Multi-class Resume Classification Framework for Skill Extraction

MSc Research Project
Data Analytics

Anwesha Karmakar
Student ID: x21195838

School of Computing
National College of Ireland

Supervisor: Paul Stynes, William Clifford, Eugene McLaughlin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……Anwesha Karmakar……………………………………………………………….…………

Student ID: ……x21195838……………………………………………………………………………..……

Programme: ……MSc Data Analytics……………………………… **Year:** …2023……………………..

Module: ……MSc Research Project……………………………………………………………….………

Supervisor: ……Paul Stynes, William Clifford, Eugene McLaughlin………………….………

Submission Due Date: ……14/08/2023……………………………………………………………………….………

Project Title: …..Multi-class Resume Classification Framework for Skill Extraction…….

Word Count: ……3932.……………………… **Page Count**…………13……………………………….……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …Anwesha Karmakar………………………………………………………………………

Date: …14/08/2023………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Multi-class Resume Classification Framework for Skill Extraction

Anwesha Karmakar
X21195838

Abstract

Skill extraction identifies and extracts technical and soft skills from resumes such as programming and problem solving. Current research uses machine learning and deep learning technologies for extracting skills from resumes. However, the **challenge** is in identifying high-level skills such as web development from low-level skills such as HTML. **This research proposes a Multi-class Resume Classification framework** to assist recruiters in hiring candidates with the right technical skills. **The proposed framework** combines an optimal transfer learning-based word embedding model with deep learning resume classification model. A deep learning text classification model is trained using resume corpus dataset consisting of 29,783 resumes to classify ten classes of occupations. Contextual word embedding and deep learning technique named Bidirectional Long Short-Term Memory (BiLSTM) is applied. **Results** of the model are presented based on accuracy, loss, precision, sensitivity, and specificity. This research shows promise for the proposed model in classifying resumes into different categories.

1 Introduction

Manually reviewing resumes is highly labor-intensive and lacks cost efficiency. Over the past few years, job-resume matching algorithms have emerged to streamline candidate selection and automate the hiring process. Within an organization's human resource management system, the task of identifying suitable candidates for job positions has become increasingly difficult due to the large volume of applications received. Deep learning text classification models such as transfer learning and RNN can improve the accuracy of classification of different IT roles. Although the existing methods can effectively perform skill extraction from resumes, the studies lack the extraction of high-level skills based on the low-level skills. **The aim of the research** is to improve the automation process of recruitment by improving the skill extraction technique.

The research question posed in this study investigates how a multi-class resume classification framework can be used to accurately extract and predict skills from resumes.

To address the research question, the following specific sets of research objectives are derived:

1. Investigate the state of the art broadly around skill extraction and resume classification.
2. Design a multi-class resume classification framework.
3. Implement multi-class resume classification framework.
4. Evaluate the framework using accuracy and precision.

The major contribution of this research is a novel multi-class resume classification framework that combines an optimal transfer learning model with self-attention mechanism and deep learning model that improves skill extraction. A minor contribution of this research is the application of hybrid deep neural network employing Bidirectional LSTM, which will enhance the process of skill extraction. The resume corpus dataset used in the study consists of 29,783 resumes with ten classes of IT occupations namely front-end developer, network administrator, web developer, project manager, database administrator, security analyst, software developer, systems administrator, python developer, java developer.

This paper discusses deep learning models used for multi-class resume classification and application of transfer learning for skill extraction in section 2 related work. The research methodology is discussed in section 3. The design components of multi-class resume classification are discussed in section 4. Section 5 illustrates the implementation of the research. The evaluation and results are presented in section 6. Section 7 discusses the conclusion of research and focuses on future work.

2 Related Work

Automatic skill extraction and classification of resumes are making the recruitment system more efficient. This can be beneficial for both recruiters and job seekers as the process is much faster than manual processing. Peña et al. (2023) in their study have showed that application of machine learning techniques can reduce bias and improve fairness in recruitment system. Numerous research has been performed over the past years to automatically extract skills from resumes. Research works on skill extraction and classification is discussed in section 2.1. Section 2.2 discusses related works performed using BERT based transfer learning. Works on bidirectional LSTM is discussed in section 2.3.

2.1 Works on Skill Extraction and Classification

Current research in resume skill extraction is implementing deep learning models such as Recurrent Neural Network (RNN) (Xu et al.; 2020), Convolutional Neural Network (CNN) (Nasser, Sreejith and Irshad; 2018; Ayishathahira, Sreejith and Raseek; 2018) and Deep Neural Network (DNN) (Habous; 2021) to improve the automatic extraction process. Nasser, Sreejith and Irshad (2018) have implemented CNN with conv1D, MaxPooling1D, dropout and dense layers to classify resume in several technical and non-technical categories.

Jiechieu and Tsopze (2020) have used skip-gram algorithm for word embedding and CNN to classify resumes into ten skill categories. The model was trained using resume corpus dataset. The dataset consisted of almost 30000 resumes of ten categories namely front-end developer, network administrator, web developer, project manager, database administrator, security analyst, software developer, systems administrator, python developer, java developer. Two classes of the dataset, network administrator and system administrator are closely related. The idea of their work is to predict high-level skills from resumes even if those are not explicitly mentioned. The dataset used is divided in ten sub-datasets and ten base classifiers are trained for each class. Their proposed technique yielded 90% accuracy which shows promising results. However, this approach may not be suitable for a greater number of classes present in the data as the number of CNNs involved will rise which will eventually increase the model complexity.

