Automatic Text Scoring

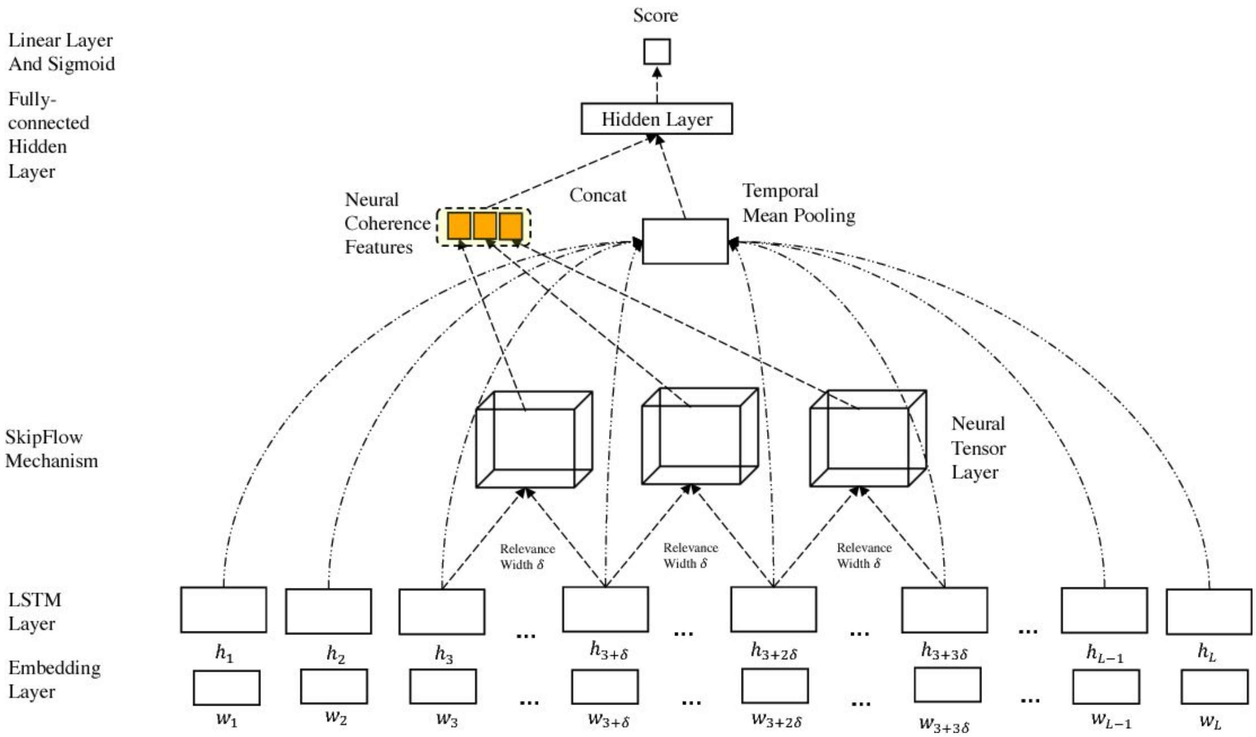
30.03.2018

G. Rama Rohit Reddy - 201525083

Yashwanth Regatte - 201525036

Implementation

In this project we would like to primarily implement paper "**SKIPFLOW: Incorporating Neural Coherence Features for End-to-End Automatic Text Scoring**" and we would like to dwell more into the topic if time permits.



Some of our ideas to implement apart from paper are:

- Adding additional layers in SkipFlow model like
  - **Convolution layer** before LSTM layer because it might be beneficial for the network to extract *local features* from the sequence before applying lstm operation. This optional characteristic can be achieved by applying a convolution layer on the output of the Embedding layer.
  - **Use of Bi-directional LSTM:** Training the LSTM in a uni-directional manner (i.e from left to right) might leave out important information about the sentence. For example, our interpretation of a word at some point t_i might be different once we know the word at t_{i+5}. An effective way to get around this issue can be to train the LSTM in a bidirectional manner.
  - Try different **Pre-trained word embeddings.**
  - Try **doc2vec** as additional feature.
- Use of state-of-art features and concatenate them with output of fully connected layer and try to increase the accuracy, if we have enough time.
  - Length Features:
    - Number of words;
    - Number of sentences;
    - The average sentence length, taking the number of words as a consideration;
    - Average word length, considering the number of characters;

