Inteligencia Artificial 24(74), 117-132
doi: 10.4114/intartif.vol24iss74pp117-132

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Uniqueness meets Semantics: A Novel Semantically Meaningful Bag-of-Words Approach for Matching Resumes to Job Profiles

Seba Susan[A], Muskan Sharma, Gargi Choudhary
Department of Information Technology, Delhi Technological University, Delhi, India.
[A]seba_406@yahoo.in

Abstract In an increasingly competitive world, the automated screening of resumes of job applicants is the need of the hour given the large numbers of such resumes in career portals on the World Wide Web. Resume classification is a subset of the document classification problem in which the keywords extracted from the resume document play a significant role in determining the job profile. In this paper, we explore the combination of the concept of the uniqueness of a keyword based on its higher number of occurrences in a particular resume class and the concept of semantics by representing the unique keywords using word embeddings that capture semantic similarities between resume documents. The aim is to extract semantic representations of only those keywords that occur more frequently in one class than in any other class. The unique elite keywords, as they are called, are extracted from each resume document and passed as input to a Bidirectional long short-term memory (BiLSTM) for classification. Our experiments on two publicly available resume datasets, with distinctive job profiles, prove that the proposed approach outperforms the state of the art by a significant margin, establishing the efficacy of our approach. Our code is made available online at: https://github.com/Muskankalonia/Unique-Elite-Bag-of-Words-Approach-On-Resume-Classification

Keywords: Resume classification, document, keyword, uniqueness, semantics, word embeddings, term frequency

1 Introduction

Classifying documents based on the keywords they contain is a well-researched problem [1, 2, 3, 4, 5]. Two documents are assumed to belong to the same category if they have high mutual keyword content, as asserted by Heaps way back in 1973 [1]. The choice of the keywords that represent the text document, therefore, has a crucial role to play in the classification performance [2]. Keywords are filtered based on their correlation with the class label [3] or simply on the basis of their frequencies of occurrences in the documents [4]. One-hot encoding, term frequency (TF), and term frequency-inverse document frequency (TF-IDF) are examples of bag-of-words (BoW) representations of the text in a document [5]; these are then classified using machine learning algorithms like the artificial neural network or decision trees. In the BoW approach, the keywords are compared simply on the basis of their occurrences and not on their actual meanings. On the other hand, the use of semantically meaningful word embeddings such as GloVe [6] and Word2Vec [7] facilitates semantic similarity matching between documents for text classification [8, 9]. Recurrent neural networks such as the Long Short-Term Memory (LSTM) [10] or transformers [11] are typically used to extract useful information from the sequence of word embeddings emanating from each document [12].

Resumes are documents prepared by applicants to a job, that contain their personal details, educational qualifications, skills pertaining to the job being applied for, and work experience. Resume classification [13, 14] is a subset of the document classification problem. The classes or categories in resume datasets represent different job profiles such as accountant, teacher etc. that have different educational and skill set requirements [15]. Fig. 1

ISSN: 1137-3601 (print), 1988-3064 (on-line)

illustrates the process flow of resume classification, presented from the perspective of a document classification task. The steps comprise of text-preprocessing followed by feature extraction (BoW or word embeddings) and classification (machine learning classifiers or LSTM/transformers).

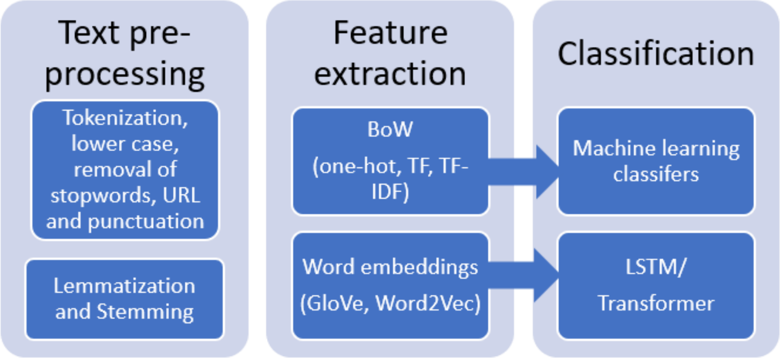

Text pre-processing
Tokenization, lower case, removal of stopwords, URL and punctuation
Lemmatization and Stemming
Feature extraction
BoW (one-hot, TF, TF-IDF)
Word embeddings (GloVe, Word2Vec)
Classification
Machine learning classifers
LSTM/ Transformer

Figure 1. Process flow for the resume classification task

Resumes typically contain some highly suggestive keywords that are related to a particular job profile. These may be keywords associated with the educational qualification, skill set or job experience of the candidate. It is therefore important to identify and filter out the most discriminative keywords that would help to distinguish between resume categories. The identification of keywords that are unique to a particular resume class would, therefore, serve as a useful indicator of the job category. The authors in [16] introduced the concept of uniqueness while shortlisting keywords by counting their relative occurrences across classes and computing keyword entropy across classes. A low entropy across classes indicates a significant keyword which has high frequencies of occurrences in one class, and low frequencies of occurrences in other classes. We further explore this concept in our current work where we concentrate on the specific case study of resume classification. We first identify the "unique elite" keywords that are unique to a particular resume class in which they occur in relatively higher numbers as compared to other resume classes. The unique elite keywords are extracted and concatenated across all classes after removing redundancy. Furthermore, we propose the use of GloVe and Word2Vec word embeddings to capture the semantics of the unique elite keywords extracted from each resume document. Word embeddings are feature vectors laid out in "meaning space" such that words similar in meaning lie near each other in the embedding space. The objective is to create semantically meaningful representations of selected or filtered keywords in a resume document that would aid in effective classification. The rest of this paper is organized as follows. Section 2 reviews some related work in literature, section 3 presents the methodology followed in our experiments, section 4 discusses the experimentation and the results, and the paper is concluded in section 5.

2 Related work

A scrutiny of literature on resume classification reveals that there are two main approaches adopted to perform the task: - (1) BoW approach (2) word embeddings. The BoW approach is a most popular feature extraction technique, that includes one-hot encoding, TF, and TF-IDF. Supervised machine learning models are trained on these features for classifying resumes to different categories. Ali et al. in a recent work [13] compared the performance of support vector machine (SVM), Naïve Bayes, K-Nearest Neighbor and Logistic regression classifiers for resume classification using TF-IDF feature vectors. The authors in [17] used a combination of TF-IDF and machine learning classifiers to implement a resume recommendation system. The linear SVM classifier outperformed the logistic regression, Naïve Bayes and random forest classifiers in the classification task. Cosine similarity was used to rank the resumes as per the job profile.