2.2 Works on BERT

Many researchers have utilised transfer learning in their research due to its promising results. Bidirectional Encoder Representations from Transformers (BERT) (Ravichandiran; 2021) model has been pre-trained on English Wikipedia data that covers all possible words. Barducci et al. (2020) have implemented BERT as contextual embedder to extract information from Italian resumes. Their research has shown significant improved result in extracting information using BERT in comparison to other research works. James, Kulkarni and Agarwal (2023) have implemented BERT for performing skill extraction and ranking of resumes. The model has been proved to deliver improved results than previous models used.
Wings, Nanda and Adebayo (2021) have proposed a context-aware approach for extracting hard and soft skills from job descriptions using different embedding techniques. The research compared the performance of word2Vec, FastText and BERT for word embedding and highlighted the performance of BERT as superior. Their study indicates that transformer-based context-specific technique is better for skill extraction. The original BERT base uncased model was fine-tuned using TensorFlow and Hugging face.
BERT have been used extensively in numerous research for contextual meaning extraction. Vukadin et al. (2021) have performed resume information extraction from resumes in multiple languages. Multilingual pre-trained BERT model was applied to perform contextual vector embedding. Their findings show that the performance of model improves with a greater number of BERT layers. Tallapragada et al. (2023) have implemented BERT vectorization to extract contextual information from resumes. The above studies validate the usage of BERT as contextual word embedder for resume skill extraction.

2.3 Works on Bidirectional LSTM

Bidirectional Long-Short Term Memory is an improved version of RNN. It is one of the most popular deep learning models for textual analysis. This model has the ability to capture contextual features and semantic information due to its capability of capturing long-term dependencies. Liu et al. (2021) proposed resume parsing technique based on multi-label classification. They have compared few different techniques such as BPNN, CNN, BiLSTM and CRNN to classify resume texts. The combined BiLSTM model was able to classify the resumes accurately 96% of time with one layer on LSTM and 100 epochs. This study shows that combining BiLSTM can deliver improved accuracy than previous techniques.
Xu (2022) implemented combined CNN and BiLSTM with attention mechanism to extract word-level features from resumes. BiLSTM with attention mechanism proved to be performing better in classification in terms of accuracy, AUC and F1 score. However, the performance of the model could be improved by fine tuning the parameters of BiLSTM.
In another study of resume information extraction, a combined technique of BERT, BiLSTM and CRF is proposed (Li; 2021). Different combinations of classification techniques were examined among which combination of BiLSTM, BERT and CRF performed best. Although the study achieved 91% precision, model performance could be enhanced by fine tuning the parameters. Su, Zhang and Lu (2019) had performed a study on information extraction of large resume corpus data. The study compared four different classification models where BiLSTM performed best achieving 98% accuracy. The above studies motivate the use of BiLSTM as deep learning classification model in this research.

In conclusion, it is seen that semantic embeddings such as word2Vec, FastText, and classification models such as CNN, CRNN, BPNN, and BiLSTM with attention mechanism have been employed to enhance contextual and semantic richness of automatic skill extraction process by past researchers. However, transformer based contextual embedding and BiLSTM have been largely underutilised in resume classification. Jiechieu and Tsopze (2020) investigated the impact of convolution filters to visualise and classify resumes based on skills, but their work was limited as they used CNN which is incapable of considering future information or sequence structure of words, thus does not perform well for longer text sequences. Moreover, the study utilised word2Vec which is incapable of producing dynamically informed word representations for word embedding. For addressing this gap, this work investigates a novel resume classification approach leveraging textual information as word embedding adopting transfer learning to capture deeper contextual information. It also adopts BiLSTM to consider both past and future information for resume classification and this combined method of resume classification does not appear to have been used in any works.

3 Research Methodology

The research methodology consists of five **steps namely** data gathering, data pre-processing, data transformation, data modelling, evaluation and results as shown in figure 1.

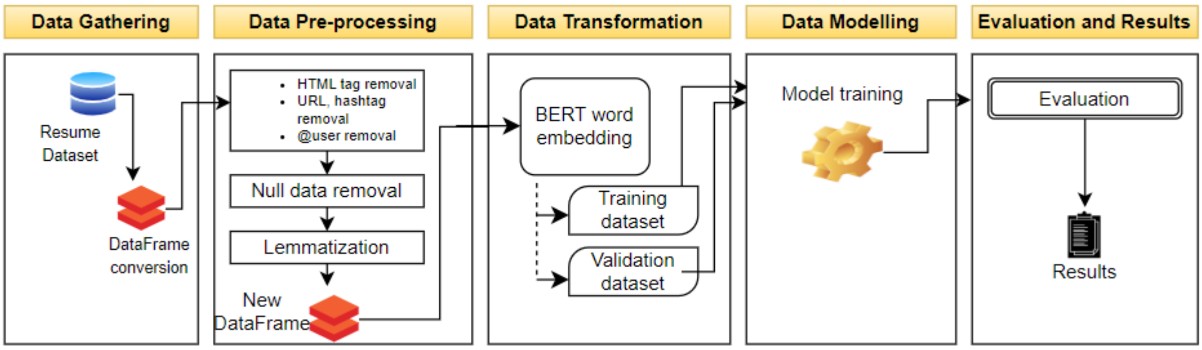

Figure 1: Research Methodology

The first step, *Data Gathering* involves collecting and converting the dataset[1]. The original unstructured dataset containing texts was converted to a structured data frame using pandas library.

