Augmenting Textual Qualitative Features in Deep Convolution Recurrent Neural Network for Automatic Essay Scoring

Tirthankar Dasgupta, **Abir Naskar**, **Rupsa Saha** and **Lipika Dey**

TCS Innovation Lab, India
(dasgupta.tirthankar, abir.naskar, rupsa.s, lipika.dey)@tcs.com

Abstract

In this paper we present a qualitatively enhanced deep convolution recurrent neural network for computing the quality of a text in an automatic essay scoring task. The novelty of the work lies in the fact that instead of considering only the word and sentence representation of a text, we try to augment the different complex linguistic, cognitive and psychological features associated within a text document along with a hierarchical convolution recurrent neural network framework. Our preliminary investigation shows that incorporation of such qualitative feature vectors along with standard word/sentence embeddings can give us better understanding about improving the overall evaluation of the input essays.

1 Introduction

The quality of text depends upon a number of linguistic factors, corresponding to different textual properties, such as grammar, vocabulary, style, topic relevance, clarity, comprehensibility, informativeness, lexical diversity, discourse coherence, and cohesion (Crossley et al., 2008)(McNamara et al., 2002). In addition, there are deep cognitive and psychological features, such as types of syntactic constructions, grammatical relations and measures of sentence complexity, that make automatic analysis of text quality a non-trivial task.

Developing tools for automatic text quality analysis have become extremely important to organizations that need to assess writing skills among adults and students on a regular basis. Because of the high participation in such assessments, the amount of time and effort required to grade the large volume of textual data generated is too high to be feasible by a human evaluator. Manual evaluation processes by multiple evaluators may also be prone to erroneous judgments due to mutual disagreements between the evaluators. Therefore, developing a means through which such essays can be automatically scored, with minimum human interference, seem to be the best way forward to meet the growing demands of the education world, while keeping inter-evaluator disagreements to a minimum. Automatic Essay Scoring (AES) systems have thus been in the research focus of multiple organizations to counter the above issues (Landauer, 2003).

A typical AES system takes as input an essay written on a specific topic. The system then assigns a numeric score to the essay reflecting its quality, based on its content, grammar, organization and other factors discussed above.

A plethora of research have been done to develop AES systems on various languages (Taghipour and Ng, 2016; Dong et al., 2017; Alikaniotis et al., 2016; Attali and Burstein, 2004; Chen and He, 2013; Chen et al., 2010; Cummins et al., 2016). Most of these tools are based on regression methods applied to a set of carefully designed complex linguistic and cognitive features. Knowledge of such complex features have been shown to achieve performance that is indistinguishable from that of human examiners. However, since it is difficult to exhaustively enumerate all the multiple factors that influence the quality of texts, the challenge of automatically assigning a satisfactory score to an essay still remains.

Recent advancement in deep learning techniques have influenced researchers to apply them for AES tasks. The deep multi-layer neural networks can automatically learn useful features from data, with lower layers learning basic feature detectors and upper levels learning more high-level abstract features. Deep neural network models, however, do not allow us to identify and extract those properties of text that the network identi-

Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, pages 93–102
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

fies as discriminative (Alikaniotis et al., 2016). In particular, deep network models fail to take into account integral linguistic and cognitive factors present in text, which play an important role in an essay score assigned by experts. Such models emphasizes a simple uniform paradigm for NLP: "*language is just sequences of words*". While this approach has rapidly found enormous popularity and success, its limitations are now becoming more apparent. Gradually researchers stressing towards the importance of linguistic structure and the fact that it reduces the search space of possible outputs, making it easier to generate well-formed output (Lapata, 2017). Dyer (Dyer, 2017) also argued for the importance of incorporating linguistic structure into deep learning. He drew attention to the inductive biases inherent in the sequential approach, arguing that RNNs have an inductive bias towards sequential recency, while syntax-guided hierarchical architectures have an inductive bias towards syntactic recency. Several papers noted the apparent inability of RNNs to capture long-range dependencies, and obtained improvements using recursive models instead (Chen et al., 2017).

In order to overcome the aforementioned issues, in this paper we propose a qualitatively enhanced deep convolution recurrent neural network architecture for automatic scoring of essays. Our model takes into account both the word-level and sentence-level representations, as well as linguistic and psychological feature embeddings. To the best of our knowledge, no other prior work in this field has investigated the effectiveness of combining word and sentence embeddings with linguistic features for AES tasks. Our preliminary investigation shows that incorporation of linguistic feature vectors along with standard word/sentence embeddings do improve the overall scoring of the input essays.