Ramraj and Sivakumar proved that the combination of TF-IDF with character-level convolutional neural networks yielded better results than that of the conventional machine learning algorithms [18]. LinkedIn profile descriptions were used to screen the resumes in this case. Use of N-grams was found to boost the accuracy of text classification in [19]. The authors in [20] found that the combination of unigrams with bigrams achieved the best performance using the random forest classifier. The BoW approach, however, does not capture the semantic similarity between keywords among resume documents since it is based on the morphological form of the word and not on its meaning. The use of word embeddings resolves this issue. Word embeddings are alternative feature representations of textual data, that are extracted sequentially from the word sequence in the text document [21]. They are semantically meaningful feature vectors that do not rely on the morphological form of the word but on its meaning. Examples of word embeddings include GloVe [6], Word2Vec [7] and FastText [22]. Word embeddings are given as input to the LSTM or transformer for learning the sequence of words in the input document [23].

Several researchers have explored word embeddings for resume classification. Some notable and distinctive works in this direction are: - (a) GloVe word embeddings with convolutional neural networks [24] (b) Word2Vec word embeddings with Bidirectional LSTM with attention [25] (c) GloVe word embeddings with deep neural networks [26] (d) GloVe word embeddings with graph neural networks [27]. In [28], Zu and Wang claimed that the combination of Bidirectional LSTM with convolutional neural network and conditional random fields is the best classifier to classify word embedding sequences extracted from text blocks in a resume. In another work, the combination of BERT transformer with conditional random fields was found to achieve the best performance [29]. In a recent work, Wings et al. emphasize the importance of contextual information for classifying skill sets in resumes [30]. One problem with word embeddings is the existence of multiple senses of a word that highlights the importance of the morphological representation of the word [32]. In this paper, we integrate both concepts of lexicality and semantics for resume document classification.

3 Methodology

In this section, we provide the motivation behind the adopted approach for resume classification, and the detailed steps in the computation of the semantically meaningful feature representations incorporating both lexicality and semantics in the same framework. We also outline the steps in the classification process.

3.1 Motivation

We propose the confluence of two diverse, popular perspectives for feature extraction from textual data: - the number of occurrences of the morphological form of the word [33, 34] and the semantics or meaning of the word [35, 36]. For the former, we make use of term frequencies to identify the unique keywords in a document, and for the latter, we consider semantically meaningful GloVe and Word2Vec word embeddings for representing the unique keywords filtered in the first phase.

In the current work, we take up the specific case study of resume classification where the task is to classify the resume of candidates to different job profiles that constitute the resume categories. A recent work on resume classification [38] successfully adapted the elite keywords, introduced in [16], along with machine learning classifiers, for resume classification. The elite keywords were shortlisted based on their frequencies of occurrences in a particular document class. Our current research advances on the existing work by:- (i) further filtering the elite keywords to extract the unique elite keywords that are exclusive to a particular class, followed by concatenation across classes after eliminating redundancy (ii) representing the sequence of unique elite keywords extracted from each resume document by GloVe and Word2Vec word embeddings to make the overall feature representation semantically meaningful (iii) learning the sequential word embeddings using BiLSTM which is known to process temporal information effectively to classify documents. The aim is to use semantic representations of only those discriminative keywords that occur more frequently in one class than in any other class. This would help to eliminate misleading keywords which though high in count may not singularly represent a resume class.

3.2 Steps of the proposed methodology

The pipeline of the proposed method is shown in Fig. 2, illustrating the various steps followed in the experimentation.

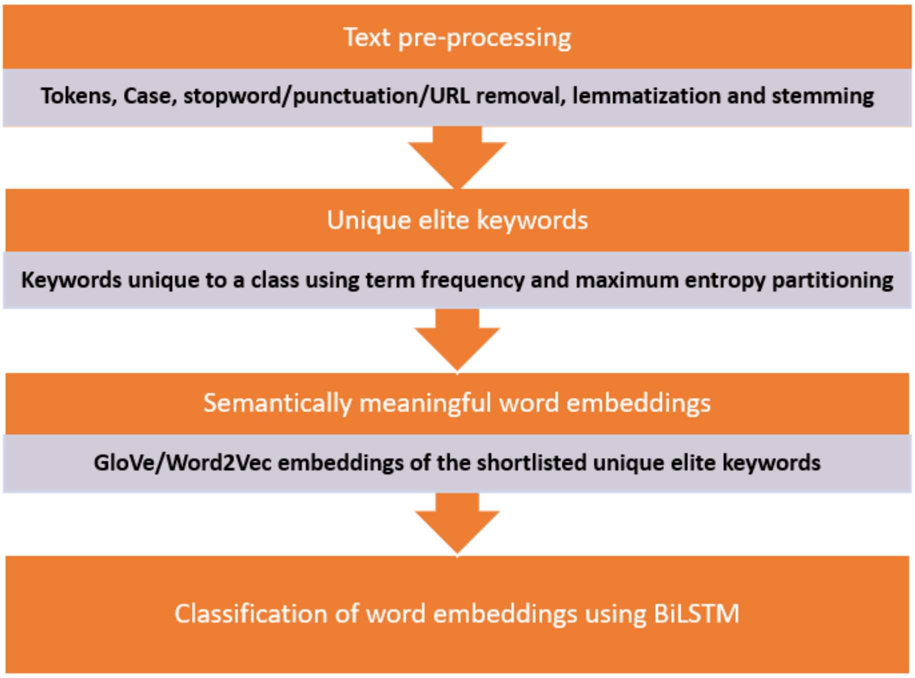

Text pre-processing
Tokens, Case, stopword/punctuation/URL removal, lemmatization and stemming
Unique elite keywords
Keywords unique to a class using term frequency and maximum entropy partitioning
Semantically meaningful word embeddings
GloVe/Word2Vec embeddings of the shortlisted unique elite keywords
Classification of word embeddings using BiLSTM

Figure 2. Pipeline for the proposed method

Step 1: The first step is text pre-processing that comprises of natural language processing techniques used to remove noise, and refine the text to a form suitable for classification. This constitutes tokenization of the text, converting all tokens to lower case, removal of punctuation, stopwords, URLs, hashtags, mentions, extra whitespaces, numbers and non-English characters. This is followed by lemmatization and stemming procedures that convert each word to its root form.

Step 2: The text pre-processing step is succeeded by the extraction of the unique elite keywords based on the computation of the relative term frequency and the Maximum Entropy Partitioning (MEP) algorithm, as per the procedure outlined in [16]. The first step is the identification of elite keywords from each resume class that are distinguished by their higher frequencies of occurrences in that particular class. These are then filtered to extract the unique elite keywords that comprise of those elite keywords which contribute to low entropy across classes. The MEP algorithm is used in all stages for determining the threshold of the optimal keyword subset (elite, unique elite). The procedure is explained in more detail below.

