Evaluation of essays using incremental training for Maximizing Human-Machine agreement

Dhananjay Ambekar
123059008

Guided By
Prof. D. B. Phatak

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

October, 2014
Table of Contents

1 Introduction

2 Types of assessments in MOOCs
   - Self & Peer Assessments
   - AI Grading (or ML Grading)

3 Current Research in AES (Automated Essay Scoring)
   - AES

4 Training Algorithms
   - Bayesian Independence Classifiers
   - K-nearest Neighbour
   - GBRT

5 Experiments

6 Problem Formulation
Introduction to topic
Areas of Interest

- Massive open on-line courses (MOOCs)
- Grading essays in MOOCs
- Automated essay grading
- Correlation between Human-Machine grading
Self Assessments
Students grade assignments or tests based on a teacher’s guidelines. *Rubrics* are used.

Peer Assessments
Student grades submissions of other students after his completion. A process for calibration may or may not be applied.
AI Grading (aka ML Grading)

- MOOCs have a large number of submissions with varied data, making ML grading an apt choice.
- Uses Machine Learning techniques along with NLP for text preprocessing.
- Part of the data is graded by instructors and used to train models for grading further.

![Figure: Process of automatic Evaluation of essays](image-url)
Discussion on automated Scoring

- Essays are subjective in nature, best sources to judge understanding of student
- Multiple graders will produce different grading for same data
- Produces inconsistency among grades, Automated models are consistent (might not be accurate)
- If trained well, automated scoring is consistent with human graders
Tools for AES

- **Project Essay Grade (PEG)**
  Essays graded based on *Proxes*.
  Ex. Count of prepositions, Special words.
  No grammar check using NLP techniques.

- **Intelligent Essay Assessor (IEA)**
  Statistical Tool
  Grading based on frequencies of words

- **Educational Testing Service (ETS-I)**
  *Lexicon* is created using NLP rules
  Discourse Structure is weighted during grading

- **Bayesian Essay Test Scoring system (BETSY)**
  Mainly used for grouping of essays on few point scale
  Bernoulli Models used for text classification
### Summary of Performances

<table>
<thead>
<tr>
<th>System</th>
<th>Technique</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEG</td>
<td>Statistical</td>
<td>0.87</td>
</tr>
<tr>
<td>IEA</td>
<td>Algebra</td>
<td>85-91</td>
</tr>
<tr>
<td>ETS-I</td>
<td>NLP</td>
<td>93-96</td>
</tr>
<tr>
<td>BETSY</td>
<td>Bernoulli/NLP</td>
<td>80</td>
</tr>
<tr>
<td>Automark</td>
<td>NLP/ML</td>
<td>93-96</td>
</tr>
<tr>
<td>C-Rater</td>
<td>NLP/ML</td>
<td>80</td>
</tr>
</tbody>
</table>
Evaluation of essays using incremental training for Maximizing Human-Machine agreement
Current Research in AES(Automated Essay Scoring)
AES

Estimating accuracy against Human grading

Cohen Kappa Coefficient

\[ k = \frac{P(a) - P(e)}{1 - P(e)} \]

Pr(a) is the relative observed agreement among raters
Pr(e) is the hypothetical probability of chance agreement

Precision

\[ P = \frac{\text{No. of cases of agreement}}{\text{No. of evaluations by Human} + \text{No. of auto evaluations}} \]
Steps in Feature Extractions

- **Prepossessing of input data**
  Involves removal of non-ASCII characters.
  Processing punctuation such as period comma with spaces
  In this step essay is truncated with some limit of words, for example essays with more than 2500 words will be truncated to this limit.