The rest of the paper is organized as follows: Section 2 describes the recent state of art in AES systems. Our proposed Linguistically informed Convolution LSTM model architecture is discussed in Section 3, while section 4 has further details on generation of linguistic feature vectors. In section 5, we cover the experimentation and evaluation technique, reporting the obtained results in section 6, and finally concluding the paper in section 7.

2 Related Works

A plethora of attempts have been taken to develop AES systems over the years. A detailed overview of the early works on AES is reported in (Valenti et al., 2003). An Intelligent Essay Assessor (Foltz et al., 1999) was proposed more recently that uses Latent Semantic Analysis to compute the semantic similarity between texts. Lonsdale and Strong-Krause (Lonsdale and Strong-Krause, 2003) used the Link Grammar parser (Sleator and Temperley, 1995) to score texts based on average sentence-level scores calculated from the parser's cost vector. In Rudner and Liang's Bayesian Essay Test Scoring System (Rudner and Liang, 2002), stylistic features in a text are classified using a Naive Bayes classifier. Attali and Burstein's e-Rater (Attali and Burstein, 2004), includes aspects of grammar, vocabulary and style among other linguistic features, whose weights are fitted by regression. A weakly supervised bag-of-word approach was proposed by Chen et al. (Chen et al., 2010). A discriminative learning based approach was proposed by Yannakoudakis et al. (Yannakoudakis and Cummins, 2015) that extracts deep linguistic features and employs a discriminative learning-to-rank model that out-performs regression. Recently, Farra et al. (Farra et al., 2015) utilized variants of logistic and linear regression and developed scoring models. McNamara et al.'s hierarchical classification approach (McNamara et al., 2015) uses linguistic, semantic and rhetorical features. Despite the existing body of work, attempts to incorporate more diverse features to text scoring models are ongoing. (Klebanov and Flor, 2013) demonstrated improved performance by adding information about levels of association among word pairs in a given text. (Somasundaran et al., 2014) used the interaction of lexical chains with discourse elements for evaluating the quality of essays. Crossley et al. (Crossley et al., 2015) identified student attributes, such as standardized test scores, and used them in conjunction with textual features to develop essay scoring models. Readability features (Zesch et al., 2015) and text coherence have also been proposed as a source of information to assess the flow of information and argumentation of an essay (Chen and He, 2013). A detailed overview of the features used in AES systems can be found in (Zesch et al., 2015). Some attempts have been made to address different aspects of essay writing, like argument strength and organization, independently, through designing task-specific features for each aspect (Persing et al., 2010; Persing and Ng, 2015). There has been a lot of recent work in deep neural network models based on continuous-space representation of the input and non-linear functions. Recently, deep learning techniques have been applied to text analysis problems including AES systems (Alikaniotis et al., 2016; Dong and Zhang, 2016; Dong et al., 2017; Taghipour and Ng, 2016), giving better results compared to statistical models with handcrafted features (Dong and Zhang, 2016). Both recurrent neural networks (Williams and Zipser, 1989; Mikolov et al., 2010) and convolution neural networks (LeCun et al., 1998; Kim, 2014) have been used to automatically score input essays. In comparison to the work of Alikaniotis et al. (Alikaniotis et al., 2016) and Taghipour and Ng (Taghipour and Ng, 2016) that uses single-layer LSTM (Hochreiter and Schmidhuber, 1997) over the word embeddings for essay scoring, and Dong and Zhang (Dong and Zhang, 2016) used a two-level hierarchical CNN structure to model sentences and documents separately. More recently, (Dong et al., 2017) et al. proposed a hierarchical attention based CNN-LSTM model for automatic essay scoring.

Although the deep learning based approaches are reported to be performing better than the previous approaches, the performance may yet be bettered by the use of the complex linguistic and cognitive features that are important in modeling such texts. Our proposed system, takes into account both word and sentence level embeddings, as well as deep linguistic features available within the given text document and together learns the model. The detail architecture and working of the model is depicted in the following sections.

3 The Qualitatively Enhanced Convolution Recurrent Neural Network

As mentioned earlier, neural network based models are capable of modeling complex patterns in data and do not depend on manual engineering of features, but they do not consider the latent linguistic characteristics of a text. In this section, we will present a deep neural network based model that takes into account different complex linguistic, cognitive and psychological features associated within a text document along with a hierarchical convolution network connected with a bidirectional long-short term memory (LSTM) model (Hochreiter and Schmidhuber, 1997) (Schmidhuber et al., 2006). We will begin the model architecture by first explaining about generating the linguistic and psychological feature embeddings that will in turn be used by the neural network architecture.

