Evaluating the Effectiveness of TF-IDF and Word Embedding Models with Cosine Similarity to Predict Plot Similarity Between Books and Films

Jackson Reinhart Jakob Lamber Jean Luis Adrover Yixiao Zhang
New York University
[jar10020, jml10005, ja4146, yz7999]@nyu.edu

Abstract

This paper presents novel methods to engage modern audiences with books through predicting their relevance from movies. We first developed a baseline system on top of the CMU Movie Summary Corpus and the CMU Book Summary Corpus, utilizing Term Frequency-Inverse Document Frequency (TF-IDF) and Cosine similarity. Our goal was to identify thematic and narrative connections across the popular mediums. To this end, we leveraged genre as a primary indicator of relatedness, scoring our system on its overlap. Early evaluations of our system revealed its limitations in capturing nuanced connections, prompting a series of enhancements. We performed stricter pre-processing (removing proper nouns), accounted Jaccard Score into our vectors, and utilized both Word2Vec for embeddings and DistiliBERT. Comparative assessments indicated that DistilBERT, combined with Cosine Similarity, significantly outperforms other models in detecting intricate narrative similarities. This paper demonstrates great potential for connecting modern audiences to books through film and sets a precedent for further cross-media recommendation systems.

1 Introduction

While generalized semantic analysis of text has become accurate, it still struggles with extracting complex, abstract ideas such as plot structure or thematic elements from texts. We aim to develop a foundational method of textual narrative analysis to predict the relatedness of two pieces of text relative to their plot structures.

The world has become a war for user attention. Cinematic giants like Disney and Warner Bros. dominate the space through their expensive productions of epic stories where we follow characters larger than life. While these blockbusters draw the attention of audiences worldwide, their high production costs and massive appeal have overshadowed a quieter, yet equally significant cultural cornerstone: books.

Americans on average are reading less and watching more (for The Arts, 2002). Polling shows the average number of books Americans read per year has fallen to a thirty year low, at 12.6 per year in 2021 compared to 15.3 in 1990. Simultaneously, the amount Americans have spent at the box office has increased year over year, as has tickets sold, both rising nearly forty percent from 1995 to 2019 (Jones). Most concerning of all is the slow decrease in average grade twelve standardized reading scores across the country that has occurred since 1998 (NAEP). And so we ask the question: what if we could create a connection between reading and watching that helps to grow their common audience instead?

It is to that end that we developed our system. With an accurate method of deriving narrative structure in text and an accurate method of comparing those plots to others could come an accurate way of relating texts through their plot structures. Thus, our goal is a system that, given a plot summary of a film, could generate a list of related books. Our system will engage new audiences, make reading more accessible, more culturally relevant, and further ameliorate the common problem of not knowing what to read.

2 Data

We made use of the CMU Movie Summary Corpus and the CMU Book Summary corpus as our corpora of plot summaries. The two corpora consist of plot summaries of various books and films, scraped from their respective Wikipedia page's plot section, ensuring a consistent format and style across summaries and between the two corpora. The former contains 42,306 movie plot summaries

Table 1: Genres Pre-Normalization

Media	Genre
The Sorcerer's Apprentice	Family Film, Fantasy, Adventure, World Cinema
A Clockwork Orange	Science Fiction, Novella, Speculative fiction

Table 2: Genres Post-Normalization

Media	Genre
The Sorcerer's Apprentice	Fantasy, Adventure
A Clockwork Orange	Science Fiction, Speculative fiction

and associated metadata (genres, runtime, languages, etc.), and the latter contains 16,559 book plot summaries with similar metadata (genres, author, year, etc.). Each plot summary is a comprehensive, multi-paragraph summary of the major narrative points of each piece of media.

In the evaluation of our system, we used genre as an indicator for relatedness. This presented a challenge with our data, as despite having similar themes and story elements, films and books both have genres that are intrinsic to their format. That is to say, the intersection of the set of all genres in the movie corpus with the set of all genres in the book corpus was not equal to the cardinalities of each corpus; there were items in each that were not in the other. To solve this, we decided to use only the genres in the intersection of the two genre sets for our evaluation, and so we normalized the data for each film and book to remove genres outside that intersection. For example: in tables 1 and 2, post-normalization the genres "Family Film," "World Cinema," and "Novella" are removed.

