National Research University Higher School of Economics

Faculty of Computer Science

Programme 'Master of Data Science'

MASTER'S THESIS

Applying Machine Learning and Natural Language Processing for Automated Teachers' Resume Screening

Student: Niiaz Gainutdinov

Supervisor: Maksimovskaya Anastasia Maksimovna

Moscow, 2023

Contents

Abstract

The teachers' resume classification is one of the most crucial tasks of the hiring manager of the school as it will directly affect the quality of the education and parents' satisfaction level. Automating the selection process of the right candidate can be achieved with the help of Machine Learning and Natural Language Processing techniques.

The purpose of the work is to find a more accurate algorithm for classifying the teachers' resumes and selecting the relevant skills according to the school requirements.

We show that the classification task of the collected teachers' resumes' dataset with 8 different categories was managed by the fine-tuning of different hyperparameters of the Bidirectional Encoder Representations from Transformers (BERT) [35] model, which outperformed the BiLSTM, Linear Support Vector Machine (LinearSVC) [30], K-Nearest Neighbor (KNN) [26], Logistic Regression [29], Multinomial Naive Bayes and XGBoost on weighted f1 metric achieving the 0.932. The mentioned beaten models, except BiLSTM, were experimented on the Bag-of-Words, Term Frequency Inverse Document Frequency, and Word2Vec vectorization techniques.

To identify the relevant candidate among classified resumes for a particular teaching position, the skills extraction according to the school standards, in our case Cambridge International School requirements, was performed with the open source project Spacy [1] by adding the skills set for each teacher.

1. Introduction

1.1 Problem statement

Hiring qualified teachers is a crucial task for educational institutions, but manually reviewing resumes can be time-consuming and prone to bias. In this thesis, we propose a solution to this problem by applying machine learning (ML) and natural language processing (NLP) techniques for automated teacher resume screening.

The hiring manager's peak period at school is between March and May when there might be around 200 - 300 new resumes in the inbox from different countries, and backgrounds, which takes lots of effort to review the resumes in more detail and separate them according to the school's needs.

The workflow of the school HR Manager is to manually read the resume, and determine the position according to the teachers' requirements if there is any particular relevant experience and education. The HR Manager needs to identify the relevant skills according to the school requirements and teaching position even if the resume is classified as a particular teacher. For instance, at the Cambridge International School, the right math teacher candidate besides having the relevant experience need to have certificates from the Cambridge International Education (CIE) courses, and the CIE exam preparations experience as well.

As the process of classifying and identifying the relevant skills is time-consuming, the possible best candidates might be hired by other schools. In this regard, the school administration is trying to assist the hiring manager by spending additional time on their main duties. As a result, during that hiring period the effectiveness of the school management, and the operational work quality is decreasing as well.

To avoid missing the most suitable candidates and the deterioration in the quality of the education process, school institutions are getting the service of third parties like hiring agencies or websites, which are using automated resume screening techniques to select the right candidate and providing the resumes to the HR Manager of the school.

1.2 Goals

In this work, we aim to learn and evaluate different Machine Learning and Natural Language Processing techniques to classify the resume and provide the skills according to the predefined requirements of the school. In this case, we will be considering the Cambridge International School teachers' standards.

It is known that in general there are 3 different approaches to text classification tasks:

1. Bag-of-Words based (BoW) models, where the word frequency is taken into account and represented as a vector, later the obtained vectors are used in the classifiers.
2. Sequence-based models like BERT [35], and LSTM [9].
3. Graph-based models like TextGCN [10].

The 'related work' section consists of various studies on implementing the ML models on text classification tasks, where it can be seen, that scientists were trying to find the right model for the particular datasets.

In our work, we are aiming to

1. Find the most accurate ML model for the teachers' resume classification.
2. Implement the NLP tools to extract the teachers' skills from the resume to identify the right candidate according to the school requirements.

1.3 Subtasks

Inspired by the studies [34] comparing the BoW models and Pretrained Language Models on the text classification tasks, our work is focusing on the first 2 approaches. During our studies, we experimented with the BoW, TF-IDF, and Word2Vec techniques before implementing the produced vectors to classifiers.

Machine Learning models in the text classification tasks were compared in the [6], [8], [20], which were the ground of our experiments with the Logistic Regression, LinearSVC, KNN, MultinomialNB, and Random Forest classifiers. The study with the XGBoost classifier [7] in the text classification was taken into account by including the model to compare with others. The BERT [35] model was compared with other classification models in various studies [14],[20], [31], where it was outperformed in some datasets, but also gained the best result in others. Known as the state-of-the-art model, BERT was included in our studies taking into account the fine-tuning advice [31].

To evaluate the effectiveness of the ML techniques, we conducted a study involving a dataset with 4119 teacher resumes from indeed.com, and used classifications models to determine the position of the candidate and NER techniques to extract and analyze the teacher's skills. The required skills list for each teacher position were obtained from the International School of Laos HR Manager and confirmed with the administration team.

We demonstrate how ML can help in the classification of resumes and how NLP can be used to extract relevant information from resumes, such as teachers' skills, and how this information can be used to select the right candidates who are a good fit for a particular teaching position.

The results of the experiments conclude that the BERT [35] classification model outperformed the LinearSVC [30], KNN [26], Logistic Regression [29], and XGBoost [22] models, achieving the f1 weighted score of 0.937. The proposed model is used for the next step to gather the skills with the Spacy [1] documentation to provide the skills matching score. We do believe that the developed algorithm will be beneficial for schools to reduce time on searching for the right candidates and select the most relevant teacher among others.

2. Related works

Throughout the years data science researchers were learning various methods and techniques for classifying the text and extracting the features from it. ML text classification techniques were employed in the email classification tasks to detect spam [15], recommendation systems for Human Resource Management [24] and E-Commerce [13], Sentiment analysis [11,12], Banking and Exchange Stock Markets [16]. The ML techniques in our work are related to Human Resource Management.

Text classification models can be classified as Bag-of-Words (BoW)-based, sequence-based, and graph-based models, which were mentioned in the study [14]. For a long time, the BoW-based techniques were widely used in text classification, recently the focus changed to the sequence-based approaches, and after to the graph-based models.

2.1. BoW-based models

Classical machine learning models that are based on a BoW-based input are extensively discussed in two surveys [17] and other comparison studies [20]. During the research work, the authors described in detail all steps of preprocessing the data: cleaning from the unnecessary and stop words, noise removal, tokenization, stemming, and lemmatizations. The word embeddings concepts BoW, TF-IDF [39], Word2Vec [40], Glove [41], FastText [42], and context2vec [43] were explained for the feature extraction steps of the text classification models.