3.1 Generating Linguistic and Psychological Feature Embeddings

We have used different linguistic and psychological features available within a text to augment them with the deep neural architecture.

The **psychological features** used in this work are mostly derived from Linguistic Information and Word Count (LIWC) tool (Tausczik and Pennebaker, 2010). The rapid development of AI, Internet technologies, social network, and elegant new statistical strategies have helped usher in a new age of the psychological study of language. By drawing on massive amounts of text, it is indeed possible to link everyday language use with behavioral and self-reported measures of personality, social behavior, and cognitive styles (Tausczik and Pennebaker, 2010). LIWC is a text analysis tool that counts words in psychologically meaningful categories. Empirical results using LIWC already demonstrated its ability to detect meaning in a wide variety of experimental settings, such as to show attentional focus, emotionality, social relationships, thinking styles, and individual differences.

The linguistic features we use to make our model linguistically informed are: *Part of Speech(POS)* (Manning et al., 2014), *Universal Dependency relations* (De Marneffe et al., 2006) , *Structural Well-formedness*, *Lexical Diversity*, *Sentence Cohesion*, *Causality* and *Informativeness of the text*.

The **lexical diversity** of a given text is defined as the ratio of different unique word stems (types) to the total number of words (tokens). According to Jarvis's model (Jarvis, 2002), lexical diversity includes six properties that are measured by the indices discussed in Table 1.

We device a novel algorithm to determine **cohesion** between sentences in a document. The algorithm follows the following steps: a) identify the GloVe word embeddings(Pennington et al., 2014) of each constituent word of two sentences $S_1, /S_2$.

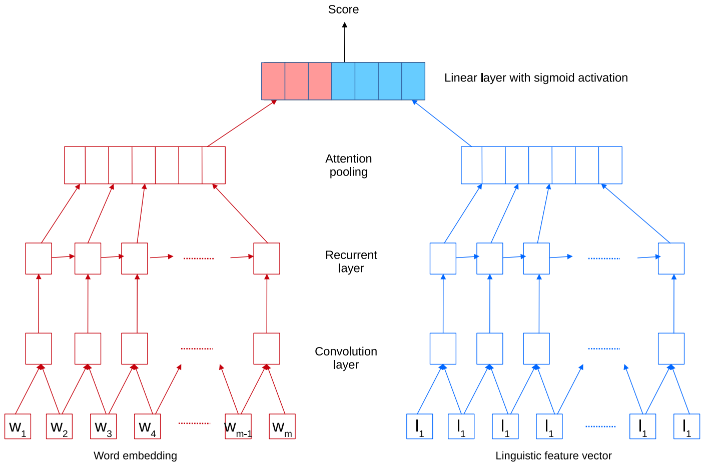

Figure 1: Overview of the qualitatively enhanced convolution recurrent neural network for AES.

Table 1: Lexical Diversity Indices

Property	Measure
Variability	Measure of Textual Lexical Diversity (MTLD)
Volume	Total number of words in the text
Evenness	Standard deviation of tokens per type
Rarity	Mean BNC rank
Dispersion	Mean distance between tokens of type
Disparity	Mean number of words per sense

b) create sentence embeddings by computing a tensor product between the individual word embeddings. For example, given two sentences S_1 and S_2 $S_1 = w_1, w_2..., w_i$ and $S_2 = w'_1, w'_2, ...w'_j$, where $w_1, w_2, ...w_k$ and $w'_1, w'_2...w'_k$ are the word embeddings of S_1 and S_2. Sentence embedding $SE(S_1)$ is $(w_1 \bigotimes w_2) \bigotimes w_3)... \bigotimes w_k)$. Where $\bigotimes$ refers to the tensor product of each adjacent word embedding pairs in S_1. Similarly for sentence S_2. c) define $A\ and\ B$ as the number of word embeddings in $S_1\ and S_2$ respectively. d) the cohesion score between S_1 and S_2 can be computed as $coh(S_1, S_2) = \frac{(S' + Sim(p_1, p_2))}{N_1 + 1}$ The expression N_1 represents $A \cup B$. S' and S'' are computed as: $S' = \sum_{\forall w_i \in C_1} S_{w_i}$ Where, $S_{w_i} = \max_{\forall w'_j \in C_2}(Sim(w_i, w'_j))$ p_1 and p_2 are sentence embeddings of S_1 and S_2 respectively, and $Sim(x, y)$ is the cosine similarity between two vector V_i and V_j.