After normalization, we were left with fifty three distinct genres in our corpora with which to evaluate our results.

3 Scoring and Evaluation

Scoring the output of such a system proved to be challenging. Given the absence of a definitive 'answer key' to reference concerning proper book recommendations from a movie summary, we required a method to approximate the relevance and accuracy of our recommendations. And so, we honed in on structuring scoring metrics through one piece of available meta-data in *both* sets: genre.

We believe that genre characterizes these two mediums in a manner particularly relevant here. Genre, in short, articulates an overarching narrative tone. This high level representation of a film or book is not only available in both sets but comparable as well. Through comparing genres, we can gauge if the two share similar narrative frameworks and thus appeal, making genre a practical proxy for assessing recommendation relevance.

To quantify the similarity between the genre sets we employed two metrics:

1. Accuracy was used as a lenient signifier of performance. We chose to define correct outputs as those sharing at least one genre between each other and incorrect as no genre overlap. And so, accuracy was the percentage of outputs that correctly produced one such match.

2. Jaccard score was used as a strict signifier of performance. It is defined as follows:

$$\text{Jaccard Score} = \frac{|A \cap B|}{|A \cup B|}$$

where A and B are the sets of genres associated with the movie and the book, respectively.

This score measures the similarity between two sets by dividing the size of the intersection by the size of the union of the sets. In our case, this calculation provides a quantitative and more precise assessment of genre-based similarity.

4 Initial Methodology Using TF-IDF and Cosine Similarity

4.1 TF-IDF and Cosine Similarity

TF-IDF followed by cosine similarity is a common language processing paradigm that we developed as a baseline for our system. Combining these two metrics allows us to assess document similarity in a refined and effective manner. Outlined below we further explain the approach.

In short, TF-IDF is the combination of two representative methods of textual analysis: Term Frequency, Inverse Document Frequency. The former tracks the number of times that a term appears in a document. This value is then normalized by the

length of the document. The equation is defined as follows:

$$TF = \frac{O_{t,d}}{T_d}$$

Where $O_{t,d}$ is the total number of occurrences of the term t in a document d, and T_d is the total number of terms in d.

The latter gives a weight relative to how commonly a word is used. If a word is more frequent across documents, the lower its score. It is therefore a less important word for characterizing documents. The equation is defined as follows:

$$IDF_{t,d} = \log \frac{N}{\mid d : t_i \in d \mid}$$

Such that N is the total number of documents in the corpus, and $\mid d : t_i \in d \mid$ is the number of documents in the corpus in which the term t occurs at least once. TF-IDF is the final product of both these values.

Putting the two together:

$$tfidf = TF \cdot IDF$$

TF-IDF thus translates to a high value for terms with large term frequencies in one document coupled with a low frequency across the entire set of documents. This gives weights that accurately reflect how well a term can inform our classification.

Cosine similarity on the other hand is a method for guaging similarity between vectors and is defined as follows:

Given two n-dimensional vectors of attributes, A and B, we may define the cosine similarity as their dot product divided by the product of their magnitudes:

$$\text{Cosine Similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Where:

- $\mathbf{A} \cdot \mathbf{B}$ represents the dot product of vectors $\mathbf{A}$ and $\mathbf{B}$.
- $|\mathbf{A}|$ and $|\mathbf{B}|$ denote the Euclidean norms (or magnitudes) of vectors $\mathbf{A}$ and $\mathbf{B}$ respectively.

The resulting value with a range from -1 to 1 represents the similarity between the two vectors quantified by the cosine of the angle between the vectors projected in a multi-dimensional space.

For our purposes, vectors A and B both are composed of components A_i or B_i which reflect weighted term frequencies. Our resulting scores may be understood as follows:

- 0 denotes orthogonal vectors: no similarity
- 1 denotes parallel vectors: the same vector
- -1 denotes anti-parallel vectors
  - Since it is impossible to have a negative count of words, all elements of these vectors are zero or positive. Therefore dot product will also remain positive and -1 will never be output in our system.

