Automated Reading Comprehension Clustering

Charles Hathaway

October 29, 2015

1 Executive Summary

This project will focus on discussing and exploring the effectiveness of various language features when applied to clustering books based on reading comprehension level. It will utilize previous works in the area, which although may not be perfectly applicable, have a somewhat obvious connection to the objective. To test the features, we will attempt to generate 3 clusters from plain-text books freely available online; child-friendly, young adult, and adult books.

The timetable for this project is given at the end of the document.

2 Goal

The goals of this project are:

- Exploration of existing reading comprehension features and clustering tools
- Testing of new and existing reading comprehension features
- Analysis of results and discussion
- Interface to allow others to utilize the tool

3 Background and Motivation

There are several systems currently used to classify books based on reading comprehension level, for numerous applications ranging from selecting books for classrooms, to measuring an individuals literacy skills for both medical (autism, dyslexia, etc.) and educational purposes. The goal of this project is to enable a larger selection of books for these purposes by automated the system of clustering books by grade level, which is currently manually done at the cost of the publisher.

4 System Architecture and Approach

Following many other NLP applications, this project will utilize a pipeline approach. This pipeline is laid out in figure 6. The design is further described in the box chart, displayed in figure 6.

In order to accelerate development, the coding will primarily be done in Python. The project will utilize various open source toolkits, including the NLTK, which provides a maximum entropy clustering framework. This will allow me to spend more time focusing on selecting effective features, and less time remaking the wheel.

5 Deliverables

The deliverables of this project will have 3 key parts:

- A technical report, detailing the results of key experiments and output of several configurations of the systems
- A command-line interface that allows the scanning and clustering of a multitude of books
- A web interface which outputs the cluster a particular uploaded books belongs to

6 Required Resources

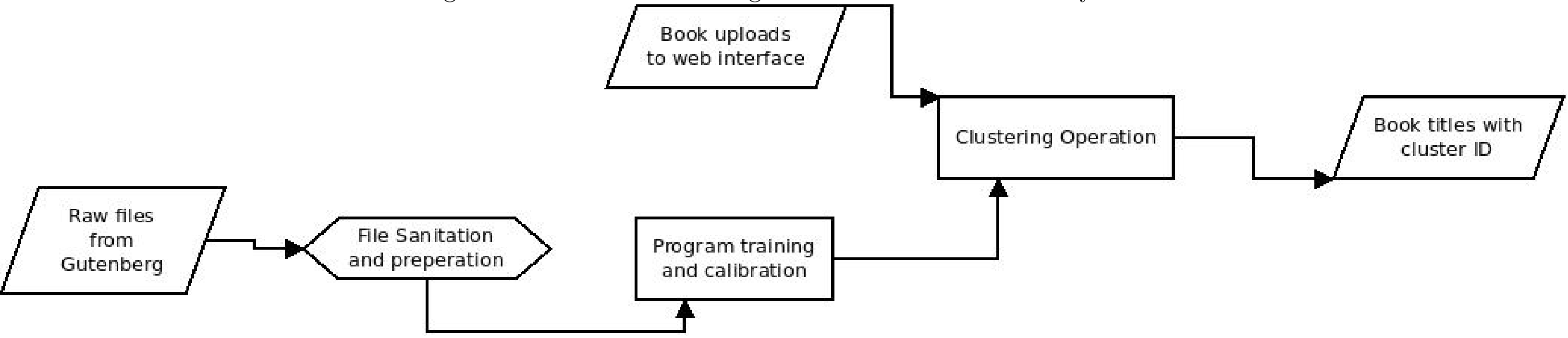
The most important resources for this project is the books themselves. To that end, Project Gutenberg will be mined to provide both the training and testing data sets. They have an extensive list of childrens books, which we will use as the basis for the first clustering. Ideally, we will end up with a clustering such that childrens book all fall below the average reading difficulty, at average - 1 standard deviation. For the second clustering, we will determine it by picking a range in between the children books and the highest threshold; ideally in the range of average $\pm$ 1 standard deviation. The last section, adult/advanced readers, will consist of books deemed more complicated than the average + 1 standard deviation.

It should be noted that the Project Gutenberg repository is large; upwards of 650GB. Most books in the project have multiple translations, which contributes to the large size (in addition to multiple translations, they also have multiple format). To make this project more achievable, we will limit ourselves to the fiction category; which has it's own subcategory, children's fiction.

Lastly, we need a toolkit which provides many resources to help calculate the clusters given our feature set. For this purpose, we will use the Python Natural Language Toolkit (NLTK), which is a free and open source library intended for these types of application [1].