The resume classification study [21] suggests the TF-IDF vectorizer as more suitable for feature extraction and vector representation as the results were perfect on almost all of the classifiers. During the studies, authors implemented 9 Machine Learning classification models: K Nearest Neighbors [26], Multinomial Naive Bayes, Bernoulli Naive Bayes [27], Gaussian Naive Bayes [28], Logistic Regression (LogR) [29], Linear Support Vector Classifier (LinearSVC), Support Vector Classifier (SVC) [31], Nu-Support Vector Classifier, Stochastic Gradient Descent (SGD). During the experiments with data consisting of 962 resumes with 25 categories, the LinearSVC, SGD, LogR, and SVC performed with a 1.00 precision score and had the accuracy of 99.6 %, 99.6%, 99.3%, and 99.3% respectively. The results of all 318 analyses on the test data concluded that the LSVC outperformed all other models with nearly 98% and 1.0 precision.

The TF-IDF techniques experimented with unigrams, bigrams, and trigrams, as a result, the trigrams were included in the feature vectors in the study of [34] on comparing the Pretrained Language Models (PLM) and the Support Vector Machine (SVM) text classification methods' efficiency. The BoW-based approach TF-IDF was used for preprocessing the text for the LinearSVM classifier, which outperformed the sequence-based approaches BERT [35], DistilBERT [36], RoBERTa [37], XLM [38]. The BERT transformers model was the closest one to the LinearSVM model in terms of the f1-score. During the experiments on the BBC News, 20NewsGroup, Consumer Complains and IT support tickets datasets BERT model obtained 0.79, 0.97, 0.92, 0.85 respectively, while the LinearSVM achieved 0.79, 0.98, 0.93, 0.82. The authors concluded that the pre-trained models do not provide significant gains over the Linear SVM classifier. However, the BERT fine-tuning techniques were not mentioned and there also should be next experiments with the maximum length, the batch size, and other hyperparameters.

2.2 Sequence-based models

An important innovative technique and the turning point in the improvement of the NLP technologies was the novice approach of the Bidirectional Encoder Representations from Transformers (BERT) [19]. By performing a huge amount of pre-training operations in an unsupervised manner and automatically mining the knowledge in semantic, BERT learns to produce word vectors that are contextualized and have a global semantic representation. BERT-like models are suited for large datasets as they can parallelize computation [14]. The effectiveness of BERT-like models in text classification is demonstrated by Galke and Scherp [20]. However, very few works take them into account when classifying texts. According to the studies [7], the BERT-based transformers provided better results compared with the CNN, LSTM, BiLSTM models on the short-text classification task.

Various fine-tuning techniques of the state-of-the-art, BERT [31] was experimented on the uncased pretrained model on 8 text classification datasets. Authors considered several factors that can be tuned to achieve better results of the final model. First factor is the length of the sentence since the maximum sequence length of BERT is 512. Authors implemented the truncating techniques to adjust the length of the long text. Assuming that the most valuable parts are at the start and the end of the text the only head, only tail and head (128 tokens) + tail (382 tokens) methods were evaluated, where the last method outperformed others. The second factor is selecting the most effective layer. There are an embedding layer, encoder with 12 layers, and a pooling layer in the official BERT-base model. Authors achieved the best results with the last layer with the max pooling. The third factor is selecting the appropriate learning rate to have a better optimizer. During the experiments, the learning rate 2e-5 is found to be necessary to make BERT combat the forgetting matter, while with the learning rate 4e-4 the training set fails to converge.

2.3 Skills extraction

Skills extraction was studied in various studies and the widely used one is the open-source project Spacy [1], which provides the framework to work easily with the text data. The toolkit provides the entity ruler with standard categories, and is also flexible on adding the new categories.

After reviewing the above-mentioned studies, we came up with the idea to compare the LinearSVC, MultinomialNB, Logistic Regression, K-Nearest Neighbor, and XGBoost classifiers over the BoW, TD-IDF, Word2Vec techniques, and the BERT, BiLSTM models on the text classification. After obtaining the best classifier, the resume will be evaluated additionally on the skills required by the Cambridge International School teacher's skills standards for a particular subject teacher.

3. Methodology

3.1. Dataset

Collecting the data

For the experiments the teachers' resumes were collected from the indeed.com platform by parsing with the Chrome webdriver and Selenium. The final dataset of 4119 resumes consists of 8 different teachers' positions: art teacher, math teacher, ICT teacher, kg teacher, music teacher, science teacher, PE teacher, and ESL teacher.

All the resumes consisted of the experience of being a particular teacher to keep the relevance of obtained data, and all of them were applying for a job as a particular teacher.

For each category around 500 resumes were collected to have a balanced dataset.

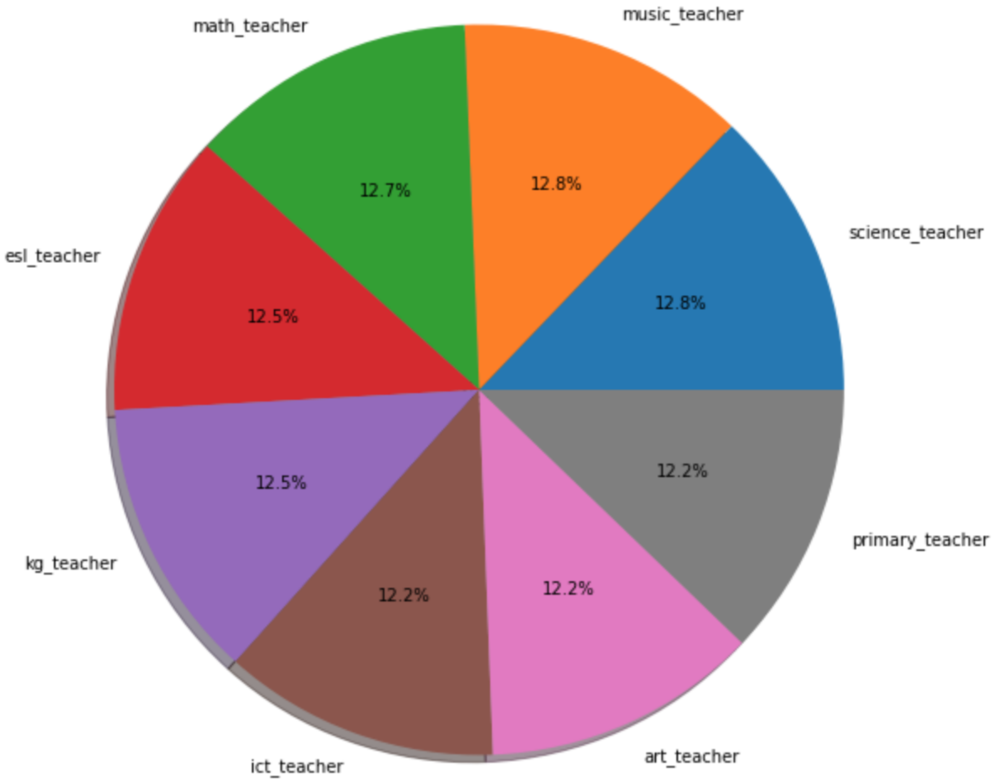


Figure 1. The distribution of teachers' resumes in the dataset.

