Milestone II-Final Proposal:

Text Difficulty Prediction

Project Description

Apply supervised and unsupervised learning techniques on wikipedia text to predict sentences which will need to be simplified for readers to make it easier to understand. Readers may include students, children, adults with learning/reading disability, and non-native English speakers.

Rough Timeline:

Week 1- 2: Finalize Project Proposal

Week 3-4: Refine Project Proposal, Explore data

Week 5-6: Build models, Prepare Visualizations

Week 7-8: Fine tune outcome and write final report.

Part A: Supervised learning

Feature Engineering: After preprocessing unstructured data (i.e removing digits, lowercasing words etc) we will explore helper resources (given) to see which new features should be built to understand text difficulty (Examples):

a. Length of sentence: From Professor Kevyn's paper, I learned that text usually contains many short sentences (under 10 words) which could result in higher Mean RMS error and causing lower model accuracy. Hence, this would need to be explored further.
b. Use of stopwords: Some early learners (grade 4 students) may find use of stopwords slightly more difficult than the higher grader learners. Hence, by removing stopwords, are we introducing model bias, would need to be explored.
c. Use of distinct words (synonyms) with same meaning in multiple sentences: It is also possible that some sentences have less frequently used words that share the same meaning as more frequently used words in other sentences, causing text difficulty.
d. Other features: There could also be other features such as POS tags, syllables, difference in tokenization approach (TF-IDF Vs Word2vec) which may impact model accuracy, to be explored.

Learning Approaches and Evaluation Measure: After obtaining engineered features as per above, we would like to use **Naïve Bayes, Random Forest and Recurrent Neural Network (LSTM, GRU)** to explore how these features and hyperparameters impact model accuracy as well as evaluate **precision-recall, AUC-ROC curve** to see how many true Vs false predictions are made by each model. It's critical that we evaluate this, because if a model with high accuracy score misclassifies difficult sentences as simple, then it would pose difficulty for many readers in comprehending the text, which would be an undesirable outcome. Hence, avoiding false positives would be more important than false negatives in this case. This is one of the reasons we have selected above learning approaches to see how a baseline naïve bayes model (given by instructor) would compare with the new features and other algorithms. We particularly chose RF because it's fast, does not require dimensionality reduction and has many hyperparameters to tune. While LSTM can capture long-term dependencies between word sequences and GRU can do the same using less computational power, which would be an interesting comparison to learn.

External Tools Resources: In addition to helper resources, I might use additional NLP libraries such as wordnet, word embeddings (gensim), regex for preprocessing and building feature sets.

Data Visualization: Distribution of words in difficult sentences, line chart of model accuracy relative to hyperparameters and features, plot of evaluation curves (precision-recall, AUC-ROC curve).

Part B (Unsupervised learning)

Data Structure and Data Manipulation: For unsupervised learning, since data will not contain any target labels, we may use topic modeling to find hidden structure in the data and then compare it with Part A (Supervised Learning). We might also explore ways to tag unknown words (UNK) to deal with missing data, as well as look for similarity relationships between simple and complex words using jaccard or cosine similarity as seem fit at the time of exploration.

Learning Approaches and Evaluation:
We plan to use a few clustering algorithms (like K-means, Affinity Propagation etc) to analyze clusters in the feature sets and then compare their cluster quality. We will use silhouette score (as an example) to evaluate cluster quality in terms of density and separation from other clusters and then plot it.

Data Visualization:

We may use dimensionality reduction techniques (PCA, MDS) to compare clustering quality using scatter plots as well as make use of interactive word cloud visualization to show text classification.

(Prashant Sanghal)