The second step, *Data Pre-processing* involves data cleaning by performing html tags removal, URL, hashtags and @user removal and lemmatization. Texts containing unnecessary information such as html tags, URL, hashtags and @user were removed. Resumes containing no labels or skills were removed. Text segments were normalized using WordNet lemmatizer to get the root forms of words. After data pre-processing, the resume corpus dataset contained 29035 text resumes that were saved in a new data frame.

The third step, *Data Transformation* involves word embedding, tokenization, padding of data and splitting the dataset in training and testing datasets. Word embedding involves extracting the contextual meaning of textual data. The resume corpus dataset consists of ten classes namely front-end developer, network administrator, web developer, project manager, database administrator, security analyst, software developer, systems administrator, python developer, java developer. Word embedding was applied to the resume data using transformer model. A pretrained transformer model named BERT base uncased was imported

[1] https://github.com/florex/resume_corpus

from the library of TensorFlow. Transfer learning was employed to perform deep information extraction. Bert tokenizer was used to tokenize the words of each resume. These included add special tokens = True, max length = 512, pad to max length = True, return attention mask = True, truncation = True. Zero padding was performed to make the maximum length of token to 512. The embedded dataset was split into a ratio of 80:20 with random state = 42 to training and testing sub-datasets.

The fourth step, *Data Modelling* involves model training and model optimization for achieving the best results. The model was trained with the splitted training dataset (23228 resumes) and validated with training dataset (5807 resumes). LSTM, dense, flatten and dropout layers were imported from the library of TensorFlow for model training. The bidirectional LSTM model was trained using one Bi-LSTM layer, 8 dense layers, 2 dropout and 1 flatten layer with training data size of (23228, 768). The neural network model was trained with 100 epochs, categorical cross entropy loss function, ReLU activation function, SoftMax activation function and accuracy as a metric. The performance of the trained model was validated using the testing dataset. The model was optimized using Adam optimizer and a different set of learning rates. The graph of accuracy, loss and confusion matrix were plotted for both training and validation. Accuracy, precision, sensitivity, specificity of all the classes were generated for performing evaluation.

The fifth step, *Evaluation and Results* involves evaluating the performance of deep learning model by using accuracy, loss, sensitivity, and specificity. The results were compared with training and validation. The optimal value of learning rate of Adam optimizer was selected by experimentation and was applied for skill classification.

4 Design Specification

The multi-class resume classification framework **architecture** combines a transfer learning-based word embedding model with a deep learning resume classification model as shown in figure 2. **The components of** the transfer learning-based word embedding model include resume samples, transfer learning model and tokenizer as discussed in section 4.1. Components of deep learning resume classification model are discussed in section 4.2.

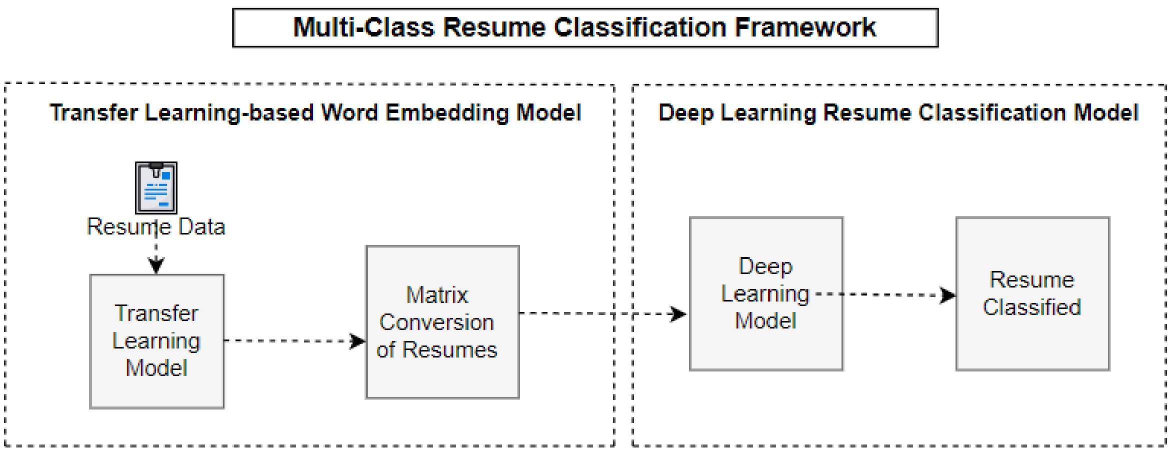

Figure 2: Multi-Class Resume Classification Framework Architecture

4.1 Transfer Learning Based Word Embedding Model

Transfer learning-based word embedding was applied on resume data after performing preprocessing steps. The transfer learning model of the framework is based on BERT base uncased model. Each word of resumes was converted to tokens and unique IDs are assigned to each token. The token inputs were converted to a vector to form matrix representation of input ids and attention masks. The input ids were then converted to a NumPy array for model training.

4.2 Deep Learning Resume Classification Model

Deep learning resume classification model classifies the vectorized resumes in the respective classes based on the low-level skills present in the resumes. The deep learning resume classification model consists of one bidirectional LSTM layer, 8 dense layers of size (2048 and 1024 and 256), 2 dropout layers of 0.2 and 0.5 to reduce the chances of overfitting and one flatten layer. The output of dense layer was reshaped in the size of (64, 32) for input of bidirectional LSTM layer. The final dense layer produced an output of 10 classes, which classifies the resumes in respective classes. The deep learning model then checks the accuracy, precision, sensitivity, specificity of the match between predicted and actual classes.

