IRE Major Project Report

Automatic Text Scoring

Team

Akhilesh Soni (201530001)

Sagar Thakur (20172103)

Sai Teja Reddy (201564086)

Suma Reddy Duggenpudi (201525145)

Aim of the Project :

The aim of this project is to develop and end-to-end model which can perform **automatic text scoring on essays** and to create an **user interface** which enhances the usage of the model developed.

Introduction :

Automated Text Scoring (ATS) systems are targeted at both alleviating the workload of the teachers and improving the feedback cycle in educational systems. Traditionally, the task of ATS has been regarded as a machine learning problem which learns to approximate the marking process with supervised learning. This mainly involves trying to extract some handcrafted and standard features of the essays and then passed into a machine learning based classifier. These features were mostly some basic features like essay length, sentence length, grammar correctness, readability etc.

Deep learning based ATS systems demonstrated that neural network architectures such as LSTM and CNN are capable of outperforming systems that extensively required handcrafted features. However, **these models do not consider Logical Flow and coherence over time**.

Semantic compositionality is modelled within the recursive operations in LSTM model which compresses the input text repeatedly within the recurrent cell. In this case, **the relationship between various consecutive sentences (multiple points) in the essay cannot be captured** effectively. But essays are typically long sequences of sentences and **the complete information lies in the continuity of these sentences** one after the other. But this pushes the limits of memorization capability of LSTM.

So finally the main aim of this implementation is to solve the above problems. This first is to alleviate the inability of current neural network architecture to model the flow of the essay, **coherence of the text and semantic relatedness** over the course of the essay. And the second is to **easing the burden of the recurrent model**.

And accordingly, the model tries to read the essay and **capture the semantic relationship between two points of an essay using a neural tensor layer**. And finally, multiple features of semantic relatedness are aggregated across the essay and used as auxiliary features of prediction.

The semantic relationship between two points is important because it can be an indicator of the writing flow and textual coherence. And these auxiliary features aim to capture the **logical and semantic flow of the essay**. And secondly, these additional parameters from the external tensor serve as **an auxiliary memory for the network** and which can in turn improve the performance of the deep architecture by allowing access to intermediate states. And finally, the architecture performs sentence modeling and semantic matching in a unified end-to-end framework.

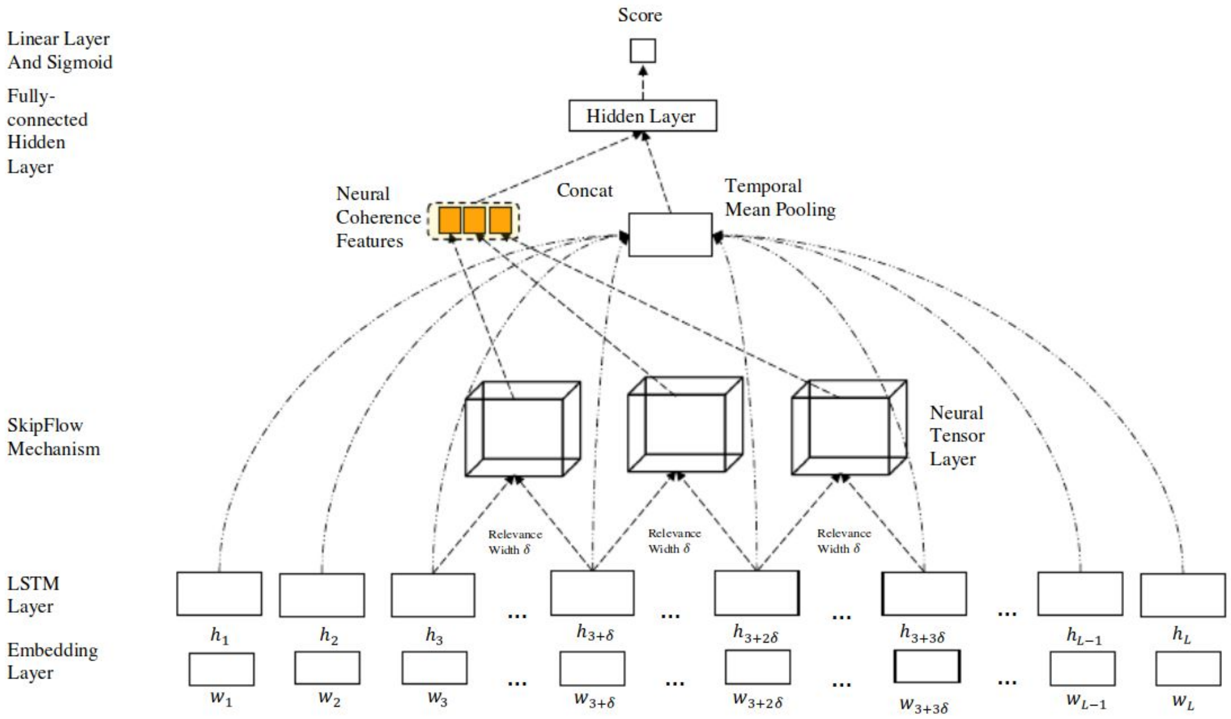
About the model - SKIPFLOW LSTM :

In the proposed architecture, **SKIPFLOW LSTM** adopts parameterized tensor compositions to model **the relationships between different points within an essay**.