(i) *Elite keyword extraction*: The term frequency TF_i is defined as the total count of a keyword i in a document class (resume category).

$$TF_i = count\left(keyword_i\right)\Big| class \tag{1}$$

The relative or normalized term frequency is a probability value p_i and is computed for each keyword i as

$$p_i = \frac{TF_i}{\sum_{i=1}^{n} TF_i} \tag{2}$$

Here, n is the total number of distinctive keywords in a document class. $\{p_i\}$ is a complete probability distribution such that the sum of the probabilities is equal to one. The normalized term frequencies in each document class are now sorted in the descending order. The MEP algorithm is then used to shortlist the elite keywords comprising of the topmost partition, following the procedure shown in Fig. 3.

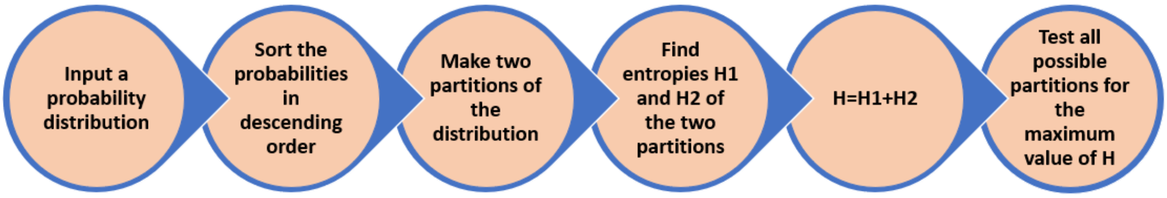

Input a probability distribution
Sort the probabilities in descending order
Make two partitions of the distribution
Find entropies H1 and H2 of the two partitions
H=H1+H2
Test all possible partitions for the maximum value of H

Figure 3. Maximum entropy partitioning (MEP) procedure to shortlist elite keywords from a document. The input probability distribution corresponds to the normalized term frequencies $\{p_i\}$ computed from each document class as per Equation (2). The elite keywords correspond to the topmost partition obtained after MEP.

(ii) *Unique Elite keyword extraction*: The unique elite keywords constitute of those elite keywords that are specific to a particular class. The primary motive of extracting unique elite keywords is to represent the text document in a more condensed manner. In the current set of experiments targeting resume classification, the resume classes correspond to job profiles, each of which are characterized by some popular keywords that are unique to the job profile. Hence, the unique elite keywords are an ideal choice for text representation in resume documents. The procedure for extracting the unique elite keywords is detailed below.

The entropy of an elite keyword j is determined using the relative or normalized term frequency f_k computed across document classes.

$$H_j^{(Elite)} = -\sum_{k=1}^{c} f_k \log f_k \tag{3}$$

where, the relative or normalized term frequency is a probability value f_k given by

$$f_k = \frac{p_k}{\sum_{k=1}^{c} p_k} \tag{4}$$

Here, c is the number of classes, and p_k is computed from Equation (2). The entropy of the elite keyword is now converted to a probability value q_j by normalizing the entropies in a document class as

$$q_j = \frac{H_j^{(Elite)}}{\sum_{j=1}^{e} H_j^{(Elite)}} \tag{5}$$

Here, e is the total number of elite keywords extracted from a document class. $\{q_j\}$ is a complete probability distribution such that the sum of the probabilities is equal to one. A unique elite keyword is now defined as the elite keyword that is unique to a particular class. In other words, since its occurrence is high in one particular class and low in all others; hence, it will contribute to a low value of entropy (Equation (3)) across all classes. The normalized entropies $\{q_j\}$ in each document class are now sorted in the ascending order. The MEP algorithm is then used to shortlist the unique elite keywords, as per the procedure shown in Fig. 4. The topmost partition corresponding to the elite keywords that contribute towards the lowest values of entropy in a document class is termed as the unique elite keywords. The unique elite keywords are computed from each class in the above manner, and concatenated across all classes after removing redundancy.

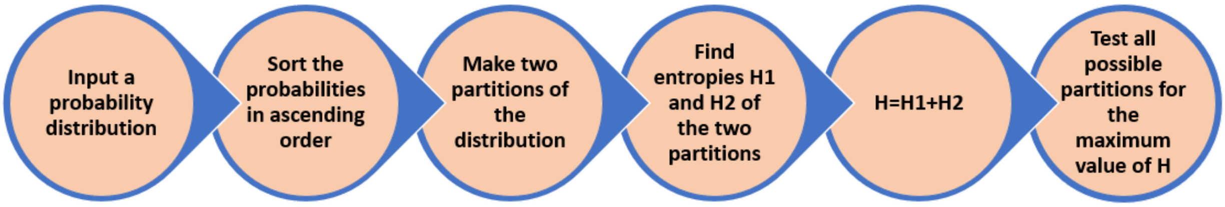

Input a probability distribution
Sort the probabilities in ascending order
Make two partitions of the distribution
Find entropies H1 and H2 of the two partitions
H=H1+H2
Test all possible partitions for the maximum value of H

Figure 4. Maximum entropy partitioning (MEP) procedure to shortlist unique elite keywords from a document. The input probability distribution corresponds to the normalized entropies $\{q_j\}$ computed from each document class as per Equation (5). The unique elite keywords correspond to the topmost partition obtained after MEP.

Step 3: In the original work [16], the unique elite keywords were used to derive the term frequency feature vectors that were subsequently classified using machine learning algorithms. In contrast, in the current work, we further pass the extracted unique elite keywords extracted from each resume document to a word embedding layer wherein the unique elite keywords are represented by either (i) 300-dimensional GloVe word embeddings or (ii) 300-dimensional Word2Vec word embeddings. GloVe and Word2Vec embeddings are word representations in a distributional semantic space. Words with similar meanings have similar (closer) word representations facilitating semantically meaningful text representation and classification. The words are transformed into feature vectors in the distributional semantic space so that similar meaning words are represented by similar vectors, aiding in text understanding and classification. Finally, the sequence of word embeddings is passed as input to a BiLSTM for classification purpose. LSTM was introduced in 1997 [10] by Hochreiter and Schmidhuber. It is distinctive from other machine learning algorithms since it accepts a time sequence of feature vectors at its input and hence is apt as a classifier for text since it accepts a sequence of words at its input. Bidirectional LSTM or BiLSTM processes the word sequence both in the forward and reverse directions. This is an added advantage over LSTM that processes the information in the forward direction only. We also explore the suitability of BiLSTM with attention mechanism for the current classification task.

4 Results

4.1 Experimental setup

All experiments were performed using Python (3.7 version) software on a 2.6 GHz Intel Core PC. For facilitating future research work and promoting reproducibility of research, we have made our code available online[1]. Three-fold cross validation is used for all experiments, for a 70:30 train-test split ratio. All experiments were performed on two publicly available resume datasets that we refer to as Dataset-1 and Dataset-2. The first dataset (Dataset-1) is collected from the website livecareer.com and is available online[2]. There are twenty four resume categories in this dataset corresponding to twenty four job profiles. The class population profile is uneven with most of the categories having more than 100 samples, while the “Automobile” and “BPO” categories contain only 36 and 22 samples, respectively. There are a total of 1738 training samples and 744 test samples for a train-test split ratio of 70:30. The details of the twenty four classes are shown in Table 1.