To indicate presence of **causality**, we use the semantic features as identified by Girju (Girju, 2003) - nine noun hierarchies (H(1) to H(9)) in WordNet, namely, *entity, psychological feature, abstraction, state, event, act, group, possession, and phenomenon*. A single feature Primary Causal Class (PCC) is defined for a word w_i. If $w_i \in H_i$ where H_i is as defined, $PCC = H_i$, else $PCC = null$. Another feature, Secondary Causal Class(SCC) is also defined. This takes value $H(i)$ if any WordNet synonym of the word belongs to $H(i)$, and is $Null$ otherwise.

The **informativeness of a text** refers to how much information is present in a text with respect to a given collection. We have introduced an information theoretic approach towards determining such informativeness in text. We consider each document d, represented by a bag-of-word as, $< (q_1, w_1), (q_2, w_2), ..., (q_n, w_n) >$ where q_i is the i^{th} unique term in document d and w_i is the corresponding weight computed with respect to a collection of documents C. The Informativeness score $NS(d, C)$ of each new text document d, is computed with respect to the collection C, indicating the informativeness of d amongst C. In the described context, we declare a document d_i as informative when the corresponding $NS(d_i, C)$ is higher than a threshold θ. We have defined the informativeness of d in terms of its information content (IC). Information content is a heuristic measure for term specificity and is a function of term use. Our idea is to therefore use it as an estimator of informativeness *an informative document is more likely to use unique vocabulary than other documents*. We compute the information content of a document in terms of its Entropy. We define the entropy of a text T, with N words out of which n are unique, as:$E_T(p_1, p_2, ..., p_n) = \frac{1}{N}\sum_{i=1}^{n}(p_i * (\log_{10} N - \log_{10} p_i)$. $p_i(i = 1...n)$ is the probabilistic measure of the specificity of the i^{th} word in the T. The technique to compute term specificity is discussed below. In order to avoid the problem of zero probabilities, we have used linear interpolation smoothing, where document weights are smoothed against the set of the documents in the corpus. Then the probabilities are defined as:$\theta_{d_n}(q) = \lambda * \theta_d(q) + (1-\lambda) * \theta_{d_1} ... \theta_{d_n}(q)$. Where, $\lambda \in [0, 1]$ is the smoothing parameter and is the probability of term q in the corpus C. In our experiments, λ was set to 0.9.

As discussed earlier, the cornerstone of our informativeness prediction engine is to compute the rarity of a document, which can, in turn, be computed by determining the rarity of individual terms. Accordingly, we have applied the principle of Inverse Document Frequency (IDF) (Karkali et al., 2014). Aggregating all the IDF of the terms of a given document may led us to a better estimator of the documents Informativeness. IDF is originally defined as,$IDF(q, C) = \log(\frac{N}{df_q})$ where, q is the term in hand, df_q is the document frequency of the term q across the corpus C and N is the total number of documents in the collection. On the other hand, in probabilistic terms IDF can be computed as:$IDF_p(q, c) = \log(\frac{N-df_q}{df_q})$.

3.2 Model architecture

The proposed linguistically informed convolution recurrent neural network architecture that we have used in this paper is illustrated in Figure 1. In the next few subsections, we describe each layer in detail.

Generating Embeddings: Pre-trained GloVe word vector representations of dimension 300 have been used for this work (Pennington et al., 2014) for the word embeddings. Similarly we have constructed a pre-trained sentence vectors. The Sentence vectors from each input essay is appended with the vector formed from the linguistic features identified for that particular sentence.

Convolution Layer: Since convolution networks works best in determining local features from texts, it is important to feed each of the generated word embeddings to a convolution layer. Accordingly, the convolution layer applies a linear transformation to all K windows in the given sequence of vectors. We perform a zero padding to ensure the same dimensionality between the input and output vectors. Therefore, given a word representations $X_1, X_2, ...X_l$, the convolution layer first concatenates these vectors to form a vector $\bar{x}$ of length $l.d_{LT}$ and then uses $Conv(\bar{x}) = W.\bar{(}x) + b$ to calculate the output vector of length d_c. Where, **W** and **b** are the weights that the network learns.

