# Automated Reading Comprehension Clustering

Charles Hathaway, David Hedin

December 16, 2015

**Abstract**

Determining the level of readability of documents, especially books, has lots of application in the domain of education. It helps to quantify and group books which may be used at a particular reading level, therefore enhancing the classroom experience for both instructors and students. In this paper, we ran an experiment with a variety of books obtained from Project Gutenberg [1] organized by the project maintainers into a 2 groups; books for children, and adult fiction. To further enhance the analysis of this project, we also used our features to try and cluster books into clusters defined by the Flesch-Kincaid readability metrics [2].

## 1 Introduction

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. In this paper, we analyze the results of utilizing several existing features to classify books in addition to a variety of novel features we created. The primary goal is to cluster books in groups representing the original designation of books; adult fiction and children fiction.

Given an input of 3244 books (1416 children, 1938 adult, with 110 books overlapping) we achieved an $B^3$ F-score of 0.629; which is significantly less than the 0.821 base score. In the conclusion we discuss reasons for why this may be.

## 2 Previous Works

Although not extensively, we did evaluate and learn from a number of previous works. Most significantly, we borrowed features from work done by Feng at al. [3]. Feng focused on the concept of documents being difficult to read due to "items fall[ing] out of memory before they can be semantically encoded". With this in mind, most of the novel metrics they designed focus on the number of entities mentioned in a document. Their work was motived to help organize documents by readability for adults with intellectual disabilities.

Our work is different than this in that we are using a much different data source (they were focused on news articles, where we are focused on fiction books). In addition, instead of focusing on adults with intellectual disabilities directly, we are instead more interested in examining books in relation to each other, with the assumption that easier to read books would be easier for both adults with ID, and developmentally delayed children.

### 2.1 Flesch-Kincaid Readability Calculations

To calculate the Flesch–Kincaid readability score, we apply the following function:

$$206.835 - 1.015(\frac{total\_words}{total\_sentences}) - 84.6(\frac{total\_sylllables}{total\_words})$$

To calculate the Flesch–Kincaid readability grade, we apply the following function:

$$0.39(\frac{total\_words}{total\_sentences}) + 11.8(\frac{total\_sylllables}{total\_words}) - 15.59$$

## 3 Methodology and system design

The system is written in Python, and utilizes the Natural Language ToolKit (NLTK) [4]. The toolkit was trained using a portion of the Penn Treebank and the Conll2000 corpus. The portion used was provided by NLTK via it's distribution manager (nltk.download).

Once the corpus was downloaded and configured, we built the system in 3 stages. The first was retrieving the document to be analyzed; this was done ahead of time to verify the availability of the resources, and allow us to develop without worrying about going over any kind of bandwidth quota. However, now that development is mostly complete, this step can be skipped and documents can be downloaded directly from a Gutenberg mirror.

The next step is to process the data. This includes using the trained POS tagger, chunker, and NE tagger on all documents, running a variety of feature functions on the documents, then recording the results. Our system writes data to a CSV file as it is processed, and uses a "Book" object to cache the results of each computation between features. We manually reviewed the data to look for outliers that would imply a "book" was not really a book. After looking at the data, we found several bad data points, which included README files, audio book description files and 2 books in Chinese. These greatly skewed the data so they were removed so that clustering would yield reasonable results.

And lastly, we score the results of the system. This process involves reading in the CSV file and training our clustering algorithmn, then clustering our test data. Ultimately, we determine the accuracy of the clustering using a $B^3$ scoring system.

To get a better idea of how each feature contributes to the final score, we run the system multiple times with every combination of features.

### 3.1 System Usage

To setup your system, please follow the instruction in the README.md file provided in the repository.

Once configured, simply run "python src/main.py –help" to get a list of options and configuration settings. The simplest invocation of this command requires you to specify 2 files; the list of adult books, and the list of children books. It will then print the clustering results with the default feature-set to standard out.

