Cross-Lingual Automated Essay Scoring - NLP Final Project 2023

Sharon Mordechai (ID. 304866551)

Submitted as final project report for the NLP course, IDC, 2023

1 Abstract

Automated essay scoring (AES) systems have been widely adopted to efficiently assess and evaluate written content. However, existing AES systems cannot often accurately score essays in languages other than English, restricting their widespread applicability in multilingual contexts.

To address this limitation, this research project proposes the development of a novel cross-lingual automated essay scoring system tailored explicitly for the Hebrew language.

Leveraging the power of Transfer Learning, we will build upon a pre-trained language model in English and fine-tune it to create a robust and accurate scoring model for Hebrew essays.

The project will be divided into multiple stages, starting with the creation of a new language model in English, followed by the fine-tuning process using a large-scale dataset of English essays. To enable cross-lingual adaptation, the dataset will undergo back translation to convert it into Hebrew. Finally, we will employ Transfer Learning techniques to adapt the fine-tuned English model to the Hebrew dataset, enabling it to accurately score Hebrew essays.

The project's successful completion will pave the way for more accessible and efficient automated essay scoring systems across various languages.

1.1 Related Works

Several noteworthy research papers have contributed to the progress of AES using Deep Learning techniques.

[1] Christopher M. Ormerod, Akanksha Malhotra, and Amir Jafari *"Automated essay scoring using efficient transformer-based language models"*.

[2] Dimitrios Alikaniotis, Helen Yannakoudakis, Marek Rei *"Automatic Text Scoring Using Neural Networks"*.

[3] Kaveh Taghipour, Hwee Tou Ng *"A Neural Approach to Automated Essay Scoring"*.

2 Solution

2.1 General approach

The general approach for solving the problem of Cross-Lingual Automated Essay Scoring involves a multi-step process. Initially, we perform data preprocessing, including cleaning the data by expanding contractions and ensuring uniformity in the text. The primary objective of data preprocessing is to prepare the dataset for further analysis and model training.

In the model development phase, we aim to explore a diverse set of approaches, starting with traditional machine learning models to serve as baseline benchmarks for gauging the performance of simpler techniques. Following this, we delve into neural network models, renowned for their capacity to capture complex patterns in data.

The main objective of this approach is to systematically compare and evaluate the performance of the diverse set of models, ranging from traditional machine learning to advanced neural network architectures.

After identifying the best-performing English model for automated essay scoring, we pursued cross-lingual transfer learning to adapt the model for the Hebrew language. To address cross-lingual challenges, we used back-translation to convert English essays into Hebrew, performed Hebrew essay embedding, and fine-tuned the model. This resulted in a robust automated essay scoring system for the Hebrew language.

By analyzing the results, we aim to identify the most effective model for cross-lingual automated essay scoring, providing valuable insights for further research and practical implementations.

2.2 Dataset

The dataset used for this study is sourced from the Kaggle competition *"The Hewlett Foundation: Automated Essay Scoring"* which seeks to develop an automated scoring algorithm for student-written essays.

The dataset comprises 12,976 essays categorized into eight different essay types, each corresponding to different prompts. Each essay is scored by two raters on a varying scale, depending on the specific prompt. In cases where the two raters differ significantly in their grading, a third rater is brought in to evaluate the essay, and the final score is determined based on the third rater's assessment.

The scoring process involves computing the final score using either an average or the sum of the first two raters' scores or using the score assigned by the third rater, if applicable. Due to the substantial variation in the score ranges among different essay sets (e.g., one set may have scores in the range [0, 4], while another set may have scores in the range [10, 60]), the scores are normalized to the interval [0, 10]. This normalization facilitates the training process and simplifies the task by making it easier to predict the final score.

2.3 Design

The design of our cross-lingual automated essay scoring system involved selecting appropriate model architectures and embeddings, addressing technical challenges, and ensuring efficient performance. To achieve the best evaluation results, we adopted a classification model, as essay scores were normalized within the range of 0 to 10, enabling accurate evaluation using precision, recall, and F1-score metrics.

For text embeddings, we leveraged the BERT (Bidirectional Encoder Representations from Transformers) tokenizer and *"bert-base-uncased"* model variant. After conducting extensive analyses, we have discovered that the BERT model's contextual embeddings offer a profound comprehension of essay content, surpassing simpler text embedding methods like Doc2Vec, Word2Vec, or tf-idf, and resulting in significant improvements in the accuracy and interpretability of our cross-lingual automated essay scoring system.