5 Implementation

The multi-class resume classification framework was **implemented** to classify resumes in 10 classes according to the skills of resumes. The development environment used for implementation of code was python 3.8.10. Keras v2.11.0 was used for model building and TensorFlow v2.9.1 was used as backend. The experiment was carried out on cloud-based machine having 200 GiB RAM, 1x NVIDIA A10 GPU, 24 GB vRAM per GPU and 30 vCPUs. The necessary modules such as NLTK, transformers, TensorFlow, np_utlis were installed. All the necessary packages were imported from libraries. Pre-processing was performed by removing HTML tags from data using html parser of Python's BeautifulSoup package. URLs, hashtags and @users were removed using regular expressions. Lemmatization was performed using WordNet lemmatizer to get the lemma of words.

Transfer learning-based BERT word embedding was performed to extract deep features from processed resumes. The encoded tokens were converted to NumPy array in input id and attention masks. Input ids contained all the resumes in a matrix form which can be used to perform classification. The architecture of Bidirectional LSTM model consists of 14 layers having one bidirectional LSTM layer with 128 neurons, 8 dense layers of 2048, 1024 and 256 neurons, one flatten layer and two dropout layers of 0.2 and 0.5 to prevent overfitting. The model was trained with 100 epochs and batch size was set to default. The accuracy and loss function were calculated after every epoch on training and validation set for checking the performance of model. Adam optimizer was utilised to minimize the loss function and the optimal learning rate was found to be 0.00000753 delivering the best results.

6 Evaluation

The aim of this experiment is to compare BERT transfer learning and BiLSTM based deep learning resume classification with resume corpus dataset. Bidirectional LSTM model with several dense layers was trained to identify and classify resumes into ten different classes namely front-end developer, network administrator, web developer, project manager, database administrator, security analyst, software developer, systems administrator, python developer, java developer. For evaluating the performance of the model, accuracy and loss of model on training and validation data are compared. Confusion matrix of training and

validation data is generated to calculate the actual number of true positive, false positive, true negative and false negative cases, thereby calculating precision, recall or sensitivity and specificity.
This study performs two experiments where the first experiment to replicate the state of the art implementing CNN is discussed in section 6.1. Section 6.2 discusses the effectiveness of the combined transfer learning and deep learning model.

6.1 Experiment 1: Effectiveness of CNN model in classifying resumes

The state-of-the-art technique around resume skill classification was implemented as a part of replicating the base work. The performance of model is evaluated using accuracy, precision and recall. The previous work achieved an accuracy of 90.22% with word2Vec and CNN. The accuracy achieved in replicating the work was lower than the actual work. The results of experiment 1 is shown in the figure 3. The graph shows that the accuracy obtained in the task is 79.62% for 100 epochs of model training. The accuracy curve shows a slight gap between training and validation accuracy, which indicates overfitting of training data.

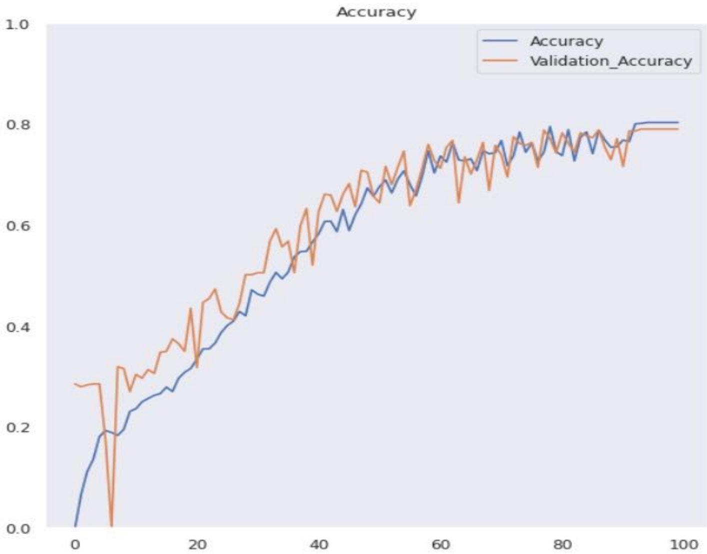

Figure 3: Accuracy curve of Experiment 1.

The difference in accuracy between past and present implementation is primarily because of lack of detailed specifications provided in the pre-processing techniques employed by researchers and usage of different hardware configuration which can reduce accuracy.

6.2 Experiment 2: Effectiveness of combined transfer learning model BERT and BiLSTM in classifying resumes

The combined transfer learning and deep learning model is evaluated with metrics such as accuracy, loss, precision, sensitivity, and specificity to validate the performance.

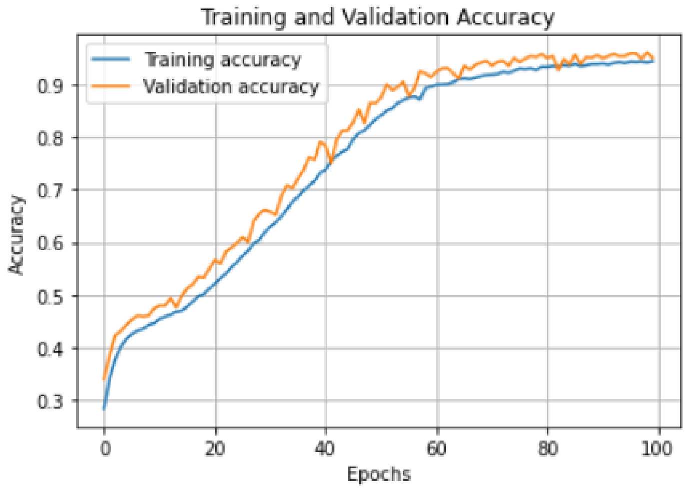

Figure 4: Accuracy curve of training and validation data.

