Name: Nicholas Ramkissoon
Net ID: nr1610
Final Project Proposal

Problem Statement: Document classification of Japanese texts by reading difficulty/complexity level for the purpose of learning the language

Background/Definitions: Japanese Language Proficiency Test (JLPT) - standard exam for Japanese language learners, there are 5 levels of difficulty labeled 1 through 5. 5 being the easiest and 1 being the most difficult. The JLPT classifies several, but not all, vocabulary words, grammatical constructions, and kanji characters.

Papers related to topic:

1. https://www.jstage.jst.go.jp/article/jnlp/16/4/16_4_4_3/_pdf - A Corpus-based E-learning System for Japanese Vocabulary
    a. Paper deals with a similar topic: assisting Japanese language learning by classifying text based on difficulty
    b. The methods for classifying document reading complexity in this paper are similar in that I will be using the same standard for determining the difficulty of a word (standard defined by the Japanese Language Proficiency Test (JLPT)).
    c. At the sentence level, I intend to factor in grammatical complexity which is not considered in this paper. The paper only considers the vocabulary JLPT level of the words in the sentence and the JLPT level of the kanji characters in the sentence to determine the overall sentence difficulty.
2. https://www.aclweb.org/anthology/W16-4912.pdf - Japanese Lexical Simplification for Non-Native Speakers
    a. Paper deals with determining the complexity of a word and replacing it with a word of lower complexity without changing sentence meaning.
    b. For words not listed in JLPT, authors use a linear regression model to classify the complexity of words using features such as unigram frequency, presence in children's corpora, and if word is likely to be a technical term. The linear regression is trained using words already listed in JLPT.
    c. I'll adopt the method of using linear regression for predicting word complexity, but I will use different features. I will train the data using words listed in JLPT, but the features for my model will be unigram frequency in a standard corpus, average JLPT level of the kanji characters in the word (JLPT classifies roughly 2000 characters already),

and the amount of kanji characters in a word. I'll explain the reasoning for using different features in detail in the paper itself.

3. https://www.aclweb.org/anthology/C16-1159.pdf - Grammatical Templates: Improving Text Difficulty Evaluation for Language Learners
    a. Paper deals with parsing documents for specific grammatical templates/constructions and then determining text difficulty based on those grammatical features.
    b. Parsing for grammatical templates is shown to be an accurate predictor of text difficulty in the paper (87.7% accuracy). I will also be parsing for grammatical templates in my system, but will use the results from parsing in a different way. After collecting the grammatical templates of a document, the authors of the paper use a "multilevel linear classification" algorithm to predict sentence difficulty. The algorithm essentially starts at the easiest difficulty and checks if a document is at that level and works its way up the difficulty levels. I intend to use TF-IDF to rank documents by grammatical difficulty.

Problem Solving Strategy: System Project

System Input: Japanese text of varying difficulty taken from varying sources including news websites, JLPT practice websites, etc.

System Output: An assigned JLPT level for each input document

Evaluation Metric: Accuracy = correctly classified documents / total documents. In this case, the documents are classified by JLPT levels 1 through 5

Training Data/Corpora/Other Resources:

1. Balanced Corpus of Contemporary Written Japanese (BCCWJ) - word unigram frequencies will be calculated using this corpus to be used to determine the difficulty of words that are not listed in JLPT
2. JLPT standards for vocabulary, grammar, and kanji difficulty - the linear regression model for predicting word complexity will use JLPT vocabulary lists as training data, and the features of the model dealing with individual kanji difficulty in a word will rely on JLPT classifications available for roughly 2000 kanji. Grammar constructions

Development/Test Corpus: Japanese text that has already been given a JLPT classification. Past JLPT exams and problem sets are publicly available. There are also several JLPT practice websites such as https://japanesetest4you.com/ that

provide sample reading material for every JLPT level. I will compile material from these sources and split it into a development and test corpus.

High-Level System Architecture:

The system will consist of 2 components:

1. A component for evaluating text based on word/vocabulary difficulty
2. A component for evaluating text based on grammatical difficulty

Decoupling the grammatical classification from the word classification allows for flexibility for weighting each of the results when determining the overall classification (easier words can be used with more complex grammar patterns and vice versa). Once both components are implemented, the optimal weights can be determined by running on the development corpus.

Both components will use TF-IDF and cosine similarity to classify documents into JLPT levels. For each JLPT level, there will be a vector of words/grammar structures representative of that level. This vector will be used as a query against all of the documents in the input. The documents will be parsed and tokenized into vectors of words/grammar structures. For a given document vector, the JLPT query vector with the highest cosine similarity score will indicate that document's word/grammar JLPT level.

Baselines for Evaluation:

1. As a first, low baseline, the system should be able to score a higher accuracy than a system that only considers the difficulty of the vocabulary in a sentence. A simple system that assigns the average of the JLPT levels of the words in a document to the document itself should be easily beaten.
2. Because the system has several layers, during development I will baseline the system as a proof of concept before implementing subsequent layers. For example, I would test the system on the development corpus once the system can tokenize documents for JLPT listed words, then test again once the linear regression model for unlisted words is implemented and the query vectors are augmented with the new unlisted words. On the grammar side, I will implement regular expressions for parsing a handful of the most relevant grammar structures of each JLPT level, and as time permits, continue to add more structures.