Identifying Maturity Rating Levels of Online Books

Eric Brewer and Yiu-Kai Ng
Computer Science Department, Brigham Young University, Provo, Utah 84602, USA
Email: brewer@byu.edu, ng@compsci.byu.edu

***Abstract*—With the huge amount of books available nowadays, it is a challenge to determine appropriate reading materials that are suitable for a reader, especially books that match the maturity levels of children and adolescents. Analyzing the age-appropriateness for books can be a time-consuming process, since it can take up to three hours for a human to read a book, and the relatively low cost of creating literary content can cause it to be even more difficult to discover age-suitable materials to read. In order to solve this problem, we propose a maturity-rating-level detection tool based on neural network models. The proposed model predicts a book's content rating level within each of the seven categories: (i) *crude humor/language*; (ii) *drug, alcohol, and tobacco use*; (iii) *kissing*; (iv) *profanity*; (v) *nudity*; (vi) *sex and intimacy*; and (vii) *violence and horror*, given the text of the book. The empirical study demonstrates that mature content of online books can be accurately predicted by computers through the use of natural language processing and machine learning techniques. Experimental results also verify the merit of the proposed model that outperforms a number of baseline models and well-known, existing maturity ratings prediction tools.**

I. Introduction

A recent report has observed that people in the USA spent on average over ten hours per day viewing, listening to, or reading some form of digital media in 2018[1]. Similar reports reveal the same truth holds in other countries such as the U.K.[2] The ever-increasing amount of time spent on digital media poses the question: "What effect does consumption of digital media have on its consumers?"

Various studies have been conducted on the myriad of effects of digital media consumption. Some of these studies reveal that exposure to mature content within digital media can instigate behavior related to that content. For example, Gentile et al. [9] discover strongly positive correlations between media violence exposure and verbal and physical aggression in elementary school students, whereas Coyne et al. [7] report that exposure to profanity in media can cause increased leniency towards the use of profanity, followed by its increased use of profanity, then finally, increased tendencies towards acts of aggression. A study conducted by Bleakly et al. [1] verify a bi-directional causal relationship between sexual content exposure and sexual behavior in teenagers. Separate studies by Sargent et al. [18] and Titus-Ernstoff et al. [20] link tobacco exposure in media to tobacco use in youths.

To combat some of the negative effects of mature content in digital media, various organizations have proposed and implemented *content rating systems*. Organizations which implement these systems perform two main functions. First, an item within a digital medium is gathered and *rated* according to its mature content. Ratings are meant to be objective and unbiased and are not to be construed as a judgment on the quality of that media item. Second, the assigned content rating is prominently *displayed* alongside the media item, designed to be visible to the consumer at the point of purchase. A content rating for a media item provides information about the level of maturity of that item's content and ultimately helps that consumer decide whether to view that media.

TABLE I
Content ratings given to films by the MPAA

Rating	Name	Description
G	General Audiences	All ages admitted
PG	Parental Guidance Suggested	Some material may not be suitable for children
PG-13	Parents Strongly Cautioned	Some material may be inappropriate for children < 13
R	Restricted	< 17 requires accompanying parent or adult guardian
NC-17	No One 17 and Under Admitted	

Today, most forms of digital media—films, television, video games, music, and mobile applications—may request (or are required to obtain) a content rating from an authoritative rating system in the country where that media is published. For example, the MPAA[3] provides content ratings to films released in the USA using a system based on age-appropriateness as depicted in Table I.

As it stands, literature is freely published without any unified rating system, i.e., consumers may not be informed of any mature content in a book before they read it. Though third-party book rating systems, e.g., CommonSenseMedia, have been developed, none have become adopted standards to which book publishers can turn.

In this paper, we consider the computer-aided *literary content analysis* through the use of natural language processing (NLP) and machine learning techniques. We frame the problem as a traditional *supervised machine learning problem* where our model takes as input the title and text of a book and predicts as output the *maturity rating levels* for that book. Our work, which benefit the reading community as a whole, specifically parents and children, is a solid example of overcoming the challenge of extracting textual features over very large documents and of a multi-modal deep learning architecture.