## 4 Linguistic Features

Based on previous work, a list of possible features was generated that could be used as a starting point for clustering the data. As well as this list, a new list of features was created with other features that may be useful for clustering.

| |
|---|
| Average number of words per sentence |
| Average number of syllables per word |
| Percentage of words with more than 3 syllables |
| Average number of noun phrases per sentence |
| Average number of common and proper nouns per sentence |
| Average number of verb phrases per sentence |
| Average number of adjectives per sentence |
| Average number of conjunctions per sentence |
| Average number of prepositional phrases per sentence |
| Total number of noun phrases in document |
| Total number of common and proper nouns in document |
| Total number of verb phrases in document |
| Total number of adjectives in document |
| Total number of conjunctions in document |
| Total number of prepositional phrases in document |
| Number of entity mentions in document |
| Number of unique entities in document |
| Average number of entity mentions per sentence |
| Average number of unique entities per sentence |

Table 1: List of possible features from previous work [3]

From Table 1 and Table 2, a number of features were selected and grouped together to form 6 possible metrics for clustering the data. A correlation between nouns and initial results for clustering was the reason for deciding that features that used nouns should be in their own group. These clusters are documented in tables 3-8

| |
|---|
| Average word length in document |
| Total number of unique words in document |
| Ratio of unique words to total number of words in document |
| Ratio of proper nouns to common nouns in document |
| Length of document |
| Average number of proper nouns per sentence |
| Total number of proper nouns in document |
| Total number of passive sentences in document |
| Average number of prepositional phrases per sentence |
| Total number of prepositional phrases in document |

Table 2: List of possible new features

| |
|---|
| Total number of noun phrases in document |
| Total number of proper nouns in document |
| Total number of common and proper nouns in document |
| Ratio of proper nouns to common nouns in document |

Table 3: Document Wide Noun Features

| |
|---|
| Average number of noun phrases per sentence |
| Average number of proper nouns per sentence |
| Average number of common and proper nouns per sentence |

Table 4: Sentence Wide Noun Features

| |
|---|
| Total number of verb phrases in document |
| Total number of adjectives in document |
| Total number of conjunctions in document |
| Total number of prepositional phrases in document |

Table 5: Document Wide Non-Noun Features

| |
|---|
| Average number of verb phrases per sentence |
| Average number of adjectives per sentence |
| Average number of conjunctions per sentence |
| Average number of prepositional phrases per sentence |

Table 6: Sentence Wide Non-Noun Features

| Length of document |
|---|
| Average word length in document |
| Total number of unique words in document |
| Ratio of unique words to total number of words in document |

Table 7: Document Wide Statistics Features

| Average number of words per sentence |
|---|
| Average number of syllables per word |

Table 8: Sentence Wide Statistics Features

# 5 Results

Originally, we ran all combinations of features to find the best combination. This was time consuming, and not very telling since there were a few features which clearly were superior to others. To help make more sense of the data, we grouped the features (as described above) and ran all combinations of those groups. This gave us much more manageable output, which is recorded in the table below. The "combination of features" column gives the table ID of the features applied.

| f-score | recall | precision | # in cluster 1 | # in cluster 2 | features |
|---|---|---|---|---|---|
| 0.629 | 0.792 | 0.521 | 411 | 2833 | 4 |
| 0.607 | 0.753 | 0.508 | 559 | 2685 | 3 |
| 0.607 | 0.753 | 0.508 | 559 | 2685 | 3+4 |
| 0.607 | 0.753 | 0.508 | 559 | 2685 | 3+6 |
| 0.607 | 0.753 | 0.508 | 2685 | 559 | 3+8 |

Table 9: Results of clustering: Gutenberg vs Features

Table 9 shows the results of running a $B^3$ scorer on the K-means clustering of the various feature groups we generated. The results are not great; much less than the baseline.

| f-score | recall | precision | # in cluster 1 | # in cluster 2 | combination of features |
|---|---|---|---|---|---|
| 0.587 | 0.705 | 0.503 | 755 | 2489 | flesch_kincaid_grade |
| 0.584 | 0.701 | 0.500 | 2475 | 769 | flesch_kincaid_score+flesch_kincaid_grad |
| 0.561 | 0.650 | 0.493 | 2300 | 944 | flesch_kincaid_score |

Table 10: Results of clustering: Gutenberg vs Kincaid

Table 10 shows the results of attempting to cluster the books by their Kincaid readability score. This helps to demonstrate how inconsistent the Gutenberg clustering is with any other metric; perhaps suggesting that the Gutenberg metrics are incorrect.

