International Journal of Management, IT & Engineering
Vol. 13 Issue 04, April 2023,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

An Innovative Hashing Scheme and BiLSTM-based Dynamic Resume Ranking System

1st Sandeepkumar Racherla
Amazon
Data Scientist
Seattle, Washington, United States
sandeep.racherla1@gmail.com
ORCID: 0000-0002-8155-5643

Prathyusha Sripathi
Amazon
Data Scientist
Seattle, Washington, United States
prathyushasripathi111@gmail.com
ORCID: 0000-0001-9260-3906

***Abstract*—Content-based-filtering dependent job recommender systems are failing to cope with the dynamic nature of the job market by processing the massive amount of data the Human Resources (HR) professionals handle for talent hunting. Deep Learning (DL)-based job recommender systems capable of dynamically establishing a relation between candidate profile and job description is a promising solution to this challenge. This paper proposes an innovative hashing scheme-based word embedding dependent recommendation system using a Bidirectional Long-Short Term Memory (BiLSTM) network to intelligently sort the applicants' resumes according to the job description and requirements. The proposed system ranks the resumes against job description based on skills required with 89.93% accuracy, 88.11% precision, 89.17% recall, 6.79% R^2, 6.05% RMSE, and 7.31% MAE. The proposed recommendation system effectively handles dynamic job market requirements and the rapid rate of new candidate entrance. This noble three-module system architecture beats the challenges of automating talent acquisition responsibilities of HR managers in today's vibrant and rapidly changing job market.**

***Keywords*—Resume Ranking, CV Shortlisting, Deep Learning, BiLSTM, Word Embedding**

I. INTRODUCTION

Talent acquisition is one of the responsibilities of Human Resource (HR) officials [1]. The resume ranking is at the heart of this process [2]. It starts with preparing the job description and circular publication. Then the interested candidates apply with their resumes. The HR hiring managers need to review these resumes to rank the eligible and potential candidates [3]. It sounds simple in this context. However, ever-changing job descriptions, rapid rate of change of new job seekers, a large number of resumes for a single circular, and the improper ratio of relevant and irrelevant resumes make the talent acquisition process very competitive for HR professionals [4]. The content-based-filtering dependent job recommender systems used to be an effective solution to assist in this process [5]. However, this system is failing to cope up with the vibrant and rapidly changing nature of the job market [6]. A noble recommendation system has been developed and presented in this paper to meet the market demand and HR professionals' criteria for resume ranking.

The performance of Deep Learning (DL)-based solutions depends on the dataset's quality [7]. Recommender systems, also known as recommendation systems, are a sub-branch of information filtering syst. Usually, they are used to recommend items to the users [8]. A DNN-based recommendation system trained with a proper dataset identifies the motives of the users based on their interaction with the system and recommends items the user may be interested in [9]. This system consists of two parts - feature extraction and prediction. The feature extraction part extracts the feature from the user interaction, and the prediction part predicts the probability of the user being interested in some available items [10]. The system learns from the interaction, correct and incorrect prediction, and eventually becomes better at recommending the appropriate items [11].

A. Research Gap Identification

The longer a recommendation system performs on a similar task, the better it becomes in that task. Moreover, the more enriched the dataset gets, the better performance it can ensure
[12]. It works well for e-commerce sites, Social media, and e-Learning platforms [13]. However, the ever-evolving and changing environment and shorter interaction period are significant challenges to the overall efficiency of such systems [14]. That is why despite the massive success of DNN-based recommendation systems [15], it needs to improve in assisting HR managers in automating the candidate shortlisting process. A recommendation system still works well if the job description remains the same every time. However, the job market is more vibrant than ever, and job descriptions change rapidly [16].

Furthermore, a job circular stays online for a short period. At the same time, the number of new job seekers changes rapidly [17]. That is why personalized recommendation systems must work better in this vibrant environment [18]. On the other hand, the customized PageRank algorithms work well for old users with enough interaction with the system. There are well-developed and personalized PageRank algorithms [19]. However, there is a research gap that focuses on a system that works well for rapidly changing new and existing users. Data sparsity is another challenge in this domain. When cosine similarities or relational methods are used for filtering and ranking, the different data instances exhibit a heterogeneous nature [20]. As a result, the data becomes sparse distorting identifiable or measurable patterns. Incorporating advanced behavioral similarities with the existing DNN-based recommendation systems is another research gap.

B. Technology Selection

The deep learning-based recommendation systems are adequate to abridge the research gaps identified in the previous section [21]. There are numerous DNN models available. The Feed-Forward Neural Network (FFNN) architecture [22] and text as a bag of words [23] are potential solutions to text classification. The Convolutional Neural Networks (CNN), specially designed for image classification, are also helpful in text classification. However, it is specific to computer vision, and NLP-based solutions [24]. Most of the job descriptions come in pure text form. As a result, the CNN for this task can be ignored [25]. The Recurrent Neural Network (RNN) architecture uses the texts as a sequence of words. It maintains the sequence [26]. That is why the semantics and parts of speech relations are preserved [27]. As a result, the text classification exhibits human intelligence when RNN is used.