[1] https://github.com/Muskankalonia/Unique-Elite-Bag-of-Words-Approach-On-Resume-Classification

[2] https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset

Table 1: Resume categories and number of keywords extracted from Dataset-1

Resume Categories (Job profiles)	**Unique elite keywords**	**Total Keywords**
Accountant	1157	5084
Advocate	1291	5530
Agriculture	1054	4662
Arts	1093	5227
Apparel	1130	5384
Automobile	907	3227
Aviation	1419	6222
Banking	1265	5396
BPO	694	2481
Business-Development	1182	5424
Chef	1338	5674
Construction	1245	5514
Consultant	1378	6202
Designer	1193	5770
Digital-Media	1021	5150
Engineering	1407	6310
Finance	1114	5051
Fitness	1203	5575
Healthcare	1113	5948
HR	1025	4538
Information-Technology	1478	6275
Public-Relation	1301	6239
Sales	1127	5090
Teacher	1036	4636

Like Dataset-1, Dataset-2 is also a publicly available dataset [39], that is available online[3]. However, it is a multi-label dataset; the six job profiles contained in this dataset (such as “Security Analyst”, “Database Administrator”) are closely related and belong to the computer science/IT domain. In contrast, Dataset-1 comprised of resumes belonging to a diverse background (such as “Construction”, “Sales”, “Teacher”) as observed from Table 1. We consider only single-label resumes from Dataset-2 for our experiments. The total number of training samples is 8353 while the number of test samples is 3850 for a train-test split ratio of 70:30. The dataset is imbalanced, but not severely so, with class populations ranging between 1400 to 2400. The total number of keywords and the number of unique elite keywords extracted from each class of Dataset-2 are shown class-wise in Table 2.

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

Table 2: Resume categories and number of keywords extracted from Dataset-2

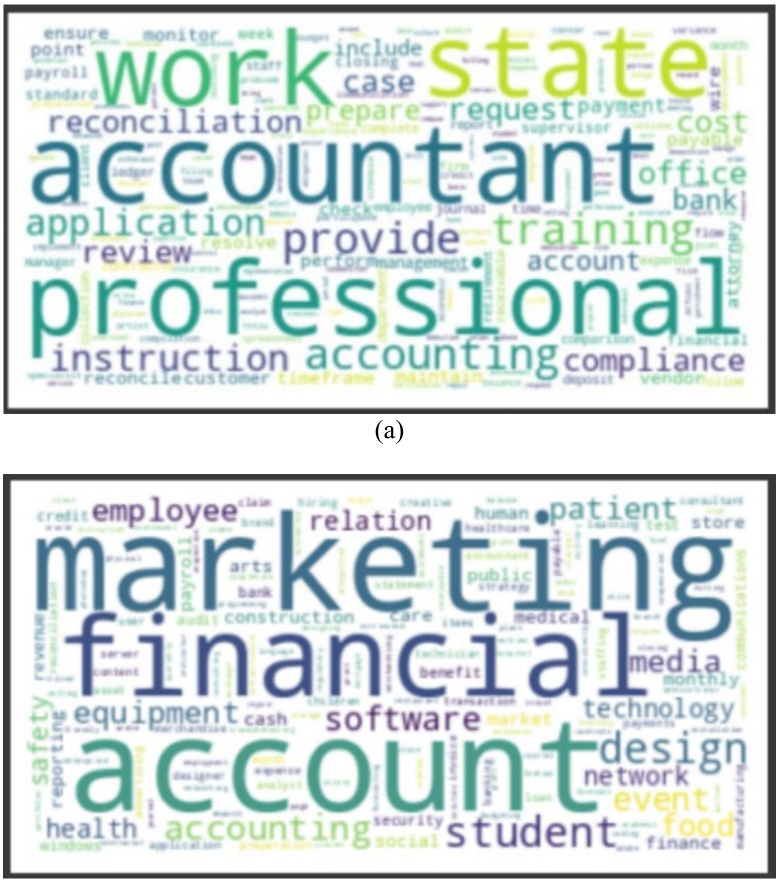
Resume Categories (Job profiles)	Unique elite keywords	Total Keywords
Database Administrator	3616	21010
Network Administrator	3166	25210
Project Manager	4786	17407
Security Analyst	3071	23502
Software Developer	4450	17369
Systems Administrator	4335	23211

On comparison, the number of unique elite keywords is found significantly lower (by around a factor of 1/5) than the total number of keywords as noted from both Tables 1 and 2. This highlights the significance of the proposed approach that serves to filter out the useful keywords that aid in effective classification, thereby reducing the feature dimensionality resulting in fast execution.

4.2 Discussion on results

The number of unique elite keywords individually shortlisted from each class is shown in Table 1 for Dataset-1, and in Table 2 for Dataset-2. The total number of unique elite keywords concatenated across all classes, after removing redundancy, amounts to 8336 for Dataset-1 and 9986 for Dataset-2. The different resume categories like “Accountant” or “Teacher” (Dataset-1) correspond to different job profiles. Hence the classification task at hand is to categorize a resume document to one of the job profiles based on the keywords they contain. The task is more challenging in the case of Dataset-2 in which all six job profiles belong to the computer science/IT domain, and categories like “Database Administrator”, “Network Administrator” and “Systems Administrator” appear much related. Since a resume may contain both useful and irrelevant information, a filtering mechanism is required prior to the feature vectorization. The unique elite keywords serve this purpose, and also manage to reduce the dimensionality to a significant extent, as shown in Table 1 and Table 2 that compare the total number of keywords occurring in a class with the number of filtered unique elite keywords.

The word clouds derived from the “Accountant” class of Dataset-1 are shown in Fig. 5. The size of the words in the word cloud indicates the frequency of the keyword. The word cloud constructed using all 5084 keywords in Fig. 5 (a) highlights “work”, “state”, “professional”, “application”, “training”, which are not relevant to the accountant class, and hence would lead to misleading results. Fig. 5 (b) illustrating the word cloud of 1157 unique elite keywords, on the other hand, highlights words like “marketing”, “financial”, account” etc. that are more relevant to the “Accountant” category and would help in more effective resume categorization.