Preprocessing the data

The data preprocessing was operated with the Natural Language Toolkit (NLTK) [18] and the regular expressions packages. With the help of regular expressions, the punctuations, hashtags, and extra whitespaces were removed. The dataset was obtained from indeed.com and most resumes were gathered from the USA teachers or from teachers that experienced teaching in the USA. As a result, the dataset consisted of a lot of states' full names and short abbreviations. All the states' names and abbreviations were removed aiming to avoid bias with the real-life teachers' resumes.

The NLTK package tools helped to remove the English stop words. The next step was to implement the stemming technique, which helps to keep the same form for the words. For instance, the words "studying", and "studied" will be changed to "study". The suffixes replacement of the words was achieved with the lemmatization tool of the NLTK package.

Below the used vectorization and classification techniques are briefly described.

3.2 Vectorization techniques

Bag-of-Words

The first and simplest is the Bag-of-Words (BoW) model, which builds a vocabulary from a dataset of resumes and calculates the frequency of words in each document. By getting the words as a feature, the resumes were represented by a vector with the same length as the vocabulary. The BoW model has its limitation as the dimension increases with the increase of the words, which can be partially handled with preprocessing techniques like removing stop words, punctuations, stemming, and lemmatization.

Term Frequency - Inverse Document Frequency (TF-IDF)

Taking into account that the focus only on the frequency of the token is not the best representation for text, the frequency of the common words in the dataset is found with little predictive power in the TF-IDF [39] model. The frequency of the term in the document (TF) is taken into account with the inverse document frequency (df) of the term among all documents (N). The value of a word increases with the increase in count, but it is inversely proportional to the frequency of the word in the dataset.

$$W(d,t) = TF(d,t) * log(\frac{N}{df(t)}) \tag{1}$$

The term frequency improves the recall metric and the inverse document frequency the precision of the word embedding. The problems with common term words in the dataset were overcome by the TF-IDF, but the similarities are not taken into account.

Word2Vec

Word Embedding is a technique, which learns the features from the text by taking the dataset words and mapping them to vectors of real numbers. The word appearance before or after another word is taken to calculate the vectors with the probability distribution. Words will be close in the vector space if words of the same context appear together in the corpus. In practice can be seen different embedding techniques: Word2Vec [40], Glove [41], and FastText [42].

In our work, the Word2Vec techniques were used, which provide two architectures.

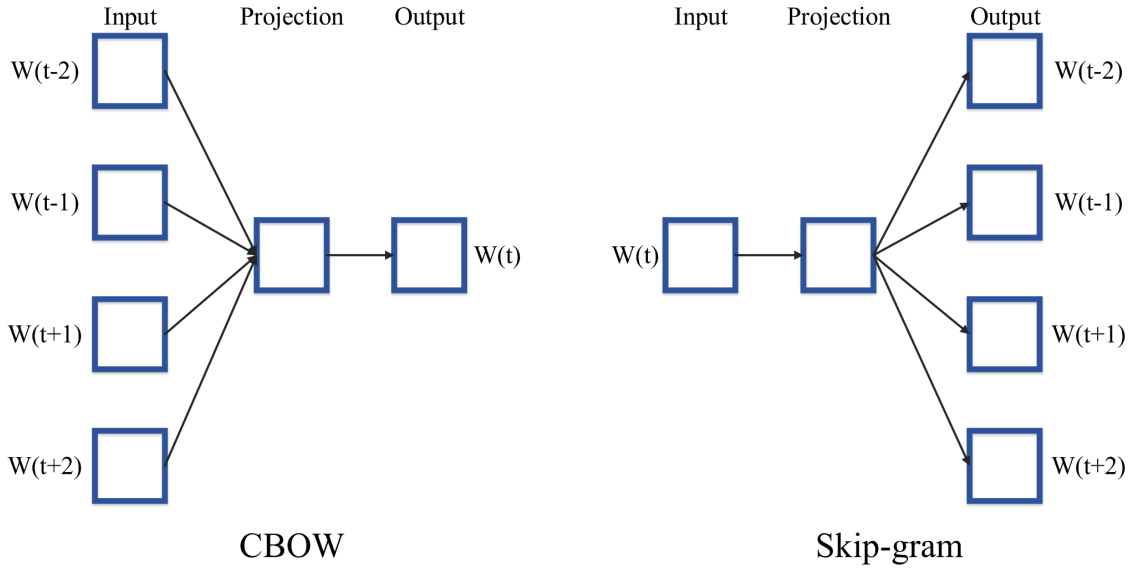


Figure 2. The CBOW and Skip-gram [23].

The continuous bag-of-words (CBOW) model by taking the context predicts the current word, and the Skip-gram by taking the current word predicts surrounding words.

3.3 Classification models

After having the vector representations there are different classification models that have been used. According to the studies [21], the most used models are Support Vector Machine, K-Nearest Neighbor, Logistic Regression, and Multinomial Naive Bayes. We also would like to include the XGBoost [7] and Random Forest [4] to compare with models as it is also used for the text classification tasks.

Multinomial Naive Bayes

Based on conditional probability, the Naive Bayes (NB) classifier is finding the probability of a vector that represents the class. Based on the strong independence between the features, for all the given classifiers the NB classifier computes the probability with the conditional probability. Multinomial Naive Bayes is from the family of Naïve Bayes classifiers that is used for the all-pairs' multinomial distribution [27].

K-Nearest Neighbor (KNN)

The algorithm of the KNN is based on calculating the Euclidean distance equation between data points and finding the k-nearest to assign the label according to the greatest neighboring data points [26].

Logistic Regression

The logistic function is used for the classification task with the threshold value in the Logistic Regression, which is considered to be one of the easiest models in the studies [30].

Linear Support Vector Machine

The model is aiming to find the best line to separate two classes, and known as being the simplest form of Support Vector Machine (SVM). SVM is finding the linear hyperplane o separate two classes. If the data can't be separated linearly it will not give precise results. The least square Support Machine classifier is also well known as Linear SVM [31].

XGBoost

XGBoost is from the family of gradient-boosted decision tree classifiers. During the boosting, the trees are built sequentially. The aim of each subsequent tree, which is called the base or weak learners, is to reduce the errors of the previous tree. The weak learners provide the prediction information enabling the boosting technique to develop strong learners. The benefits of XGBoost are its scalability, which drives fast learning through distributed and parallel computations and efficient memory usage.

3.4. BERT

BERT is a Transformer, which is the mechanism with attention technique that learns from the connection of the context between words in a given text. The Transformer, in general, consists of an encoder and decoder. First one reads the input text and the second produces a prediction for the task. As the goal is generating a language representation model, BERT needs only the encoder part. To work with the BERT, the sequence of tokens is converted to the vectors and then taken into the neural network. The preprocessing step consists of token embeddings, segment embeddings, and position embeddings.

During the token embeddings, the [CLS] token is added at the beginning and the [SEP] token is inserted at the end of each sentence. The segment embedding is a marker indicating the sentences in order to distinguish between sentences. The position embedding is used to add a token to identify its sentence position.