Job circular and application process involves both real-time and offline processing [28]. The RNN requires substantial computing resources if the input sequence becomes lengthier. Moreover, it becomes complicated when long-term dependencies are considered. It is a significant drawback of RNNs in real-time applications [29]. The solution to the limitations of RNN is Long Short-Term Memory (LSTM) neural networks. The LSTM networks promise to process long and short sequences in real-time [30]. However, there is still a gap in identifying the semantic structure of the job description to relate it to the candidate profile. The solution to abridge this gap is one of the novel contributions of this paper.

C. Word Embedding Analysis

The DL-based text classifiers require word vectors [31]. The process of creating a wordvector is called word embedding. It is a process of vectoring the word tokens to mathematically assign distance among different words so that different similarity measures can be applied [32]. There are different word embedding methods. These methods have been discussed and analyzed below [33]:

1) ***Word2Vec:*** The Word2Vec algorithm is a neural network dependent algorithm. It uses an existing corpus of text to discover the associations among words. The network learns to distinguish the words by assigning unique numbers representing vectors. The vectors consist of amplitude and direction that preserve the semantic meaning and syntactic structure. The cosine similarities among the word vectors are used to retrieve the semantic similarities and classify the words. The similarity is measured using equation 1 [34].

$$S(i,j) = \frac{Vec_i \times Vec_j}{|Vec_i||Vec_j|} \tag{1}$$

2) ***Bag of Words (BoW):*** As the name suggests, in this embedding technique, a text is represented as a bag of words. The grammatical rules, word sequence, and multiplicity are ignored in this method. The equation 2 defines the structure [23].

$$BoW = \sum_{i=1}^{N} Word_i : i \tag{2}$$

It is famous for document classification based on the frequencies of certain words. However, the BoW is not an appropriate solution for resume ranking because of not consider the word sequence.

3) ***Continuous Bag of Words (CBoW):*** The CBoW method is not based on the word frequency only. It is a contextual approach. The distribution representation of the context of a specific corpus is vectorized in this model [35].

4) ***Autoencoder:*** Autoencoder is a DNN used to encode unlabelled data. Once encoded, the data can be regenerated from the code. It is more famous for dimensionality reduction. However, also used for word embedding. The autoencoder consists of two components - the decoded message x and the encoded message x'. These are considered in the Euclidean space where $x = \mathbb{R}^m$ and $x = \mathbb{R}^n$ where m and n are the Euclidean coordinates. Both encoding and decoding are done using the multilayer perceptron layer defined in equation 3 [36].

$$E_\phi(x) = \sigma(Wx + b) \tag{3}$$

In this paper, the σ is the sigmoid activation function, W is the weight matrix, and b is the bias.

5) *TF-IDF:* The TF-IDF is the abbreviation for Term Frequency-Inverse Document Frequency. In the BoW method, frequency plays a significant role in embedding. The TF-IDF is an extension of the BoW method. In this embedding, the words are organized according to their significance. The significance is controlled by the weighted sum assigned to them during embedding. The equation 4 defines the procedure [37].

$$TF - IDF = TF(w,d) * IDF(w,D) \tag{4}$$

Here, w means the frequency of the word, d is the document where it showed up, and D is the entire dataset. Once the term frequency is calculated, the inverse of it is convolved with it. The inverse is calculated using the IDF (w, D) function. The word embedding analysis shows that the existing effective methods construct the word vector directly using the tokens. It works well as a general solution and is not a practical solution for resume ranking. In this paper, instead of wordvector, hash-vectors have been used, which is another noble contribution of this experiment. The following list highlights the contribution of this research:

1. Semantic structure-based resume recommendation system development to handle the dynamic and rapidly changing environment

2. Hash-vector dependent noble word embedding technique for Deep Learning-based classifiers.
3. BiLSTM-based processing module design and implementation
4. Achieving a remarkable performance of 89.93% accuracy, 88.11% precision, 89.17% recall, 6.79% R^2, 6.05% RMSE, and 7.31% MAE.

The rest of the paper has been organized into four different sections. The second section discusses the methodology. We presented the experimental results and evaluation in the third section. The paper has been concluded in the fourth section.

II. Methodology

The introduction highlights the research gap. This research has been conducted to abridge the gap and develop an innovative and effective solution for resume ranking. The methodology starts with the hashing of the datasets. Then the hash vectors are sent to the BiLSTM network to train it. The probability predicted by the network is used to rank the resumes. The overview of the proposed methodology has been illustrated in figure 1.