II. Related Work

Algorithmically capturing the semantics of words, sentences, or documents is a challenging task for computer algo-

[1] nielsen.com/us/en/insights/reports/2018/q1-2018-total-audience-report.html

[2] emarketer.com/content/time-spent-with-media-plateaus-in-uk

[3] Motion Picture Association of America, www.mpaa.org/film-ratings/

rithms. Since natural language is replete with imagery, polysemy, and loose syntax, and requires a broad understanding of the world, building powerful language models is very difficult.

Chen et. al. [4] have investigated the discrepancies between maturity ratings for applications published through two of the largest mobile app services, Apple App Store and Google Play. In their study, they build a maturity rating classifier using a vocabulary of human-selected seed terms extracted from user reviews. Their work showed promise in that mature content can indeed be accurately detected in text.

Following the work done by Chen et al., Hu et al. [10] perform an analysis of mature content in mobile applications. They hand-pick *sensitive* words from the App Store and Google Play maturity rating policy descriptions, which were augmented with similar words. Their data set, however, significantly differs from ours in terms of size—an app description is on the order of a few paragraphs, whereas a book in our data set contains hundreds of paragraphs.

A study similar to ours has been conducted by Wanner et. al. [21] who estimate mature contents of books by compiling a hand-picked list of *sensitive* terms for each of six topics: *war*, *crime*, *sex*, *horror*, *fantasy*, and *science fiction*. They then extend their lists of sensitive words with synonyms and hypernyms taken from WordNet [15]. Our work, however, avoids using manually compiled word lists in the prediction phase on detecting mature content in books.

Other related works include Razavi et al. [16] who detect offensive/abusive phrases in text messaging through the Internet/cellular phones, Yenala et al. [22] who address the problem encountered by online discussion fora on abusive and discourteous comments made by users, and Merayo-Alba et al. [14] who present an approach using NLP in detecting violent content on text documents circulated on social networks. None of them, however, focuses on the maturity rating level problem.

III. Document Classification

Our task qualifies as document classification, since it is to classify the level of maturity for a book given a textual input—the text of the book. In order to perform numerical analysis on text, i.e., presenting the text to a machine learning classifier, the entirety of the text is transformed somehow into numerical values using *vectors* to represent all of the individual word (or character) tokens in the document. We present the preprocessing step, i.e., *tokenization*, and discuss the merits and drawbacks of the quantification approach.

A. Tokenization

To a computer, a document is simply an ordered sequence of symbols where each symbol is represented by a unique byte value or sequence of bytes values. Before using the word vector approach, an ordered sequence of symbols must be divided into smaller, more meaningful parts, usually as an ordered sequence of markers, i.e., punctuation, and words, collectively called *tokens*. Since in the English language, words are separated by a space, a simple first step would be to divide the document into parts by splitting on one or more whitespace symbols. In fact, each tokenization decision can directly affect the size of the vocabulary and the representations of documents as bag-of-words or word vectors[4]. For our task, we reduce the size of the vocabulary as much as possible, which would increase the likelihood that any particular token in a document could be uniquely represented. Hence, we converted all words to lowercase and treated end-of-sentence markers as separate tokens.

B. Word Vectors

A collection of static *word vectors*, also called a word *embedding*, can be obtained using algorithms such as *word2vec*, *GloVe*, or *fastText* over a text corpus. Such algorithms construct vectors, or an embedding, for word tokens by first creating a mapping from tokens to arbitrary points in an n-dimensional space, then iteratively optimizing these vectors in such a way that tokens which frequently occur in similar *contexts* such as `car` and `automobile` (measured by the co-occurrence of surrounding tokens) should have similar vector representations by cosine similarity or euclidean distance, or both. After word vectors are sufficiently optimized, one would expect that different tokens which are used in similar contexts should have similar vector representations, i.e., a cosine similarity close to 1, and/or a small euclidean distance. For our work, we will obtain word vectors for all of the tokens in our data set using an existing *word2vec* algorithm[5]. We highlight two specific neural network architectures, *recurrent neural networks* (RNN) and *convolutional neural networks* (CNN), which utilize static word vectors.