Long short-term memory In AES systems, the surrounding context is of paramount information. While typical LSTMs allow the preceding elements to be considered as context for an element under scrutiny, we prefer to use bidirectional LSTMs (Bi-LSTM) networks (Graves et al., 2012) that are connected so that both future and past sequence context (i.e. both preceding and succeeding elements) can be examined. Corresponding to each input text, we determine the word embedding representation (W_e) of each word of the text and the different linguistic feature embeddings (W_l). The input to the Bi-LSTM unit is an embedding vector E which is the composition of W_e and W_l, i.e. $\overrightarrow{E} = \overrightarrow{W_e} \bigotimes \overrightarrow{W_l}$

Activation layer: After obtaining the intermediate hidden layers from the Bi-LSTM layer $h_1, h_2, ..., h_T$, we use an attention pooling layer over the sentence representations. The attention pooling helps to acquire the weights of sentence contribution to final quality of the text. The attention pooling over sentences is represented as: $a_i = \tanh(W_a.h_i + b_a)$, $\alpha_i = \frac{e^{w_\alpha \cdot a_i}}{\sum e^{w_\alpha \cdot a_i}}$, $O = \sum(\alpha_i.h_i)$. Where W_a, w_α are weight matrix and vector respectively, b_a is the bias vector, a_i is attention vector for i-th sentence, and α_i is the attention weight of i-th sentence. O is the final text representation, which is the weighted sum of all the sentence vectors.

The Sigmoid Activation Function: The linear layer performs a linear transformation of the input vector that maps it to a continuous scalar value. We apply a sigmoid function to limit the possible scores to the range $[0, 1]$. The mapping of the linear layer after applying the sigmoid activation function is given by $s(x) = sigmoid(w.x + b)$. Where, x is the input vector, w is the weight vector, and b is bias value. We normalize all gold-standard scores to $[0, 1]$ and use them to train the network. However, during testing, we rescale the output of the network to the original score range and use the rescaled scores to evaluate the system.

Table 2: Statistics of the Kaggle dataset; Range:score range and Med: median scores.

Set	#Essays	Genere	Avg. Len.	Range	Med.
1	1783	ARG	350	2-12	8
2	1800	ARG	350	1-6	3
3	1726	RES	150	0-3	1
4	1772	RES	150	0-3	1
5	1805	RES	150	0-4	2
6	1800	RES	150	0-4	2
7	1569	NAR	250	0-30	16
8	723	NAR	650	0-60	36

Table 3: Hyper-parameters

Layer	Parameter Name	Parameter Value
Lookup	Word embedding dim	50
CNN	Window size	5
	No. of filters	100
Bi-LSTM	Hidden units	100
Dropout	Dropout rate	1.0
	Epochs	200
	Batch size	10
	Initial learning rate η	0.001
	Momentum	0.9

4 Experiments

4.1 Dataset

An Automated Student Assessment Prize (ASAP) contest was hosted at Kaggle in 2012. It was supported by the Hewlett Foundation, aiming to explore the capabilities of automated text scoring systems (Shermis and Burstein, 2013). The dataset released consists of around twenty thousand texts (60% of which are marked), produced by middle-school English-speaking students, which we use as part of our experiments to develop our models. In order to train and test the proposed models, we have used the same dataset as published at the Kaggle challenge. Table 2 reports some of the basic statistics about the dataset. Due to the unavailability of the testing set, we have performed a 7-fold cross validation to evaluate our proposed models. In each fold, 80% of the data is used for training, 10% as the development set, and 10% as the test set. We train the model for a fixed number of epochs (around 8000) and then choose the best model based on the development set. We have used the NLTK toolkit to perform various NLP tasks over the given dataset. For ease of experimentation, we have further normalized the expert scores (gold-standard scores) to the range of $[0, 1]$. During testing, we rescale the system-generated normalized scores to the original range of scores and measure the performance.

4.2 Training and parameter estimation

For a given learning function our goal is to minimize the mean squared error (MSE) rate. Accordingly, we have used the RMSProp optimization algorithm (Dauphin et al., 2015) to minimize the mean squared error (MSE) loss function over the training data. This is represented as: $MSE(s^*, s) = \frac{1}{N} * \sum_{i=1}^{N}(s_i - s_i^*)^2$. Therefore, given N training samples and their corresponding expert generated scores p_i^* normalized within a range of $[0.1]$, the model computes the predicted scores p_i for all training essays and then updates the network parameters such that the mean squared error is minimized.