The accuracy curve of proposed model on training and testing data is shown in Figure 4. Accuracy indicates how effectively a model can predict the outcome during each iteration of optimization. The graph shows significant increase of accuracy from 0.32 on 1st epoch to over 0.9 with increasing epochs. **This result indicates that** the combined BERT and Bidirectional LSTM model is able to accurately classify resumes at 96.18% on 100th epoch into 10 different categories.

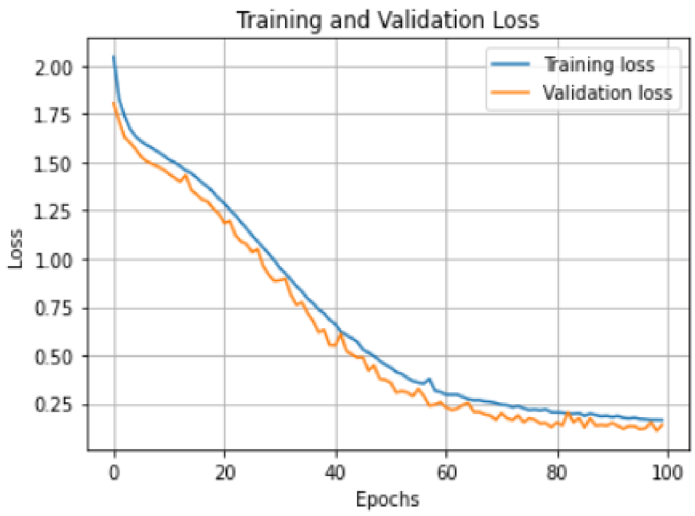

Figure 5: Loss curve of training and validation data.

Figure 5 shows the loss curve of applied model on training and testing data. Loss describes how effectively or poorly a model's forecast is after each iteration of optimization. Therefore, a model seems to perform better if the amount of loss is low. The graph shows steep and significant decrease in loss function with increasing number of epochs from 2 on 1st epoch to 0.09 on 100th epoch. **The results show promise** for this model with accuracy of 96.18% and loss function of 0.09 on 100th epoch on resume data based on the estimated loss of training and validation data.