IV. Our Maturity-Level Predictor

Our maturity-level predictor can be treated as a supervised machine learning document classification and retrieval problem, since it is to classify the *level of maturity* for each book given its textual input that is represented as a vector, and retrieves the ones that match the user's maturity level. To give consumers insight toward the suitability of an item of literature as a whole, an *overall* maturity rating is derived from the categorical maturity rating levels.

A. Data Collection

Prior to building a predictive model, we obtained both the input and output data. We began by collecting the output data, i.e., content ratings for books. After searching through various third-party, online rating systems for literature, we settled on the one that was the most robust and complete—Book Cave[6]. Book Cave publishes a database of over 11,000 books, each of which is given at least one overall content rating. Ratings for books on Book Cave are supplied by humans who have acknowledged that they have read through the entire book at least once. We treat these ratings as *ground-truth*.

Book Cave users assign content ratings to books across the following seven categories: (i) *crude humor/language*; (ii)

[4]A popular algorithm for tokenizing documents is the Stanford tokenizer (https://nlp.stanford.edu/static/software/tokenizer.shtml).

[5]https://radimrehurek.com/gensim/models/word2vec.html

[6]https://mybookcave.com/mybookratings/

TABLE II
MERGED RATING LEVELS WITHIN THE *profanity* CATEGORY. A CLASS ID VALUE AND DERIVED ORDINAL CLASS REPRESENTATION BASED ON THIS ID ARE SHOWN FOR EACH MERGED RATING LEVEL.

Rating Level	Overall Rating Yield	Class ID	Class Ordinally
None	All Ages	0	0 0 0
Mild language	Mild(+)	1	1 0 0
Some profanity (6-40) Moderate profanity (41-100)	Moderate Moderate+	2	1 1 0
Significant profanity (101-200) Significant profanity (201-500) Extensive profanity (501+)	Adult Adult Adult+	3	1 1 1

drug, alcohol, and tobacco use; (iii) *kissing*; (iv) *profanity*; (v) *nudity*; (vi) *sex and intimacy*; and (vii) *violence and horror*. An overall rating for a book is determined by the maximum rating yielded by the assigned categorical rating levels over all categories. The possible overall ratings for books are: *All Ages*, *Mild*, *Mild+*, *Moderate*, *Moderate+*, *Adult*, and *Adult+*. If more than one user has rated a book, all of the assigned overall ratings are *averaged*, rounding up to the nearest overall rating. We collected the overall ratings and categorical rating levels for all (∼11,000) books in the Book Cave database.

The next, more difficult step came when we set out to collect book texts. Luckily for us, the web page for over 98% of books in the Book Cave database also includes a hyperlink to the Amazon Store web page for the Kindle (electronic) version of that book. From the Amazon Kindle Cloud Reader[7] web page we managed to collect the entire texts in English for over 6,000 books. The texts were already broken up into paragraphs, which is an important aspect. These book data provided a sufficiently-sized data set to build a predictive model.

B. Content Rating Levels

The content of each book in the Book Cave database has been given a categorical rating level in each of the seven maturity categories. The overall maturity rating for a book is derived as simply the *maximum* overall rating yield over the seven categorical rating levels. To encode labels to be used by our predictive model, we grouped rating levels within each category based on their corresponding *base* (i.e., not including a '+') overall maturity rating as shown in Table II.

Because the categorical rating levels naturally represent increasing magnitudes of mature content rather than distinct topics or themes, we follow the approach described by Frank et al. [8] by representing merged categorical rating levels *ordinally*, that is, as a binary vector instead of as an integer, or nominal, value as seen in Table II in the column labelled *Class Ordinally*. The intuition behind the binary ordinal representation is that books with relatively high categorical rating levels may include harsher words in their vocabularies than those with lower rating levels, i.e., vocabularies of former books yield the supersets of vocabularies of latter books. Thus, for a particular category for a book, each '1' in the binary vector expresses that book contains *at least* that magnitude of maturity for this category. A machine learning model benefits from this binary vector representation more so than from the original integer value because the model is free to independently learn whether certain subsets of vocabulary terms are associated with different elements in the vector. Thus, we chose to train our model using the ordinal representation of rating levels as a vector of bits rather than as an integer. (See the column labeled *Class Ordinally* in Table II as depicted in Figure 1.)