This in turn generates neural coherence features that can support predictions. As said above, **it mainly models coherence and semantic relatedness over time**. This is done in the following way :

- It reads the essay and **models semantic relationships between two points** of an essay using Neural Tensor Layer.
- It generates **Neural Coherence Features by performing semantic matching** k-times while reading.
- It models the relationship between distant states with additional parameters which **enhances memorization and improves performance** of the deep architecture by allowing access to intermediate states. This eases the burden and provides protection against vanishing gradient by exposing hidden states to deeper layers.
- Most importantly, it performs **semantic matching and semantic modeling.**

Model Architecture of SKIPFLOW:



- **Embedding layer:** The model accepts an easy and the target score as a training instance. And the embedding layer is used to represent each essay as a fixed-length sequence in which we pad all sequences to the maximum length. This helps us obtain word embeddings.
- We have used Glove Embeddings in our Model, instead of a training an embedding layer.
- **Long Short-Term Memory (LSTM):** The sequence of word embeddings obtained from the embedding layer is then passed into a long short-term memory network. At every time step *t*, LSTM outputs a hidden vector h_t that reflects the semantic representation of the essay at position *t*. And to select the final representation of the essay, a temporal mean pool is applied to all LSTM outputs.

- **Neural Tensor Layer:** A tensor layer is used to model the relationship between two LSTM outputs. It is a parameterized composition which is defined using :

  $$s_i(a,b) = \sigma(u^T f(v_a^T M^{[1:k]} v_b + V[v_a, v_b] + b))$$

  The vector outputs of LSTM at two time steps of δ -width apart are passed through neural tensor layer and it returns a similarity score that determines the coherence feature between the two vectors. The usage of bilinear product enables interaction between vectors through a similarity matrix. This enables a rich interaction between hidden representations. Moreover, the usage of multiple slices(k) encourages different aspects of this relation to be modeled.

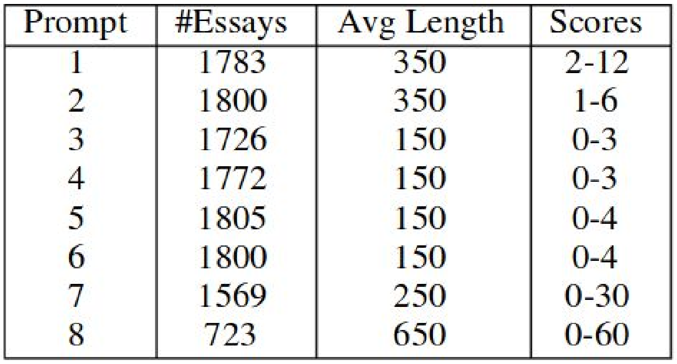
- **Fully-Connected Hidden Layer:** All the scalar values that are obtained from the tensor layer are concatenated together to form the neural coherence feature vector. The essay representation which obtained from a mean pooling over all hidden states is then concatenated with the coherence feature vector. This is then given as an input to the fully connected hidden layer.
- **Linear Layer with Sigmoid:** This is the final layer. The output at this layer is the normalised score of the essay.
- **Learning and Optimization:** Network optimizes the mean-squared error.

Tools Used:

- **Keras + Tensorflow** - for implementing the model and training the model using the ASAP dataset.
- **NLTK Toolkit** - for preprocessing the data.
- **Glove Embeddings** - for the embedding layer

Dataset:

We use the **ASAP (Automated Student Assessment Prize) dataset** from kaggle for experimental evaluation. This dataset contains **eight essay sets** and each of it has its own unique characteristics. The average length of each essay varies from 150 to 650 words. The statistics of the ASAP dataset are as follows :



Prompt	#Essays	Avg Length	Scores
1	1783	350	2-12
2	1800	350	1-6
3	1726	150	0-3
4	1772	150	0-3
5	1805	150	0-4
6	1800	150	0-4
7	1569	250	0-30
8	723	650	0-60

In the above table, the first column represents the type of essay from the eight types. The second column is about the number of essays of each type. The third column gives the information about the average length of essays of that particular type and the last column, scores, denotes the range of possible marks in the dataset.

Evaluation Metric:

Firstly, the experimental setup is a 5-fold cross validation to evaluation all systems with a 60-20-20 split for train, validation and test sets. We trained all models for **1000 epochs or lesser** depending on whether the model keeps improving its performance. For word processing, we used **tokenizer of NLTK library**. The evaluation metric used was the **Quadratic Weighted Kappa (QWK)** which measures agreement between raters and it is a commonly used metric for ATS systems.

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}$$

Goals Achieved

1. Finished the implementation of the complete end-to-end system.
2. Compared the results of the paper with the metrics obtained for the built system.
3. *Built an **interactive demo*** (web-based) for users to input essays and get the predicted ATS score.

Results:

Type of essay	**SKIPFLOW LSTM Architecture QWK Score**
1	80.1
2	66.67
3	70.84
4	84.71
5	80.88
6	78.83
7	76.47
8	68.81
Average	75.91

Conclusion:

Thus this system serves the purpose of score prediction for a given essay. This has been mainly done by **incorporating the intuition of textual coherence** in neural ATS system. SKIPFLOW architecture adopts **parameterized tensor compositions** to model the **relationships between different points** within an essay, generating **neural coherence features** that can support predictions. These neural coherence features when combined with LSTM sentence representations can produces significantly better results.

Demo/Video: https://youtu.be/bm6BvYx6AAQ

Code : https://github.com/Saiteja-Reddy/Automatic-Text-Scoring

Website: https://saiteja-reddy.github.io/ATS/

References:

- SKIPFLOW: Incorporating Neural Coherence Features for End-to-End Automatic Text Scoring Paper .
- Automatic Student Assessment Prize (ASAP) Dataset from Kaggle
- Automatic Text Scoring Using Neural Networks