Our approach employed the pooler layer for machine learning (ML) models, creating a dataset with (768,) dimensions, derived from the hidden state of the "[CLS]" token in the BERT model. However, for neural network (NN) models, we utilized the last layer of the BERT model, which had (512,768) dimensions, representing the final contextualized embeddings for each token in the essay. By using the last layer, our NN models could capture more detailed information and semantic context from the essays, leading to improved performance. For the *"BertForSequenceClassification"* model, we utilized the input ids and attention masks obtained from the BERT model. The input ids represented the tokenized essay input, while the attention masks allowed the model to focus on relevant tokens and disregard padding tokens during processing.

We explored various ML models, including Support Vector Classifier, Random Forest, AdaBoost, Logistic Regression, K-Nearest Neighbors, Decision Tree, Gaussian Naive Bayes, and Linear Discriminant Analysis. Grid Search was used for hyperparameter tuning, enhancing model training and evaluation.

Moreover, we explored various neural network (NN) models. First, we fine-tune the BERT Classifier, utilizing its pre-trained capabilities on English text data. Subsequently, we experiment with constructing various networks, such as constructing a Convolutional Neural Network (CNN) model, exploring the Long Short-Term Memory (LSTM) model, and lastly, exploiting the power of attention mechanisms, which represents the state-of-the-art approach in NLP tasks.

For Hebrew training, we used the best-performing English model. To address cross-lingual challenges, we performed back-translation using the *"Helsinki-NLP/opus-mt-en-he"* model to translate English essays into Hebrew. We then utilized *heBERT*, a BERT model for the Hebrew language, to perform essay embedding and trained and evaluated using the optimized model.

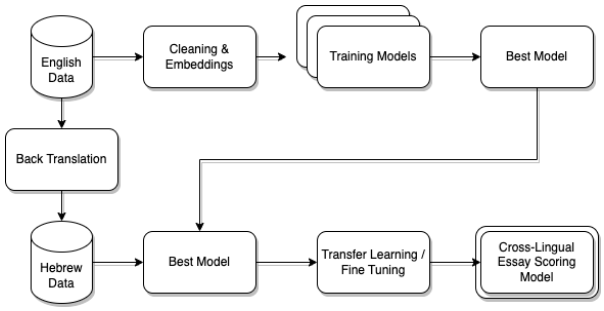

Figure 1: Process Design Overview

The research encountered several technical challenges: 1. Selecting the most suitable data embedding model (Doc2Vec, Word2Vec, tf-idf, BERT, RoBERTa, or XLNet) and size (base or large BERT). After extensive experimentation, we found the base BERT model provided comparable results while being computationally efficient. 2. Determining the ideal level of data cleaning operations before using the BERT model. We concluded that BERT required minimal text cleaning to avoid losing context and achieve better results. 3. Defining models and hyperparameters for improved evaluation results, which involved time-consuming experimentation. 4. Finding optimal hyperparameters for the translation model proved to be particularly difficult. Achieving accurate translation of English essays into Hebrew required fine-tuning the translation model to capture the nuances and patterns specific to Hebrew language essays.

Regarding time complexity, training the models required several weeks of computation. Moreover, the translation of data from English to Hebrew consumed a substantial amount of time. To expedite the training process, we utilized GPU resources, which significantly reduced the training time for each model. The training time for individual models varied and can be found in the *Results* section.

3 Experimental results

In this section, we present the experimental results for Cross-Lingual Automated Essay Scoring.

Our approach includes two main phases: (1) Automated Essay Scoring using various machine learning models, BERT sequence classification, and neural network architectures ranging from simple to state-of-the-art designs, within the context of the English language data; (2) Leveraging Transfer Learning to adapt the best-performing English language model to the Hebrew language, achieved through fine-tuning.

In all the experiments, we utilized the Cross Entropy Loss as the criterion and employed the AdamW optimizer with a learning rate of 1e-3 and weight decay of 0.001.

3.1 Experiment 1: Machine Learning Models

In the first stage of our experiments, we prepared the dataset by extracting pooler embeddings of size (768,) from BERT, and subsequently splitting each element in this vector into a dedicated column. This data preprocessing allowed us to utilize the dataset with various machine-learning models. To optimize the performance of each model, we employed a grid search, to determine the best configuration for each classifier.

The hyperparameters considered for the classifiers were as follows: Support Vector Classifier (SVC): Kernel, C, and gamma. Random Forest Classifier: Number of estimators and maximum depth of trees. AdaBoost Classifier: Number of estimators and learning rate. Logistic Regression: Penalty and regularization parameter C. K-Nearest Neighbors Classifier: Number of neighbors and weighting scheme. Decision Tree Classifier: Maximum depth of the tree and minimum samples required to split a node.

After determining the optimal hyperparameters through grid search, we proceeded to train and evaluate the machine learning models using the selected configurations. To ensure robustness in our experimental setup, we split the dataset into training and testing sets, utilizing an 80-20 train-test split ratio. In our evaluation, we consider multiple metrics, including precision, recall, and F1 score, alongside accuracy, to assess model performance; however, particular emphasis is placed on accuracy as the primary metric for result analysis.