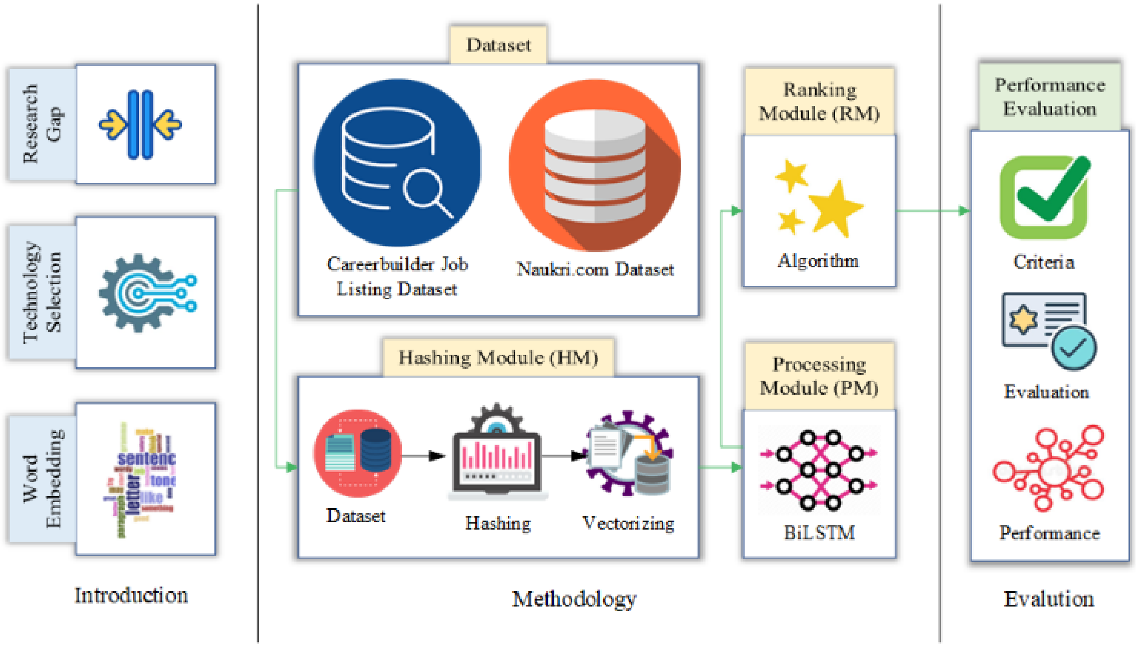

Figure 1: The overview of the proposed methodology

A. System Structure

The proposed methodology processes the document and the query by a specially designed neural network that handles them separately and later merges the signals together to make the final decision. The proposed system is defined as S(q; Φ) and the working principle is governed by equation 5.

$$S: q \in D \rightarrow y \in RD \tag{5}$$

According to 5, the system queries(q) on the entire document(D) and the output(y) establishes a relation(R) with document(D). The system tries to pair up the candidates' skills with the relevant skills of the job description and expresses it in probability calculated by equation 6.

$$P(d|Q) = \frac{\exp(sig \times R(Q,d))}{\sum d \in D \exp(sig \times R(Q,d'))} \tag{6}$$

Here is equation 6, the sig represents the sigmoid function, which is the activation function of the layers used in the proposed network. The skill of the candidate is d, the query performed on the job description (D) is Q, and the P (d|Q) is the probability to rank the candidate based on the skills. The proposed system uses the log-loss function defined by equation 7 to measure the loss during the learning process.

$$L(i,j) = -log \times \alpha \times P(d+Q) \tag{7}$$

Here in equation 7, the L(i, j) is the layer L of the multilayer perceptron model located at index (i, j) and the α is the learning rate.

Table 1: THE TERM FREQUENCY IDENTIFICATION BY THE HM

TF	Software	Engineering	with	Excellent	Salary	and	Benefits
d1	1	1	0	0	0	0	0
d2	1	1	1	2	1	1	1

B. Module View of the System

The proposed system has been designed as a combination of three modules. They are the Hashing Module (HM), the Processing Module (PM), and the Ranking Module (RM).

1) The Hashing Module (HM): The word embedding analysis presented in the introduction of this paper shows that converting the text into vectors using the existing state-of-art

2) methods is not a feasible solution to the resume ranking challenge because of the dynamic nature of the job description and short-term interaction of the new candidates entering the job market at a rapid rate [38]. In this paper, an innovative hashing method has been employed in the Hashing Module to process the input text sequence for the PM. The HM handles large datasets smoothly because of Word N-gram scaling.

Case Study of HM Application: In the case study, the d1 and d2 are two English sentences. These sentences are experimental. These are not any real job descriptions. They have been carefully chosen to explain the HM module's operations properly.

- d1: Software Engineering Salary
- d2: Software Engineering with excellent salary and excellent benefits.

The HM identifies the term frequency first. It has been listed in table I. Now the hashes are generated using Word N-gram as follows:

- d1: software #engineering
- d2: software #engineering #with #excellent #salary engineering #with #excellent #salary #and with #excellent #salary #and #excellent excellent #salary #and #excellent #benefits

The value of N in the N-gram dynamically changes within the range of 1 to 5. If the lowest value is not applicable, then the immediate next value is assigned to N using the algorithm 1.

