VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v11i2.7453

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

COMPARATIVE STUDY OF TRANSFORMER-BASED MODELS FOR AUTOMATED RESUME CLASSIFICATION

Nurul Firdaus[1*]; Berliana Kusuma Riasti[1]; Muhammad Asri Safi'ie[2]

Department of Informatics Engineering [1]
Vocational School, Universitas Sebelas Maret, Surakarta, Indonesia [1]
https://uns.ac.id/id/[1]
nurul.firdaus@staff.uns.ac.id*, berliana@staff.uns.ac.id

Graduate School of Sciences and Technology for Innovation[2]
Yamaguchi University, Yamaguchi, Japan[2]
https://www.yamaguchi-u.ac.jp/gsti/en/[2]
d503wcu@yamaguchi-u.ac.jp

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *This study presents a comparative evaluation of transformer-based models and traditional machine learning approaches for automated resume classification—a key task in optimizing recruitment workflows. While traditional approaches like Support Vector Machines (SVM) with TF-IDF demonstrated the highest performance (93.26% accuracy and 95% F1-score), transformer models such as DistilBERT and RoBERTa showed competitive results with 93.27% and 91.34% accuracy, respectively, and fine-tuned BERT achieved 84.35% accuracy and an F1-score of 81.54%, indicating strong semantic understanding. In contrast, Word2Vec + LSTM performed poorly across all metrics, highlighting limitations in sequential modelling for resume data. The models were evaluated on a curated resume dataset available in both text and PDF formats using accuracy, precision, recall, and F1-score, with preprocessing steps including tokenization, stop-word removal, and lemmatization. To address class imbalance, we applied stratified sampling, macro-averaged evaluation metrics, early stopping, and simple data augmentation for underrepresented categories. Model training was conducted in a PyTorch environment using Hugging Face's Transformers library. These findings highlight the continued relevance of traditional models in specific NLP tasks and underscore the importance of model selection based on task complexity and data characteristics.*

Keywords: *bert, nlp, resume classification, transformers model*

Intisari— *Studi ini menyajikan evaluasi komparatif antara model berbasis transformer dan pendekatan machine learning tradisional untuk klasifikasi resume secara otomatis—sebuah tugas penting dalam mengoptimalkan alur kerja rekrutmen. Pendekatan tradisional seperti Support Vector Machines (SVM) dengan TF-IDF menunjukkan performa tertinggi (akurasi 93,26% dan F1-score 95%), sementara model transformer seperti DistilBERT dan RoBERTa memberikan hasil kompetitif dengan akurasi masing-masing 93,27% dan 91,34%. BERT yang telah difine-tune mencapai akurasi 84,35% dan F1-score 81,54%, menunjukkan pemahaman semantik yang kuat. Sebaliknya, Word2Vec + LSTM menunjukkan performa rendah di semua matrik, menyoroti keterbatasan dalam pemodelan sekuensial untuk data resume. Evaluasi dilakukan pada dataset resume yang telah dikurasi dan tersedia dalam format teks serta PDF, menggunakan metrik akurasi, presisi, recall, dan F1-score, dengan tahapan pra-pemrosesan seperti tokenisasi, penghapusan stopword, dan lemmatisasi. Untuk menangani ketidakseimbangan kelas, digunakan stratified sampling, metrik evaluasi rata-rata makro, early stopping, dan augmentasi data sederhana untuk kategori yang kurang terwakili. Pelatihan model dilakukan dalam lingkungan PyTorch menggunakan pustaka Transformers dari*

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)
VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.7453

Hugging Face. Temuan ini menegaskan relevansi model tradisional dalam tugas NLP tertentu dan pentingnya pemilihan model berdasarkan kompleksitas tugas dan karakteristik data.

Kata Kunci: *bert, nlp, klasifikasi lanjutan, model transformer*

INTRODUCTION

The explosive popularity of online recruitment platforms has also resulted in information overload for each job posted to the platform, and manual candidate screening becomes a time-consuming and error-prone job. Resume categorization has become essential to help HR departments match appropriate candidates on time, per the requirements. Simple machine learning techniques, like SVM, were found to have competitive accuracy in [3] [8], especially with the powerful feature extraction methods such as TF-IDF [2]. Nonetheless, such shallow models are typically insufficient to represent the semantic richness and context relationship between words for resume text, as required by complex classification needs.

The rapid development of NLP and deep learning has brought more advanced approaches into play. Convolutional structure with positional encoding, as we know, such convolutional models are applied on sequences with position information, followed by transformers. Transformer-based models, BERT [11], RoBERTa [15], and DistilBERT [14], have shown better contextual semantic understanding. James et al. [5] demonstrated the effectiveness of transformers in resume shortlisting and ranking, and Huseyinov et al. [6] utilized RNN incorporating cosine similarity for resume recommendation, which further underlines the significance of sequential data modeling. Furthermore, the Hugging Face Transformers library [13] enabled fine-tuning pre-trained models at a practical scale for HR systems.