	ⓘ	Name	Duration	Start	Finish	Predecessors	Resource Names
1	▦	Scrape PGB	2 days	10/29/15 8:00 AM	10/30/15 5:00 PM		
2		Sanitize Data	2 days	11/2/15 8:00 AM	11/3/15 5:00 PM	1	
3		**Iteration 1**	**9 days**	**11/4/15 8:00 AM**	**11/16/15 5:00 PM**		
4		Develop Features 1	3 days	11/4/15 8:00 AM	11/6/15 5:00 PM	2	
5		Apply Features 1	3 days	11/9/15 8:00 AM	11/11/15 5:00 PM	4	
6		Run Expirements 1	3 days	11/12/15 8:00 AM	11/16/15 5:00 PM	5	
7		**Iteration 2**	**9 days**	**11/17/15 8:00 AM**	**11/27/15 5:00 PM**		
8		Develop Features 2	3 days	11/17/15 8:00 AM	11/19/15 5:00 PM	3	
9		Apply Features 2	3 days	11/20/15 8:00 AM	11/24/15 5:00 PM	8	
10		Run Expirements 2	3 days	11/25/15 8:00 AM	11/27/15 5:00 PM	9	
11		Write Report	3 days	11/30/15 8:00 AM	12/2/15 5:00 PM	7	
12	▦	Create Final Present...	3 days	12/3/15 8:00 AM	12/7/15 5:00 PM	11	

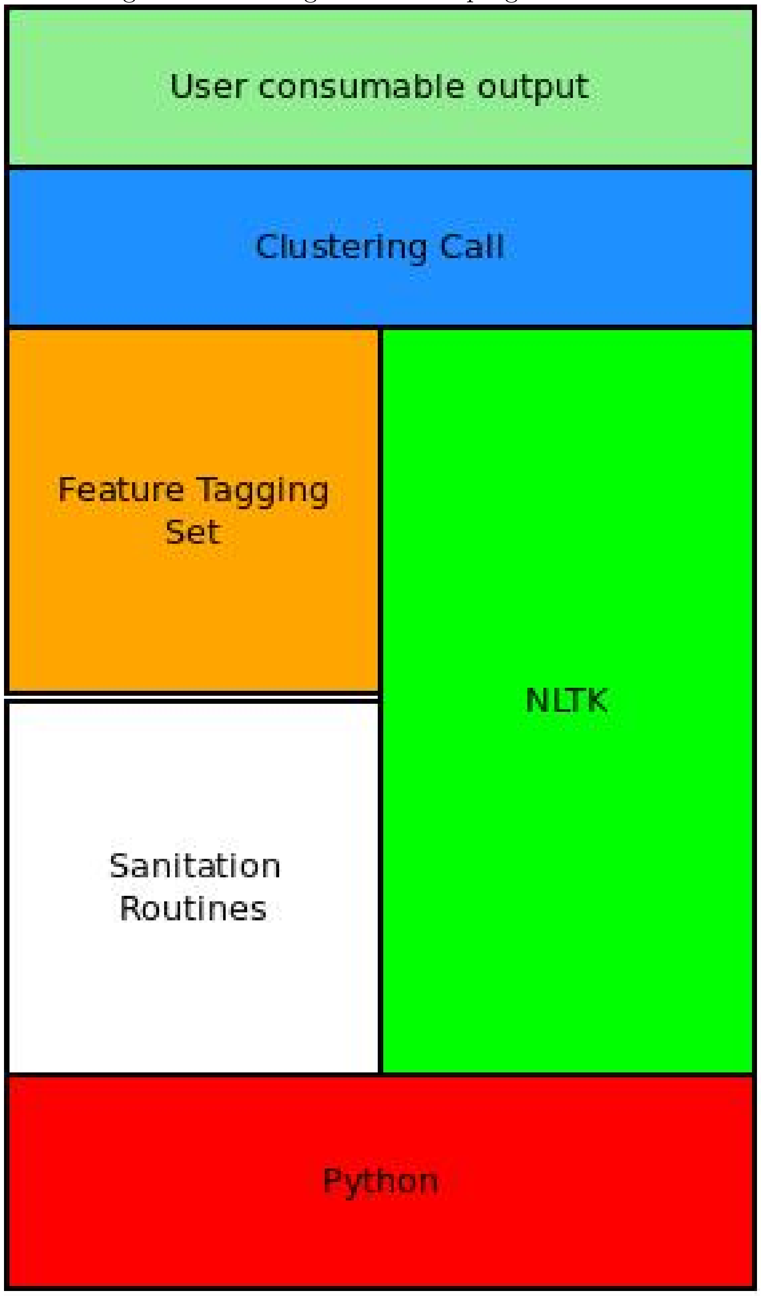
Figure 1: Flowchart indicating flow of information within system


Book uploads to web interface
Clustering Operation
Book titles with cluster ID
Raw files from Gutenberg
File Sanitation and preperation
Program training and calibration

References

[1] A. Aoki, K. Hayashi, K. Kishida, K. Nakakoji, Y. Nishinaka, B. Reeves, A. Takashima, and Y. Yamamoto, "A case study of the evolution of jun: An object-oriented open-source 3d multimedia library," in *Proceedings of the 23rd International Conference on Software Engineering*, ser. ICSE '01, Toronto, Ontario, Canada: IEEE Computer Society, 2001, pp. 524–533, ISBN: 0-7695-1050-7. [Online]. Available: `http://dl.acm.org/citation.cfm?id=381473.381535`.

Figure 2: Box diagram of intra-program modules


User consumable output
Clustering Call
Feature Tagging Set
NLTK
Sanitation Routines
Python