Algorithm 1 Word N-gram Value Assignment

Require: $r \leftarrow [1, 2, 3, 4, 5]$
Ensure: $1 \leq N \leq 5$
  $k = 0$
  $N = 1$
  for $i = 0; i < 5; i{+}{+}$ **do**
    $k \leftarrow r[i]$
    if $k > N$ **then** $N \leftarrow k$
    end if
  end for
  Return N

The word hashes are dynamically generated using the N selected by the algorithm 1.

3) *The Processing Module (PM):* The Processing Module (PM) uses a Bidirectional Long-Short Term Memory (BiLSTM) network [39]. The analysis presented in the Technology Selection section of the introduction justifies that the LSTM network is well-suited for the resume ranking we are developing in this paper. However, it has been observed that the BiLSTM networks perform better than LSTM for a large volume of data [40]. That is why the BiLSTM network illustrated in figure 2 has been used in research.

Table 2: THE NETWORK ARCHITECTURE DETAILS

Layer Type	Output Shape	Parameters
Embedding	(None, 50, 16)	12000
Bidirectional	(None, 256)	194460
Dropout	(None, 256)	0
Dense	(None, 1)	564

The network architecture details have been listed in table II. The embedding layer, which receives the signal from the HM module, processes the input signal with 12,000 learning parameters. The bidirectional layer contains 194,460 parameters. And the dense layer consists of 564 parameters. Together there are 207,024 learning parameters in the network. The proposed network uses a Log-Loss function to measure the loss during the learning process defined by equation 8 [41].

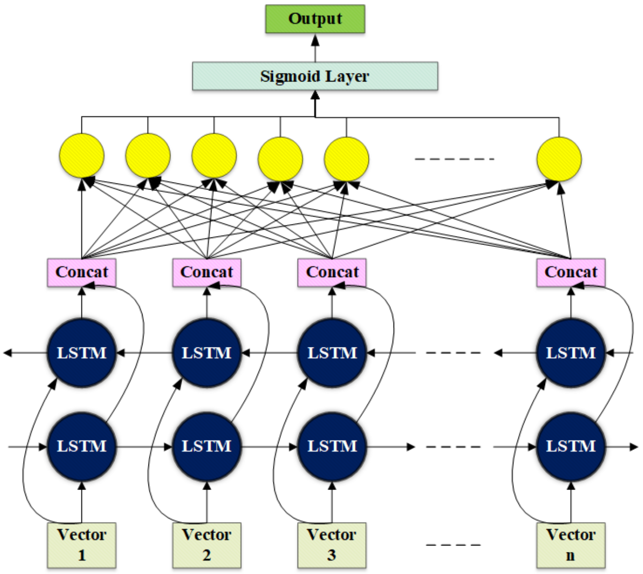

Figure 2: The BiLSTM Network Architecture

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} HM(y_i) \times log\left(p\left(HM(y_i)\right)\right) + \left(1 - HM(y_i)\right) \times log\left(1 - p(y_i)\right) \quad (8)$$

Here, HM (y) is the label of the training dataset assigned by the hashing module, and N is the total number of datasets. In this paper, we experimented with three different optimization algorithms to train the BiLSTM network. First, we experimented with the Adaptive Gradient Descent (Adagrad) optimizer. This algorithm updates the weight by equation 9 [42].

$$W_t = W_{t-1} - \eta_t' \frac{\delta L}{\delta w(t-1)} \qquad (9)$$

Here in the equation 9, the W_t is the updated weight, and W_{t-1} is the updated weight. The η_t' is defined by equation 10.

$$\eta_t' = \frac{\eta}{\sqrt{\alpha_t + \epsilon}} \qquad (10)$$

The second optimizer we experimented with is the Root Mean Square Propagation (RMS Prop) optimizer.

It calculates the average of squared gradients for each weight. After that, it divides the graduate by the square root of the mean square. First, the output from the network v(w, t) is calculated using the equation 11 [43].

$$v(w,t) = \gamma v(w,t-1) + (1-\gamma)\left(\delta Q_i(w)\right)^2 \qquad (11)$$

After that, the v(w, t) is used to update weight using the equation 12.

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{v(w,t)}}\delta Q_i(w) \qquad (12)$$

After getting the experimental results from the Adagrad and RMS-Prop algorithms, we experimented with the network using the Adaptive Moment Estimation (ADAM) algorithm. The adaptive momentum used to update weight is calculated using the equation 13 [44].

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\left[\frac{\delta L}{\delta w_t}\right] v_t \qquad (13)$$

$$R_s = \sum_{i=1}^{N} s[i] \qquad (14)$$

Here is equation 14, the s is the skillset specified by the HR hiring managers. The skill set of a particular candidate is a subset of the Rs defined by the equation 15.

$$s \in R_s \qquad (15)$$

The comparison between the s and the R_s is made through the probability score obtained from the processing unit. The higher the probability, the higher the resume rank for a particular job circular.