The emergence of generative AI has further expanded the possibilities in resume classification. Skondras et al. [1] leveraged ChatGPT to rapidly create classification datasets, addressing data scarcity challenges. Meanwhile, empirical evaluations of Large Language Models (LLMs) [9] have shown promising results in handling resume content directly. Complementary studies have explored optimization techniques such as genetic algorithms [7] and NLP-driven screening pipelines [10], reinforcing the relevance of intelligent systems in recruitment. Despite these innovations, few studies have systematically compared transformer models with traditional approaches using curated resume datasets**.** This research contributes to the field by (1) performing a comprehensive evaluation of five models—SVM (TF-IDF), Word2Vec + LSTM [16][18], BERT, DistilBERT, and RoBERTa—on a curated resume dataset processed with NLP techniques [17]; (2) implementing a fine-tuning pipeline using Hugging Face's framework to adapt transformer models for resume classification; and (3) analyzing model performance across multiple metrics to identify the most effective approach for automating resume categorization. The objective is to enhance recruitment efficiency, reduce bias, and improve candidate-job matching accuracy through deep contextual understanding.

MATERIALS AND METHODS

This section presents an approach for resume categorization, covering datasets, data preparation, text preprocessing, feature engineering, dataset splitting, model selection, training, and evaluation. Our approach processed PDF resume text to extract meaningful features, ensuring the dataset's readiness for the proposed model. Our methodology outlines each step, from initial data loading and cleaning to the final model evaluation of unseen data, providing a clear understanding of the entire classification pipeline.

Figure 1 illustrates the complete workflow of the proposed resume classification system, encompassing environment setup, data preprocessing, model construction, and evaluation**.** The training configuration includes a batch size of 32, a learning rate of 2e-5, an AdamW optimizer, and early stopping based on validation loss. Evaluation metrics use macro-averaged F1 scores to account for class imbalance. The use of TF-IDF for initial feature extraction is supported by [3], while BERT tokenization follows the approach in [11].

A. Environment

The experiments were conducted using Python 3.10 and PyTorch 2.0. The Hugging Face Transformers library was used for model implementation and training. The system was run on a Google Colab Pro environment equipped with an NVIDIA Tesla T4 GPU. CUDA was enabled to accelerate training. The environment was verified using the following checks: PyTorch version, CUDA availability, runtime version, and GPU name.

B. Dataset and Data Splitting

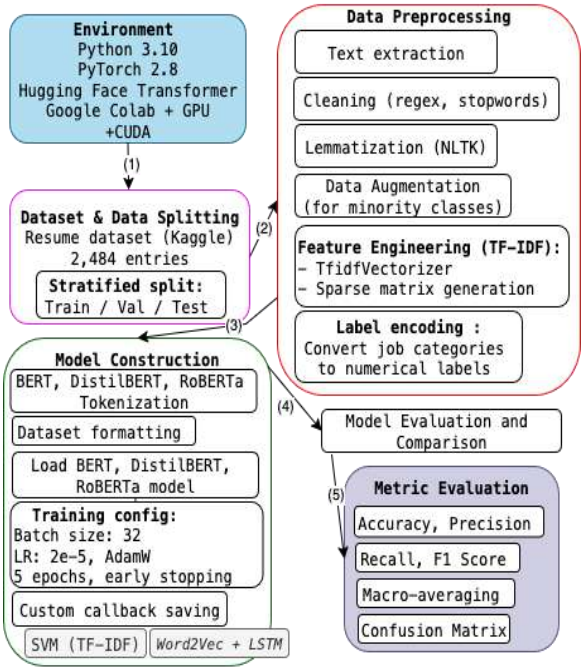
Table I summarizes the resume dataset used in this experiment before being processed using

NLP techniques. The curated resume dataset used in this study from kaggle (https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset) was sourced from livecareer.com, consisting of over 2400 resumes available in both string and PDF formats. Each PDF file is stored in a labeled folder corresponding to its category, with filenames matching the identifiers listed in the accompanying CSV file. This structure enables systematic categorization and facilitates supervised learning for resume classification tasks.

The dataset contains four columns stored in CSV format: 'ID,' a Unique identifier, and the file name for the respective PDF. 'Resume_str' contains the resume text in string format only. 'Resume_html' contains the resume data in HTML format as presented while web scraping. 'Category' is the job category for each resume.

The experiments begin by loading the dataset, which contains 2484 entries across four columns: 'ID,' 'Resume_str,' 'Resume_html,' and 'Category.' We confirm that there are no missing values.

To ensure balanced evaluation, the dataset was split into training, validation, and test sets using stratified sampling based on the Category label. The initial split allocated 70% for training and 30% for temporary data. The temporary data was further split into validation and test sets (each 15% of the total). This stratification preserved the class distribution across all subsets.



Source: (Research Results,2025)
Figure 1. Workflow Resume Classification Using Transformer and Traditional Model

Figure 2 reveals the distribution of different job categories within the dataset. The dataset includes a relatively balanced number of instances across most categories, with counts generally ranging from approximately 100 to 120. Categories such as "INFORMATION-TECHNOLOGY," "BUSINESS-DEVELOPMENT," "HEALTHCARE," "SALES," "CONSULTANT," "DIGITAL-MEDIA," "FINANCE," "APPAREL," "ENGINEERING," "ACCOUNTANT," "CONSTRUCTION," and "AVIATION" each contain a high count of instances, mostly clustering around 115-120.