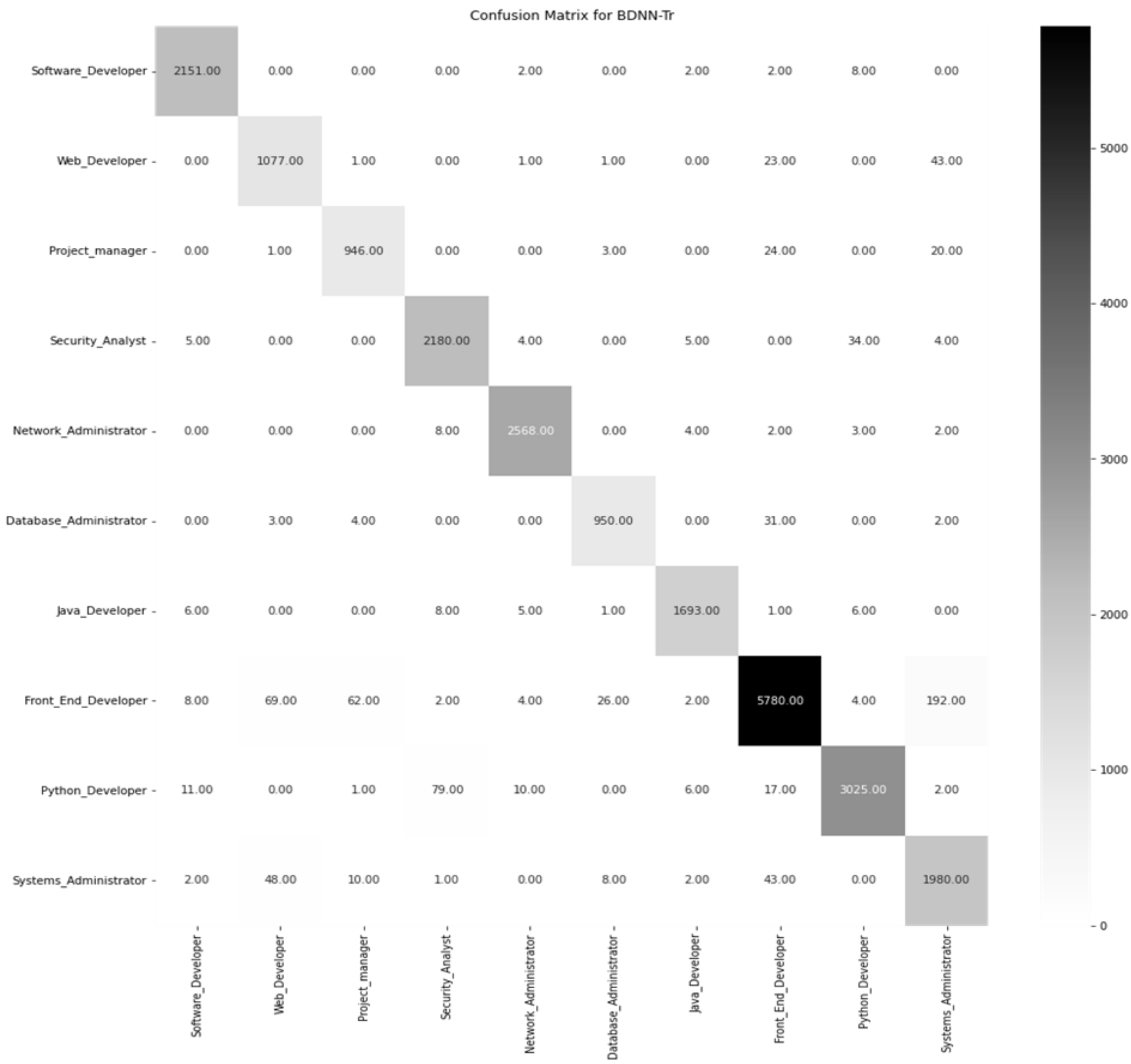

Figure 6: Confusion matrix of training data.

Figure 6 and 7 shows confusion matrix of training and testing dataset respectively. This gives a comparison between actual and predicted values in class wise distribution of 10 different resume categories. Confusion matrix provides the values of true positive, false positive, false negative and true negative, which is used to calculate metrics such as precision, sensitivity, and specificity.

Figure 8 presents some significant evaluation metrics considered for validating the performance of model. The class wise accuracy, precision and sensitivity is around 99% for most of the classes. The class front end developer has achieved least accuracy of 97.96% while software developer and java developer has achieved maximum accuracy of 99.8%. The average precision and sensitivity of all classes are at around 99%. Specificity of classes are ranged between 93% to 99.4% with software developer at maximum of 99.4% and web developer having lowest (93.3%) specificity. These values indicate that the number of false positive and false negative predictions are comparatively lower.

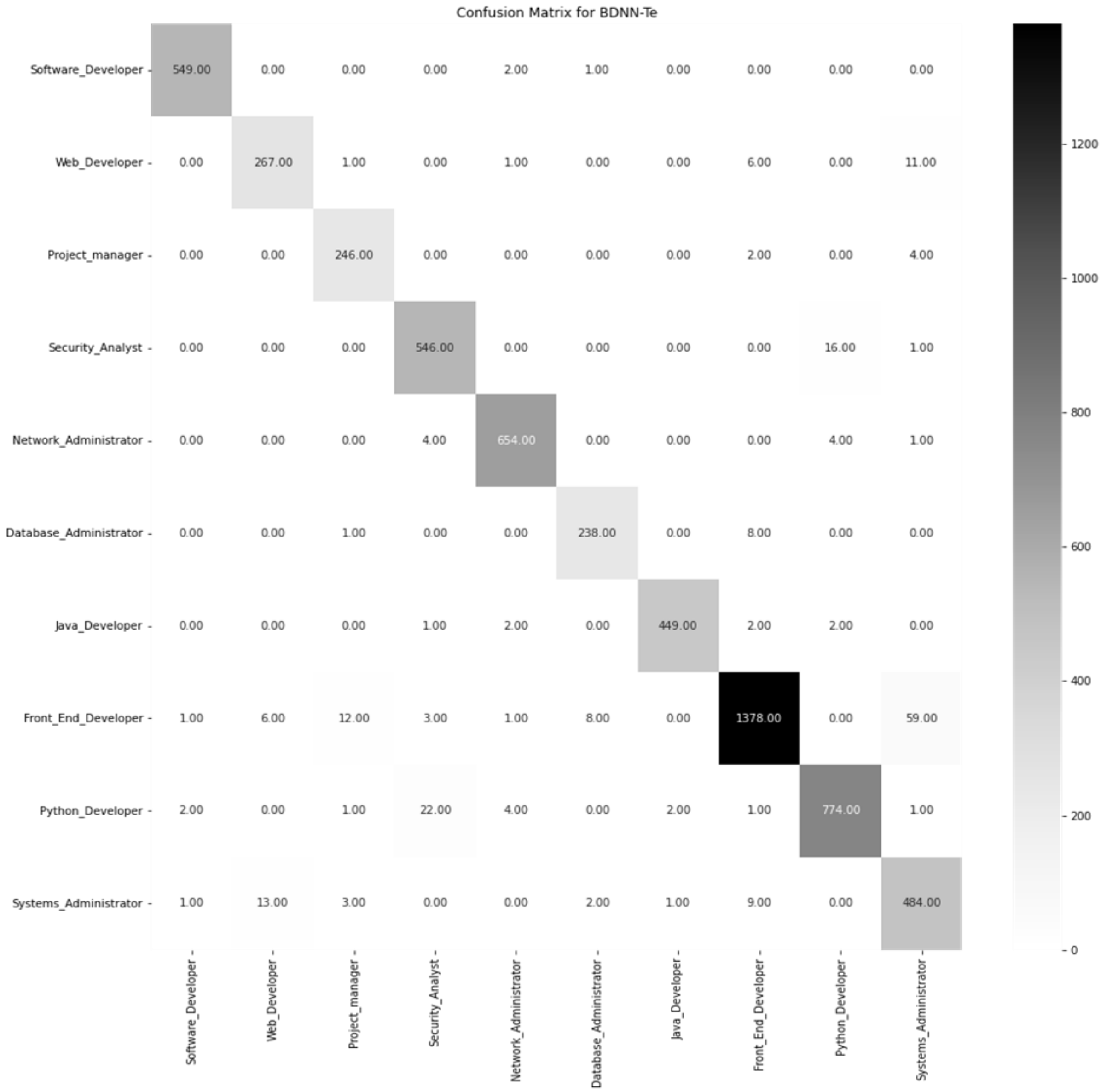

Figure 7: Confusion matrix of test data.

