Autograder: *Automated Classification of Documents to Grade School Level*

Kiley Yeakel, Stephanie Tzeng **{kileyy10, stzeng} @ stanford.edu**

Predicting

For teachers, finding reading content of the appropriate grade level can be challenging. We aim to provide a suite of machine learning algorithms which can automatically classify a text passage to its associated grade level. We explored 3 different classes of algorithms – Kneser-Ney Smoothing with N-grams, Naïve Bayes, and RNN – and determined their ability to classify sentences and entire articles. For classifying articles as "easy" or "hard" we found Naïve Bayes (NB) and logistic regression to provide the best accuracy (**92.4%**). The language model was able to achieve **84.7%** accuracy for binary classification. For classifying articles to their specific grade level (multi-class classification), we found the best performance again with NB and a median calculation at **40.9%**. The language model was able to achieve **26.6%** accuracy for multi-class classification as well as **63.5%** accuracy within 1 grade level. In all cases explored the RNN was found to be extremely overfit, implying we did not have enough data for a RNN approach.

Data

Our dataset was provided by Newsela, an education tech company which "translates" news articles into varying grade levels – from 2nd through 12th grade. The database consisted of ~2,155 articles each of which were translated into 5 different grade levels – for a total of 10,755 articles at varying grade levels. The text strings were preprocessed to remove artifacts such as html tags, then tokenized. Preprocessing depended on the model utilized.

Original text (assumed 12th grade):

```
The group pushing to replace President Andrew Jackson with a
woman on the $20 bill has revealed its final four candidates
after     more     than     256,000     votes     were     placed.
```

4th grade translation:

```
 A group wants to take President Andrew Jackson's face off the
$20 bill. It wants to put a woman on the bill instead. It has
four final choices of women to go on the $20 bill.
```

Number of Articles per Grade Level

Grade	2	3	4	5	6	7	8	9	12
Num. of Articles	283	616	1730	1488	1171	1365	1152	862	2096

Features

Articles were broken down in to groupings of sentences of various lengths. Text segments were then tokenized into individual words. For the Kneser-Ney and Naïve Bayes algorithms, we then computed N-grams of lengths varying from 1 to 5 words, forming a dictionary for our article corpus. For the Naïve Bayes approach, we ignored n-grams which appeared in less than 5 text segments or more than 1000 text segments.

Models

Kneser-Ney Smoothing on 5-grams

- Generates language models: probability distributions for each grade level
- When out of vocabulary 5-grams appear in the validation set, interpolation is used, which mixes weighted probabilities from all the 5-gram, 4-gram, down to the unigram counts
- Model that produced lowest perplexity score was the prediction of the sentence. Entire article was then classified into grade level and easy/hard classification by looking at **means** and **linear regression**

Naïve Bayes + Linear Classifier

Layer 1: Naïve Bayes Model

TF-IDF vectors for the text segments were used to train a Naïve Bayes model.

- Text segment lengths = 1 sent., 2 sent., 3 sent., entire article
- Max N-gram lengths = 1, 2, 3

Layer 2: Secondary Classifier

Four different algorithms were considered: **mean, median, linear regression, logistic regression**

For the linear regression and logistic regression algorithms, the features were the percentage of the article predicted to belong to a particular class.

Recurrent Neural Network

Different architectures of RNN long short-term memory (LSTM) networks were tested.

- Number of nodes = 20, 100, 200
- Number of layers = 1, 2, 3
- L_1 and L_2 regularization = 0, 0.01

L_1 and L_2 regularization combinations were applied to the input, recurrent, and bias weights. In all cases, the RNN was severely overfit.

Results

Binary Classification Results

Type	Seg. Length	n-gram	LC	Test Err.	Train Err.
NB w/ orig	3	2	None	0.225	0.177
NB w/o orig	Article	1	None	0.158	0.148
NB w/ orig	2	3	Logistic	0.076	0.036
NB w/o orig	1	3	Linear	0.118	0.045
Grade LM w/ orig	Article	5	Mean	0.153	0.809
Grade LM w/o orig	Article	5	Mean	0.266	0.117
Grade LM w/ orig	Article	5	Linear	0.171	0.035
Binary LM	Article	5	Mode	0.202	0.0003

Multi-class (Grade Level) Classification Results

Type	Seg.	n-gram	LC	Test Err.	Train Err.
NB w/ orig	1	2	Median	0.591	0.372
NB w/o orig	1	2	Logistic	0.625	0.257
NB w/ orig	3	2	None	0.704	0.592
NB w/o orig	3	2	None	0.716	0.538
Grade LM w/ orig	Article	5	Mean	0.734	0.632
Grade LM w/o orig	Article	5	Mean	0.757	0.642

Discussion

- The Naïve Bayes algorithm with secondary linear classifier performed the best despite being the simplest algorithm; best performance for binary classes
- For binary: NB layer had higher accuracy when given longer text segments; whole article performed best
- For multiclass: NB improved with longer sentence segments and longer N-grams; Multiclass NB identified two "clusters" but did not do as well classifying individual grades
- For secondary classification layer: linear regression performed better at predicting the grade "range" of a passage

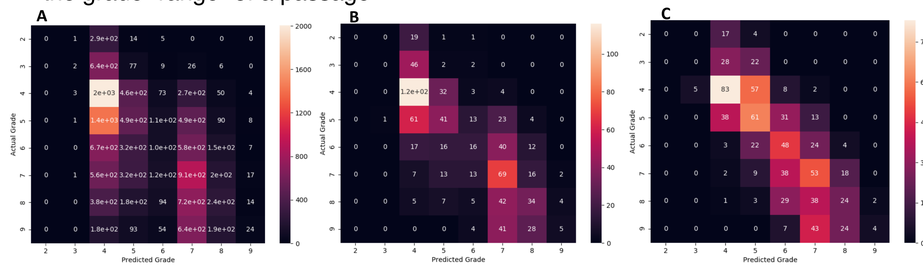

Figure 1. (A) Confusion matrix for NB predictions per text segment (3 sentence length, n-gram length = 2) as compared to results after applying Logistic Regression (B) and Linear Regression (C) to get a prediction for an entire article.

- N-gram smoothing achieved a 63.5% accuracy for multi-class classification within +/-1 grade level
- N-gram smoothing did not perform as well as Naïve Bayes in predicting a binary classification of easy vs. hard.

Conclusion: Both models use a version of n-grams, NB as a generative model and the smoothed n-grams approach finding the most similar language model (i.e., probability distribution). ***Complexity of text can be fairly well detected by looking at probable chained words.***

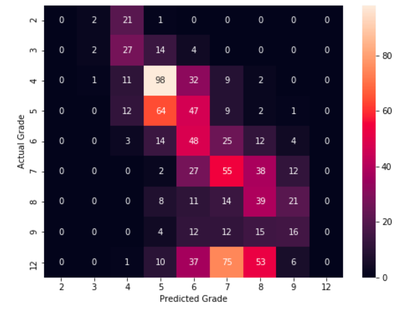

Figure 2. Confusion matrix for Kneser-Ney Smoothing per article, using means

Future

Data limitation was a major impediment to achieving better results, particularly in the case of the RNN where model overfitting became unavoidable due to our small dataset. With more time, we would:

- Gather more data and retest our algorithms
- Explore using BERT[1] to get pre-trained contextual embeddings of our text samples rather than GloVe[2] embeddings; implement this as our embedding in the RNN

References

[1] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K., 2018, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv:1810.04805.
[2] Pennington, J., Socher, R., and Manning, C. D., 2014, "GloVe: Global Vectors for Word Representation.