3.2 Experiment 2: BERT for Sequence Classification Model

In the second experiment, we fine-tuned the BERT for Sequence Classification model on English language data. The dataset was prepared using input ids and attention masks from the BERT tokenizer, enabling the model to comprehend textual information effectively. For evaluation, we split the dataset into training, validation, and testing sets with an 80-20 split ratio and utilized a training data loader with a batch size of 64 for efficient processing. To optimize the BERT model for the target task, specific layers, including the classifier, pooler, and the last transformer encoder layers (layers 9,10, and 11), were selectively fine-tuned while keeping others frozen. This approach allowed the model to retain valuable pre-trained knowledge and adapt to the new task. The performance of the fine-tuned BERT model was evaluated based on accuracy and loss metrics, measuring the overall correctness of predictions and training convergence, respectively.

3.3 Experiment 3: Neural Network Models

In the third experiment, we undertook the task of running neural network models, ranging from simple architectures to state-of-the-art complex ones. Before model training, the dataset was split into train, validation, and test sets using an 80-20 split ratio. Subsequently, we implemented custom data loaders for each set, ensuring efficient handling during model training and evaluation with a batch size of 64. To address the issue of imbalanced data, we devised

an approach that balanced the dataset by sampling a specified number of rows per class, ensuring equal representation across all classes. By employing batch-based training with balanced data, we aimed to improve model performance and mitigate the effects of class distribution imbalance.

A collection of initial neural network models were created, including Linear layers network, CNN, LSTM, and the attention mechanism. The hyperparameters for each model were manually defined, using a trial-and-error method.

(1) A basic linear neural network with a single fully connected layer is used. (2) A neural network with three fully connected layers. It applies ReLU activation and dropout regularization after each layer to enhance non-linearity and prevent overfitting. (3) A Convolutional Neural Network (CNN) architecture. It applies a single 2D convolutional layer with a kernel size of 3, followed by ReLU activation, max-pooling, and dropout regularization operations. (4) A Long Short-Term Memory (LSTM) neural network, utilizing sequential information from embeddings. It consists of two LSTM layers with 100 and 200 hidden units, respectively, and dropout regularization (0.2) is applied after each LSTM layer. (5) A neural network architecture, integrating attention mechanisms. It includes a Positional Encoding module and a Multihead Attention module. The model further consists of two fully connected layers with 512 units each, utilizing ReLU activation and dropout regularization, followed by a linear layer.

For the evaluation of these models, precision, recall, f1, and accuracy metrics were measured. However, we have decided to focus on the accuracy and loss metrics for assessing the model's performance.

3.4 Automated Essay Scoring in the Hebrew Language

In the second part of our experiments, we focused on Automated Essay Scoring in the Hebrew Language, leveraging Transfer Learning from English Language Data. Due to the absence of Hebrew essay scoring data, we employed back translation on the English dataset and converted the essays to Hebrew embeddings using heBERT with the same size as English embeddings (512, 768), and we split it into training, validation, and test sets using an 80-20 split ratio. For transfer learning, we fine-tuned the model by selectively setting the last layers (including layer 11) parameters as trainable while freezing other layers. This process allowed us to adapt the pre-trained English model to the specific task of Automated Essay Scoring in Hebrew. In this phase, precision, recall, f1, accuracy, and loss metrics were measured for evaluating the performance of the Hebrew model. However, our primary focus for the evaluation step was on the accuracy and loss metrics.

3.5 Results

In our experimental results (Table 1), we first evaluated various machine-learning algorithms which demonstrated poor performance in comparison to the neural network models.

Among the neural network models, the BERT For Sequence Classification model exhibited outstanding performance, achieving a test loss of 0.602 and an impressive test accuracy of 84.01%. It demonstrated the capacity to comprehensively understand essay scores effectively but required significant training time.

Further exploring the results from a loss perspective, the Attention Linear Model also showcased promising performance with a test loss of 0.997, providing a viable alternative to BERT while reducing the computational burden.

Interestingly, from an accuracy standpoint, the simple Linear Model stood out, attaining an accuracy of 79.78%, closely followed by the CNN model, which achieved an accuracy of 79.69%.

Model	Test Loss	Test Accuracy	Execution Time (minutes)
SVC	3.27	54.00	0.76
Random Forest	4.02	47.68	0.81
AdaBoost	8.15	16.87	2.65
Logistic Regression	3.36	52.88	0.03
K-Nearest Neighbors	5.01	41.29	0.008
Decision Tree	5.63	34.70	0.13
Gaussian Naive Bayes	12.02	10.67	0.005
Linear Discriminant Analysis	3.24	54.16	0.031
Bert For Sequence Classification	**0.60**	**84.01**	**57.01**
Linear (1 layer)	5.03	**79.78**	1.01
Linear (3 layers)	1.09	57.69	4.39
Linear (2 layers) + Attention	**0.99**	60.23	9.39
CNN	4.69	**79.69**	6.68
LSTM	2.03	50.08	2.85

Table 1: Model Performance Comparison Results

The second phase of our experiments focuses on the Transfer Learning process from the best-performed Essay Scoring model developed for the English language to adapt the model for the Hebrew language, thereby achieving Cross-Lingual Automated Essay Scoring.