4.2 Implementation

For implementation of our analysis framework, we leveraged the *scikit-learn* Python library (Pedregosa et al., 2011). Our decision was based on the library's widespread acceptance in the natural language processing space and its ease of use.

To initially transform our text summaries into vectors we can compare we employed *scikit-learn's TfidfVectorizer* which computes our TF-IDF weighted vectors described above for each term in the corpus, producing a multi-dimensional vector of unique terms.

After transforming the documents, we again employed *scikit-learn's* implementation of cosine similarity. This function compares the vectors in our weighted multi-dimensional space, measuring the cosine of the angle between the two, resulting in our similarity score.

This method was scaled to the entirety of both datasets. TF-IDF was performed on the combined set of documents and cosine similarity was used to specifically identify similar works *cross-medium*: from movie to book.

4.3 Results

Using this method, our system had promising results. It measures a 45.79% accuracy and an average Jaccard Score of 0.062. This is what we will be using as a baseline understanding of our system performance.

Importantly, the accuracy, while seemingly quite low, is still better than blind guessing, suggesting that the system does have a foundational ability to match books and movies based on genre overlap. This is significant considering the complexities that need to be grasped through a simple

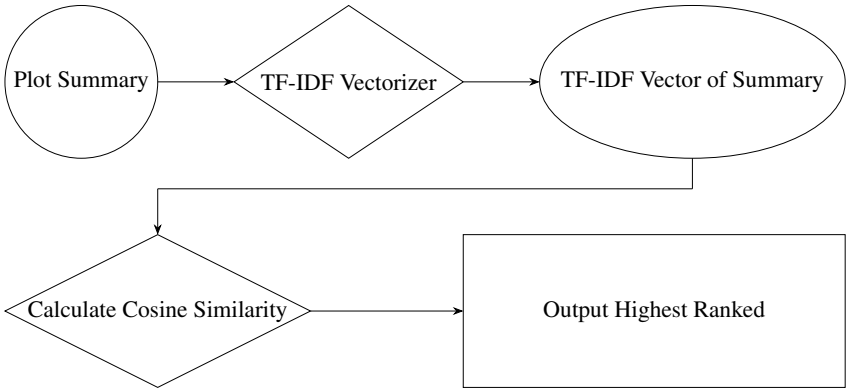

Figure 1: Architecture of Initial Implementation

summary. However, the fact that more than half of the predictions failed to identify any genre overlap indicates substantial room for improvement. This could be attributed to the inherent limitations of the TF-IDF model in capturing deeper semantic meanings rather than just superficial textual similarities.

The average Jaccard Score of 0.062 further illustrates the challenges faced by the initial system. A low Jaccard Score suggests that while the system can occasionally identify correct genres, it generally predicts only a small subset of overlapping genres. This indicates that the system's successful predictions are not comprehensive and fail to capture the full spectrum of thematic elements that could potentially link books and films more accurately.

The disparity between the accuracy and the Jaccard Score suggests that while accuracy can indicate a general alignment in genre, it does not reflect the depth or the quality of that alignment. The Jaccard Score, by providing a ratio of the intersection to the union of predicted and actual genres, offers a more nuanced view of how well the system's predictions align with the true genre composition of the texts.

These initial results however necessitate more advanced methodologies to improve both metrics accordingly. In the subsequent subsection, "Improvements on Our Initial Methodology," we will explore various strategies and refinements aimed at addressing these shortcomings.

5 Improvements on Our Initial Methodology

Initial improvements were made by adjusting the parameters of the Term Frequency-Inverse Document Frequency (TF-IDF) model. The modifications included adjusting the *min_df* and *max_df* parameters of the vectorizer to exclude terms appearing in an excessively high number of documents or only in a few, thus refining the feature space to better capture significant textual features without the interference of too common or too rare terms. The best results were achieved with *min_df* and *max_df* values of 0.01 and 0.9 respectively. All other values, especially when *min_df* was increased, resulted in lower accuracy and Jaccard scores. [1].