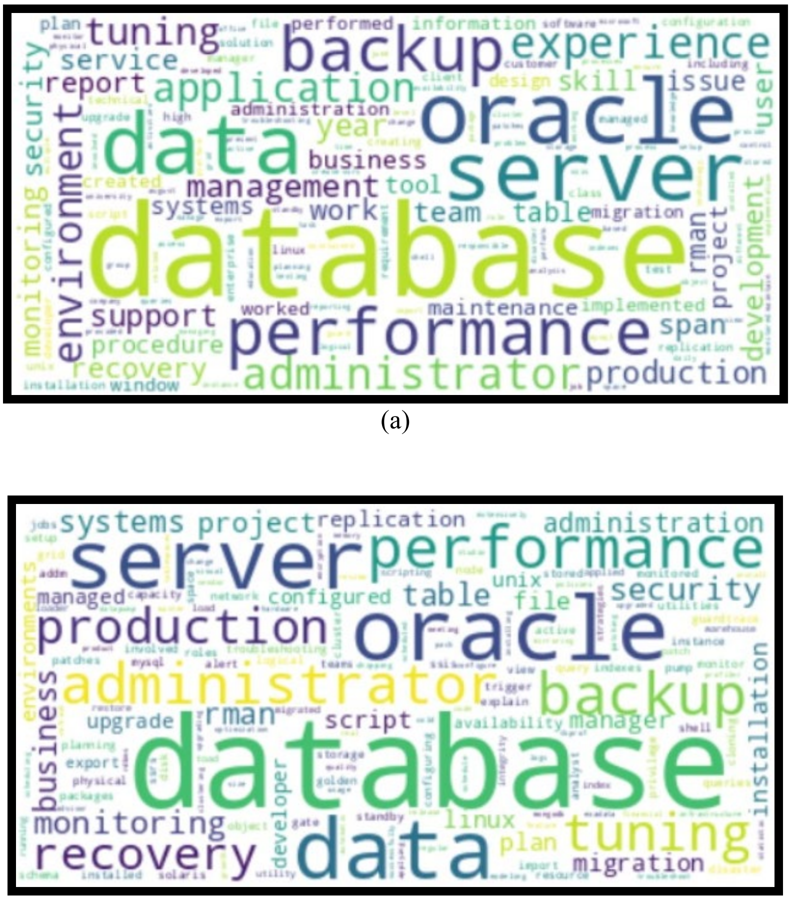


(a)

(b)

Figure 5. Word clouds derived from “Accountant” class of Dataset-1 for (a) all keywords (b) unique elite keywords

Likewise, the word clouds constructed using all the keywords and the unique elite keywords for the “Data Administrator” class in Dataset-2 are shown in Fig. 6 (a) and Fig. 6 (b), respectively. The most popular words highlighted in both word clouds in Fig. 6 are “database”, “data”, “oracle”, “server”, “backup” and “administrator” which are indicative of the nature of the job of a data administrator. However, the word cloud in Fig. 6 (b) corresponding to the unique elite keywords is more technically relevant and refined; it is free of keywords like “application” and “experience” occurring in the word cloud in Fig. 6 (a), that are irrelevant and not exclusive to the post of a data administrator.



(a)

(b)

Figure 6. Word clouds derived from “Data Administrator” class of Dataset-2 for (a) all keywords (b) unique elite keywords

The classification results for Dataset-1 are summarized in Table 3, while the results for Dataset-2 are summarized in Table 4. We compared our results with various combinations of BoW feature representations such as one-hot encoding, TF and TF-IDF with the random forest classifier [2, 14, 13]. We also compared the results with the recently published work on resume classification using elite keywords classified using the random forest classifier [38]. Substituting the random forest classifier with logistic regression resulted in a drop in accuracy by 1-2%. We also compared our results with that of GloVe and Word2Vec word embeddings using BiLSTM with and without attention mechanism [6, 12, 7, 25, 37]; the attention mechanism has known to boost the accuracies of sequence learning models in the past.

In case of Dataset-1 (Table 3), we achieved the highest accuracy of 70.64%, F1-score of 0.6485 and weighted F1-score of 0.6988, for the combination of unique elite keywords, GloVe embeddings and BiLSTM. The accuracy was reduced to 66.08% on adding the attention layer indicating that unlike chatbots [11, 31], the context information in the input sequence of words in resumes does not contribute useful information when the job profiles are vastly different in terms of job description, as is the case with Dataset-1. The unique elite keywords when classified using the random forest classifier [16] gave an accuracy of 65.14%. On the other hand, word embeddings (GloVe and Word2Vec), when directly used for representing the text in a resume document and classified using LSTM, gave poor accuracy (in the range of 30-40%), indicating that morphological forms of keywords need to be considered for effective classification. There is, therefore, no advantage in extracting semantically meaningful representations of all the keywords in a resume, since some of the text in the resume may be noisy and not indicative of the job profile. This highlights the significance of filtering of unique elite keywords prior to extracting the semantic representations which is the procedure adopted in this paper.

In case of Dataset-2 (Table 4), the highest accuracy of 93.77%, F1-score of 0.9375 and weighted F1-score of 0.9378, is achieved for the combination of unique elite keywords, GloVe embeddings and BiLSTM with attention. The attention layer proved to be beneficial in case of Dataset-2, in which the job profiles or resume categories are highly inter-related, all belonging to the computer science/IT domain. Hence capturing the contextual information in the sequence of filtered keywords using the attention mechanism proved to be beneficial in case of Dataset-2. The worst performance for Dataset-2 was for the word embeddings (GloVe - 81.95%, and Word2Vec - 80.16%) when they were used directly to represent the text in the resume without any filtering procedure. The performance of GloVe was better than for Word2Vec for both the datasets.

Table 3: Classification results of various methods for Dataset-1

Model	Test accuracy	F1-score	Weighted F1-score
TF + random forest [2]	58.98%	0.5625	0.5965
One-hot encoding + random forest [14]	54.55%	0.5021	0.5501
TF-IDF + random forest [13]	55.36%	0.557	0.592
Elite keywords + random forest [38]	62.60%	0.5673	0.6158
Unique elite keywords + random forest [16]	65.14%	0.5772	0.6256
GloVe +BiLSTM [6]	40.21%	0.3759	0.4006
GloVe + BiLSTM + attention [12]	44.77%	0.3994	0.4404
Word2Vec +BiLSTM [7]	37.66%	0.3356	0.3601
Word2Vec + BiLSTM + attention [25]	40.61%	0.3612	0.3902
Elite keywords + GloVe + BiLSTM (Ours)	69.30%	0.6171	0.6736
Elite keywords + GloVe + BiLSTM + attention (Ours)	69.03%	0.6526	0.6859
Unique elite keywords + GloVe + BiLSTM (Ours)	70.64%	0.6485	0.6988
Unique elite keywords + GloVe + BiLSTM + attention (Ours)	66.08%	0.6092	0.6482
Elite keywords + Word2Vec + BiLSTM (Ours)	69.57%	0.653	0.6922
Elite keywords + Word2Vec + BiLSTM + attention (Ours)	56.66%	0.5214	0.5504
Unique elite keywords + Word2Vec + BiLSTM (Ours)	70.10%	0.6277	0.6781
Unique elite keywords + Word2Vec + BiLSTM + attention (Ours)	60.18%	0.57	0.5994