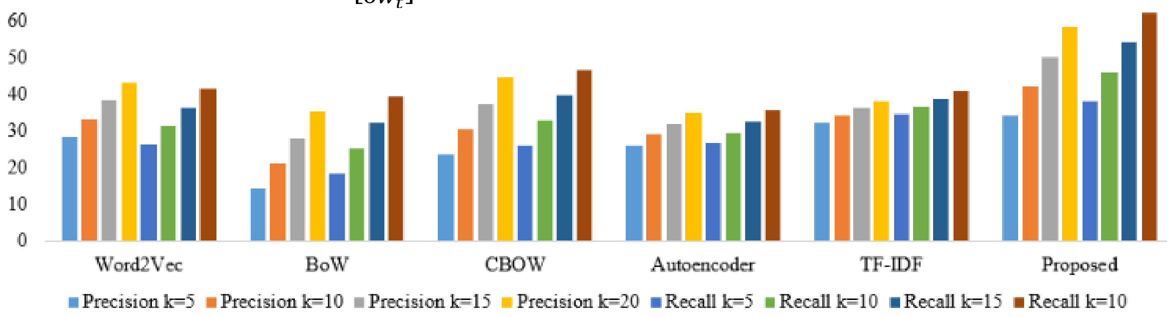

Figure 3: The performance analysis of different word embedding approaches

Based on the experimental results, which have been presented in the Experimental Results and Evaluation section, the ADAM has been used in this research for better performance.

Table 3: THE PERFORMANCE EVALUATION ON WORD EMBEDDING ALGORITHMS

	Precision				Recall			
Model	k=5	k=10	k=15	k=20	k=5	k=10	k=15	k=10
Word2Vec	28.17	33.42	38.17	43.2	26.74	31.3	36.34	41.15
BoW	14.12	21.7	28.04	35.08	18.41	25.78	32.41	39.2
CBOW	23.41	30.96	37.41	44.71	25.76	32.42	39.71	46.63
Autoencoder	25.62	28.14	31.68	34.1	26.79	29.14	32.47	35.5
TF-IDF	32.14	34.1	36.41	38.04	34.61	36.04	36.78	40.79
Proposed	34.13	42.16	50.34	58.67	37.34	45.16	53.47	61.88

4) The Ranking Module (RM): The ranking module uses the prediction from the processing module to rank the candidates' resumes. It involves the candidates' skillset and the required skillset mentioned in the job description. The skillset necessary is known from the job description. The candidates' skillset is unknown. The skillset needed in the job description is expressed as equation 14.

Algorithm 2 The resume ranking algorithm

Input: Query, Q; Skills, S; Job Title, T;
Output: Rank, R;
$D \leftarrow$ HM(S, T)
$d \leftarrow$ HM(resume(s))
$P(d|Q) \leftarrow query(PM(d, D), Q)$
$R \leftarrow P(d|Q) \times 100$
Return R

III. Experimental Results and Evaluation

A. *Evaluation Metrics*

There are a variety of job descriptions in the experimenting datasets. Some descriptions are almost identical, some are similar, and some are unique. Because of the nonuniform nature of the dataset, an overall performance evaluated using the state-of-the-art regression evaluation metrics R Squared (R^2), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) is not an appropriate approach [45]. That is why we have taken a different approach to evaluating the performance of the proposed resume ranker.

We have already discussed the dynamic nature of the job market in the introduction. We created a similar vibrant environment in the experimental setup by randomly selecting the top 15% of the data from the test dataset. Then we observed the top N -recommendations where N = 5, 10, 15, 20. We used precision and recall of the network on the test dataset for different values on N to evaluate the effectiveness of the network.

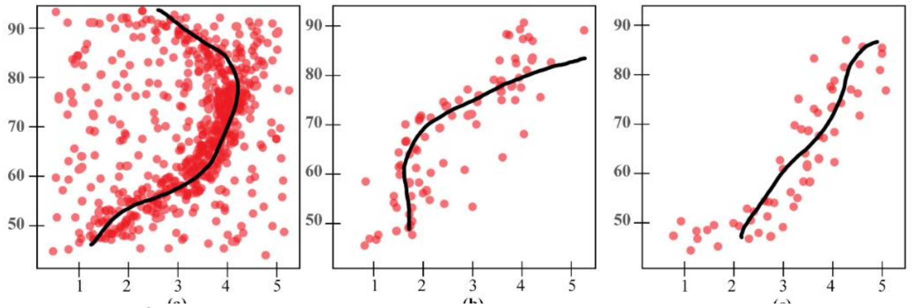

Figure 5: (a)R^2 on 7000 instances, (b) RMSE on 500 instances, and (c) MAE on 500 instances

B. Evaluation of Word Embedding

The hash-based word embedding is one of the noble contributions of this paper. The first experiment has been conducted to determine this embedding scheme's effectiveness. The precision and recall for N = 5, 10, 15, 20 have been listed in table III. The experimental results demonstrate that the proposed hash-based word embedding performs better than other popular word embedding schemes for the experimenting scenario. The comparison is more vivid in figure 3. Although the performance is better than other embedding algorithms, it is still insufficient to rank the resumes properly. It has been observed that changing the range of N to 3 to 5 in the Word N-gram significantly improves precision and accuracy. The improved results have been listed in table IV.