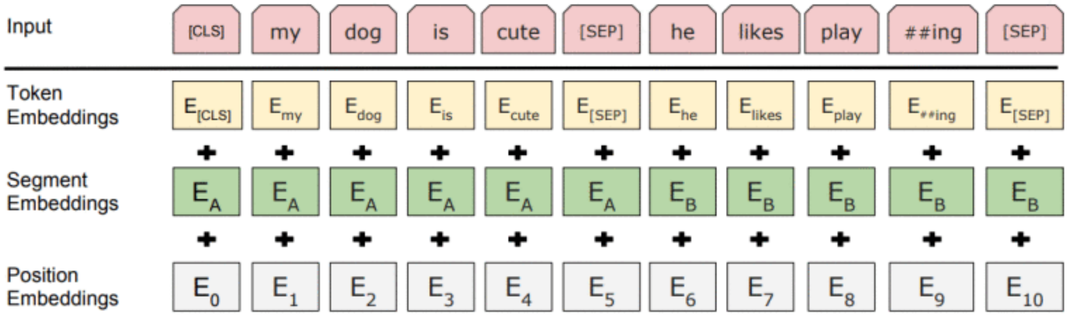

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 3. Input representation in BERT, which takes token embeddings, segment embeddings, and position embeddings [35].

By getting the output as a sequence of vectors, BERT uses the masked language modeling by replacing the words with the [MASK] token, running the whole sequence through the BERT encoder based on attention. By taking into account the context of other non-masked words in the sentence finds only the masked words predictions. The prediction of the next sentence is done by taking two sentences with the help of the [SEP] token.

BERT has 4 pre-trained versions: Bert-base, Bert-large, cased, and uncased for both. In this work, we will be using the bert-uncased.

3.5. BiLSTM

A bidirectional LSTM is a sequence processing model that consists of two LSTMs, the first model takes the input as it is, and the second model takes a backward direction copy of the sequence. This special architecture of BiLSTM effectively increases the quantity of data available to the network, giving the algorithm better context.

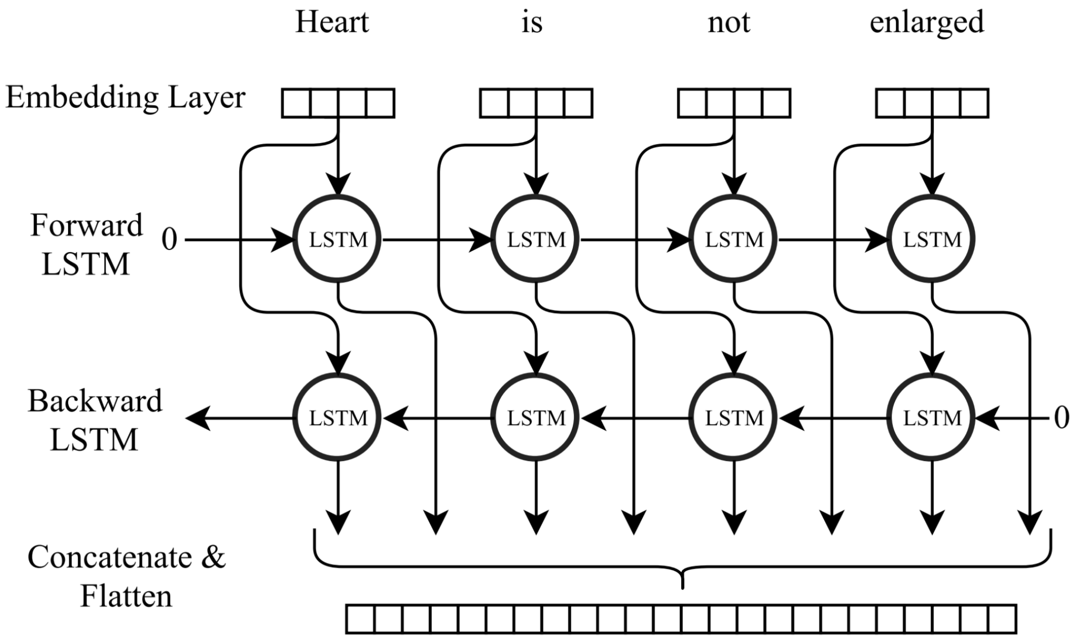


Figure 4. The BiLSTM architecture [2].

There are two LSTM models— one working in a forward direction, and the other in a reverse direction. Having them enables the BiLSTMs additional training by passing the text sequence twice. Therefore, the BiLSTM model completes additional training on a given dataset than LSTM, which helps to offer better predictions.

4. Experiments

The taken parameters for the BoW vectorizer and TF-IDF vectorizer:

N-grams have experimented with unigrams, bigrams, and trigrams. The analyzer is set to ‘word’, so features should be made of word n-grams. The max_df was set to 0.8, so we will ignore tokens that appeared more than 80% of all available. The min_df was set to 10 to ignore tokens that appeared less than 10 times.

Splitting the dataset

The dataset was separated with the train_test_split tool of the scikit-learn package [25], and the stratified parameter was specified for the teacher positions. The ratio is 80% train size and 20% test size.

BoW and TF-IDF with Classifiers

According to the study [34] the selection of the n-grams is affecting the results, where among (1,1), (2,2), (3,3) n-grams the (3,3) n-grams based models performed better. In our work, 5 experiments with (1,2), (1,3), (2,2), (2,3), and (3,3) n-grams were evaluated for each classifier to check the performance of the model.

For each classifier LinearSVC, MultinomialNB, LogisticRegressino, KNN, and XGBoost the cross-validation was performed with the default 5 folds and with default parameters for each classifier.

In figures 5 and 6 below it can be seen that the (1,2) n-grams and the (1,3) n-grams models are more accurate for most of the models. The XGBoost model performed better with BoW and TF-IDF vectorizer achieving the f1-weighted of 0.927 with (1,3) n-grams and 0.928 with (1,2) n-grams respectively.

We experimented with the max_depth (3,4,5) and the learning rate (0.01, 0.1, 0.2, 0.3) parameters of the XGBoost model over the TF-IDF vectorizer in order to have better results. The default parameters with a max depth of 3 and a learning rate of 0.1 achieved the highest result of 0.928. (Figure 7)

model_name	**accuracy**	**precision_weighted**	**recall_weighted**	**f1_weighted**	**n-grams**
XGBoost	**0.927534**	**0.928296**	**0.927534**	**0.927507**	**(1, 3)**
Logistic Regression	0.877461	0.878816	0.877461	0.877612	(1, 3)
LinearSVC	0.876705	0.878053	0.876705	0.876602	(1, 3)
MultinomialNB	0.754163	0.759663	0.754163	0.754665	(1, 3)
KNN	0.567152	0.594490	0.567152	0.568223	(1, 3)

Figure 5. The highest scores were obtained by classifiers with the BoW vectorizer.

model_name	**accuracy**	**precision_weighted**	**recall_weighted**	**f1_weighted**	**n-grams**
XGBoost	**0.928675**	**0.929868**	**0.928675**	**0.928549**	**(1, 2)**
LinearSVC	0.898710	0.900754	0.898710	0.898737	(1, 3)
Logistic Regression	0.878224	0.881137	0.878224	0.878591	(1, 2)
MultinomialNB	0.775018	0.780964	0.775018	0.774843	(2, 3)
KNN	0.637321	0.642809	0.637321	0.632505	(1, 2)

Figure 6. The highest scores were obtained by classifiers with the TF-IDF vectorizer.

model	**accuracy**	**precision_weighted**	**recall weighted**	**f1_weighted**	**max length**	**lr_rate**
XGBoost	**0.928675**	**0.929868**	**0.928675**	**0.928549**	**3**	**0.1**
XGBoost	0.927158	0.927982	0.927158	0.926989	3	0.3
XGBoost	0.92564	0.926679	0.92564	0.925565	4	0.2
XGBoost	0.923743	0.924982	0.923743	0.923636	4	0.1
XGBoost	0.921467	0.922516	0.921467	0.92123	5	0.1
XGBoost	0.917294	0.919243	0.917294	0.917179	3	0.01

Figure 7. Results of experiments with the XGBoost model over the TF-IDF vectorizer.

