Anmol Gupta, Annie Hwang, Paul Lisker, Kevin Loughlin — Milestone #2
CS109a

Essay #1: *Predicting Grammaticality on an Ordinal Scale*

The 2 main contributions of this project were: 1) a "state-of-the-art" approach for predicting grammaticality on an ordinal scale *adapting* other techniques that have already been found, and 2) conducting "realistic evaluations" to predict sentence-level grammaticality.

The purpose of the project was to find a model that best predicts the grades of sentences written by non-native English speakers that were evaluated by the expert judges. The essay explains that sentences were first evaluated by theses experts they used an ordinal scale (0-4: 4-most grammatical, 1-least grammatical, 0-incomplete/other), rather than a binary one, because the "distinguish between grammatical and ungrammatical is not simply binary". Then, they used **ridge regression** (l2-regularized linear regression) along with **5-fold cross-validation** to find the correct tuning λ value by evaluating λ 10{−4,...,4} and using the one with the highest R^2 value. The model had four features/predictors: Spelling features, n-gram Count and Language Model, Precision Grammar Features, PCFG Parsing Features. After doing some feature ablation, they realized that the most **important predictor** was the "n-gram frequencies and whether the link parser can fully parse the sentence". The project ended up producing the best, most "realistic evaluation of methods for predicting sentence-level grammaticality to date."

Essay #2: *Neural Networks for Automated Essay Grading*

This article offers an insight into how neural networks can be used to outperform other NLP machine-learning techniques. Because the paper is rather technical, I needed to do some other reading first, in order to have a basic sense of how neural networks function. While I don't claim to be an expert, my understanding is the following.

In many machine learning problems, the predicted answer stems from a set of inputs, on which we perform linear transformations to put an expected output. A simple example of this is predicting the probability that event A will occur based on inputs X_1, X_2, and X_3, which each have a weight (i.e. importance in calculation) of some w_i. Then we could say that our prediction for the probability of event A occurring is P(A) = w_1 * X_1 + w_2 * X_2 + w_3 * X_3. Ultimately, we improve our model by tuning the weights based on feedback.

While this model is strong in many situations, it limits us to linear transformations. For example, what if P(A) needs to be the max of two quantities, Y_1 and Y_2, that are themselves linear functions? Because the max function is nonlinear, we can't simply apply a linear weight to it and sum the values. This is where neural networks become particularly useful.

A neural network is modeled after neuron interactions in the brain, where systems of neurons send signals to each other (based on some initial input), and that input is transformed and propagated through layers of neurons until a final decision is output (or "fired"). Rather than maintaining the initial simplistic model of some "unit" taking in inputs and linearly transforming them into outputs, we now have layers of units for different steps of the transformations.

The input is generally passed (after a possible linear transformation) to some nonlinear "hidden" layer of neurons, which complete transformations on their inputs, and pass their outputs to some final output layer that makes the ultimate decision on what the prediction is, perhaps based on another linear transformation. The "hidden" layer can in fact be multiple layers of transformations, but we needn't worry about that for our basic understanding.

We essentially have a graph of nodes and links by layer, where the nodes represent a "transformer" or "function" that takes input and produces output based on some internalized function, and where the value of the node is determined by its output. The links are the weights applied to the input when passed into a node.

When the output is given, we can calculate the error from a pre-known true value, and propagate this error back through the network to update the weights of the links, continually training the network until the weights converge to an acceptable representation of their optimal values.

This particular paper found that by using neural networks that took features such as essay length, spelling accuracy, and vocabulary complexity (among others), one could achieve better prediction accuracy than previously used linear models. It remains to be seen whether or not we will implement a neural network due to its complexity, but we known have an interesting set of features to track, as well as a known successful method for NLP and essay grading.

Essay #3: Automated Essay Scoring by Maximizing Human-machine Agreement

This article talks about how current methods of automated essay scoring can be supplemented with analyzing the agreement of human and machine raters. Today models using classification, regression, and preference ranking, are used to predict the quality of essay submissions and general written works based on certain requirements. However, we can use list-wise learning to rank algorithms based on human vs. machine grading to create a rating model.

To test the machine-human agreement, the approach consisted of first collecting essays graded by individuals. Then, using list-wise learning to rank algorithms, a function was created that had certain features examined through the human-graded essays. Using that trained function, the model was able to grade all of the essays inputting into the function, and mapped them to scores ranging from 1-6.

The benefit of using a list-wise method of learning is that it takes the entire dataset to train the algorithm, while other methods, such as using a point-wise approach, scores independently of other items -- so a function is created based on factors the individual thinks is important in essays, and the scores for each essay are calculated based on those criteria. For the list-wise method, the features that we are looking at within the essays are categorized as follows: lexical, syntactical, grammar and fluency.

Testing the function on the essays, the range of the R^2 value was calculated to be 0.7-0.8. The variance of the scores was also calculated using five-fold cross-validation and comparing against human graders, which showed a very little difference. This demonstrated the robustness of the approach being used. From this conclusion, it's shown that a generic rating model works best when trying to calculate the grades of

essays. By categorizing the criteria, and examining the essays as an entire database, it was able to train the list-wise learning function to create a model that most accurately represented human-based grades.

Essay #4: Task-Independent Features for Automated Essay Grading

This paper seeks to assess the task-dependence of features used by state-of-the-art approaches to automated essay grading and their transferability on English and German datasets, as well as determining which models transfer more successfully across tasks. Currently, automated essay grading methods rely on measuring certain features, such as length features (e.g., sentence length, word length), occurrence features (e.g., occurrence of commas, quotations, references), syntax features (e.g., syntactic variations, subordinate, causal, and temporal clauses), style features (e.g., formality), and several other features. However, these features are not always task-independent. For example, while sentence length and word length may be indicators for a writer's complexity, they are largely task-independent (i.e., the writing prompt should not determine the sentence length and word length). On the other hand, a measure of formal references may feature more prominently in source-based prompts. This study sought to, in part, determine the transferability of models dependent on task-dependent features to other tasks.

This study proceeded by first categorizing the strongly task-dependent and weakly task-dependent features. To proceed, the investigators re-implemented an essay grading system that performed preprocessing that learned a model of essay quality from the extracted features. Then, in the second major step, the model was applied to grade the essays, which included 10-fold cross validation.

Ultimately, this research showed that training on one task with all the features and then grading on another task presents significant losses, with some "remarkable exceptions" where a model trained on one task performed better on a different task than a model performed on that different task and used on itself. On the other hand, by using only weakly source-dependent features in the training of the model, the model was more accurate when transferred to separate tasks. Nevertheless, though this reduced-feature training model was more successful, it still presented quite high losses in task transferences, which suggests that some different feature discrimination might be useful.