	class	TN	FP	FN	TP	Accuracy	Precision	Recall or Sensitivity	F1 Score	Specificity
0	Software_Developer	543	21	2	5241	0.996039	0.996009	0.999619	0.997811	0.962766
1	Web_Developer	257	22	47	5481	0.988118	0.996002	0.991498	0.993745	0.921147
2	Project_manager	252	3	34	5518	0.993628	0.999457	0.993876	0.996659	0.988235
3	Security_Analyst	567	11	94	5135	0.981918	0.997862	0.982023	0.989880	0.980969
4	Network_Administrator	636	27	6	5138	0.994317	0.994773	0.998834	0.996799	0.959276
5	Database_Administrator	226	10	7	5564	0.997072	0.998206	0.998743	0.998475	0.957627
6	Java_Developer	428	12	5	5362	0.997072	0.997767	0.999068	0.998417	0.972727
7	Front_End_Developer	1349	124	37	4297	0.972275	0.971952	0.991463	0.981611	0.915818
8	Python_Developer	723	73	28	4983	0.982607	0.985562	0.994412	0.989967	0.908291
9	Systems_Administrator	475	48	91	5193	0.976063	0.990841	0.982778	0.986793	0.908222

Figure 8: Class wise evaluation metrics of prediction.

The results of all the experiments are shown in figure 9. The results are compared using accuracy, precision, recall or sensitivity and specificity. The graph illustrates that experiment 2 achieved better results in terms of all the metrics. The experiment achieved an accuracy of 96.18%, 97.45% precision, 98.12% sensitivity and specificity of 97.56%.

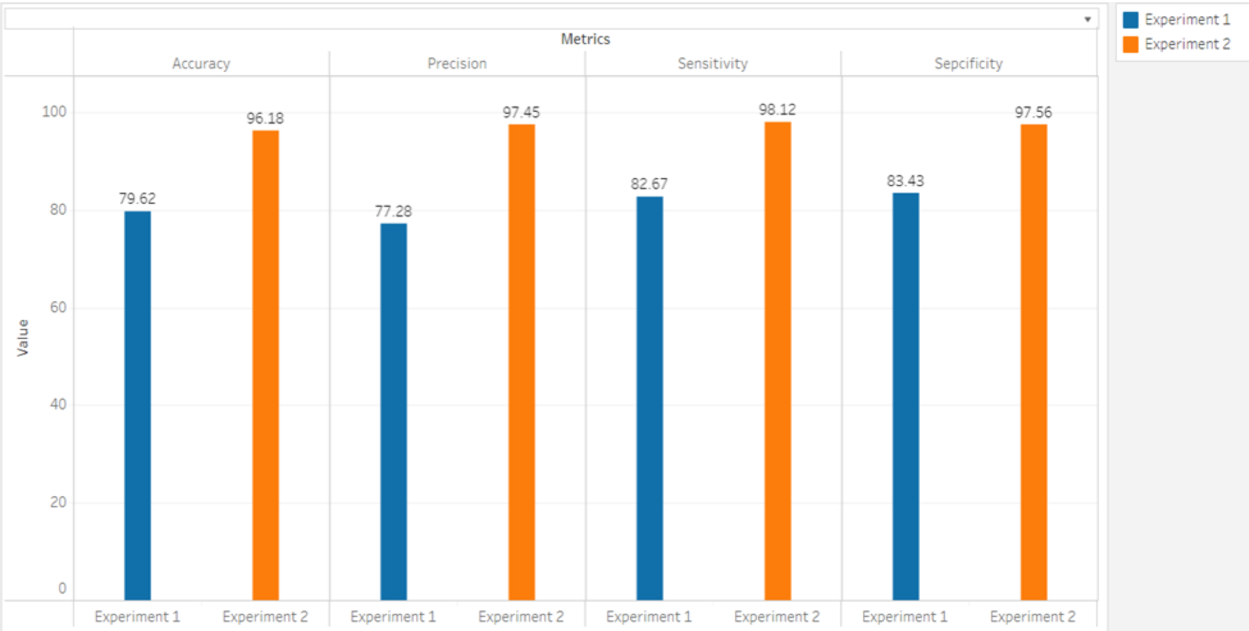

Figure 9: Summarization of results.

6.3 Discussion

This study finds the implementation of multi-class resume classification framework combining an optimal transfer learning-based word embedding model with deep learning resume classification model to be effective in classifying resumes. BERT is a powerful transformer-based model which can capture contextual information. However, the model captures information in a parallel manner, which can lead to loss of sequential information. BiLSTM can effectively capture sequential dependencies from textual data, which leads to better text processing. Moreover, the complexity of self-attention mechanism of BERT makes it harder to process very long sequences of data. On the other hand, BiLSTM is designed to handle longer sequences of data. BiLSTM layer can also prevent overfitting by acting as a regularizer. The combination of BERT and BiLSTM layer makes an ensemble method, which reduces the loss and biases leading to better performance of the overall model. Hence, BiLSTM combined with BERT lead in better performance compared to BERT.