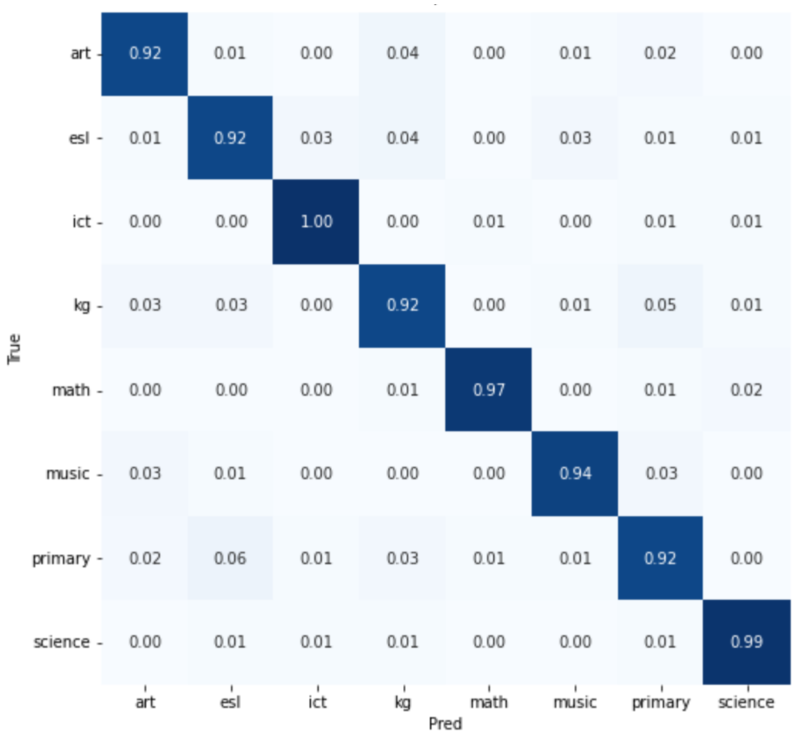


Figure 8. The confusion matrix of the XGBoost model with TF-IDF

Word2Vec

The following results were obtained after experiments with classifiers and the pre-trained word2vec model “GoogleNews-vectors-negative300”:

model_name	accuracy	precision_weighted	recall_weighted	f1_weighted
Logistic Regression	0.624425	0.640363	0.624425	0.627439
LinearSVC	**0.705998**	**0.707068**	**0.705998**	**0.703216**
KNN	0.492776	0.526306	0.492776	0.477034
XGBoost	0.614181	0.623035	0.614181	0.616170

Figure 9. The highest scores were obtained by classifiers with Word2Vec.

We can notice that the LinerSVC achieved the highest scores for all metrics by reaching the maximum 0.703 for the f1_weighted, which is 0.204 less than the XGBoost with BoW and 0.205 less than XGBoost with TF-IDF.

BiLSTM

We used the 2 BiLSTM layers for the model with the softmax activation function at the end. During the experiments, we were trying to find the best parameters for the embedding dimension (128, 256, 512, 1024, 2048), vocabulary size (20000, 50000, 10000, 20000), the unit of layers of 1st BiLSTM (32, 64, 128, 256) and the second BiLSTM layer (32, 64, 128, 256). The loss was specified as 'sparse_categorical_crossentropy', and the optimizer 'adam' during the compile stage. In the table below the 5 best results are presented.

Model parameters	**1**	2	3	4	5
embedding_dim	**256**	512	512	1024	1024
vocab_size	**50,000**	20,000	20,000	20,000	20,000
BiLSTM 1 (x)	**64**	128	128	128	512
BiLSTM 2 (x)	**32**	128	128	128	512
Dense 1 (x)	**16**	32	64	64	128
Dense 2 (x)	**8**	8	8	8	8
f1_weighted	**0.7339**	0.7268	0.7229	0.7221	0.7205

Figure 11. Experiments with BiLSTM model

The f1_weighted with 0.7339 was the highest score among other models. The heatmap confusion matrix from the seaborn package illustrates the true and predicted probabilities for each position.

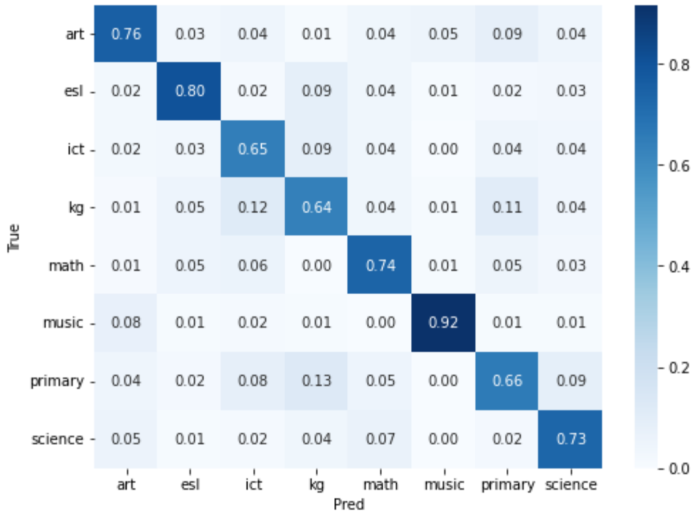


Figure 12. The confusion matrix of the BiLSTM model.

BERT

Based on the studies [31], we held experiments with different approaches to find out the best hyperparameters for our model. We have experimented with the batch size (3,4,5,6,7,8,9,10,16), max length (150, 250, 400, 450, 512), and the learning rate (1e-5, 2e-5, 5e-5).

Model	**batch size**	**max_length**	**lr_rate**	**f1_weighted**
1	**8**	**512**	**1e-05**	**0.937**
2	10	512	1e-05	0.9334
3	3	512	2e-05	0.9308
4	16	512	1e-05	0.9297
5	3	450	1e-05	0.9272

Figure 13. Results of the experiments with BERT with batch size, max length, and learning rate

Additionally, the max length has experimented with the slicing, which was mentioned in the [31] studies. We experimented with the tail, head, head+tail, or taking the middle of the text with different combinations.