5.1 Proper Noun Removal

The initial methodology employed TF-IDF for vectorizing text data, followed by cosine similarity for assessing the relevance between documents. While effective, the approach required enhancements to better handle specific textual elements such as proper nouns, which often skewed similarity assessments due to their frequency rather than their relevance to text meaning. In Table 4, evidence of this is obvious. Our initial system recommends ten books it finds related to the movie *A Clockwork Orange*. The first and highest ranked result is the book the film was based on, a sign that our system does successfully act as expected as far as recommending source material goes; but eight of the other nine results are books from the spy novel series *Alex Rider*. Evidently, the connection our system makes in this case is that the main character's name in both *A Clockwork Orange* and the *Alex Rider* series is Alex, a connection not exceptionally relevant to our goal of predicting plot similarity.

To address this, proper nouns were directly ig-

[1] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Film	Recommended Books
A Clockwork Orange	*A Clockwork Orange* ***Stormbreaker*** ***Skeleton Key*** *Little Boy Blue* ***Double Cross*** ***Eagle Strike*** ***Point Blanc*** ***Ark Angel*** ***Snakehead*** ***Scorpia***

Table 4: Our initial system's book recommendations for *A Clockwork Orange*, in order of most relevant, descending. *Alex Rider* series novels bolded.

nored in the improved system through the use of the *spaCy* Python library (Honnibal and Montani, 2017). After tokenizing the normalized plot summaries, our system utilizes *spaCy* for part-of-speech tagging to identify proper nouns. These nouns were then subsequently removed from the texts before vectorization. This refinement aimed to reduce noise within the data, ensuring that the TF-IDF model emphasized thematic and contextual elements rather than entity-specific terms.

5.2 Dual Comparison With Jaccard Similarity

To augment the cosine similarity measurement, a Jaccard similarity coefficient was introduced as an additional comparative layer assessing genre overlap. This method provided a means to quantify the similarity based on the presence and absence of shared keywords between documents, offering a complementary metric to the angle-based cosine similarity. This dual-measurement approach allowed for a more nuanced analysis of text similarities, for different aspects of text relationship and relevance.

5.3 Results

In the improved system, we observe an advancement over the initial setup in terms of accuracy and Jaccard score, as detailed in Table 6. The modifications have effectively increased the average Jaccard score from 0.0618 to 0.0697 and improved accuracy from 46% to 49.43%.

6 Exploring Different Vectorization Methodologies

6.1 Word2Vec Embedding

To further improve the system, we incorporated the Word2Vec model (Mikolov et al., 2013).

Word2Vec utilizes the technique word embedding to represent words as vectors in a continuous space. In this space, words that have similar meanings are positioned next to each other. This model excels at recognizing and preserving the details of language in a vast collection of texts, which is extremely helpful for our objective of comparing movie plot summaries with book plot summaries. Word2Vec's neural network design enables it to efficiently learn word associations from extensive datasets, resulting in a more dynamic and contextually aware representation of text.

We trained the model on our corpora. This training enabled Word2Vec to capture a diverse range of narrative elements and themes that were previously dominated by the high-frequency yet contextually restricted words that are usually emphasized by TF-IDF. Afterwards, the narrative embeddings produced by Word2Vec were used to calculate cosine similarities between texts, enabling a new recommendation system.

The performance marginally improved with the integration of Word2Vec. Prior to this integration, our system, relying solely on TF-IDF, achieved an Overall Average Jaccard Score of 0.0618 with a corresponding accuracy of 0.4579. After implementing the Word2Vec model, there was a slight improvement: the Overall Average Jaccard Score increased to 0.0822, and the accuracy increased to 0.5444. This was an 18% increase in accuracy.

The integration of Word2Vec yielded an improved performance in detecting text similarities. This improvement is reflected in a noticeable reduction of cases where no commonalities (in terms of genre) between texts were identified (the increase in accuracy), indicating a potential increase in the reliability of our analysis framework. This approach has demonstrated potential in improving the system's ability to suggest books based on movie summaries by boosting the level of detail and precision in text comparisons. However, the improvements are not without caveats. The model heavily depends on pre-trained word embeddings, which may encounter difficulties in handling newly emerged slang, culturally specific expressions, or most relevant to the task, fictional

words (such as "lightsaber") that are not adequately captured in the training data, a trade-off in its improvement over TF-IDF which lacks any such limitations.