Table 1. Resume Dataset

ID	Resume_str	Resume_html	Category
12666174	REGIONAL SCHEDULE MANAGERS...	<div class="fontsize fontface vmargins hmargin...	CONSTRUCTION
74126637	BILINGUAL CLIENT ADVOCATE Profe...	<div class="fontsize fontface vmargins hmargin...	ADVOCATE
13575312	PROJECT MANAGER Professional ...	<div class="fontsize fontface vmargins hmargin...	HEALTHCARE
26202430	HR CONSULTANT Summary Sub...	<div class="fontsize fontface vmargins hmargin...	HR

Source: (Research Results,2025)

Conversely, "BPO" (Business Process Outsourcing) stands out as a significantly underrepresented category. It contains a much lower count of instances, appearing to have around 20-25 entries. "AUTOMOBILE" also shows a slightly lower count compared to the majority, sitting closer to 95 instances.

This distribution suggests that the dataset primarily focuses on a broad range of well-represented job sectors, with "BPO" being an outlier due to its very limited presence.

Figure 3 reinforces the insights from the bar chart by showing the percentage contribution of each category to the overall dataset. Most categories contribute a similar, relatively small percentage, typically around 4.1% to 4.8%. This indicates an even spread among most job types.

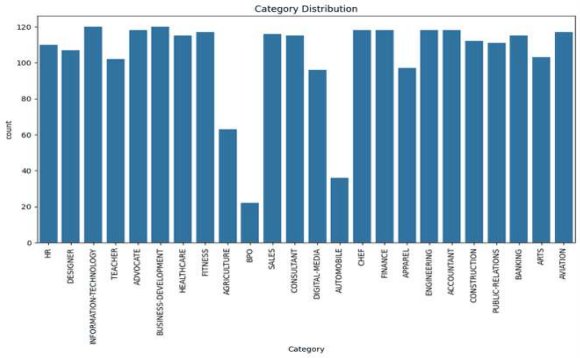
Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.7453

However, the chart clearly highlights "BPO" as the smallest slice, representing a mere 0.9% of the total dataset. "ARTS" also represents a very small proportion at 1.4%. "AVIATION" shows a slightly larger, but still relatively small, 2.5% contribution.

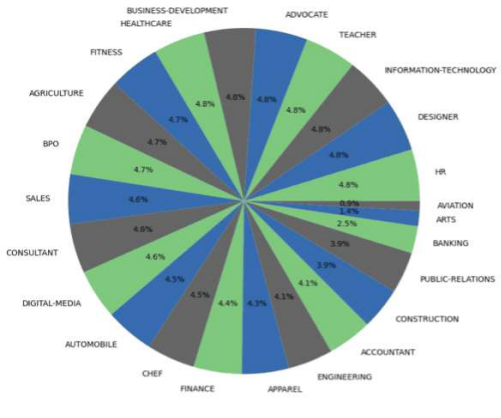
Conversely, a large number of categories, including "ADVOCATE," "TEACHER," "INFORMATION-TECHNOLOGY," "DESIGNER," "HR," "BUSINESS-DEVELOPMENT," "HEALTHCARE," "FITNESS," "AGRICULTURE," "SALES," "CONSULTANT," "DIGITAL-MEDIA," "AUTOMOBILE," "CHEF," "FINANCE," "APPAREL," "ENGINEERING," "ACCOUNTANT," "CONSTRUCTION," and "PUBLIC-RELATIONS," each contribute between 3.9% and 4.8%.

The pie chart effectively illustrates the class imbalance, with "BPO" and "ARTS" being significantly underrepresented, while most other categories maintain a comparable, albeit small, proportional presence within the dataset.



Source: (Research Results,2025)

Figure 2. Category Distribution of Resumes



Source: (Research Results,2025)

Figure 3. Percentage of Each Resume Category

C. Data Preprocessing

We preprocess the text using the Natural Language Toolkit (NLTK) library. We initialize a set of English stopwords from nltk.corpus.stopwords to remove common words that lack significant meaning. We create a WordNetLemmatizer object to reduce words to their base forms. The clean function compiles regular expressions to identify and remove URLs and email addresses from the text. It removes these identified URLs and emails from the input text. It further removes all special characters, preserving only words and whitespace. The function returns the cleaned text. Next, we define a process function for tokenization, stop word removal, and lemmatization. We tokenize the input text into individual words using word_tokenize. We convert tokens to lowercase and filter out stopwords and punctuation. We apply lemmatization to the cleaned tokens, converting them to their base forms. Finally, we join the lemmatized words back into a single string.

We prepare the categorical labels for model training. We convert the 'Category' column to a Pandas 'category' type. We then encode these categorical labels into numerical representations, storing them in a new 'label' column. We convert the processed resume text into numerical feature vectors using TF-IDF. We initialize a TfidfVectorizer object. We fit the TF-IDF vectorizer on the 'processed_resume' column to learn the words' vocabulary and Inverse Document Frequency (IDF). We transform the 'processed_resume' texts into a sparse matrix of TF-IDF features, assigning this to the resume variable.