Model	**batch size**	**max_length**	**lr_rate**	**slicing**	**f1_weighted**
1	**3**	**512**	**1.00E-05**	**[5:518]**	**0.9296**
2	10	512	1.00E-05	[5:518]	0.9295
3	10	512	1.00E-05	[15:528]	0.9261

4	3	512	1.00E-05	head(200)
				tail(312)
					0.9127

5	3	512	1.00E-05	[50:563]	0.9007

Figure 14. Results of the experiments with BERT with batch size, max length, earning rate, and slicing

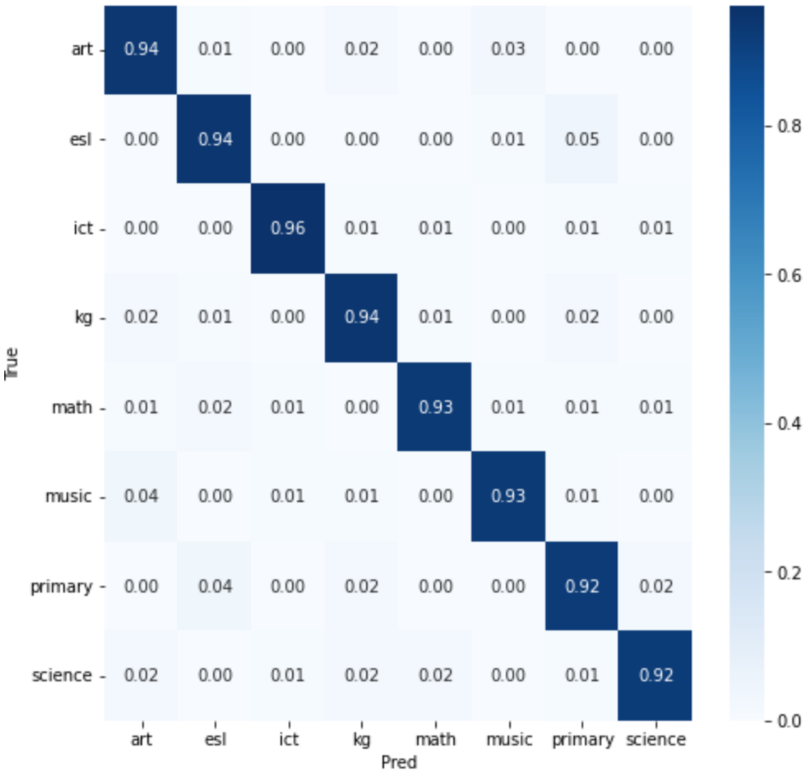


Figure 15. The confusion matrix of BERT results.

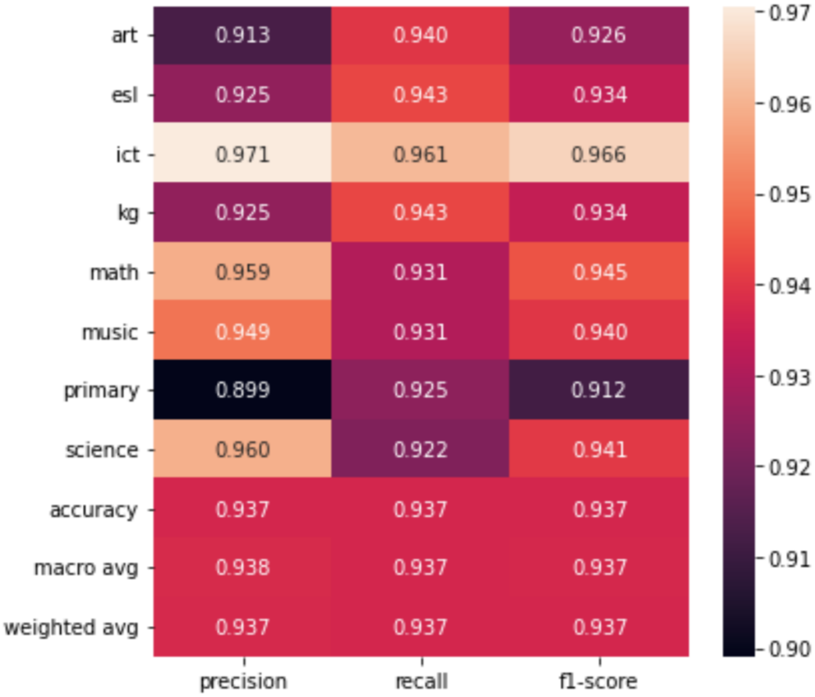


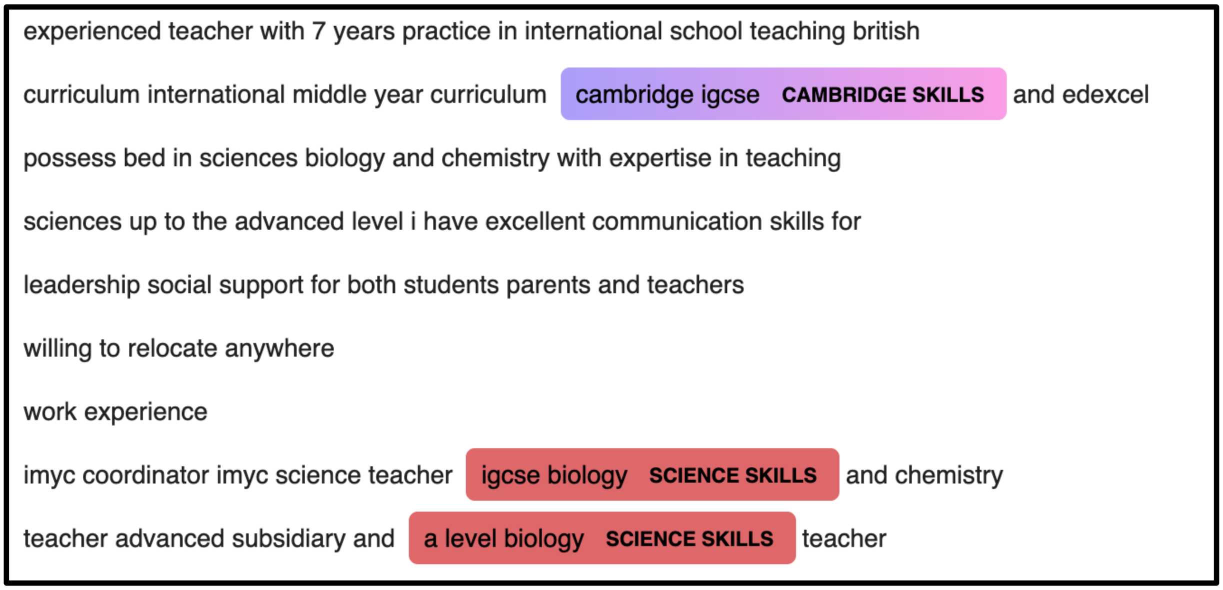
Figure 16. Classification report of BERT model

5. Skills extraction

Performing the classification tasks is beneficial to figure out the category of teachers' resumes, but not every teacher is suitable or desired by the school admin. During our work we collaborated with the management team of International School of Laos [44], authorized Cambridge International School, in order to specify the school standards and desired skills. For each teaching position, the designed skills were based on the Cambridge Assessment International Education (CIE) [45] standards, which provides the unique syllabuses with the identification numbers, professional development courses with special names. For instance, teacher can be experienced and classified as a math teacher, but not experienced with the CIE math curriculum standards and requirements. In this case, particular candidate will be asked to take additional course or to have trainings before the school year.