The obtained results (Table 2) for the Fine-Tuned BERT For Sequence Classification model in the Hebrew language are as follows: a test loss of 1.32 and a test accuracy of 47.42%. While this accuracy may not be deemed exceptional, it signifies a promising foundation for the model's performance.

Moreover, considering the model's primary objective of predicting essay scores within the precise range of 0 to 10, we conducted an additional evaluation by categorizing scores into three main classes: below average, average, and above average. The outcome of this evaluation (Table 2) demonstrates that the model successfully achieved an accuracy of 71% in predicting scores within the appropriate range, further validating the model's fundamental efficiency.

Model	Test Loss	Test Accuracy	Execution Time (minutes)
Bert For Sequence Classification - Hebrew	1.32	47.42	127.67

Range	Test Accuracy Range
Below Average [0-3]	69.60
Average [4-7]	71.00
Above Average [8-10]	72.17
Total Accuracy [All]	**71.04**

Table 2: Hebrew Model Results: Transfer Learning from English to Hebrew

To conclude, by using transfer learning, we effectively utilized the knowledge from the English model while achieving promising results on the Hebrew dataset. This demonstrates the potential of leveraging multilingual embeddings for cross-lingual applications, particularly in Automated Essay Scoring.

4 Discussion

This paper aimed to enhance the performance of Automated Essay Scoring in the Hebrew language through cross-lingual transfer learning from an English language model.

Our achieved results showcase positive performance. The English model demonstrates impressive accuracy, and the Hebrew model establishes a strong foundation while achieving good range accuracy. These results made possible through straightforward translation and meticulous model fine-tuning, establish a robust framework for Automated Essay Scoring in both English and Hebrew.

Notably, the notebook's "*Automated Essay Scoring Prediction*" section, demonstrates good predictions for English and Hebrew essays across different ranges (below average, average, and above average), and highlights the model's versatility across languages.

Throughout the experimental process, several key insights were gained: **The Challenge of Essay Scoring**: Training a classification model for essay scoring proved to be a challenging task, particularly when dealing with long texts and maintaining context across languages. The process involved crucial decisions, including text embedding and model selection, which required extensive exploration and experimentation. **The superiority of BERT Embeddings**: Among the various text embedding algorithms explored, BERT appeared as the most effective method. BERT excelled in capturing contextual information and producing accurate embeddings, with minimal text preprocessing required. **Limitations of Basic Machine Learning Models**: The application of basic machine learning models for Automated Essay Scoring resulted in suboptimal performance, indicating the need for more robust and sophisticated models capable of capturing the complexities of essay scoring. **Insights from Neural Network Models**: These results highlight the significance of neural network

models, particularly the BERT-based approach, in effectively addressing the automated essay scoring task for the English language. Surprisingly, the LSTM model did not perform well, despite its capabilities for handling sequence data in NLP tasks. On a positive note, both the attention mechanism and linear models exhibited promise as potential alternatives for this task.

In the context of this task, the lack of Hebrew language data proved to be a critical challenge. The process of translating essays from English to Hebrew while preserving their context and inherent patterns posed significant difficulties. However, despite these challenges, the use of heBERT embeddings showcased its robust capabilities in generating high-quality embeddings that effectively retained contextual information. Nevertheless, it is important to acknowledge that working with the Hebrew language in this task presented its own set of obstacles, particularly when dealing with lengthy texts like essays.

Overall, the exploration of various techniques and model architectures in this study shed light on the potential of transfer learning and deep learning approaches for cross-lingual Automated Essay Scoring. The use of BERT embeddings and the implementation of Linear models with attention mechanisms showed promising results in enhancing essay scoring performance in the English and Hebrew languages.

4.1 Future Work

In future work, we aim to collect a reliable dataset of Hebrew essays with corresponding scores. Furthermore, an additional normalization process for the scores can enhance the evaluation process's accuracy and fairness (defining 3 distinct essay classes: "poor/below average" [0-3], "average" [4-7], and "good" [8-10] essays). Moreover, we will explore the potential of adding more layers to the BERT classifier during fine-tuning, avoiding the freezing of the last layers, to access more nuanced patterns and features and potentially achieve superior performance in essay scoring.

5 Code

The code for this research project, including the implementation of the models, data processing, evaluation, and prediction, can be found in the following link.