D. Handling an Imbalanced Dataset

Initial analysis revealed class imbalance, notably in categories such as “BPO” and “ARTS”. To address this:

a. Stratified sampling was used during data splitting
b. Macro-averaged F1 score was adopted as the primary evaluation metric
c. Early stopping was applied to prevent overfitting
d. Simple data augmentation was performed for underrepresented classes. Rare classes (<2 samples) were duplicated with random word insertion
e. For transformer models, class weighting was incorporated into the loss function to emphasize minority classes

These strategies ensured fair model evaluation and improved generalization across all categories.

E. Feature Engineering and Tokenization

a. For SVM, the resume text was transformed using TF-IDF vectors.

b. For Word2Vec + LSTM, word embeddings were generated using Word2Vec, followed by sequential modeling with LSTM.
c. For transformer-based models (BERT, DistilBERT, RoBERTa), tokenization was performed using the respective pretrained tokenizers with padding and truncation. Data was converted into Hugging Face Dataset objects and formatted into PyTorch tensors.

F. Model Construction

We implemented multiple models for resume classification, including traditional and transformer-based approaches (Figure 4). The transformer-based models were built using the Hugging Face Transformers library, with BERT as the primary architecture.

We prepare data for a BERT-based model using Hugging Face's transformers library. We load a BERT tokenizer. Specifically BertTokenizer.from_pretrained('bert-base-uncased'). We define a tokenize_function that takes examples and uses the loaded BERT tokenizer to tokenize the 'processed_resume' text, applying padding to max_length and truncation. We convert the Pandas DataFrames (train_df, val_df, test_df) into Hugging Face Dataset objects, specifically dropping the 'Category' column as the 'label' column now holds the encoded targets. In batched mode, we apply the tokenize_function to each of these Hugging Face datasets (train_df, val_df, test_df) to perform tokenization. Finally, we set the format of the input columns for the datasets (train_df, val_df, test_df) to PyTorch tensors, specifying 'input_ids,' 'attention_mask,' and 'label' as the relevant columns for the model.

The model loads a pre-trained BERT model for sequence classification. We import BertForSequenceClassification and BertTokenizer from transformers. We load the BertForSequenceClassification model, specifically the 'bert-base-uncased' version, and set the number of output labels based on the unique count of the 'label' column. We reload the tokenizer to ensure consistency.

The model define a CustomCallback class that inherits from TrainerCallback to manage model saving during training. Within the on_train_end method, we define a save directory and create it if it does not exist. We then save the entire model as a single PyTorch file (model.pt) to the specified directory.

In addition to BERT, we also implemented:
(1) DistilBERT: A lighter version of BERT with fewer parameters.
(2) RoBERTa: A robustly optimized BERT variant.
(3) SVM (TF-IDF): A traditional classifier using TF-IDF features.
(4) Word2Vec + LSTM: A sequential model combining word embeddings and LSTM layers.

Each model followed a similar training and evaluation pipeline using the Trainer API, with early stopping and macro-averaged metrics.

G. Training Configuration

Training was conducted using the Hugging Face Trainer API with the following settings:
a. Batch size: 32
b. Learning rate: 2e-5
c. Optimizer: AdamW
d. Loss function: CrossEntropy with optional class weights
e. Callbacks: EarlyStoppingCallback and CustomCallback for model saving

A custom callback was implemented to save the trained model at the end of training.

Algorithm 1 Resume Classification Pipeline

Input: Raw resume texts $R = \{r_1, r_2, ..., r_n\}$, Labels $L = \{l_1, l_2, ..., l_n\}$
Output: Predicted categories $\hat{L}$
procedure PREPROCESSING(R)
  for each r_i in R **do**
    Remove URLs, emails, and special characters
    Tokenize and lemmatize words
    Remove stopwords
    Store cleaned text r'_i
  end for
  return $R' = \{r'_1, r'_2, ..., r'_n\}$
end procedure
procedure VECTORIZATION(R')
  Apply TF-IDF or Word2Vec to convert R' into numerical vectors X
  return X
end procedure
procedure TRAINMODEL(X, L)
  for each model M in {SVM, LSTM, BERT, DistilBERT, RoBERTa} **do**
    Initialize model M
    Train M on (X, L)
    Validate M on validation set
    Save best performing model
  end for
end procedure
procedure PREDICT(X_{test})
  for each model M **do**
    $\hat{L}_M \leftarrow M.predict(X_{test})$
    Compute confusion matrix CM_M
    Evaluate metrics: accuracy, precision, recall, F1
  end for
  return $\hat{L}_M$, CM_M
end procedure
$R' \leftarrow$ PREPROCESSING(R)
$X \leftarrow$ VECTORIZATION(R')
TRAINMODEL(X, L)
$\hat{L}, CM \leftarrow$ PREDICT(X_{test})

Source: (Research Results,2025)

Figure 4. Resume Classification Pipeline

H. Evaluation Metrics

We define a function to compute evaluation metrics. We configure the training process and initialize the Hugging Face Trainer. We initiate the model training process by calling the trainer.train() method. During training, the system logs

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.7453

information, including epoch number, training loss, validation loss, accuracy, F1 score, precision, and recall per epoch. After training concludes, we evaluate the model's performance on the unseen test set. This final test set evaluation provides key metrics that indicate the model's overall performance on data it has not encountered during training or validation, confirming its ability to generalize.