Table 4: THE IMPROVEMENT AFTER CHANGING THE RANGE

Embedding	N	Precision	Recall
Proposed	5	57.48	62.17
Proposed	10	65.41	76.86
Proposed	15	78.1	82.35
Proposed	20	88.96	92.74

Figure 4 shows the improved precision and recall at different values of N . No significant variations between precision and recall are observed. Both of these are equally improved after changing the range of N from the Word N-grams.

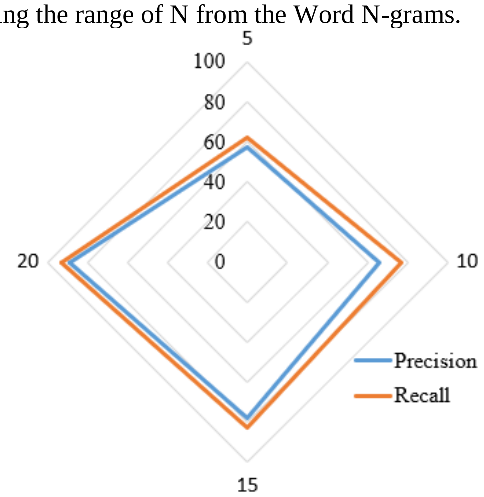

Figure 4: The improved precision and recall

Figure 4 shows the improved precision and recall at different values of N. No significant variations between precision and recall are observed. Both of these are equally improved after changing the range of N from the Word N-grams.

C. The Network Performance Evaluation

The network performance has been evaluated using R Squared (R^2), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). 8000 instances from the test dataset have been used for this evaluation. To plot the R^2, RMSE, and MAE, relevance scores from 1 to 5 have been prepared based on the 2D spatial distribution of the prediction made by the network expressed in percentage. For ease of visualization, the 7000 instances for R^2, 500 instances for RMSE, and 500 instances for the MAE have been illustrated in figure 5. The values have been calculated from the same number of instances mentioned earlier. The visualization shows that the hypothesis fits properly for all three data distributions. That means the network is well-trained for correct prediction.

In this experiment, we used k-fold cross validation at k = 5. The values of the precision, recall, accuracy, R^2, RMSE, and MAE have been listed in table V at different values of k. The average performance is listed in the table as well. The accuracy, precision, and recall at each fold are almost similar. The R^2, RMSE, and MAE are also stable at every fold. That means the network is not overfitting and good at generalization.

Table 5: THE AVERAGE PERFORMANCE WITH 5-FOLD CROSS VALIDATION

K	Accuracy	Precision	Recall	R Square	RMSE	MAE
1	88.17	87.94	88.45	6.92	5.47	7.48
2	89.67	88.75	89.76	7.02	6.02	7.91
3	90.78	88.05	89.7	6.47	6.75	6.98
4	89.99	87.62	89.9	6.84	5.97	7.04
5	91.05	88.19	88.03	6.74	6.04	7.12
Average	89.932	88.11	89.17	6.798	6.05	7.31

Figure 6 demonstrates the insignificant variations at the different folds. The R2, RMSE, and MAE are significantly lower than the accuracy, precision, and recall. That is why there is a sudden change in the slope on 6.

The variations among the R2, RMSE, and MAE on the figure are ignoble. The experimental results and evaluation demonstrate the excellent performance of the proposed system in ranking the resumes against the job description depending on the skill matching. The proposed system is robust, dynamic, and correctly rank resumes with less than 7% R^2 error.