The 10% data kept for development is used to identify the different hyper-parameters for the models. There are several hyper-parameters that need to be set. We use the RMSProp optimizer with decay rate (ρ) set to 0.9 to train the network and we set the base learning rate to 0.001. The mini-batch size is 64 in our experiments and we train the network for 400 epochs. We have also make use of dropout regularization (Srivastava et al., 2014) to avoid over-fitting. We also clip the gradient if the norm of the gradient is larger than a threshold. We do not use any early stopping methods, instead, we train the neural network model for a fixed number of epochs and monitor the performance of the model on the development set after each epoch. Once training is finished, we select the model with the best QWK score on the development set. During training, the norm of the gradient is clipped to a maximum value of 10. We set the word embedding dimension (d_LT) to 50 and the output dimension of the recurrent layer (d_r) to 300. For the convolution layer, the window size (l) is set to 5 and the output dimension of this layer (d_c) is set to 50. The details of the hyper-parameters are summarized in Table 3.

4.3 Evaluation

In past literature, a number of techniques were used to measure the quality of AES systems. This includes Pearson's correlation r, Spearman's ranking correlation ρ, Kendall's Tau and kappa, and quadratic weighted kappa (QWK). (Alikaniotis et al., 2016) proposed to evaluate their model in terms of the first three parameters, whereas works of (Taghipour and Ng, 2016; Dong and Zhang, 2016; Dong et al., 2017) uses QWK as the evaluation criteria. This is primarily due to the fact that

Table 4: Comparing the performance of the present system with that of the state-of-the-art

Models/Prompts	1	2	3	4	5	6	7	8	AVG QWK
EASE (BLRR)	0.761	0.606	0.621	0.742	0.784	0.775	0.730	0.617	0.705
CNN	0.797	0.634	0.646	0.767	0.746	0.757	0.746	0.687	0.722
LSTM	0.775	0.687	0.683	0.795	0.818	0.813	0.805	0.594	0.746
LSTM-CNN	0.821	0.688	0.694	0.805	0.807	0.819	0.808	0.644	0.761
LSTM-MoT	0.818	0.688	0.679	0.805	0.808	0.817	0.797	0.527	0.742
CNN-CNN-MoT	0.805	0.613	0.662	0.778	0.800	0.809	0.758	0.644	0.734
LSTM-CNN-att	0.822	0.682	0.672	0.814	0.803	0.811	0.801	0.705	0.764
Qe-C-LSTM	0.799	0.631	0.712	0.711	0.801	0.831	0.815	0.695	0.786

Table 5: Comparing performance of the proposed model taking all the prompts together with that of the existing models

Models	Pearson's r	Spearman's ρ	RMSE	Cohen's κ
doc2vec	0.63	0.62	4.43	0.85
SVM	0.77	0.78	8.85	0.75
LSTM	0.60	0.59	6.80	0.54
Bi-LSTM	0.5	0.70	7.32	0.36
word2vec + Bi-LSTM	0.86	0.75	4.34	0.85
SSWE+ Bi-LSTM	0.92	0.80	3.21	0.95
SSWE+ Two-layer Bi-LSTM	0.96	0.91	2.40	0.96
Qe-C-LSTM	0.97	0.94	2.1	0.97

the Automated Student Assessment Prize (ASAP) competition official criteria takes QWK as evaluation metric.

The QWK statistics or its other variants are widely used to measure inter-rater agreement of the annotators or experts. In our case inter-raters refer to the human rater and the system predicted ratings. QWK is modified from kappa which takes quadratic weights. The quadratic weight matrix in QWK is defined as: $W_{i,j} = \frac{(i-j)^2}{(R-1)^2}$, where i and j are the reference rating (assigned by a human rater) and the system rating (assigned by an AES system), respectively, and R is the number of possible ratings.

An observed agreement score O is calculated such that $O_{i,j}$ refers to the number of essays that receive a rating i by the human rater and a rating j by the AES system. An expected score E is calculated as the outer product of the two ratings. Finally, given the three matrices W, O, and E, the QWK value is calculated as:$\kappa = 1 - \frac{\sum(W_{i,j} * O_{i,j})}{\sum(W_{i,j} * E_{i,j})}$

5 Results

We evaluate the performance of our proposed model by comparing it with some of the well known state-of-the-art models. These models are: a) the publicly available 'Enhanced AI Scoring Engine' (EASE[1]). EASE is based on hand-crafted linguistic features and regression methods including support vector regression (SVR) and Bayesian linear ridge regression (BLRR). In the present paper we have used only the BLRR model as our baseline systems due to its improved performance in comparison to the SVR model. b) The LSTM-MoT models proposed by (Taghipour and Ng, 2016). c) the Attention-based Recurrent Convolution Neural Network model proposed by (Dong et al., 2017). d) The hierarchical CNN (CNN-CNN-MoT)(Dong and Zhang, 2016) and e) the hierarchical CNN layer with LSTM along with an additional attention layer (CNN-LSTM-att) (Dong and Zhang, 2016) (Dong et al., 2017) as our baselines.