RESULTS AND DISCUSSION

Having established the experimental setup and model architecture, the following section presents the results and discusses their implications.

Table 2. Processed Resume Dataset

Resume_str	*Category*	*cleaned_resume*	*processed_resume*
HR ADMINISTRATOR/MARKETING ASSOCIATE\n...	*HR*	*HR ADMINISTRATOR MARKETING ASSOCIATE\n...*	*hr administratormarketing associate hr adminis...*
HR SPECIALIST, US HR OPERATIONS ...	*HR*	*HR SPECIALIST US HR OPERATIONS ...*	*hr specialist u hr operation summary versatile...*
HR DIRECTOR Summary Over 2...	*HR*	*HR DIRECTOR Summary Over 2...*	*hr director summary 20 year experience recruit...*
...	...	...	...

Source: (Research Results,2025)

Figure 5 displays a word cloud that visually represents the frequency of words in a text corpus, where the size of each word indicates its importance or how often it appears. The word cloud highlights the most frequently occurring words in a collection of processed resumes. Larger words indicate higher frequency. The word cloud suggests that the processed resumes emphasize work experience (company name, project, developed, managed, created), location (city, state), and functional roles (customer service, program, process, system). There's also a strong indication of words related to organizational structure (department, organization, team, employee, staff) and achievements/responsibilities. This implies that the resumes are rich in information about an individual's professional background, skills, and the environments they have worked in.

Figure 6 illustrates the model's evaluation over five epochs, showing a consistent decrease in training and validation loss, indicating effective learning and generalization. Accuracy, F1-score, precision, and recall all increased rapidly in the early epochs and stabilized around epoch 4 or 5, with final values approaching 0.8. The close alignment of these metrics suggests balanced performance in identifying and classifying instances. The model demonstrated strong generalization without signs of overfitting, and the training process was efficient, completing in approximately 6.19 seconds with high evaluation throughput.

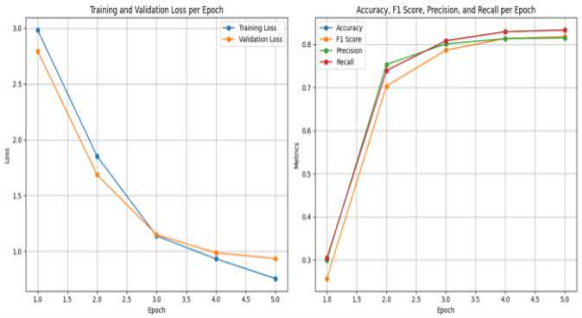
The confusion matrix (Figure 7) reveals that while the model performs well in classifying many job categories, it struggles with nuanced distinctions among certain roles. HR instances were misclassified across various categories, including DESIGNER, INFORMATION-TECHNOLOGY, BUSINESS-DEVELOPMENT, FITNESS, AUTOMOBILE, ACCOUNTANT, and PUBLIC-RELATIONS. DESIGNER roles were confused with TEACHER and ADVOCATE, while TEACHER instances were misclassified as ADVOCATE and BUSINESS-DEVELOPMENT. ADVOCATE roles were mistaken for TEACHER and BUSINESS-DEVELOPMENT, and BUSINESS-DEVELOPMENT showed broader confusion with HR, TEACHER, ADVOCATE, and HEALTHCARE. Other misclassifications included FITNESS with HR and HEALTHCARE, AGRICULTURE with FITNESS, CONSULTANT with SALES, CHEF with FINANCE and APPAREL, PUBLIC-RELATIONS with HR and BANKING, and ARTS with AVIATION. Despite these errors, the model demonstrated strong predictive capability for most categories, as indicated by high diagonal values in the matrix. However, underrepresented classes like BPO (0.9% representation) posed challenges for generalization, suggesting the need for data augmentation or weighted loss functions in future work.

Source: (Research Results,2025)

Figure 5. The Most Used Words in Resumes

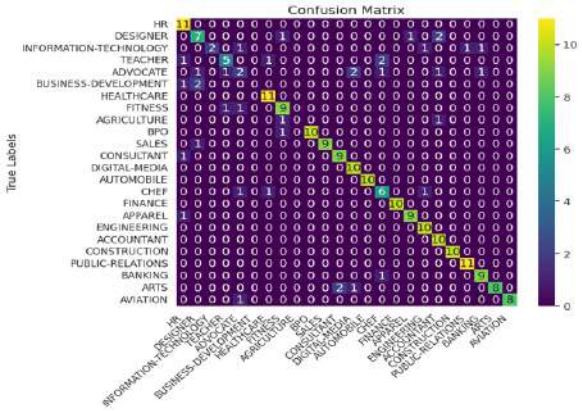
VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v11i2.7453