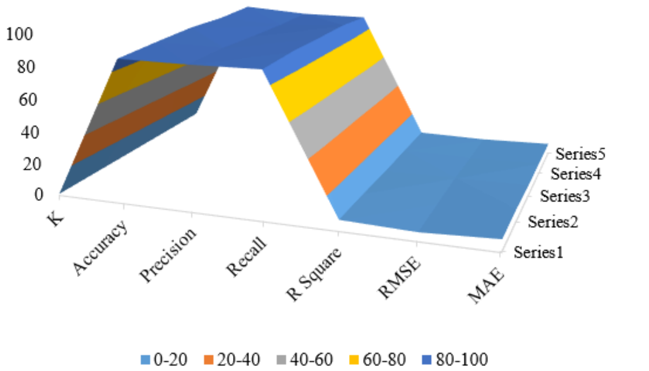

Figure 4: The performance visualization

IV. Conclusion

The innovative hashing scheme and BiLTSM-based dynamic resume ranking system is an extension of the content-based-filtering dependent job recommender system. The key to technological sustainability is to evolve with the change. The job market change rate from both requirements and operational perspectives is the highest that every human civilization has experienced. The system presented in this paper has been designed to help HR professionals automate their recruitment process by automatically ranking the candidates' resumes. It solves the challenges imposed by the rate of rapid change of job requirements and the entrance of new candidates on Deep Learning based automation. This research has examined and presented the analysis of the current context and possible technology to develop a dynamic, robust, automatic, and accurate resume ranking system. Based on the analysis, an innovative hashing-based word embedding scheme has been developed in a three-module system. This BiLSTM-dependent system ranks the resumes with 89.93% accuracy, 88.11% precision, 89.17% recall, 6.79% R^2, 6.05% RMSE, and 7.31% MAE. The proposed system has been cross-validated and examined in different ways, and the experimental results, along with thorough evaluation, demonstrate that it is a potential solution to rank resumes in today's rapidly changing job market.

References

[1] H. H. Larsen and C. Brewster, "Line management responsibility for hrm: what is happening in europe?," Employee relations, 2003.

[2] A. Mohamed, W. Bagawathinathan, U. Iqbal, S. Shamrath, and A. Jayakody, "Smart talents recruiter-resume ranking and recommendation system," in 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), pp. 1–5, IEEE, 2018.

[3] V. Yakubovich and D. Lup, "Stages of the recruitment process and the referrer's performance effect," Organization science, vol. 17, no. 6, pp. 710–723, 2006.

[4] A. Vatousios and A. Happonen, "Transforming hr and improving talent profiling with qualitative analysis digitalization on candidates for career and team development efforts," in Intelligent Computing, pp. 1149–1166, Springer, 2022.

[5] T. V. Yadalam, V. M. Gowda, V. S. Kumar, D. Girish, and M. Namratha, "Career recommendation systems using content based filtering," in 2020 5th International Conference on Communication and Electronics Systems (ICCES), pp. 660–665, IEEE, 2020.

[6] Y. Zheng and D. X. Wang, "A survey of recommender systems with multi-objective optimization," Neurocomputing, vol. 474, pp. 141–153, 2022.

[7] Faruqui, N., Yousuf, M. A., Chakraborty, P., & Hossain, M. S. (2020). Innovative automation algorithm in micro-multinational data-entry industry. In Cyber Security and Computer Science: Second EAI International Conference, ICONCS 2020, Dhaka, Bangladesh, February 15-16, 2020, Proceedings 2 (pp. 680-692). Springer International Publishing.

[8] A. Gunawardana, G. Shani, and S. Yogev, "Evaluating recommender systems," in Recommender systems handbook, pp. 547–601, Springer, 2022.

[9] S. Ahmadian, M. Ahmadian, and M. Jalili, "A deep learning based trust-and tag-aware recommender system," Neurocomputing, vol. 488, pp. 557–571, 2022.

[10] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems," Knowledge-Based Systems, vol. 211, p. 106545, 2021.

[11] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in Proceedings of the 24th international conference on world wide web, pp. 278–288, 2015.

[12] L. Almonte, E. Guerra, I. Cantador, and J. De Lara, "Recommender systems in model-driven engineering," Software and Systems Modeling, vol. 21, no. 1, pp. 249–280, 2022.

[13] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: recommendation models, techniques, and application fields," Electronics, vol. 11, no. 1, p. 141, 2022.

[14] A. Iana, M. Alam, and H. Paulheim, "A survey on knowledge-aware news recommender systems," Semantic Web Journal, 2022.

[15] H. Wang, D. Amagata, T. Makeawa, T. Hara, N. Hao, K. Yonekawa, and M. Kurokawa, "A dnn-based cross-domain recommender system for alleviating cold-start problem in e-commerce," IEEE Open Journal of the Industrial Electronics Society, vol. 1, pp. 194–206, 2020.

[16] M. Almgerbi, A. De Mauro, A. Kahlawi, and V. Poggioni, "A systematic review of data analytics job requirements and online-courses," Journal of Computer Information Systems, vol. 62, no. 2, pp. 422–434, 2022.

[17] C. Clegg and C. Spencer, "A circular and dynamic model of the process of job design," Journal of Occupational and Organizational Psychology, vol. 80, no. 2, pp. 321–339, 2007.

[18] S. Khatwani and M. Chandak, "Building personalized and non personalized recommendation systems," in 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), pp. 623–628, IEEE, 2016.

[19] T. Gr ̈utzmacher, T. Cojean, G. Flegar, H. Anzt, and E. S. Quintana-Ort ́ı, "Acceleration of pagerank with customized precision based on mantissa segmentation," ACM Transactions on Parallel Computing (TOPC), vol. 7, no. 1, pp. 1–19, 2020.

[20] N. Sano, N. Machino, K. Yada, and T. Suzuki, "Recommendation system for grocery store considering data sparsity," Procedia Computer Science, vol. 60, pp. 1406–1413, 2015.