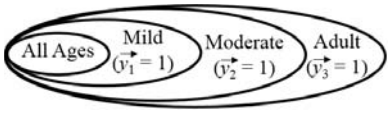

Fig. 1. Ordinal class representation of rating levels

Because the data set has not been curated in any way, the distributions of categorical rating levels are not balanced, i.e., some classes are heavily imbalanced and thus produce a biased accuracy score. We have remedied this by also reporting the mean squared error score (see Section V).

C. Neural Networks

We tested a few different configurations of neural networks for our approach. Each of our network configurations contains two main components which we call *modules*. The first module is the *paragraph encoder*, which takes as input a sequence of word tokens of a single paragraph and produces a fixed-length vector. The second module is the *aggregation* module, which aggregates, or summarizes, all vectors produced by the paragraph encoder for all paragraphs of a book.

The job of the *paragraph encoder* is to learn a representation of any fixed-length token sequence as another fixed-length feature vector. The text of each book in the Book Cave database was already divided into paragraphs for us. We experimented with two different neural network implementations as the *paragraph encoder*. The first is a *recurrent neural network* (RNN), and the second is a *convolutional neural network* (CNN). Note that the paragraph CNN contains only the convolutional layer that looks at adjacent words (n-grams) in a paragraph, whereas the paragraph RNN looks at each word in a paragraph sequentially and can detect long-range co-occurrences of words. The paragraph CNN+RNN aggregates the vectors resulting from both a CNN and an RNN layer.

1) Recurrent Neural Network (RNN) Module: Using a recurrent neural network (RNN) alongside word vectors would more accurately capture contextual patterns (i.e., those that can be observed sequentially) in book text than would a bag-of-words representation, since bag-of-words models do not preserve the order of tokens in the text, which we assume is relevant. Our RNN module used as the paragraph encoder is constructed as follows, and its structure is depicted in Figure 2.

Input. The input to the network is a sequence of 128 tokens representing a single paragraph, since *fewer* than 4% of paragraphs in our data set contained more than 128 tokens.

Embedding Layer. This layer simply map each word token in the sequence to its corresponding vector representation

[7]https://read.amazon.com/

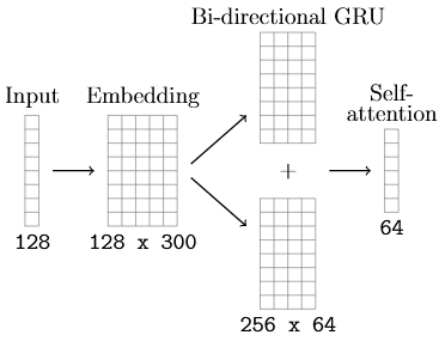

Fig. 2. Output dimensions at each step of the RNN Module

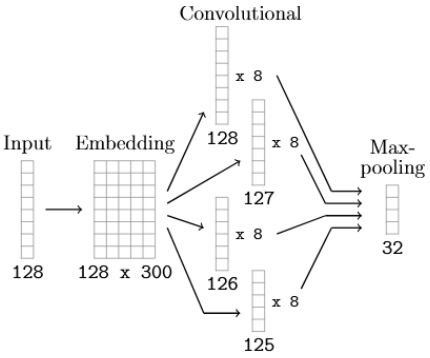

Fig. 3. Output dimensions at each step of the CNN Module

in the 300-dimensional *GloVe* embedding. The result is a sequence of vectors, each of which is of length 300, or, a matrix with 128 rows and 300 columns.

Bi-directional GRU Layer. A *bi-directional* GRU additionally processes the input sequence in the backwards direction. Processing text sequences in both directions tends to lead to better performance [19] with the intuition that contextual clues in text may not always appear in the direction they are written. In addition, we have chosen 64 as the number of dimensions of each hidden state vector. The result is a sequence of 256 vectors (128 from each of the directions) each with 64 elements.

Self-attention Layer. This layer learns a relative weighting function for all hidden state vectors produced by the bi-directional GRU, where more informative vectors are given more weight, and returns a weighted sum of those vectors. Its output is a single vector of size 64.