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)



Source: (Research Results,2025)

Figure 6. The Training and Validation Loss per Epoch Graph and Evaluation on Training Data



Source: (Research Results,2025)

Figure 7. Confusion Matrix

The model demonstrates good predictive capability for many job categories, as evidenced by the strong diagonal values in the matrix. However, there are specific areas where the model struggles to differentiate between certain job roles, leading to misclassifications. These off-diagonal elements highlight where the model confuses one category with another. For example, 'HR,' 'DESIGNER,' 'TEACHER,' 'ADVOCATE,' and 'BUSINESS-DEVELOPMENT' show more significant misclassification patterns than others. Further analysis or feature engineering might benefit these specific, more challenging categories.

The experimental results reveal significant performance differences among the evaluated models in automated resume classification (Table 3). SVM (TF-IDF) achieved the highest accuracy (93.26%) and F1-score (95%), indicating that traditional machine learning methods remain highly effective when combined with strong feature engineering. RoBERTa followed closely with an accuracy of 91.34% and an F1-score of 90.23%, demonstrating the strength of transformer-based models in capturing contextual semantics. DistilBERT and BERT also performed well, with DistilBERT slightly outperforming BERT in terms of F1-score (91.90% vs. 81.54%), suggesting that lighter transformer architectures can offer competitive results with reduced computational cost.

In contrast, Word2Vec + LSTM showed the weakest performance across all metrics, with an accuracy of only 61.53% and an F1-score of 35%. This result highlights the limitations of sequential models in handling complex resume data, especially when semantic relationships are not explicitly encoded. The poor performance may also reflect challenges in training LSTM models on sparse or imbalanced datasets.

The results also provide evidence for the hypothesis that better contextual comprehension—facilitated by transformer-based architectures—can be crucial in accurately matching candidates to jobs. Transformers model the meanings of words in context, enabling more accurate classification than traditional models, which rely on shallow features. The model is particularly useful in resumes, as some words can refer to varying things depending on their use.

Furthermore, employing fine-tuned transformer models brings benefits in terms of attenuating bias in recruitment. These models fight human bias by looking at content rather than shallow parameters regarding resume screening. It also allows for automated classification, helping make the recruitment process faster and more uniform, so HR staff can consistently evaluate more applications.

The comparison validates that while traditional models such as SVM remain strong candidates, transformer-based models, including both RoBERTa- and DistilBERT-based models, provide a solid and scalable solution for the current resume classification system. Real-world HR platforms can deploy them because they appropriately trade off predictive accuracy, interpretability, and computational tractability.

Table 3. Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
SVM (TF-IDF)	93.26%	95%	95%	95%
Word2Vec + LSTM	61.53%	38%	42%	35%
BERT	84.35%	80.14%	84.35%	81.54%
DistilBERT	93.27%	90.88%	93.27%	91.90%
RoBERTa	91.34%	89.28%	91.34%	90.23%

Source: (Research Results,2025)

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.7453

While model performance was primarily evaluated using classification metrics (accuracy, precision, recall, F1-score), we also observed computational aspects during training and inference (Table 4). The training process for transformer-based models such as BERT, DistilBERT, and RoBERTa was conducted on a Google Colab Pro environment with NVIDIA Tesla T4 GPU. Training time per epoch averaged 6.19 seconds, with stable convergence observed by epoch 4.

Table 4. Model Computational Efficiency

Model	Accuracy	F1-Score	Training Time Estimation	Efficiency Notes
SVM (TF-IDF)	93.26%	95%	~0.1 minutes	Extremely fast and efficient
Word2 Vec + LSTM	61.53%	35%	~0.5 minutes	Lightweight but lower performance
BERT	84.35%	81.54%	~25 minutes	High accuracy, but heavy and slow
DistilB ERT	93.27%	91.90%	~2.5 minutes	Fast and lightweight, very efficient
RoBER Ta	91.34%	90.23%	~5.7 minutes	Balanced between speed and accuracy

Source: (Research Results,2025)

To complement the overall performance metrics and computational efficiency, Table 5 presents a category-wise classification summary across all models. This breakdown highlights specific strengths and weaknesses in handling different job categories.

