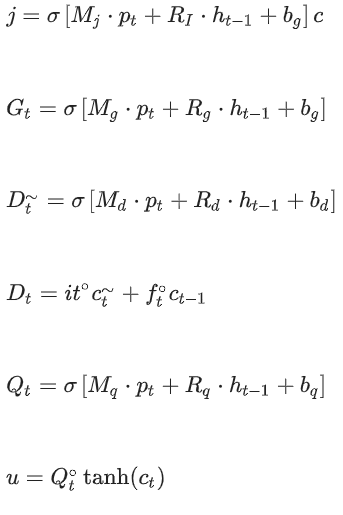
* Documentation of dataset – Learning Agency lab – Automated Essay Scoring 2.0
  + The competition dataset comprises about 24000 student-written argumentative essays. Each essay was scored on a scale of 1 to 6 ([Link to the Holistic Scoring Rubric](https://storage.googleapis.com/kaggle-forum-message-attachments/2733927/20538/Rubric_%20Holistic%20Essay%20Scoring.pdf)). Your goal is to predict the score an essay received from its text.
  + Rubric- [PERSUADE Rubric: Holistic Essay Scoring](https://storage.googleapis.com/kaggle-forum-message-attachments/2733927/20538/Rubric_%20Holistic%20Essay%20Scoring.pdf)
    - After reading each essay and completing the analytical rating form, assign a holistic score based on the rubric below. For the following evaluations you will need to use a grading scale between 1 (minimum) and 6 (maximum). As with the analytical rating form, the distance between each grade (e.g., 1-2, 3-4, 4-5) should be considered equal.
* **train.csv** - Essays and scores to be used as training data.
  + essay\_id - The unique ID of the essay
  + full\_text - The full essay response
  + score - Holistic score of the essay on a 1-6 scale
* **test.csv** - The essays to be used as test data. Contains the same fields as train.csv, aside from exclusion of score. (**Note**: The rerun test set has approximately 8k observations.)
* Documentation of dataset - AES
  + There are eight essay sets. Each of the sets of essays was generated from a single prompt. Selected essays range from an average length of 150 to 550 words per response. Some of the essays are dependent upon source information and others are not. All responses were written by students ranging in grade levels from Grade 7 to Grade 10. All essays were hand graded and were double-scored. Each of the eight data sets has its own unique characteristics.
  + **essay\_id**: A unique identifier for each individual student essay
  + **essay\_set**: 1-8, an id for each set of essays
  + **essay**: The ascii text of a student's response
  + **rater1\_domain1**: Rater 1's domain 1 score; all essays have this
  + **rater2\_domain1**: Rater 2's domain 1 score; all essays have this
  + **rater3\_domain1**: Rater 3's domain 1 score; only some essays in set 8 have this.
  + **domain1\_score**: Resolved score between the raters; all essays have this
  + **rater1\_domain2**: Rater 1's domain 2 score; only essays in set 2 have this
  + **rater2\_domain2**: Rater 2's domain 2 score; only essays in set 2 have this
  + **domain2\_score**: Resolved score between the raters; only essays in set 2 have this
  + **rater1\_trait1 score - rater3\_trait6 score**: trait scores for sets 7-8
* Automatic Essay Scoring (AES) models -
  + Both feature-engineered and end-to-end models are required for improvement
    - https://doi.org/10.1007/978-981-99-4932-8\_8
  + Grammar Error Correction (GEC) system is incorporated
* Bidirectional Encoder Representations from Transformers (BERT) language model with CNN
* Bidirectional Encoder Representations from Transformers (BERT) language model with LSTM networks
  + Determines semantic score, coherence score, and prompt-relevant score.
* BERT is used as a feature extractor
* Output is fed into a CNN and LSTM combination
  + Captures contextual information and the structure of the essays
    - Grammatical errors, semantics, coherence, prompt relevance, etc.
* Bayesian linear ridge regression
* Methodology (two-staged training)
  + [A Hybrid Approach Towards Automated Essay Evaluation based on Bert and Feature Engineering | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/9824999)
  + [[1901.07744] Automated Essay Scoring based on Two-Stage Learning](https://arxiv.org/abs/1901.07744)
  + Data Cleaning
    - Remove N/A
    - The two data sources have different grading criteria
    - Once this data is partially cleaned and essays are used to find the self-defined features for scoring, the data will be further cleaned by using techniques like stop word removal, tokenization, removal of repeated entries, etc.
  + Features = {spelling mistakes, punctuation errors, grammatical errors, word count}
  + Sentence embedding (vectorization)
    - BERT (PyTorch or Google)
    - Stop words to remove texts using BERT (768 dimensions and 12 encoder layers)
      * Pooled output is representation/embedding of CLS token passed through some layers BERT pooler, linear/dense, and activation function and contains contextualize information of whole sequence.
      * Pooled output will be used to represent text into a multi-dimensional embedding
    - Math Citations - [Fine grained sentiment polarity classification using augmented knowledge sequence-attention mechanism - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S014193312030524X)
    - The encoding process of the LSTM is described above
    - *k* = 768 means the length of sentence embedding
    - *ut*—hidden state of sentence *pt*. (*My, Ry*) where *y* = (*j*, *D*, *G*, *Q*) are the weight matrices for the input, forget, candidate, and output gates, respectively. *b* stands for the bias vectors of the specific gates. Sigmoid function = *σ* and ◦ means element-wise multiplication. Hence, for every essay, we will get the hidden state set *H* = {*u*1, *u*2,⋯, *um*}. *Hm* is passed through the dense layer to convert to scalar value. The values from dense layer output are then projected back to their respective ranges according to ASAP dataset as has different sets of essays with different scoring range mentioned in Table [2](https://link.springer.com/chapter/10.1007/978-981-99-4932-8_8#Tab2).
  + Prompt Relevancy Score
    - Score is how relevant the essay is to the question/prompt.
    - Same procedure as semantic score described above
      * i.e., after LSTM hidden layer data is fed to dense layer which gives a scalar output scaled to 0–1 range and score 0 is received by essays that are irrelevant. The scores are projected back to their actual score ranges according to the dataset.
  + Training
    - Scores in the training dataset are scaled down to 0-1 scale and rescaled to original values before the testing phase
    - A monolayered LSTM neural network with a hidden layer size of 1024 is used in the model
    - The dropout proportion is 0.5 to avoid overfitting of the model
  + Evaluation
    - Epochs are set to 10 and are compiled 5 times per fold cross validation
* Report Figures
  + Essay set descriptions [Table 1 | Autograder: A Feature-Based Quantitative Essay Grading System Using BERT | SpringerLink](https://link.springer.com/chapter/10.1007/978-981-99-4932-8_8/tables/1)
  + It is known that the performance of multi-layered and bidirectional LSTM model is subpar from [[11](https://link.springer.com/chapter/10.1007/978-981-99-4932-8_8#ref-CR11)]