6.2 DistiliBERT Embedding

In our endeavor to refine text similarity analysis for linking cinematic and literary content, the DistilBERT model was integrated to improve the performance of our existing methodologies, namely TF-IDF and Word2Vec.

DistilBERT is a distilled version of the BERT (Bidirectional Encoder Representations from Transformers) model, designed to maintain most of BERT's capabilities while being smaller and faster (Sanh et al., 2019). DistilBERT is trained via knowledge distillation, a technique that enables it to replicate the functionality of BERT while utilizing fewer parameters and requiring less processing resources. The simplified nature of this approach is advantageous for our research, as it enables the effective handling of extensive datasets containing plot summaries of movies and books, without significantly compromising performance.

The DistilBERT model was employed to generate dense embeddings from plot summaries, enhancing our system's ability to assess textual similarity. The implementation involved preprocessing text data using the DistilBERT tokenizer, which converts text into a format suitable for the model, and then computing embeddings through the model's transformer network. These embeddings represent text in a multidimensional space, efficiently capturing contextual connections between words. This allows DistilBERT to identify thematic and narrative similarities with greater precision. Unlike TF-IDF and Word2Vec, which focus on specific aspects like term frequency or word proximity, DistilBERT captures both individual word meaning and broader contextual information, leading to a more comprehensive analysis.

The text data was tokenized and encoded using the DistilBERT tokenizer. This successfully prepared the input for processing by the model. In order to enhance memory usage and computational efficiency, the encoded texts were processed in batches. The obtained embeddings were subsequently utilized to calculate cosine similarities between between movie and book summaries.

7 Comparing Methodologies

Observe the following results for the discussed similarity metrics:

- TF-IDF: Achieved an Overall Average Jaccard Score of 0.0618 with an accuracy of 0.4579.
- Word2Vec: Improved upon TF-IDF, recording a Jaccard Score of 0.0822 and an accuracy of 0.5444.
- DistilBERT: Marked a significant enhancement with a Jaccard Score of 0.1059 and an accuracy of 0.6359.

The measurements indicate that DistilBERT not only generated more comprehensive embeddings, but also better matched plot components between the two mediums, resulting in fewer cases where no similarities could be detected.

See Table 4 for full evaluation results. By including DistilBERT into our text similarity analysis methodology, we greatly improved our capacity to identify and measure similarities between movie and book plot summaries. This methodological improvement aligns with our objective to effectively calculate similarity between movie and book genres. Additional fine-tuning could be performed to fully exploit the possibilities of DistilBERT. Exploring alternative BERT models that may be more aptly suited for this specific task could yield superior results as well.

8 Related Work

8.1 Combining Vectors with Matrix Norms

Vor der Brück and Pouly propose combining word vectors using matrix norms as an improved way of measuring semantic document similarity superior to cosine similarity (vor der Brück and Pouly, 2019). This approach may offer a potential direction for enhancing our method of comparing books and movies, and suggest that relying on traditional cosine similarity might overlook unseen relationships within the data. Adopting the usage of matrix norms could possibly generate improved recommendations that better identify thematic and narrative connections between summaries.

8.2 B. Rex: A Similar System

Abrams, Gessler, and Marge developed a similar system, B. Rex, to recommend books to children

Approach	Avg. Jaccard Score of All Genre Sets	Accuracy	Percentage of Zero Scores
Initial Approach Using TF-IDF and Cosine Similarity	0.0618	0.4579	0.5421
Initial Approach with Proper Nouns Removed and Dual Jaccard and Cosine Similarity	0.0697	0.4943	0.5057
Word2Vec Model with Cosine Similarity	0.0822	0.5444	0.4556
DistiliBERT Model with Cosine Similarity	0.1059	0.6359	0.3641

Table 6: Evaluation Results of Initial Model and Improvements

based on a chat conversation they have with the system (Abrams et al., 2019). B. Rex generates its recommendations by sending data like preferred genre and author to the Goodreads API, which then returns books for the system to recommend to the user. This system is similar to ours in its general purpose, but differs in its means: we attempted to develop a new system for generating recommendations, while B. Rex was developed to better deliver already generated recommendations. B. Rex is however an excellent example of how the goal we sought can be effectively packaged and used for the promotion of literacy.