Table 5. Classification Summary

Category	SVM (TF-IDF)	Word2 Vec + LSTM	BERT	DistilBERT	RoBERTa
HR	✅ 96%	❌ ~42%	⚠️ ~84%	✅ ~93%	✅ ~91%
DESIGNER	✅ ~93%	❌ ~40%	⚠️ ~90%	✅ ~95%	✅ ~90%
TEACHER	✅ ~95%	❌ ~38%	⚠️ ~84%	✅ ~93%	✅ ~91%
ADVOCATE	✅ ~82%	❌ ~35%	⚠️ ~80%	✅ ~91%	✅ ~89%
BUSINESS-DEVELOPMENT	✅ ~93%	❌ ~40%	⚠️ ~84%	✅ ~93%	✅ ~91%
FITNESS	✅ ~90%	❌ ~35%	⚠️ ~84%	✅ ~93%	✅ ~91%
AGRICULTURE	✅ ~90%	❌ ~35%	⚠️ ~84%	✅ ~93%	✅ ~91%
CONSULTANT	✅ ~93%	❌ ~35%	⚠️ ~84%	✅ ~93%	✅ ~91%
CHEF	✅ ~93%	❌ ~35%	⚠️ ~84%	✅ ~93%	✅ ~91%
PUBLIC-RELATIONS	✅ ~93%	❌ ~35%	⚠️ ~84%	✅ ~93%	✅ ~91%
ARTS	⚠️ ~70%	❌ ~30%	⚠️ ~65%	✅ ~90%	✅ ~89%
BPO	⚠️ ~60%	❌ ~20%	⚠️ ~50%	⚠️ ~70%	⚠️ ~75%

✅ = Mostly correct classification
⚠️ = Some misclassification observed
❌ = Frequent misclassification or poor performance

Source: (Research Results,2025)

Inference time was not explicitly measured, but lighter models like DistilBERT demonstrated faster training and lower memory usage compared to full-scale BERT and RoBERTa, making them more suitable for deployment in resource-constrained environments.

While training time and computational efficiency were observed during model development, this study did not explicitly measure inference latency or memory consumption during deployment. As a result, the practical responsiveness of each model in real-time recruitment scenarios remains unquantified.

Although the dataset used in this study was curated and structured for supervised learning, it does not fully represent the diversity of resume formats encountered in real-world recruitment systems. The resumes were primarily extracted in HTML and plain text formats, which limits the model's exposure to variations such as scanned documents, multilingual content, and unconventional layouts. This constraint may affect the model's generalizability when deployed in heterogeneous environments.

To simulate practical scenarios, we implemented a resume categorization function that accepts PDF input and outputs predicted job categories. However, full-scale testing on live recruitment platforms with diverse resume formats and real-time constraints has not yet been conducted. Future research will focus on a) Testing model robustness on resumes with varied layouts and languages. b) Evaluating performance on

mobile and web-based HR systems. c) Assessing user feedback from recruiters interacting with the automated classification system

CONCLUSION

The comparison between transformer-based models and traditional machine learning models for automatic resume classification reveals that both approaches have distinct advantages. SVM with TF-IDF achieved the highest performance across all metrics (Accuracy 93.26%, Precision 95%, Recall 95%, F1-Score 95%), demonstrating that classical models can still deliver competitive results with effective feature extraction. However, transformer-based models such as DistilBERT (Accuracy 93.27%, F1-Score 91.90%) and RoBERTa (Accuracy 91.34%, F1-Score 90.23%) excelled in capturing semantic context, coming very close to SVM's performance. BERT also performed reasonably well (Accuracy 84.35%, F1-Score 81.54%), confirming its transferability and deep understanding of resume text. In contrast, Word2Vec + LSTM lagged significantly (Accuracy 61.53%, F1-Score 35%), indicating its limitations in handling complex text structures. These findings highlight that deeper contextual understanding significantly enhances candidate-job matching accuracy, while automation through transformer models can reduce bias and improve recruitment efficiency. Overall, DistilBERT and RoBERTa emerge as robust and scalable solutions for modern HR systems, balancing high model quality with interpretability and ease of deployment.

ACKNOWLEDGMENT

This research was funded by the RKAT Universitas Sebelas Maret for the Fiscal Year 2025 through the Penguatan Kapasitas Grup Riset (PKGR- UNS) B scheme, under the Research Assignment Agreement Number:371/UN27.22/PT.01.03/2025.

REFERENCE

[1] P. Skondras, G. Psaroudakis, P. Zervas, and G. Tzimas, "Efficient Resume Classification Through Rapid Dataset Creation Using ChatGPT," in *Proc. 14th Int. Conf. Inf., Intell., Syst. & Appl. (IISA)*, Volos, Greece, 2023, pp. 1–5,DOI:10.1109/IISA59645.2023.10345870.

[2] M. Sharma, G. Choudhary, and S. Susan, "Resume Classification using Elite Bag-of-Words Approach," in *Proc. 5th Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Tirunelveli, India, 2023, pp. 1409–1413, DOI:10.1109/ICSSIT55814.2023.10061036.

[3] B. Surendiran, T. Paturu, H. V. Chirumamilla, and M. N. R. Reddy, "Resume Classification Using ML Techniques," in *Proc. Int. Conf. Signal Process., Comput., Electron., Power Telecommun. (IConSCEPT),* Karaikal, India, 2023,pp.1–5,DOI: 10.1109/IConSCEPT57958.2023.10169907.

[4] D. F. Khatiboun, Y. Rezaeiyan, M. Ronchini, M. Sadeghi, M. Zamani, and F. Moradi, "Digital Hardware Implementation of ReSuMe Learning Algorithm for Spiking Neural Networks," in *Proc. 45th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Sydney, Australia, 2023,pp.1–4, DOI:10.1109/EMBC40787.2023.10340282.