The skills extraction was manually performed with the Spacy [1] package by creating new pipeline with above mentioned skills. Despite having particular skills for a subject, teachers might have the general CIE skills, which was also included to the pipeline.

On figure 16 we can see the part of the resume for a science teacher, which has 1 general Cambridge and 2 science teacher skills:



experienced teacher with 7 years practice in international school teaching british
curriculum international middle year curriculum cambridge igcse CAMBRIDGE SKILLS and edexcel
possess bed in sciences biology and chemistry with expertise in teaching
sciences up to the advanced level i have excellent communication skills for
leadership social support for both students parents and teachers
willing to relocate anywhere
work experience
imyc coordinator imyc science teacher igcse biology SCIENCE SKILLS and chemistry
teacher advanced subsidiary and a level biology SCIENCE SKILLS teacher

Figure 16. CIE skills extraction. Example with science teacher resume.

The classified resume will be accompanied with the number of unique skills extracted with the model, so it will be helpful for the hiring manager to proceed with the most relevant candidates first.

6. Results discussion

XGBoost with (1,3) n-grams employed better results over BoW vectorizer, while over the TF-IDF it performed better with (1,2) n-grams achieving 0.927 and 0.928 f1_weighted metric. It is noticeable that Logistics Regression and LinearSVC classifier were close with BoW reaching 0.87, but with TF-IDF LinearSVC was better for 0.02 position.

The highest score with Word2Vec embedding was gained with LinearSVC with a 0.7 f1 weighted score, while XGBoost performed only 0.61.

XGBoost model over TF-IDF has experimented additionally with the max length and the learning rate parameters, after which the previous result was still the highest.

BiLSTM model was experimented with different units of layers, embedding dimensions, and vocabulary size, where the best model obtained a 0.73 f1 weighted score. The model consisted of 256 embedding sizes, 50000 vocabulary size, two BiLSTM layers with 64 and 32 units, and two Dense layers with 16 and 8 units respectively.

BERT was trained with different batch sizes, max length of the input, and the learning rate, where the best model performed 0.937 f1 weighted score, which is the highest among other models. The learned studies [31] were taken into account to fine-tune the parameters.

The experiments show that XGBoost classifier with TF-IDF vectorizer and BERT model provided the best results among other classifiers. At the very first experiment with max_length 200, learning rate 1e-05, and batch size 3, BERT model performed 0.89 f1 weighted metric, which was less than XGBoost 0.928. By adjusting the parameters, the best model that outperformed XGBoost was employed with batch size 8, max length 512, learning rate 1e-05 with the final f1 weighted score 0.937.

Taking into account the fine-tuning techniques mentioned in the [31] study, we tried to find a better solution. We tried taking the head only, tail only, or the head + tail with different combinations, where the highest score was with batch size 3, max length 512, learning rate e-05, and by taking the tokens from the 6th one, which achieved the 0.9296 f1 weighted score.

Confusion matrix

The confusion matrices were employed to estimate the performance of the models and to compare the true and the predicted values for each category.

It can be seen that BiLSTM model achieved more than 0.9 only to predict the music teachers' resumes, while BERT models' predictions for all teachers' resumes are more than or equal to 0.92.

Although BERT model outperformed XGBoost over TF-IDF vectorizer by 0.09, the confusion matrix illustrates that some teachers' resumes were predicted better with XGBoost model rather than with BERT.

Moreover, XGBoost model predicted better the ict, math, music and science teacher with 1.00, 0.94, 0.95, and 0.93 respectively, but BERT model employed less prediction for the same resumes being 0.96, 0.93, 0.93 and 0.92. BERT model achieved the highest predictions on the ict teachers' resumes gaining 0.96 and others predictions are more than or equal to 0.92.

It is interesting to notice that in both models, some primary teachers' resumes were predicted as either ESL teacher or KG teacher. In practice, the school admin sometimes replaces ESL or KG teacher with a primary teacher, as their working experience and educational background matches the requirements.

Overall, BERT model shows the consistency on predictions having all the scores being at least 0.92, while XGBoost struggled with art, KG, and primary teachers' resume with f1 scores 0.88, 0.9, and 0.89 respectively.

Classification metrics of BERT

We can notice that the primary teachers' resumes precision score was the lowest among others, which tells that about 10% of primary teachers' resumes predictions belongs to other positions. The confusion matrix illustrates that 5% of them were ESL teachers. At the other hand, the recall score achieved 92.5 % score, which interprets the correctness of predicting the actual primary teachers resumes among all primary teachers resumes.

The ESL and KG scores are identical for precision and recall with 92.5% and 94.3% respectively. 4% of primary resumes were predicted as ESL and 2% as KG. In practice, it is also common to mix the resumes, as the backgrounds of the teachers might be same regarding the work experience, but different in terms of education and certifications.

The highest precision and recall scores were achieved for the ict teachers' resumes, 97.1 % and 96.1 % respectively. Mostly, the model misclassified with 1% the science and math teachers as ict teachers, and vice versa. The science teachers' resumes listed as the second with the precision score 96%, but the last with the recall score 92.2%. The confusion matrix shows that 6% of science resumes were evenly identified as art, KG, and math teachers.

The music and math teachers have the same recall score of 93.1%, while the precision metrics varies, which for math teachers' resumes were higher for 1%. 4% of music teachers' resumes were predicted as art teachers, while 3% of last resumes were predicted as music. The art teachers' resumes achieved 94% recall and 91.3% precision scores.

The extracted skills will be beneficial for the hiring managers, which will receive the number of unique skills for the candidate. A candidate resume with more unique skills will have more chance to be hired and school will save time and perspective teacher candidate.

7. Summary

Our study experimented different classification techniques based on BoW, TF-IDF and Word2Vec, where XGBoost model with TF-IDF performed better with the 92.8 f1 weighted score. With fine-tuning of the BERT model, we have outperformed the previous score and achieved 93.7% f1 weighted score, which was taken as the best classifier for teachers' resume.

From the experiments' results, we observe that adjusting the batch size and the max length parameters of BERT model had the most impact on increasing the score with the default learning rate. The best f1 score was obtained with batch size 8, max length 512 and learning rate 1e-5.

It is interesting to notice that the results of the model illustrate the real-life problems as well. For instance, classifying the ESL, KG and primary teachers' resumes taking more time for hiring manager as it is getting complicated with comparing the education and obtained experience of teachers. In most cases, ESL and KG teachers can be replaced with a primary teacher as they partially have required education and might have previous experience. The model also confused and wrongly predicted 4% of primary teachers' resumes as ESL teachers and 2% as KG.

