.

Besides, it should be noted that the books' rates are in the interval [0-10]. The data is really spare, and ratings of an item are highly dispersed. This is a challenge for any RS algorithm.

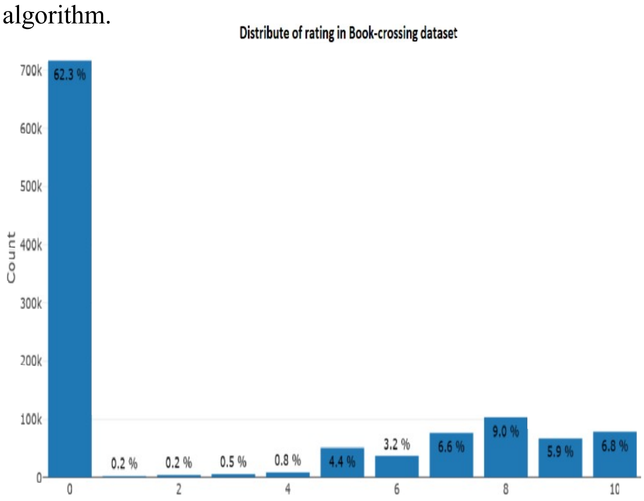

Fig. 4. Distribution of rating

B. RS performance metrics

In order to experiment, the Root Mean Square Error (RMSE) was utilized to estimate the usefulness of the proposed RS. The following formula calculates the measurement metric:

Error metric:

$$RMSE(I) = \sqrt{\frac{\sum_{i \subset I} (r_i - \hat{r})^2}{|I|}} \tag{4}$$

Where I is a set of items recommended to user u , and r_i, $\hat{r}_i$ are predicted user ratings and real ratings relatively.

RMSE is a typical measurement metric that is used to compare implemented algorithms in recommender systems. However, it should be noticed that the error metric is not all of the RS's performance. In the case of the textual embedded RS for books, it can rationally explain the recommendation based on book reviews, and this feature might be helpful in real applications indeed.

C. Experimental results

Algorithm performances

The Python programming language implemented the proposed model and used some valuable libraries. The stacked denoising autoencoder was implemented with the Theano platform and used the BaseEstimator from the "sklearn" library. Theano is a powerful open-source platform that can bring high performance to the implementation. The other task that conducted text transformation also used the "sklearn" library, mainly it used the CountVectorizer from the feature-extraction package and the standard stop-word library. The 2-gram model was chosen in the current stage of this study.

We have experienced the efficiency of the proposed models by running them with standard parameters picked from the literature review. These parameters might not be optimum for book review data; however, they were recommended for text processing in other studies. Finding the optimal set of parameters is planned for the next stage of this research.

Afterward, the RMSE results are shown in Table 2 respectively.

TABLE II. THE RMSE OF EXPERIENCED ALGORITHMS

Algorithm	RMSE
IF_CF	3.053
TE_IbCF without Text summarization	***2.717***
TE_IbCF with Text summarization	2.809

The proposed models using textual embedded features within the testing dataset can improve Book Recommender System performance. The **TE_IbCF** without Text summarization has the highest version, but the text summarization technique does not reduce the RS performance much. It means that the text summarization algorithm would be applied to increase the time performance; meanwhile, the recommender system can preserve its accuracy.

The book-crossing dataset has ratings of books that are remarkably dispersed. User reviews of books are very inconsistent; most books have the smallest rating of 0 and the highest rating of 10 (the minimum and maximum values in the measurement scale). Moreover, the bias of the 0 ratings is obvious. This point brings a complex challenge to building a proper automatic recommender system. Therefore, the improvement brought by the proposed textual embedded feature extraction model is valuable and should be applied in reality.

Finally, it should be noticed that the result in the sub-dataset of Book-crossing that includes 1000 books shows that the textual embedded feature conducted from book reviews can improve the performance of Item-based collaborative filtering. However, this testing data is not the full Book-crossing dataset that contains 10000 books; therefore, the above result can not be compared to other public studies on RS for books. The data set will be enriched in the future by adding reviews of the remaining books.

IV. CONCLUSIONS

In summary, this study has considered several improvements to the IBCF algorithm for RS in the book. The main contribution of this study is proposing a specific approach to the recommender system using embedded text. Adding the book reviews into the book similarity estimator process of an RS and using the deep learning techniques to extract meaningful textual features increases the performance of the IBCF recommender system. The stacked denoising autoencoder was implemented for textual feature extraction tasks and showed high potential. Although this research stage uses only the parameters suggested in the literature review, the results are also very encouraging.