2) Convolutional Neural Network (CNN) Module: Aside from using an RNN, we also experimented with a CNN module as the implementation of the paragraph encoder. Figure 3 shows the structure of the network.

Input. The input to the network is a sequence of 128 tokens representing a single paragraph, which is identical to RNN.

Embedding Layer. The embedding layer maps word tokens to word vectors in $\mathbb{R}^d$ ($d = 300$), which is functionally identical to that in RNN module. The output is a 128×300 matrix.

Convolutional Layer. The purpose of this layer is to extract important local sequences of word vectors. For this layer, we had to decide: (i) the number of filters, and (ii) the size of each filter, i.e., the number of consecutive word vectors to be convolved. As it is possible to choose variable filter sizes, we chose to use eight filters each of sizes 1, 2, 3, and 4. Since we process 128 tokens in each paragraph, our filter with size r = 1 will then produce a vector with 128-1+1 = 128 elements. Thus, given our filter sizes of 1, 2, 3, and 4, the outputs from this layer are eight vectors with lengths 128, 127, 126, and 125.

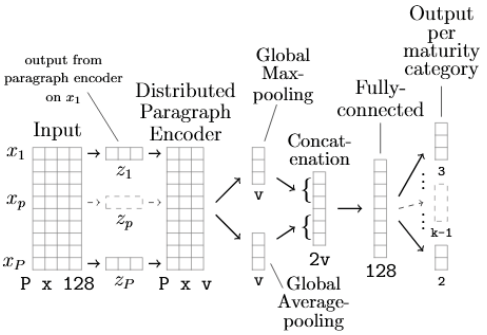

Fig. 4. Output dimensions at each Aggregation Module's step

Max-pooling Layer. The convolutional layer produces a total of 32 vectors, and some vectors vary in the number of dimensions. This layer pools the maximum scalar value from each of the 32 vectors, producing a vector with 32 elements.

3) Aggregation Module: When predicting the maturity level of a particular book, we did not want to exclude any paragraphs of that book because a high level of maturity in just one paragraph may strongly influence the maturity level of the entire book. Thus, we want our neural network to be able to draw information from every paragraph of a given book during prediction. Figure 4 shows the structure of the network.

Input. Identical to RNN and CNN modules, the input to the Aggregation module is also a sequence of 128 tokens representing a single paragraph. Multiplying to the 128 tokens is the P fixed-length feature vectors, where P is the number of paragraphs in a book.

Distributed Paragraph Encoder Layer. Either the RNN or CNN paragraph encoder outputs a fixed-length feature vector given a paragraph. This layer produces P fixed-length feature vectors. The length v of the feature vectors is 64 for the RNN paragraph encoder, and is 32 for the CNN paragraph encoder.

Global Max-pooling Layer. Global max-pooling, or max-over-time pooling, outputs the maximum value in each column of an input matrix, resulting in a vector. If we consider the input to this layer to be a P-rows $\times$ v-columns matrix, where v is the length of one vector produced by the paragraph encoder, then the output from this layer is a single vector of size v.

Global Average-pooling Layer. We used this layer since the maturity content rating for a book may be influenced not only by the most severe instance of mature content in that book, but the ones appeared throughout the book.

Concatenation Layer. The two vectors which came from the output of the *global max-pooling* and *global average-pooling* layers are concatenated to create one vector of size $2v$.

Fully-connected Layer. A fully-connected, or dense, layer simply extracts more hidden, or latent, features from the concatenation layer. The result is a vector with 128 elements.

Output Layer. This layer outputs the predicted maturity rating level as a binary ordinal vector for each maturity category.

V. Experimental Results

We present the results of our empirical study to verify the performance of our maturity rating levels prediction module. We show the results using entire book paragraphs based on the Aggregates module and the ones using individual book paragraphs based on the paragraph module. Last, but not least, we report the performance between our module and three other existing, well-known maturity-rating-level prediction tools.