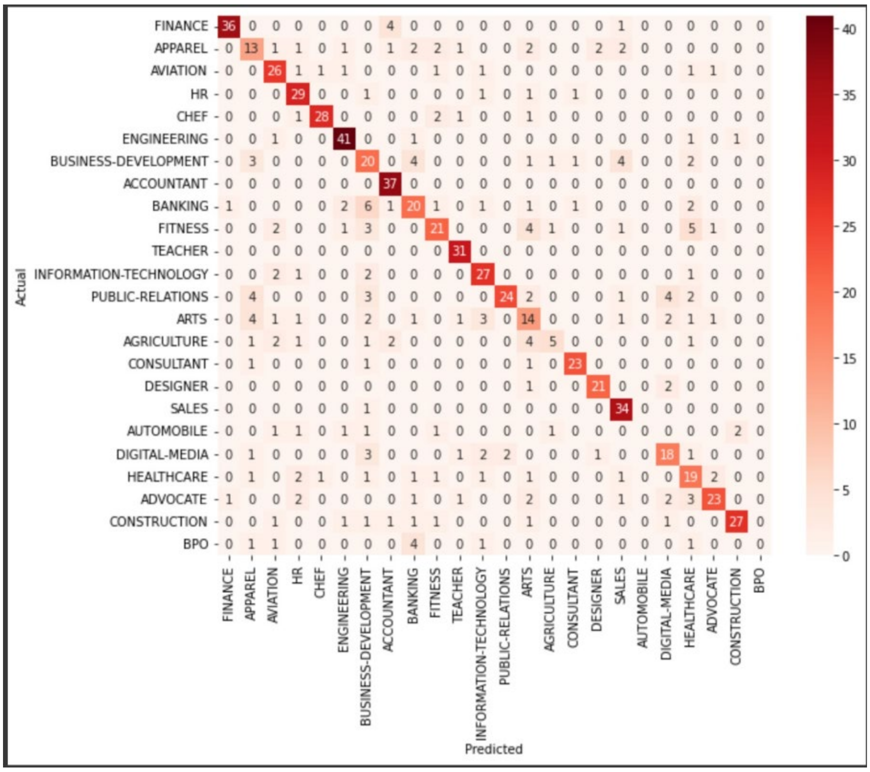
Table 4: Classification results of various methods for Dataset-2

Model	Test accuracy	F1-score	Weighted F1-score
TF + random forest [2]	84.97%	0.8439	0.8486
One-hot encoding + random forest [14]	81.81%	0.8081	0.8152
TF-IDF + random forest [13]	84.32%	0.8375	0.842
Elite keywords + random forest [38]	89.07%	0.8862	0.89
Unique elite keywords + random forest [16]	90.36%	0.9004	0.9031
GloVe +BiLSTM [6]	80.22%	0.7987	0.8023
GloVe + BiLSTM + attention [12]	81.95%	0.8164	0.8205

Word2Vec +BiLSTM [7]	78.10%	0.778	0.7809
Word2Vec + BiLSTM + attention [25]	80.16%	0.7989	0.802
Elite keywords + GloVe + BiLSTM (Ours)	89.72%	0.8954	0.897
Elite keywords + GloVe + BiLSTM + attention (Ours)	90.39%	0.9017	0.9043
Unique elite keywords + GloVe + BiLSTM (Ours)	92.79%	0.9268	0.9279
Unique elite keywords + GloVe + BiLSTM + attention (Ours)	93.77%	0.9375	0.9378
Elite keywords + Word2Vec + BiLSTM (Ours)	87.12%	0.8689	0.8705
Elite keywords + Word2Vec + BiLSTM + attention (Ours)	89.46%	0.8937	0.8947
Unique elite keywords + Word2Vec + BiLSTM (Ours)	91.78%	0.9153	0.9117
Unique elite keywords + Word2Vec + BiLSTM + attention (Ours)	93.35%	0.9314	0.9335

4.3 Confusion matrix

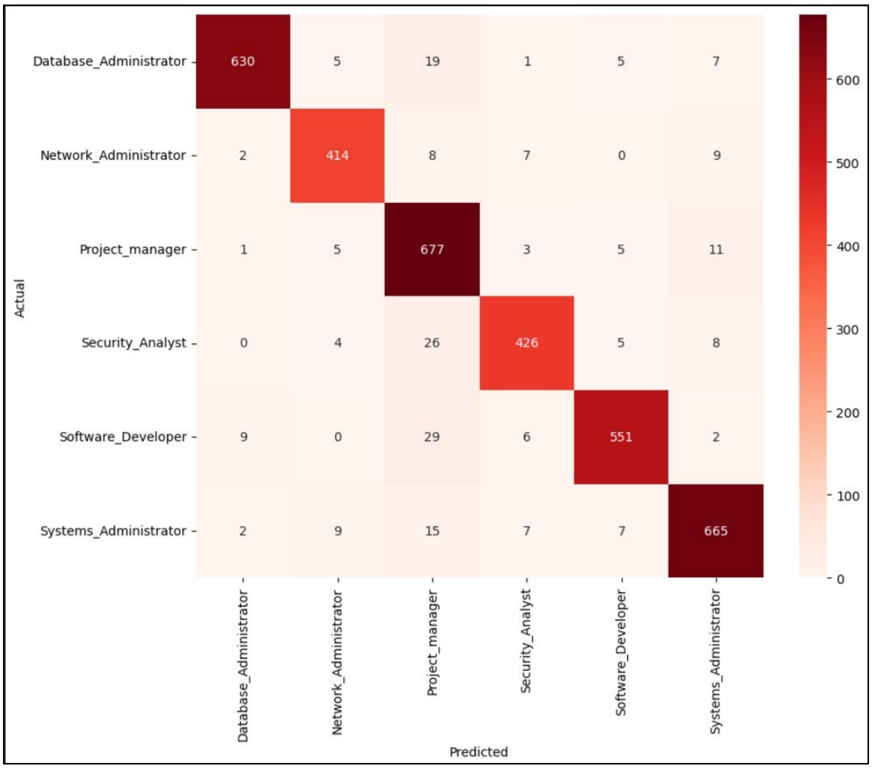
The confusion matrix showing the class-wise performance for the proposed combination of unique elite keywords and GloVe embeddings and BiLSTM that yielded the highest accuracy (70.64%) for Dataset-1 is shown in Fig. 7. The diagonal values indicate the correctly classified samples while the other cells show the mis-classification. We observe mis-classification between overlapping job profiles such as “Finance” and “Banking”, “Business-development” and “Banking” and “Sales”. Samples of some classes like “Apparel” and “Arts” are mis-classified to almost all other categories due to overlapping keywords in that class. The “Automobile” and “BPO” classes show the worst performance. This is on expected lines since the samples of these classes were too few (36 and 22, respectively) as compared to the other classes resulting in insufficient training for these two classes. Categories like “Teacher” and “Accountant” have 100% accuracy due to the absence of conflicting job profiles and non-overlap with other classes. This gives seed to the idea that resumes could be mapped to multiple job descriptions, hence investigating multi-label resume classification is the future scope of this work.