- Words with a long number of words and a number of characters greater than 7 are counted as long words;
- Stop words;
- The number of characters;

- Occurrence Features:
    - *Comma, Quotation marks, Number of exclamation marks* : The comma number reflects the internal complexity of the sentence to a certain extent; the number of quotes, reflects the student's ability to negotiate and quote; the number of exclamation marks, reflects the proportion of wonderful sentences.
- Style Features:
    - F value;
    - The total number of different forms of words and the total number of different words;
    - Average word frequency;
- Error Features:
    - Misspelled word
    - Grammatical errors

Introduction

Automated Text Scoring (ATS) systems are targeted at both alleviating the workload of teachers and improving the feedback cycle in educational systems.

Decades of ATS research follow the same traditional supervised text regression methods in which handcrafted features are constructed and subsequently passed into a machine learning based classifier. Simple and intuitive features may include essay length, sentence length. On the other hand, intricate and complex features may also be extracted, e.g, features such as grammar correctness, readability and textual coherence. However, these **handcrafted features** are often painstakingly designed, require a **lot of human involvement** and usually require **laborious implementation** for every new feature.

Semantic similarity and **textual coherence** are very important in text scoring.

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, all these neural models **do not consider Logical Flow** and **coherence over time**. **Semantic compositionality** is modeled within the recursive operations in **LSTM model** which compresses the input text repeatedly within the recurrent cell. In this case, **the relationship between multiple points in the essay cannot be captured effectively**. **Essays** are typically **long sequences** which pushes the limits of **memorization capability of LSTM**.

SKIP FLOW LSTM, adopts parameterized tensor compositions to model the relationships between different points within an essay, generating neural coherence features that can support predictions.

SKIPFLOW models **coherence** and **semantic relatedness over time** in the following way:

- It reads the essay, and models semantic relationships between two points of an essay using **Neural Tensor Layer. Semantic relationships across sentences** are commonly used as an **indicator of writing flow and textual coherence**. **Auxiliary features (generated end-to-end)** aim to capture the logical and semantic flow of an essay. This also provides a measure of semantic similarity aside from the flavor of semantic compositionality modeled by the base LSTM model.
- It generates **Neural Coherence features** by performing semantic matching k times while reading.
- By **modeling the relationship between distant states** with additional parameters can **enhance memorization and improve 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.
- It performs **sentence modeling** (compositional reading) and **semantic matching** in a unified end-to-end framework.

Model Architecture

1. **Embedding Layer:** Each essay is represented as a fixed-length sequence in which we pad all sequences to the maximum length.
2. **Long Short-Term Memory (LSTM)**: LSTM outputs a hidden vector h_t that reflects the semantic representation of the essay at position t. To select the final representation of the essay, a temporal mean pool is applied to all LSTM outputs.
3. **Neural Tensor Layer:** $s_i(a, b) = \sigma (u^T f(v_a^T M^{[1:k]} v_b + V[v_a , v_b] + b))$
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.
4. **Fully-connected Hidden Layer:** Neural coherent features concatenated with LSTM outputs (i.e sentence representation) is provided as input to this layer.
5. **Linear Layer with Sigmoid:** The output at this final layer is the normalized score of the essay.

6. **Learning and Optimization:** Network optimizes the mean-square error.

Dataset

We use the ASAP (Automated Student Assessment Prize) dataset from kaggle for experimental evaluation.

Evaluation Metric

The evaluation metric used is the Quadratic Weighted Kappa (QWK) which measures agreement between raters and is a commonly used metric for ATS systems.

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

Conclusion

SKIP FLOW 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. We also try to improve results by adding additional features and layers as mentioned above if time permits.

Milestones

I. Paper Implementation

Implement the paper mentioned above.

II. Above mentioned Ideas

Implement above mentioned ideas if time permits.

Results

Model	1	2	3	4	5	6	7	8	Average
SKIPFLOW LSTM	0.81	0.646	0.6587	0.76	0.7863	0.766	0.782	0.66	0.733