For the performance analysis, we divided the Book Cave dataset of 6,395 books into a train/validation/test set using a 60/20/20 distribution, resulting in 3837, 1279, and 1279 books in their respective sets. We trained the baseline, predictor, and our neural network models using the same train dataset. The neural networks were evaluated on the validation set during each epoch of training to monitor their improvement over time. Our neural networks quit training when the loss measured on the validation set stopped improving. The evaluation metrics of all classifiers and predictors are on the test set.

A. Baseline Classifiers

To obtain a baseline measure for our work, we used different classifiers given identical word vectors representing texts of books. For each of these baseline classifiers, i.e., *K-Nearest Neighbors (KNN)* [6], *Logistic Regression (LR)* [12], *Multinomial Naïve Bayes (MNB)* [11], *Multi-class Support Vector Machine (SVM)* [5], *Random Forest (RF)* [2], *Multi-layer Perceptron (MLP)*, [17], and *Zero Rule (Zero-R)* [3], we chose to use a *balanced bagging ensemble* [13] of that classifier to account for imbalance in the distribution of categorical rating levels. In a balanced bagging ensemble, many classifiers which implement the same algorithm are created; each individual classifier receives different subsets of the entire dataset which are balanced in terms of label distribution, i.e., the classes are equally distributed in those subsets. These balanced subsets of the data are created by randomly sampling from the entire dataset while only sampling with replacement on the minority class. Hence, each individual classifier observes a dataset that is smaller than the entire dataset, but whose class distribution is balanced, thus avoiding any bias that may have been imposed by a majority class. During prediction, for each test instance C, each individual classifier casts a "vote" for the class that it believes should be assigned to C; the class that gains the majority vote becomes the predicted class for C.

B. Results Using Entire Book Paragraphs

One of the modules that we have designed is the *Aggregation* module which aggregates all vectors produced by a paragraph encoder for *all* paragraphs of a book. We conducted an empirical study using the Book Cave dataset to evaluate the performance of the Aggregation module and each one of the baseline modules using *all* paragraphs of a book. The error ratios in predicting the maturity rating level of a book based on the aggregation and each of the baseline models are depicted in Figure 5, and the detailed analysis on their performance based on each one of the categories is shown in Figure 6. According to the analysis, the Aggregation module outperforms other

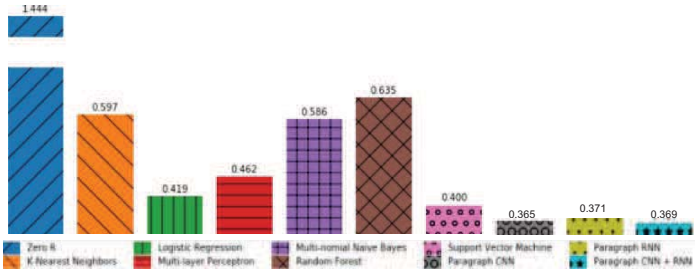

Fig. 5. Mean squared error of content rating levels for entire books for all classifiers averaged over all maturity categories