The experiment was done on the well-known book dataset, the Book-crossing subset. The results gave excellent proof of the high efficiency of deep learning techniques, including text summarization technique, bigram algorithm, and stacked denoising autoencoders. These techniques extract and present meaningful features of book reviews, and therefore the book similarity can be estimated better. The new model also has a new advantage thanks to the rational recommendation given to the reader because it concerns the book reviews of both literary analysis websites and commercial websites.

Another minor contribution of this study is the construction of an extended dataset of the Book-crossing dataset with book reviews from literary and commercial websites. Currently, no systems allow a comprehensive and automatic collection of book reviews; thus, the book reviews are collected and matched manually. amazon.com and goodread.com are among the top five famous book review websites on the globe. It guarantees the quality of book reviews and book datasets also.

However, the narrow size of the testing dataset is a limitation of this study. The study will be expanded in the future by using a new vector representation of text generated by the Bag of Word algorithm. This approach was mentioned as a high-potential technique in text processing problems. The issue of figuring out the best parameters remains as future work also.

ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2020.11.

REFERENCES

[1] Comparison of user-based and item-based collaborative filtering. [Online; accessed 17-August-2019]. https://medium.com/@wwwbbb8510/comparison-of-user-based-and-item-based/-collaborative-filtering-f58a1c8a3f1d

[2] Sarwar B, Karypis G, Konstan J, Riedl J (2001) Itembased collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, ACM, pp 285–295

[3] Yang Z, Wu B, Zheng K, Wang X, Lei L (2016) A survey of collaborative filtering-based recommender systems for mobile internet applications. IEEE Access 4:3273–3287

[4] Linden G, Jacobi J, Benson E (2001) Collaborative recommendations using item-to-item similarity mappings. [Google Patents]

[5] Deshpande M, Karypis G (2004) Item-based top-n recommendation algorithms. ACM Trans Inf Syst 22(1):143–177

[6] Ajaegbu, C. (2021). An optimized item-based collaborative filtering algorithm. Journal of ambient intelligence and humanized computing, 12(12), 10629-10636.

[7] Singh, P. K., Sinha, S., & Choudhury, P. (2022). An improved item-based collaborative filtering using a modified Bhattacharyya coefficient and user–user similarity as weight. Knowledge and Information Systems, 64(3), 665-701.

[8] Verma, M., & Rawal, A. (2022). An Enhanced Item-Based Collaborative Filtering Approach for Book Recommender System Design. ECS Transactions, 107(1), 15439.

[9] Almaghrabi, M., & Chetty, G. (2018, December). A deep learning based collaborative neural network framework for recommender system. In 2018 International Conference on Machine Learning and Data Engineering (iCMLDE) (pp. 121-127). IEEE.

[10] Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. Artificial Intelligence Review, 52(1), 1-37.

[11] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR), 52(1), 1-38.

[12] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295).

[13] Wu J, Chen L, Feng Y, Zheng Z, Zhou M, Wu Z (2013) Predicting quality of service for selection by neighborhoodbased collaborative filtering. IEEE Trans Systems, Man, and Cybernetics: Systems 43(2):428–439

[14] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). Text summarization techniques: a brief survey. arXiv preprint arXiv:1707.02268.

[15] Sagha, H., Cummins, N., & Schuller, B. (2017). Stacked denoising autoencoders for sentiment analysis: a review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7(5), e1212.

[16] Liang, J., & Kelly, K. (2021). Training stacked denoising autoencoders for representation learning. *arXiv preprint arXiv:2102.08012*.

[17] Tong, H., Liu, B., & Wang, S. (2018). Software defect prediction using stacked denoising autoencoders and two-stage ensemble learning. *Information and Software Technology*, *96*, 94-111.

[18] Patil, A., & Mahalle, P. (2021). A Building Topical 2-Gram Model: Discovering and Visualizing the Topics Using Frequent Pattern Mining. In Proceeding of First Doctoral Symposium on Natural Computing Research (pp. 11-21). Springer, Singapore.

[19] Elghannam, F. (2021). Text representation and classification based on bi-gram alphabet. *Journal of King Saud University-Computer and Information Sciences*, *33*(2), 235-242.

[20] Mori, S., Nishimura, M., & Itoh, N. (1998). Word clustering for a word bi-gram model. In *ICSLP*.