The LSTM-MoT uses one layer of LSTM over the word embeddings, and takes the average pooling over all time-step states as the final text representation, which is called Mean-over-Time (MoT) pooling (Taghipour and Ng, 2016). Next, a linear layer with sigmoid function follows the MoT

[1] https://github.com/edx/ease

layer to predict the score of an essay script. On the other hand, CNN-CNN-MoT uses two layers of CNN, in which one layer operates over each sentence to obtain representation for each sentence and the other CNN is stacked above, followed by mean-over-time pooling to get the final text representation. Similarly, the CNN-LSTM-att model uses hierarchical architecture with the CNN layer followed by an LSTM layer attached with an attention layer instead of the MoT layer(Dong et al., 2017).

Table 4 reports the comparison of the performance of our system and the existing baselines by taking the eight prompts from the Kaggle ASAP dataset individually. In general we can observe that our proposed performance of the proposed Qe-CLSTM model is comparable to that of the existing baseline systems. However, in certain cases it outperforms all the base-line models. For example, in prompt 3, 6 and 7 we have achieved an QWK of 0.712, 0.831 and 0.815 respectively as compared to the best reported average QWK score of 0.694, 0.827 and .0.811 respectively for the 10 fold run of CNN-LSTM and LSTM only.

It is worth mentioning here that all these models are compared with respect to the QWK score. On the other hand, we have also used evaluation matrices like, Pearson's correlation r, Spearman's ranking correlation ρ, RMSE scores in order to compare our model with systems proposed by (Alikaniotis et al., 2016).

Table 5 shows the comparison of the performance of our system and the existing baselines by taking all the prompts together. We have compared the systems with respect to the different models as discussed in 5. We found that that in terms of all these parameters our system performs better than the existing, LSTM, Bi-LSTM and EASE models. We have achieved a Pearson's and Spearman's correlation of 0.94 and 0.97 respectively as compared to that of 0.91 and 0.96 in (Alikaniotis et al., 2016). We also achieved and RMSE score of 2.09. We also compute a pair wise Cohen's κ value of 0.97.

Apart from scoring each of the individual essays, we also tried to analyze some of the typical cases where our model fails to predict the desired output. Figure 2 shows the general distribution of difference in average expert score and the system predicted score. We observe a minimum difference of 0 and maximum difference of 20 with median of 1 and average of 1.08. In 82% cases the difference lies between the range of [0,1].

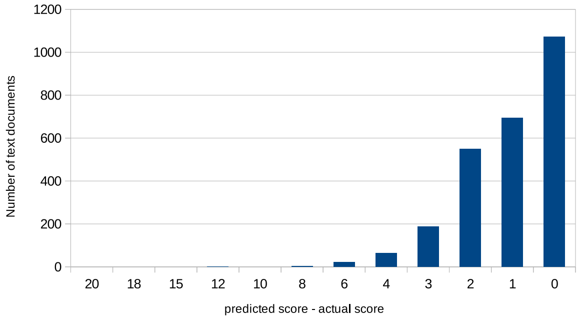

Figure 2: Distribution of difference in predicted scores with respect to the actual score

6 Conclusion

In this paper, we have proposed a novel technique that uses deep neural network model to perform Automatic Essay Assessment task. The traditional way of applying deep neural nets like CNN, LSTM or their other forms fails to identify the interconnection between the different factors involved in assessing the quality of a text. To address this issue, our method not only rely upon the pre-trained word or sentence representations of text, but also takes into account qualitatively enhanced features such as, lexical diversity, informativeness, cohesion, well-formedness etc., that have proved to be important in determining text quality. Further, we have explored a variety of neural network model architectures for automated essay scoring and have achieved significant improvements over baseline in certain cases. We would like to conclude that it is indeed possible to enhance the performance of such AES system by intelligently incorporating the supporting linguistic features into the model. One of the limitations of the present approach is that all the linguistic and qualitative features used in this work are computed off-line and then fed into the deep learning architecture. However, in principle deep learning models are supposed to learn these features apriori and perform accordingly. Therefore, one possible future directions of this work is to develop or modify the existing intermediate scores in such a way that the task specific models can automatically learn these features.

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. *arXiv preprint arXiv:1606.04289*.

Yigal Attali and Jill Burstein. 2004. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series*, 2004(2).