Abrams et al. surveyed users ($n = 8$) and found B. Rex's recommendation methodology was received well: "according to survey responses, it was for most users just slightly worse than a recommendation from a friend". We evaluated our system by calculating the Jaccard score of the genre sets of books and films determined to be related, as well as measuring if there was at least one overlapping genre between the sets. Surveying users on their experience with our system similar to Abrams et al. could help us better evaluate its effectiveness.

8.3 Narrative Alignment

Pial, Salim, Pethe, Kim, and Skiena explore the adaptation process of books into films using the Smith-Waterman local alignment algorithm and SBERT embedding distance to derive similarity between scenes and narrative units in books (Pial et al., 2023). They are then able to map specific plot points to scenes on screen and analyze the differences and similarities in source text and the derived film. This approach to quantifying units of narratives is an interesting one and a potential place to develop our system further.

8.4 Analyzing Texts of Varying Length and Style

Gong, Sakakini, Bhat, and Xiong identify problems in measuring document similarity between documents with different lengths and styles, and propose comparing such documents using hidden topics (Gong et al., 2019). Our current system was designed assuming similar length and style summaries to ensure best possible results. This assumption may not be practical in the case of widespread adoption of our system, and thus implementing this method of compensating for potentially incompatible documents could make our system more robust.

9 Future Improvements

A direction for future research involves the development of a mechanism to incorporate user feedback into the system. This would enable users to provide evaluations concerning the relevance of content recommendations, particularly for adaptations from books to movies and vice versa, based on their plot summaries. By leveraging such feedback, the system could be retrained to better recognize and align with user expectations, thereby enhancing its precision and relevance over time. This feedback mechanism could prove to be valuable, as genre alignment does not fully capture the subtleties of narrative similarity.

10 Conclusion

Analyzing and comparing semantic differences in text as abstract and complex as plot structure and narrative form proved to be a difficult, but generally achievable, challenge. Our highest scoring system was able to accurately predict relatedness between book and film plot summaries, and suggest relevant book recommendations for any given film as input. The most effective approach used the DistilBERT model to produce text embeddings that were then analyzed to calculate cosine similarities.

We hope this research can provide a foundational methodology of plot analysis and comparison with which cross-media recommendation systems can be easily and accessibly built to further improve literacy in the United States and around the world. With our system, or improvements on it, books and films can share the spotlight and no longer have their potentials for success be inversely correlated.

Our results establish a promising starting point, and we hope to stimulate similar research in the field, in the area of generating recommendations like this paper has and like the work of Pial et al. We see similar works dive more deeply into delivering those recommendations to audiences in an accessible and engaging formats as in the work of Abrams, Gessler, and Marge. Such research can help improve content recommendation systems to not only cater to user preferences but also enhance their engagement and satisfaction to any given platform.

—

References

Mitchell Abrams, Luke Gessler, and Matthew Marge. 2019. B. rex: a dialogue agent for book recommendations. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 418–421.

Tim vor der Brück and Marc Pouly. 2019. Text similarity estimation based on word embeddings and matrix norms for targeted marketing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1827–1836.

Hongyu Gong, Tarek Sakakini, Suma Bhat, and Jinjun Xiong. 2019. Document similarity for texts of varying lengths via hidden topics. *arXiv preprint arXiv:1903.10675*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Jefrey M. Jones. Americans reading fewer books than in past. Citing of a Survey. For more information visit: https://news.gallup.com/poll/388541/americans-reading-fewer-books-past.aspxmethodology-388541.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

NAEP. Naep report card: Reading. For more information visit: https://www.nationsreportcard.gov/reading/nation/scores/?grade=12.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Tanzir Pial, Shahreen Salim, Charuta Pethe, Allen Kim, and Steven Skiena. 2023. Analyzing film adaptation through narrative alignment. *arXiv preprint arXiv:2311.04020*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

National Endowment for The Arts. 2002. Reading at risk.

—