	FINANCE	APPAREL	AVIATION	HR	CHEF	ENGINEERING	BUSINESS-DEVELOPMENT	ACCOUNTANT	BANKING	FITNESS	TEACHER	INFORMATION-TECHNOLOGY	PUBLIC-RELATIONS	ARTS	AGRICULTURE	CONSULTANT	DESIGNER	SALES	AUTOMOBILE	DIGITAL-MEDIA	HEALTHCARE	ADVOCATE	CONSTRUCTION	BPO
FINANCE	36	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
APPAREL	0	13	1	1	0	1	0	1	2	2	1	0	0	2	0	0	2	2	0	0	0	0	0	0
AVIATION	0	0	26	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0
HR	0	0	0	29	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
CHEF	0	0	0	1	28	0	0	0	0	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0
ENGINEERING	0	0	1	0	0	41	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
BUSINESS-DEVELOPMENT	0	3	0	0	0	0	20	0	4	0	0	0	0	1	1	1	0	4	0	0	2	0	0	0
ACCOUNTANT	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BANKING	1	0	0	0	0	2	6	1	20	1	0	1	0	1	0	1	0	0	0	0	2	0	0	0
FITNESS	0	0	2	0	0	1	3	0	0	21	0	0	0	4	1	0	0	1	0	0	5	1	0	0
TEACHER	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0
INFORMATION-TECHNOLOGY	0	0	2	1	0	0	2	0	0	0	0	27	0	0	0	0	0	0	0	0	1	0	0	0
PUBLIC-RELATIONS	0	4	0	0	0	0	3	0	0	0	0	0	24	2	0	0	0	1	0	4	2	0	0	0
ARTS	0	4	1	1	0	0	2	0	1	0	1	3	0	14	0	0	0	1	0	2	1	1	0	0
AGRICULTURE	0	1	2	1	0	0	1	2	0	0	0	0	0	4	5	0	0	0	0	0	1	0	0	0
CONSULTANT	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	23	0	0	0	0	0	0	0	0
DESIGNER	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	21	0	0	2	0	0	0	0
SALES	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	0	0
AUTOMOBILE	0	0	1	1	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	2	0
DIGITAL-MEDIA	0	1	0	0	0	0	3	0	0	0	1	2	2	0	0	0	1	0	0	18	1	0	0	0
HEALTHCARE	0	1	0	2	1	0	1	0	1	1	0	1	0	1	0	0	0	1	0	0	19	2	0	0
ADVOCATE	1	0	0	2	0	0	0	0	1	0	1	0	0	2	0	0	0	1	0	2	3	23	0	0
CONSTRUCTION	0	0	1	0	0	1	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	27	0
BPO	0	1	1	0	0	0	0	0	4	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Figure 7. Confusion matrix for the proposed method for Dataset-1

The confusion matrix for Dataset-2 is shown in Fig. 8. In contrast to the confusion matrix plotted in Fig. 7, all classes show a good classification performance, possibly due to the more balanced class distribution of Dataset-2. It is also noted that all six classes belong to the computer science/IT domain, leading to greater chances of mis-classification as compared to Dataset-1 in which the job profiles were quite distinct from each other. The total number of mis-classification errors in each class for Dataset-1 is around 10, whereas in case of Dataset-2, the total number of misclassification errors exceeds 25 for almost all classes. The fraction of correctly classified test samples for the different classes are: - “Data Administrator”: 630/667, “Network Administrator”: 414/440, “Project Manager”: 677/702, “Security Analyst”: 426/469, “Software Developer”: 551/597, “System Administrator”: 665/705. The maximum mis-classification errors belong to the “Security Analyst” and “Software Developer” categories, both of which get mis-classified to the “Project Manager” category that is the most populated class among all other classes. The results indicate the class imbalance does create a bias in the classification results even though the dataset was not severely imbalanced. Also, the overlapping nature of the job profiles in case of Dataset-2 renders the classification task to be more challenging than in the case of Dataset-1 where all the job profiles were of a distinct nature.



Actual \ Predicted	Database_Administrator	Network_Administrator	Project_manager	Security_Analyst	Software_Developer	Systems_Administrator
Database_Administrator	630	5	19	1	5	7
Network_Administrator	2	414	8	7	0	9
Project_manager	1	5	677	3	5	11
Security_Analyst	0	4	26	426	5	8
Software_Developer	9	0	29	6	551	2
Systems_Administrator	2	9	15	7	7	665

Figure 8. Confusion matrix for the proposed method for Dataset-2

5 Conclusion

The automated screening of resumes flooding online job portals would help potential employers to select suitable candidates matching specific job profiles. Resume classification is a subset of the document classification task in which the categories are various job profiles. The resumes belonging to applicants applying for jobs may sometime bear similar content which highlights the significance of the proposed automated resume classification system that first shortlists the unique elite keywords specific to each class, and then extracts semantically meaningful word embeddings for the filtered keywords that would aid in more effective classification. More particularly, we combine the concepts of the uniqueness of a keyword in a resume class with the semantics or meanings of the filtered keywords. The unique elite keywords are first shortlisted from each class using the maximum entropy partitioning. The unique elite keywords are then concatenated across classes after eliminating redundancy. We obtain 8336 unique elite keywords for Dataset-1 and 9986 unique elite keywords for Dataset-2, by this procedure. The sequence of unique elite keywords extracted from each resume document are transformed into semantically meaningful GloVe and Word2Vec word embeddings which are then fed to a BiLSTM classifier as sequential input. We obtain an accuracy of 70.64% for the combination of unique elite keywords, GloVe embedding and BiLSTM on Dataset-1 comprising of twenty four classes belonging to diverse job profiles. We also achieved a high accuracy of 93.77% for the combination of unique elite keywords, GloVe embedding and BiLSTM with attention mechanism on Dataset-2 that comprised of six classes associated with highly inter-related job profiles in the computer science/IT domain. The use of attention layers was found to degrade the performance in case of Dataset-1 indicating that context information does not matter for unrelated job profiles. In contrast, the attention mechanism was found to boost the results for Dataset-2 underlining the significance of context in the resume documents belonging to Dataset-2 in which the job profiles are highly inter-related. The proposed approach outperforms the state of the art by a significant margin proving that it is better than the individual bag-of-words model and the word-embedding-based models popularly used in literature for the classification of text documents.

References

[1] Heaps, H. S. (1973). A theory of relevance for automatic document classification. *Information and Control*, *22*(3), 268-278.

[2] Onan, A., Korukoğlu, S., & Bulut, H. (2016). Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*, *57*, 232-247.