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. *arXiv preprint arXiv:1707.05436*.

Yen-Yu Chen, Chien-Liang Liu, Chia-Hoang Lee, Tao-Hsing Chang, et al. 2010. An unsupervised automated essay scoring system. *IEEE Intelligent systems*, 25(5):61–67.

Scott Crossley, Laura K Allen, Erica L Snow, and Danielle S McNamara. 2015. Pssst... textual features... there is more to automatic essay scoring than just you! In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 203–207. ACM.

Scott A Crossley, Jerry Greenfield, and Danielle S McNamara. 2008. Assessing text readability using cognitively based indices. *Tesol Quarterly*, 42(3):475–493.

Ronan Cummins, Meng Zhang, and Ted Briscoe. 2016. Constrained multi-task learning for automated essay scoring. Association for Computational Linguistics.

Yann Dauphin, Harm de Vries, and Yoshua Bengio. 2015. Equilibrated adaptive learning rates for non-convex optimization. In *Advances in neural information processing systems*, pages 1504–1512.

Marie De Marneffe, Bill MacCartney, Christopher Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6.

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring–an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1072–1077.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162.

Chris Dyer. 2017. Should neural network architecture reflect linguistic structure? *CoNLL 2017*, page 1.

Noura Farra, Swapna Somasundaran, and Jill Burstein. 2015. Scoring persuasive essays using opinions and their targets. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 64–74.

Peter W Foltz, Darrell Laham, and Thomas K Landauer. 1999. Automated essay scoring: Applications to educational technology. In *EdMedia: World Conference on Educational Media and Technology*, pages 939–944. Association for the Advancement of Computing in Education (AACE).

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering-Volume 12*. Association for Computational Linguistics.

Alex Graves et al. 2012. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Scott Jarvis. 2002. Short texts, best-fitting curves and new measures of lexical diversity. *Language Testing*, 19(1):57–84.

Margarita Karkali, François Rousseau, Alexandros Ntoulas, and Michalis Vazirgiannis. 2014. Using temporal idf for efficient novelty detection in text streams. *arXiv preprint arXiv:1401.1456*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Beata Beigman Klebanov and Michael Flor. 2013. Word association profiles and their use for automated scoring of essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1148–1158.

Thomas K Landauer. 2003. Automatic essay assessment. *Assessment in education: Principles, policy & practice*, 10(3):295–308.

Mirella Lapata. 2017. Translating from multiple modalities to text and back. *ACL 2017*, page 1.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Deryle Lonsdale and Diane Strong-Krause. 2003. Automated rating of esl essays. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*, pages 61–67. Association for Computational Linguistics.

Christopher Manning, Bauer Surdeanu, Mihai, Finkel John, Bethard Jenny, J. Steven, and David. McClosky. 2014. The stanford corenlp natural language processing toolkit. In *52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Danielle S McNamara, Scott A Crossley, Rod D Roscoe, Laura K Allen, and Jianmin Dai. 2015. A hierarchical classification approach to automated essay scoring. *Assessing Writing*, 23:35–59.

Danielle S McNamara, Max M Louwerse, and Arthur C Graesser. 2002. Coh-metrix: Automated cohesion and coherence scores to predict text readability and facilitate comprehension. Technical report, Technical report, Institute for Intelligent Systems, University of Memphis, Memphis, TN.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Isaac Persing, Alan Davis, and Vincent Ng. 2010. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2015. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 543–552.

Lawrence M Rudner and Tahung Liang. 2002. Automated essay scoring using bayes' theorem. *The Journal of Technology, Learning and Assessment*, 1(2).

Jürgen Schmidhuber, F Gers, and Douglas Eck. 2006. Learning nonregular languages: A comparison of simple recurrent networks and lstm. *Learning*, 14(9).

Mark D Shermis and Jill Burstein. 2013. *Handbook of automated essay evaluation: Current applications and new directions*. Routledge.

Daniel DK Sleator and Davy Temperley. 1995. Parsing english with a link grammar. *arXiv preprint cmp-lg/9508004*.

Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical chaining for measuring discourse coherence quality in test-taker essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 950–961.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891.

Yla R Tausczik and James W Pennebaker. 2010. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54.

Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education: Research*, 2:319–330.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Helen Yannakoudakis and Ronan Cummins. 2015. Evaluating the performance of automated text scoring systems. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–223.

Torsten Zesch, Michael Wojatzki, and Dirk Scholten-Akoun. 2015. Task-independent features for automated essay grading. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 224–232.