The model achieved loss function of 0.09 and 96.18% accuracy, which is higher than the state-of-the-art. The class wise accuracy and precision of predicted classes is at around 99%, which indicates efficient classification of target skills. The high accuracy of few of the classes can be explained by the fact that the respective skills can be distinguished easily from another. Some skills such as systems administrator and network administrator are closely related. Front-end developer and web developers also might share common skills, which can lead to high accuracy for any of these classes. These might impact the performance of the prediction.

A limitation of this research is that a maximum token length of 512 is taken for BERT tokenizer which may overlook some words for resume size larger than this. The BERT model was originally pre-trained with maximum 512 token length and any token length larger than this will need the model to train from scratch. On that case, pre-trained knowledge of BERT cannot be utilized. Alternatively, longformer can be used for data containing large resumes as longformer can process sequences up to 4096 tokens.

7 Conclusion and Future Work

The aim of the research was to improve automatic extraction and classification of skills from resumes. **This research proposes** a multi-class resume classification framework that combines an optimal transfer learning-based word embedding model with deep learning resume classification model. **Results demonstrate that** Bidirectional LSTM shows promises if the motivation is for accuracy, loss, precision and sensitivity. The limitation of this study is inadequate resume data only related to IT skills, which hinders the comprehensive representation of diverse real-world scenarios.

The research can potentially enhance recruitment process, especially for large software farms which receive enormous number of job applications. **This work can be improved by** optimising transfer learning-based BERT word embedding model for capturing more contextual and semantic meaning of words. Implementation of longformer can be tested as word embedding. Furthermore, a through comprehensive research can be carried out on this work by developing a resume ranking system on classifies resumes. A resume ranking system will enable the multi-class resume classification framework to sort and prioritize the resumes according to their ranking. Recruiting software companies will be benefiting from this as they can automatically select top candidates for further processing. In terms of resume classification, more research has to be performed to develop an optimal ranking system of resumes.

References

Peña, A., Serna, I., Morales, A., Fierrez, J., Ortega, A., Herrarte, A., Alcantara, M. and Ortega-Garcia, J., 2023. Human-centric multimodal machine learning: Recent advances and testbed on AI-based recruitment. *SN Computer Science*, *4*(5), p.434.

Xu, Q., Zhang, J., Zhu, Y., Li, B., Guan, D. and Wang, X., 2020, December. A block-level RNN model for resume block classification. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 5855-5857). IEEE.

Nasser, S., Sreejith, C. and Irshad, M., 2018, July. Convolutional neural network with word embedding based approach for resume classification. In *2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR)* (pp. 1-6). IEEE.

Ayishathahira, C.H., Sreejith, C. and Raseek, C., 2018, July. Combination of neural networks and conditional random fields for efficient resume parsing. In 2018 International CET Conference on Control, Communication, and Computing (IC4) (pp. 388-393). IEEE.

Habous, A., 2021. Combining Word Embeddings and Deep Neural Networks for Job Offers and Resumes Classification in IT Recruitment Domain. *International Journal of Advanced Computer Science and Applications*, *12*(7).

Jiechieu, K.F.F. and Tsopze, N., 2021. Skills prediction based on multi-label resume classification using CNN with model predictions explanation. *Neural Computing and Applications*, *33*, pp.5069-5087.

Ravichandiran, S., 2021. Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT. Packt Publishing Ltd.

Barducci, A., Iannaccone, S., La Gatta, V., Moscato, V., Sperlì, G. and Zavota, S., 2022. An end-to-end framework for information extraction from italian resumes. *Expert Systems with Applications*, *210*, p.118487.

James, V., Kulkarni, A. and Agarwal, R., 2022, December. Resume Shortlisting and Ranking with Transformers. In International Conference on Intelligent Systems and Machine Learning (pp. 99-108). Cham: Springer Nature Switzerland.

Wings, I., Nanda, R. and Adebayo, K.J., 2021. A context-aware approach for extracting hard and soft skills. Procedia Computer Science, 193, pp.163-172.

Vukadin, D., Kurdija, A.S., Delač, G. and Šilić, M., 2021. Information extraction from free-form CV documents in multiple languages. IEEE Access, 9, pp.84559-84575.

Tallapragada, V.S., Raj, V.S., Deepak, U., Sai, P.D. and Mallikarjuna, T., 2023, May. Improved Resume Parsing based on Contextual Meaning Extraction using BERT. In 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1702-1708). IEEE.

Liu, J., Shen, Y., Zhang, Y. and Krishnamoorthy, S., 2021, May. Resume parsing based on multi-label classification using neural network models. In Proceedings of the 6th International Conference on Big Data and Computing (pp. 177-185).

Xu, G., 2022, January. Human Resource Recommendation Based on Recurrent Convolutional Neural Network. In 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE) (pp. 54-58). IEEE.

Li, X., Shu, H., Zhai, Y. and Lin, Z., 2021, October. A method for resume information extraction using bert-bilstm-crf. In 2021 IEEE 21st International Conference on Communication Technology (ICCT) (pp. 1437-1442). IEEE.

Su, Y., Zhang, J. and Lu, J., 2019, December. The resume corpus: a large dataset for research in information extraction systems. In 2019 15th International Conference on Computational Intelligence and Security (CIS) (pp. 375-378). IEEE.