[21] A. H. Khan, J. Siddqui, and S. S. Sohail, "A survey of recommender systems based on semi-supervised learning," in International Conference on Innovative Computing and Communications, pp. 319–327, Springer, 2022.

[22] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," IEEE Potentials, vol. 13, no. 4, pp. 27–31, 1994. [23] C.-F. Tsai, "Bag-of-words representation in image annotation: A review," International Scholarly Research Notices, vol. 2012, 2012.

[23] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 974–983, 2018.

[24] I. Verboncu and L. Zeininger, "The manager and the managerial tools: Job description," Revista de Management Comparat International, vol. 16, no. 5, p. 603, 2015.

[25] L. R. Medsker and L. Jain, "Recurrent neural networks," Design and Applications, vol. 5, pp. 64–67, 2001.

[26] Y. Tan, X. Wang, and T. Jia, "From syntactic structure to semantic relationship: Hypernym extraction from definitions by recurrent neural

[27] networks using the part of speech information," in International Semantic Web Conference, pp. 529–546, Springer, 2020.

[28] J. Abendroth, A. Heiss, T. Jacobsen, S. R ̈ottger, and J. Kowalski, "Job mobility and job performance: beliefs about social and occupational (dis) advantages as mediators," International Journal of Occupational Safety and Ergonomics, vol. 28, no. 1, pp. 199–212, 2022.

[29] L. Rasmy, M. Nigo, B. S. Kannadath, Z. Xie, B. Mao, K. Patel, Y. Zhou, W. Zhang, A. Ross, H. Xu, et al., "Recurrent neural network models (covrnn) for predicting outcomes of patients with covid-19 on admission to hospital: model development and validation using electronic health record data," The Lancet Digital Health, 2022.

[30] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," arXiv preprint arXiv:1402.1128, 2014.

[31] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning–based text classification: a comprehensive review," ACM Computing Surveys (CSUR), vol. 54, no. 3, pp. 1–40, 2021.

[32] F. Heimerl and M. Gleicher, "Interactive analysis of word vector embeddings," in Computer Graphics Forum, vol. 37, pp. 253–265, Wiley Online Library, 2018.

[33] P. L. Rodriguez and A. Spirling, "Word embeddings: What works, what doesn't, and how to tell the difference for applied research," The Journal of Politics, vol. 84, no. 1, pp. 101–115, 2022.

[34] K. W. Church, "Word2vec," Natural Language Engineering, vol. 23, no. 1, pp. 155–162, 2017.

[35] L. Shi, A. Somnali, and P. Guan, "Using continuous bag of words," 2022.

[36] J. Su, S. Wu, B. Zhang, C. Wu, Y. Qin, and D. Xiong, "A neural generative autoencoder for bilingual word embeddings," Information Sciences, vol. 424, pp. 287–300, 2018.

[37] A. Aizawa, "An information-theoretic perspective of tf–idf measures," Information Processing & Management, vol. 39, no. 1, pp. 45–65, 2003.

[38] K. Erk, "Vector space models of word meaning and phrase meaning: A survey," Language and Linguistics Compass, vol. 6, no. 10, pp. 635–653, 2012.

[39] Achar, S.; Faruqui, N.; Whaiduzzaman, M.; Awajan, A.; Alazab, M. Cyber-Physical System Security Based on Human Activity Recognition through IoT Cloud Computing. Electronics 2023, 12, 1892. https://doi.org/10.3390/electronics12081892.

[40] Paula, L. P. O., Faruqui, N., Mahmud, I., Whaiduzzaman, M., Hawkinson, E. C., & Trivedi, S. (2023). A Novel Front Door Security (FDS) Algorithm Using GoogleNet-BiLSTM Hybridization. IEEE Access, 11, 19122-19134..

[41] Trivedi, S., Patel, N., & Faruqui, N. (2022, October). NDNN based U-Net: An Innovative 3D Brain Tumor Segmentation Method. In 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0538-0546). IEEE.

[42] Patel, N., Trivedi, S., & Faruqui, N. (2023, February). An Innovative Deep Neural Network for Stress Classification in Workplace. In 2023 International Conference on Smart Computing and Application (ICSCA) (pp. 1-5). IEEE.

[43] S. K. Turitsyn, T. Schafer, and V. K. Mezentsev, "Generalized root-mean-square momentum method to describe chirped return-to-zero signal propagation in dispersion-managed fiber links," IEEE Photonics Technology Letters, vol. 11, no. 2, pp. 203–205, 1999.

[44] Trivedi, S., Patel, N., & Faruqui, N. (2023, January). Bacterial Strain Classification using Convolutional Neural Network for Automatic Bacterial Disease Diagnosis. In 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 325-332). IEEE.

[45] Patel, N., Trivedi, S., & Faruqui, N. (2023, February). A Novel Sedentary Workforce Scheduling Optimization Algorithm using 2 nd Order Polynomial Kernel. In 2023 International Conference on Smart Computing and Application (ICSCA) (pp. 1-7). IEEE.