- **n-Grams**
  a Parts of Speech(POS) tree is created before deriving n-grams

  **Sentence:**
  “Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.”

  **Tokens:**
  [Pierre / Vinken / , / 61 / years / old/ , / will / join / the/ ...continued]
**Steps in Feature Extractions**

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNP</td>
<td>noun, proper, singular</td>
</tr>
<tr>
<td>VB</td>
<td>verb, base form</td>
</tr>
<tr>
<td>JJ</td>
<td>adjective or numeral</td>
</tr>
</tbody>
</table>

**POS-tagged tree example**

```
S
  NP-SBJ
    NP
      NNP Pierre
      NNP Vinken
    ADJP
      NNP CD
      NNP NNS
      JJ old
  MD will
  VP
    VB join
    NP
      DT the
      NNP board
      IN as
    PP-CLR
      DT a
      JJ nonexecutive
      NN director
    NP
      NNP Nov.
      CD 29
```
Steps in Feature Extractions

- **Synthetic training examples** Many essay evaluation processes also take account of synonyms of tokens generated (Ex. NLTK)
- **Feature Vector** Features are maintained for each essay in data set
  - Spelling Error Features: Avg. spelling errors per essay
  - Grammar Error Features: Counts of good n-Grams
  - Bag of stemmed words
Bayesian Independence Classifiers

These probabilistic classifier estimate a probability of document being in particular category provided that certain features are present in the document.

\[
\log P(C|D) = \log P(C) + \sum_i \left\{ \begin{array}{ll}
\log (P(A_i|C)/P(A_i)) & \text{if } D \text{ contains feature } A_i \\
\log (P(\bar{A}_i|C)/P(\bar{A}_i)) & \text{if } D \text{ does not contain feature } A_i
\end{array} \right.
\]

\(P(C)\) is probability that a document belongs to class \(C\), a **prior probability**. Class \(C\) can be a class/category of Good, Average or any class which is defined earlier while process of training.

\(P(A_i|C)\) is the **conditional probability** the document has feature \(A_i\) and given that it belongs to class \(C\).

\(P(A_i)\) is a prior probability of any document containing feature \(A_i\).

\(P(\bar{A}_i|C)\) is the conditional probability that document does not feature \(A_i\) given that class is \(C\).

\(P(\bar{A}_i)\) is the probability that document does not have feature \(A_i\).
K-nearest Neighbour

- K- essays most similar to the test essay are selected
- Score is given to essay based on weighted average of k selected essays in test set
- Similarity is calculated by querying entire document against test set
Gradient Boosting
Incomplete

- Produces a prediction model in the form of a group of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.
Subsequent Work
Experiments

Objective:

- To understand the process of automated evaluation and its components which include Natural language processing, Machine learning and Evaluation Rubrics

Setup:

- edX developer stack instance was used
- edX uses NLTK and EASE for auto-grading text data, which internally applies natural language processing and training algorithms for text classification (namely Gradient Boosting)
- Work-flow is generated from example based assessments and training is performed
- At this point a submission can be auto-graded using classifier trained in previous step
Total 10 samples were examined to assign either Good or Poor label.

<table>
<thead>
<tr>
<th>Total 10 samples</th>
<th>edX AI-ML</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Good</td>
</tr>
<tr>
<td>Human assessment</td>
<td></td>
</tr>
<tr>
<td>Good</td>
<td>6</td>
</tr>
<tr>
<td>Poor</td>
<td>2</td>
</tr>
</tbody>
</table>

**Table**: Inter grader comparison

**Cohen’s kappa Coefficient**

\[
P(a:\text{agreement}): \frac{(6+2)}{10} \\
\Pr(e:\text{random agreement}): = 0.48 + 0.08 = 0.56. \\
k = \frac{(0.8-0.56)}{(1-0.56)} = 0.56
\]
Problem Statement

Objective is to achieve the accuracy as near as human Evaluation.

- **Features contained in responses** differ adding diversity to them. (For an example - For Indian students, In an Essay written on Cow, *People worship Cow.* is an obvious feature.)

- These Features might not be captured while training by picking samples randomly because with respect to large number of students, frequency of these features might be very less and **Evaluation Rubrics** may or may not capture that content detail.

- While in training phase, another Iteration of training itself can be added to either **learn** or to **add** such features.
Future Work

- Study the process of feature extraction/language processing from textual data in order to gain some knowledge about learning features.
- Analyze the performance of different algorithms in automated essay scoring. (Currently edX-ORA2 uses Gradient Boosting for text classification.)
- Formulate a process for experimentation of Iterative training and then comparing performances using standard measures.
Thank You.