Chapter Captor: Text Segmentation in Novels

Charuta Pethe, Allen Kim, Steven Skiena
Data Science Lab

Stony Brook University
Computer Science

Introduction

In novels and literary works, authors define coherent segments by means of sections and chapters.
This improves readability for human readers, providing transition cues for breaks in the story.

Research question: Can we automatically identify natural break points in the text?
We address the task of chapter boundary identification as a proxy for the task of large-scale text segmentation.

Dataset: 9k English fiction books from the project Gutenberg corpus

Evaluation metrics:
1. ***P_k***: A penalty is computed based on whether two ends of a sliding window are in the same segment in prediction and G.T.
2. ***WindowDiff (WD)***: A penalty is computed based on the number of breaks in the window, from prediction and G.T.
3. ***F1 score***: For *exact* break prediction

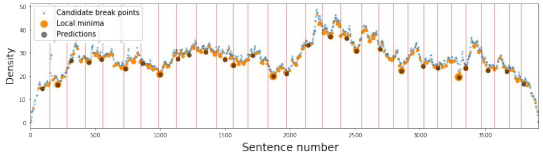
Algorithm	P_k	WD	F1
Best BBP (local)	0.303	0.384	0.447
WOC (window=50)	0.443	0.456	0.144
WOC (window=100)	0.426	0.440	0.158
WOC (window=150)	0.421	0.434	0.162
WOC (window=200)	0.420	0.433	0.164
BBP (single para.)	0.441	0.455	0.128
BBP (full window)	**0.284**	**0.305**	**0.453**

Local Segmentation Methods

Weighted Overlap Cut

Motivation: Chapters are relatively self-contained in terms of the words that they use.

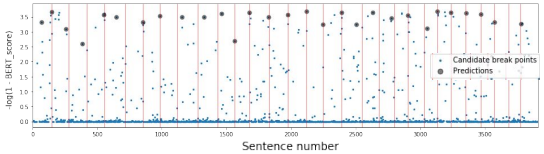
Technique: For each potential break point, compute its density as the sum of counts of common lemmas across the boundary, weighted by their distance from the break point. We expect break points to appear as local minima in the density graph.

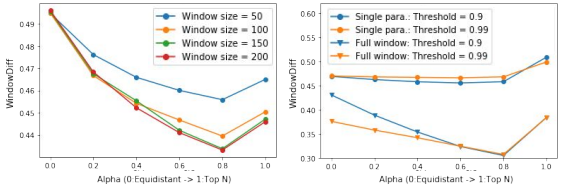


BERT for Break Prediction

Motivation: A deep neural model can detect whether a sentence is a logical continuation of the previous sentence.

Technique: Fine-tune BERT for the Next Sentence Prediction task, to predict whether the sentence after the break point is a continuation of the sentence before the break point.





WindowDiff error metric for global segmentation

Global Segmentation

Motivation: With the local segmentation approaches, the model can place two breaks very close to each other, when realistically, chapter breaks are spaced fairly apart in practice.

Technique: We use a dynamic programming approach to focus on both: predicting high probability break points, and keeping them equidistant.

We define the cost of inserting a break point at n and k breakpoints in sentences 0 to n - 1 as:

$$\text{cost}(n,k) = \min_{i \in [0,n-1]} \left(\text{cost}(i, k-1) + (1-\alpha) \frac{|n-i|}{L} \right) - \alpha \cdot s_n$$

where α is the importance given to confidence scores, and s_n is the confidence score of sentence n.

Results: Our best model achieves an F1 score of 0.453 on the task of *exact* break prediction.