To overcome the above-mentioned situations and select the right candidate with the desired skills, the skills extractions have been implemented. Our studies were in contribution with International School of Laos [44], which administration team prepared the list of desired Cambridge Assessment International Education [45] skills. The candidates resume will be provided with the amount of the unique skills to the hiring manager, so it will increase chances to hire and not lose the desired teacher.

8. Further work

The future studies will be related to fully automation of the hiring managers' resume classification procedures. We are aiming to implement the inbox chat bot assistant, which will take the list of desired teachers' positions for the current time, evaluate received resumes, and send information to the hiring manager with the new perspective teacher including skills and attached resume.

References

1. https://spacy.io/api/entityruler

2. Savelie Cornegruta, Robert Bakewell, Samuel Withey and Giovanni Montana "Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks". arXiv:1609.08409, 2016

3. Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018, arXiv:1810.04805v2

4. Xu, B.; Guo, X.; Ye, Y.; Cheng, J. An Improved Random Forest Classifier for Text Categorization. JCP 2012, 7, 2913–2920.

5. Jeffrey Pennington, Richard Socher, Christopher Manning. "GloVe: Global Vectors for Word Representation" (Pennington et al., EMNLP 2014)

6. Riya Pal, Shahrukh Shaikh, Swaraj Satpute and Sumedha Bhagwat. "Resume Classification using various Machine Learning Algorithms". ITM Web of Conferences 44, 03011 (2022)

7. Ramraj S, Nishant Uzir, Sunil R and Shatadeep Banerjee. "Experimenting XGBoost Algorithm for Prediction and Classifi cation of Different Datasets" International Science Press, Volume 9, Number 40, 2016

8. Telnoni, P.A., Budiawan, R. and Qana'a, M.: Comparison of Machine Learning Classification Method on Text-based Case in Twitter. In: Proceedings of International Conference on ICT for Smart Society: Innovation and Transformation Toward Smart Region, ICISS (2019)

9. Ruishuang Wang, Zhao Li, Jian Cao, Tong Chen, and Lei Wang. 2019b. Convolutional recurrent neural networks for text classification. In International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019, pages 1–6. IEEE.

10. Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 7370–7377. AAAI Press.

11. Parkhe V., Biswas B., "Sentiment analysis of movie reviews: finding most important movie aspects using driving factors", Soft Computing, Vol. 20, No. 9, pp. 3373-3379. 2016.

12. Bakshi R.K., Kaur N., Kaur R., Kaur G., “Opinion mining and sentiment analysis”, Proceedings of the 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 452-455, New Delhi, India, 16- 18 March 2016.

13. Sivapalan S., Sadeghian A., Rahnama H., Madni A.M., “Recommender systems in e-commerce”, Proceedings of the World Automation Congress (WAC), pp. 179-184, Kona, Hawaii, 2014.

14. Fabian Karl, Ansgar Scherp, “Transformers are Short Text Classifiers: A Study of Inductive Short Text Classifiers on Benchmarks and Real-world Datasets”. 2211.16878v2, 2022

15. Mujtaba G., Shuib L., Raj R.G., Majeed N., AlGaradi M.A., “Email classification research trends: review and open issues”, IEEE Access, Vol. 5, p. 9044-9064, 2017.

16. Theofilatos K., Likothanassis S., Karathanasopoulos A., “Modeling and trading the EUR/USD exchange rate using machine learning techniques”, Engineering, Technology and Applied Science Research, Vol.2, No. 5, pp. 269- 272. 2012.

17. Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and Donald E. Brown. 2019. Text classification algorithms: A survey. Inf., 10(4):150.

18. https://www.nltk.org/

19. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems 30 (2017)

20. Lukas Galke and Ansgar Scherp. 2021. “Bag-of-Words vs. Graph vs. Sequence in Text Classification: Questioning the Necessity of Text-Graphs and the Surprising Strength of a Wide MLP.” CoRR abs/2109.03777 (2021). arXiv:2109.03777.

21. Irfan Ali, Nimra Mughal, Zahid Hussain Khand, Javed Ahmed, Ghulam Mujtaba.“Resume Classification System using Natural Language Processing and Machine Learning Techniques”. Mehran University Research Journal of Engineering and Technology Vol. 41, No. 1, 65 - 79, January 2022.

22. Tianqi Chen, Carlos Guestrin, “XGBoost : A scalable tree boosting system”, March 9, 2016, arXiv:1603.02754[cs.LG]

23 Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. arXiv 2013, arXiv:1301.3781.

24. Al-Otaibi S.T., Ykhlef M., A., “Survey of job recommender systems”, International Journal of Physical Sciences, Vol. 7, No. 29, pp. 5127-5142, 2012.

25. https://scikit-learn.org/

26. Xu J., “An extended one-versus-rest support vector machine for multi-label classification”, Neurocomputing, Vol. 74, No. 17, pp. 3114-3124, 2011.

27. Loper E., Bird S., “NLTK: the natural language to olkit”, arXiv preprint cs/0205028, 2002

28. Kibriya A.M., Frank E., Pfahringer B., Holms G., “Multinomial naive bayes for text categorization revisited, in Australasian Joint Conference on Artificial Intelligence”, Springer, 2004.

29. McCallum A., Nigam K., “A comparison of event models for naive bayes text classification”, Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, pp. 41-48, 1998.

30. Raschka S., “Naive bayes and text classification introduction and theory”, arXiv preprint arXiv:1410.5329, 2014.

31. Chi Sun, Xipeng Qiu* , Yige Xu, Xuanjing Huang, “How to Fine-Tune BERT for Text Classification?” arXiv:1905.05583, 2020

32. Schölkopf B., Smola A.J., Bach F., “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond”, MIT Press, 2002.

33. Suykens J.A., Vandewalle J., “Least Squares Support Vector Machine Classifiers”, Neural Processing Letters, Vol. 9, No.3, pp. 293-300, 1999.

34. Yasmen Wahba, Nazim Madhavji, and John Steinbacher. “A Comparison of SVM against Pre-trained Language Models (PLMs) for Text Classification Tasks” arXiv:2211.02563, 2022

35. Devlin, J., Chang, M.W., Lee, K. and Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis. pp. 4171–4186 (2019)

36. Sanh, V., Debut, L., Chaumond, J. and Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In: Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (EMC2) co-located with the Thirty-third Conference on Neural Information Processing Systems (NeurIPS 2019), pp. 1–5 (2019)

37. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)

38. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R. and Le, Q.V.: XLNet: Generalized autoregressive pretraining for language understanding. In: Proceedings of the Advances in Neural Information Processing Systems. Vancouver. pp. 5754–5764 (2019)

39. Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* 1988, *24*, 513–523.

40. Goldberg, Y.; Levy, O. Word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv* 2014, arXiv:1402.3722.

41. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Volume 14, pp. 1532–1543.

42. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *arXiv* 2016, arXiv:1607.04606.

43. Melamud, O.; Goldberger, J.; Dagan, I. context2vec: Learning generic context embedding with bidirectional lstm. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp. 51–61.

44. https://www.isl.edu.la/

45. https://www.cambridgeinternational.org/