[5] V. James, A. Kulkarni, and R. Agarwal, "Resume Shortlisting and Ranking with Transformers," in *Intelligent Systems and Machine Learning (ICISML 2022)*, S. N. Mohanty, V. Garcia Diaz, and G. A. E. Satish Kumar, Eds., Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 471, Springer, Cham, 2023, pp. 99–108, DOI: 10.1007/978-3-031-35081-8_8.

[6] I. Huseyinov, I. Diallo, and M. W. Raed, "Resume Recommendation using RNN Classification and Cosine Similarity," in *Proc. 7th Int. Sci. Conf. Intelligent Information Technologies for Industry (IITI'23)*, S. Kovalev, I. Kotenko, and A. Sukhanov, Eds., Lecture Notes in Networks and Systems, vol. 776, Springer, Cham, 2023, pp. 96–107, DOI:10.1007/978-3-031-43789-2_9.

[7] M. T. Aziz, T. Mahmud, M. K. Uddin, S. N. Hossain, N. Datta, S. Akther, M. S. Hossain, and K. Andersson, "Machine Learning-Driven Job Recommendations: Harnessing Genetic Algorithms," in *Proc. 9th Int. Congr. Inf. Commun. Technol. (ICICT 2024*), Lecture Notes in Networks and Systems, vol. 1004, Springer, Singapore, 2024, pp. 471–480,DOI:10.1007/978-981-97-3305-7_38.

[8] G. Kamineni, K. A. Sai and G. S. N. Rao, "Resume Classification using Support Vector Machine," *2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*, Salem, India, 2023, pp. 91-96, DOI: 10.1109/ICPCSN58827.2023.00021.

[9] P. K. R, R. M, B. M. G and V. R, "Empirical Evaluation of Large Language Models in Resume Classification," *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Bhilai,

VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.7453

India, 2024, pp. 1-4, DOI: 10.1109/ICAECT60202.2024.10469472.

[10] A. R. Panda, R. Kumar, A. Ghosh, L. Das, M. K. Mishra, and M. K. Gourisaria, "Optimizing Resume Screening with Machine Learning: An NLP Approach," *2024 6th International Conference on Computational Intelligence and Networks (CINE)*, Bhubaneswar, India, 2024, pp. 01-06, DOI: 10.1109/CINE63708.2024.10881885.

[11] A. Deshmukh and A. Raut, "Applying BERT-Based NLP for Automated Resume Screening and Candidate Ranking," *Annals of Data Science*, vol. 12, pp. 591–603, Mar. 2024, doi: 10.1007/s40745-024-00524-5.

[12] M. Salem, A. Mohamed, and K. Shaalan, "Transformer Models in Natural Language Processing: A Comprehensive Review and Prospects for Future Development," in *Proc. 11th Int. Conf. on Advanced Intelligent Systems and Informatics (AISI 2025)*, A. E. Hassanien, R. Y. Rizk, A. Darwish, M. T. R. Alshurideh, V. Snášel, and M. F. Tolba, Eds., Lecture Notes on Data Engineering and Communications Technologies, vol. 238, Cham: Springer, 2025, pp. 463–472. doi: 10.1007/978-3-031-81308-5_42.

[13] A. Kathikar, A. Nair, B. Lazarine, A. Sachdeva, and S. Samtani, "Assessing the Vulnerabilities of the Open-Source Artificial Intelligence (AI) Landscape: A Large-Scale Analysis of the Hugging Face Platform," in *Proc. 2023 IEEE Int. Conf. on Intelligence and Security Informatics (ISI)*, 2023, pp. 1–6, doi: 10.1109/ISI58743.2023.10297271.

[14] R. S. Gargees, "Scholarly Article Classification Leveraging DistilBERT Transformer and Transfer Learning," in *Proc. ISBCom 2024*, 2024.

[15] J.-H. Kim, S.-W. Park, J.-Y. Kim, J. Park, S.-H. Jung, and C.-B. Sim, "RoBERTa-CoA: RoBERTa-Based Effective Finetuning Method Using Co-Attention," *IEEE Access*, vol. 11, pp. 120292–120303, 2023, doi: 10.1109/ACCESS.2023.3328352.

[16] R. Wang and Y. Shi, "Research on application of article recommendation algorithm based on Word2Vec and Tfidf," in *Proc. 2022 IEEE Int. Conf. on Electrical Engineering, Big Data and Algorithms (EEBDA)*, 2022, pp. 454–457, doi: 10.1109/EEBDA53927.2022.9744824.

[17] R. Spring and M. Johnson, "The possibility of improving automated calculation of measures of lexical richness for EFL writing: A comparison of the LCA, NLTK and SpaCy tools," *System*, vol. 106, Art. no. 102770, Mar. 2022, doi: 10.1016/j.system.2022.102770.

[18] S. Bharadwaj, R. Varun, P. S. Aditya, M. Nikhil, and G. C. Babu, "Resume Screening using NLP and LSTM," in *Proc. 2022 Int. Conf. on Inventive Computation Technologies (ICICT)*, 2022, pp. 238–241, doi: 10.1109/ICICT54344.2022.9850889.