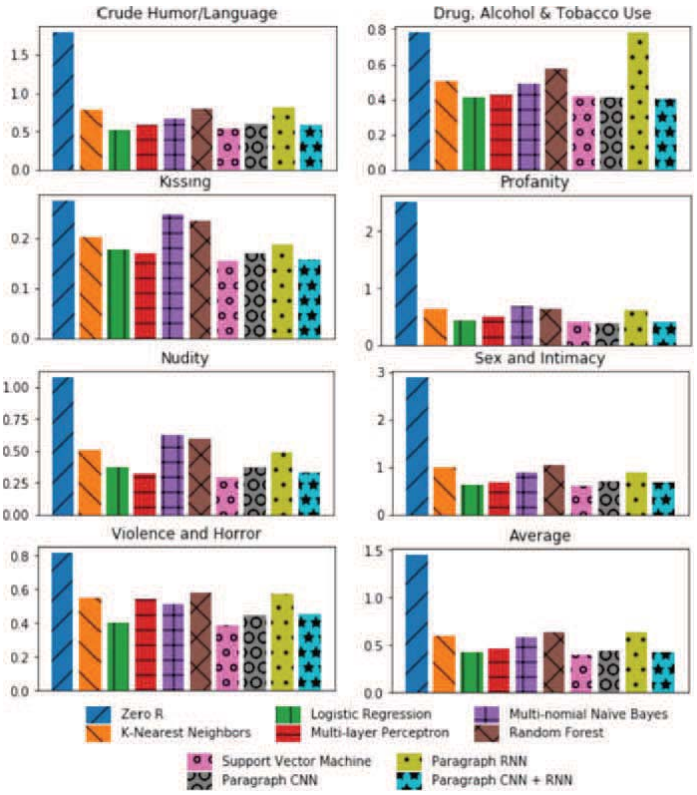

Fig. 6. Mean squared error scores of all classifiers for predicted content rating levels of entire books. All maturity categories are shown. Classifiers were trained using the same training set and evaluated on the same test set

baseline modules and the results are statistically significant based on the Wilcoxon Signed-Ranks test ($p \leq 0.04$).

C. Results Based on the Paragraph Module

We also wanted to evaluate whether or not our classifiers would be able to detect the portions of books that contained mature content. Since both the baseline classifiers and the neural network models were trained to classify the levels of mature content in texts of variable size, it is valid to give as input to the classifiers only a portion of the book instead of the entire book. We hypothesized that each of our classifiers would also be able to compute a meaningful maturity content level for individual paragraphs and groups of paragraphs of books.

As stated in Section IV-C, we have developed the *paragraph encoder* module, which requires a sequence of word tokens of a single paragraph as an input and produces a fixed-length

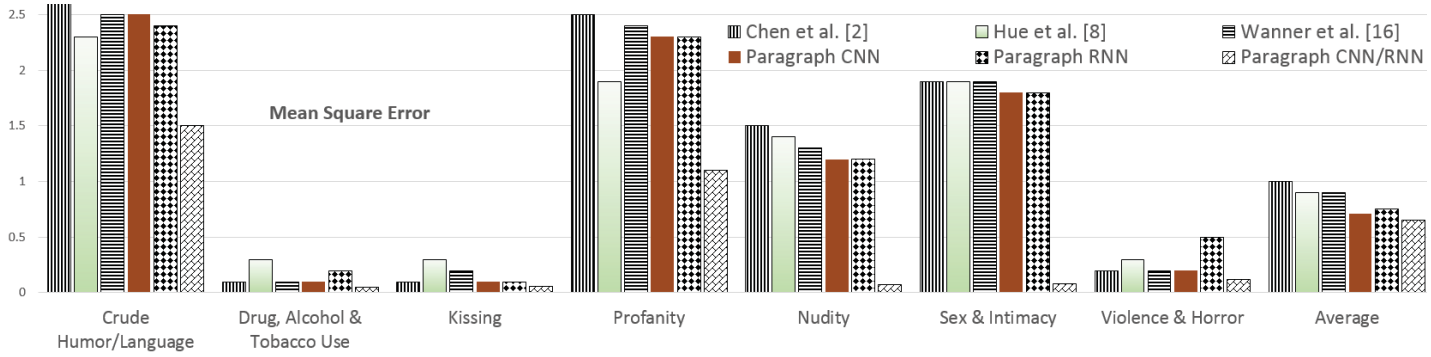

Fig. 7. Mean square error ratios of the three existing classifiers along with our neural network model based on individual book paragraphs by category

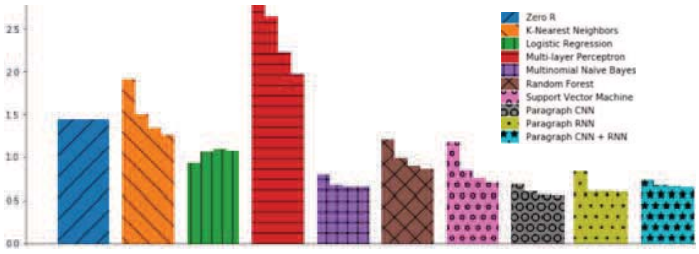

Fig. 8. Mean squared error of content rating levels after classifier predictions are aggregated over individual paragraphs averaged over all categories

vector. Figure 8 shows that the paragraph-CNN + RNN module outperforms all the baseline models averaged over all maturity categories based on the error ratios, and Figure 9 depicts the error ratios of the predicted maturity rating level of each of the individual categories, in addition to the average error ratios. These results have been verified to be statistically significant based on the Wilcoxon Signed-Ranks test ($p \leq 0.03$).