| f-score | recall | precision | # in cluster 1 | # in cluster 2 | combination of features |
|---|---|---|---|---|---|
| 0.875 | 0.900 | 0.851 | 411 | 2833 | 4 |
| 0.869 | 0.875 | 0.864 | 2644 | 600 | 4+6 |
| 0.846 | 0.815 | 0.879 | 1063 | 2181 | 4+8 |
| 0.846 | 0.815 | 0.879 | 1067 | 2177 | 4+6+8 |
| 0.845 | 0.812 | 0.880 | 1085 | 2159 | 8 |

Table 11: Results of clustering: Features vs Kincaid

Table 11 shows a strong correlation between the feature clustering and Kincaid readability score. This suggests that the two agree on the binary clustering of the books.

| f-score | recall | precision | combination of features |
|---|---|---|---|
| 0.497 | 0.595 | 0.427 | 6+8 |
| 0.483 | 0.587 | 0.410 | 4+6+8 |
| 0.481 | 0.574 | 0.414 | 8 |
| 0.478 | 0.598 | 0.398 | 4+8 |
| 0.466 | 0.594 | 0.383 | 6 |

Table 12: Results of clustering: Kincaid Grades vs Features

Table 12 shows a discrepancy in this theory; when attempting to cluster the books into the 26 grade level clusters assigned by Kincaid, we have a much smaller correlation.

# 6 Conclusion

Our best results are significantly below the baseline; this can tell us a few interesting things. Either our features are totally useless (which seems unlikely, since the Kincaid readability test has been used for years), or our corpus is not divided well or is difficult to assign features to. Given that Project Gutenberg only houses books who's copyright has expired, this leads to the interesting proposition that language has shifted so much our POS tagger failed to achieve good results; however, we can't test this (at least not easily). We have no annotated corpus' to compare our results with, and therefore no way to verify the accuracy of the POS tagger.

As for the possibility that the gold standard we are using, bookshelves as defined by the Project Gutenberg curators, being not properly categorized raises another point of failure. In Figures 1 and 2, we have 2 sample books from the Project Gutenberg collection. When presented during our proposal, it was difficult for the class to distinguish them with ease. This suggests that the categorization of these books may not be obvious or consistent.

To further support this theory, we can see that the Flesch-Kincaid score does not line up well with the Project Gutenberg clustering. However, several of our features lines up quite well with the Flesch-Kincaid score (see Table 11), which suggests that there may be a proper clustering of the books that Gutenberg did not see.

Given our results, we believe there may be more work to be done in this area. However, before the Gutenberg dataset can be fully utilized, it needs to be cleaned and resorted in a way that may more conform with current perceptions of reading difficulties.

The train from 'Frisco was very late. It should have arrived at Hugson's Siding at midnight, but it was already five o'clock and the gray dawn was breaking in the east when the little train slowly rumbled up to the open shed that served for the station-house. As it came to a stop the conductor called out in a loud voice: "Hugson's Siding!" At once a little girl rose from her seat and walked to the door of the car, carrying a wicker suit-case in one hand and a round bird-cage covered up with newspapers in the other, while a parasol was tucked under her arm. The conductor helped her off the car and then the engineer started his train again, so that it puffed and groaned and moved slowly away up the track. The reason he was so late was because all through the night there were times when the solid earth shook and trembled under him, and the engineer was afraid that at any moment the rails might spread apart and an accident happen to his passengers. So he moved the cars slowly and with caution.

Figure 1: Dorothy and the Wizard in Oz, L. Frank Baum

Steve Tolman had done a wrong thing and he knew it. While his father, mother, and sister Doris had been absent in New York for a week-end visit and Havens, the chauffeur, was ill at the hospital, the boy had taken the big six-cylinder car from the garage without anybody's permission and carried a crowd of his friends to Torrington to a football game. And that was not the worst of it, either. At the foot of the long hill leading into the village the mighty leviathan so unceremoniously borrowed had come to a halt, refusing to move another inch, and Stephen now sat helplessly in it, awaiting theaid his comrades had promised to send back from the town.

Figure 2: Steve and the Steam Engine, Sara Ware Bassett

# References


[1] (Dec. 12, 2015). Project gutenberg, [Online]. Available: `http://www.gutenberg.org/` (visited on 12/12/2015).

[2] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," DTIC Document, Tech. Rep., 1975.

[3] L. Feng, N. Elhadad, and M. Huenerfauth, "Cognitively motivated features for readability assessment," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 2009, pp. 229–237.

[4] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, Association for Computational Linguistics, 2002, pp. 63–70.