[3] Nasir, I. M., Khan, M. A., Yasmin, M., Shah, J. H., Gabryel, M., Scherer, R., & Damaševičius, R. (2020). Pearson correlation-based feature selection for document classification using balanced training. *Sensors*, *20*(23), 6793.

[4] Susan, S., Zespal, S., Sharma, N., & Malhotra, S. (2018, December). Single-keyword based document segregation using logistic regression regularized by bacterial foraging. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)* (pp. 1-4). IEEE.

[5] Carroll, J. M., & Roeloffs, R. (1969). Computer selection of keywords using word-frequency analysis. *American Documentation*, *20*(3), 227-233.

[6] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

[7] Church, K. W. (2017). Word2Vec. *Natural Language Engineering*, *23*(1), 155-162.

[8] Tulu, C. N. (2022). Experimental Comparison of Pre-Trained Word Embedding Vectors of Word2Vec, Glove, FastText for Word Level Semantic Text Similarity Measurement in Turkish. *Advances in Science and Technology Research Journal*, *16*(4), 147-156.

[9] Kim, J. K., Tur, G., Celikyilmaz, A., Cao, B., & Wang, Y. Y. (2016, December). Intent detection using semantically enriched word embeddings. In *2016 IEEE spoken language technology workshop (SLT)* (pp. 414-419). IEEE.

[10] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[11] Goel, R., Susan, S., Vashisht, S., & Dhanda, A. (2021, September). Emotion-aware transformer encoder for empathetic dialogue generation. In *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)* (pp. 1-6). IEEE.

[12] Wagh, V., Khandve, S., Joshi, I., Wani, A., Kale, G., & Joshi, R. (2021, December). Comparative study of long document classification. In *TENCON 2021-2021 IEEE Region 10 Conference (TENCON)* (pp. 732-737). IEEE.

[13] Ali, I., Mughal, N., Khand, Z. H., Ahmed, J., & Mujtaba, G. (2022). Resume classification system using natural language processing and machine learning techniques. *Mehran University Research Journal Of Engineering & Technology*, *41*(1), 65-79.

[14] Gunaseelan, B., Mandal, S., & Rajagopalan, V. (2020, December). Automatic extraction of segments from resumes using machine learning. In *2020 IEEE 17th India Council International Conference (INDICON)* (pp. 1-6). IEEE.

[15] Zhu, C., Zhu, H., Xiong, H., Ma, C., Xie, F., Ding, P., & Li, P. (2018). Person-job fit: Adapting the right talent for the right job with joint representation learning. *ACM Transactions on Management Information Systems (TMIS)*, *9*(3), 1-17.

[16] Susan, S., & Keshari, J. (2019). Finding significant keywords for document databases by two-phase Maximum Entropy Partitioning. *Pattern Recognition Letters*, *125*, 195-205.

[17] Roy, P. K., Chowdhary, S. S., & Bhatia, R. (2020). A machine learning approach for automation of resume recommendation system. *Procedia Computer Science*, *167*, 2318-2327.

[18] Ramraj, S., & Sivakumar, V. (2020, July). Real-Time Resume Classification System Using LinkedIn Profile Descriptions. In *2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)* (pp. 1-4). IEEE.

[19] Choudhry, A., Susan, S., Bansal, A., & Sharma, A. (2022, May). TLMOTE: A Topic-based Language Modelling Approach for Text Oversampling. In *The International FLAIRS Conference Proceedings* (Vol. 35).

[20] Roy, P. K., & Chahar, S. (2022, July). N-Gram Feature Based Resume Classification Using Machine Learning. In *Computational Intelligence in Communications and Business Analytics: 4th International*

Conference, CICBA 2022, Silchar, India, January 7–8, 2022, Revised Selected Papers (pp. 239-251). Cham: Springer International Publishing.

[21] Pelevina, M., Arefyev, N., Biemann, C., & Panchenko, A. (2016, August). Making Sense of Word Embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP* (pp. 174-183).

[22] Athiwaratkun, B., Wilson, A., & Anandkumar, A. (2018, July). Probabilistic FastText for Multi-Sense Word Embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1-11).

[23] Mallick, R., Susan, S., Agrawal, V., Garg, R., & Rawal, P. (2021, March). Context-and sequence-aware convolutional recurrent encoder for neural machine translation. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing* (pp. 853-856).

[24] Nasser, S., Sreejith, C., & 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.

[25] Bera, S., Ghosh, B., & Vanusha, D. (2022). Resume Classification Using Bidirectional LSTM and Attention Mechanism. In *Ambient Communications and Computer Systems: Proceedings of RACCCS 2021* (pp. 233-241). Singapore: Springer Nature Singapore.

[26] 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).

[27] Chung, Y. C., & Kuo, R. J. (2023). A domain adaptation approach for resume classification using graph attention networks and natural language processing. *Knowledge-Based Systems*, 110364.

[28] Zu, S., & Wang, X. (2019). Resume information extraction with a novel text block segmentation algorithm. *Int J Nat Lang Comput*, *8*, 29-48.

[29] Li, X., Shu, H., Zhai, Y., & 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.

[30] Wings, I., Nanda, R., & Adebayo, K. J. (2021). A context-aware approach for extracting hard and soft skills. *Procedia Computer Science*, *193*, 163-172.

[31] Goel, R., Vashisht, S., Dhanda, A., & Susan, S. (2021, January). An empathetic conversational agent with attentional mechanism. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-4). IEEE.

[32] Cheng, J., Wang, Z., Wen, J. R., Yan, J., & Chen, Z. (2015, October). Contextual text understanding in distributional semantic space. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (pp. 133-142).

[33] Pirkola, A. (2001). Morphological typology of languages for IR. *Journal of Documentation*, *57*(3), 330-348.

[34] Hancke, J., Vajjala, S., & Meurers, D. (2012, December). Readability classification for German using lexical, syntactic, and morphological features. In *Proceedings of COLING 2012* (pp. 1063-1080).

[35] Xing, C., Wang, D., Zhang, X., & Liu, C. (2014, December). Document classification with distributions of word vectors. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific* (pp. 1-5). IEEE.

[36] Kastrati, Z., Imran, A. S., & Yayilgan, S. Y. (2019). The impact of deep learning on document classification using semantically rich representations. *Information Processing & Management*, *56*(5), 1618-1632.

[37] Bahdanau, D., Cho, K. H., & Bengio, Y. (2015, January). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

[38] Sharma, M., Choudhary, G., & Susan, S. (2023, January). Resume Classification using Elite Bag-of-Words Approach. In *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1409-1413). IEEE.

[39] Jiechieu, Kameni Florentin Flambeau, and Norbert Tsopze. "Skills prediction based on multi-label resume classification using CNN with model predictions explanation." *Neural Computing and Applications* 33 (2021): 5069-5087.