D. Comparing Our Maturity Rating Level Model with Others

Besides analyzing the accuracy on predicting the maturity rating levels of books achieved by our neural network model and the baseline classifiers, we have also implemented a number of classification approaches for maturity-rating-level prediction for comparison purpose. All of these classifiers, which have been presented in Section II, share a common design methodology, i.e., using a hand-picked list of *sensitive* words for quantifying the mature contents of their respective text documents. These classifiers include (i) Chan et al.'s maturity rating model [4] that works on user reviews for applications published through mobile app services, (ii) Hu et al.'s classifier [10] that analyzes the mature content in mobile applications using maturity rating policy descriptions, and (iii) Wanner et. al.'s categorization model [21] that determines the mature content of books to decide their suitability for children. Figure 7 shows the performance evaluation of the three existing classifiers along with our neural network model based on the same dataset used for comparing the baseline classifiers. Our neural network model significantly outperforms the other three based on the Wilcoxon Signed-Ranks test ($p \leq 0.05$).

VI. Conclusions

We are interested in books published these days, since they are widely distributed over the Internet and in print whose contents affect not only adults, but children as well. In assisting users to carefully choose appropriate books to read, we have developed a natural language processing (NLP) and machine learning model which takes as input the text of a book and predicts that book's content rating within different mature themes. In building a deterministic content rating system that can be used by consumers, especially via distribution systems for e-books, such as Amazon Kindle, we can assist users, especially children, adolescents, and teenagers. in choosing books with appropriate contents to read.

Conducted empirical studies have verified that our proposed NLP and convolutional/recurrent neural network model outperforms various baseline and maturity-level classifiers in terms of accurately predicting the mature level of e-books, which is a significant contribution to the digital media community.

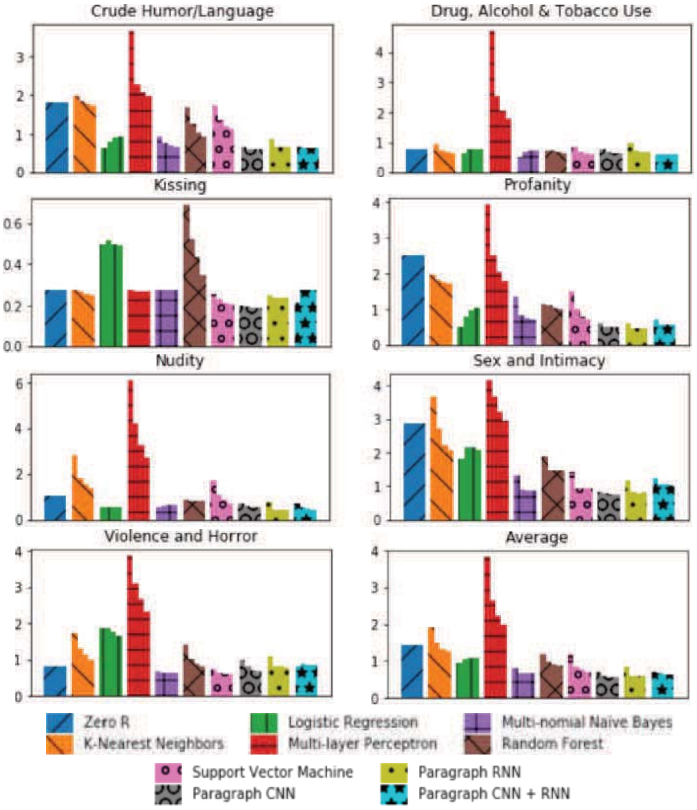

Fig. 9. Mean squared error of content rating levels after classifier predictions are aggregated over individual paragraphs for each individual category

REFERENCES

[1] A. Bleakley, M. Hennessy, M. Fishbein, and A. Jordan. It Works Both Ways: The Relationship Between Exposure to Sexual Content in the Media and Adolescent Sexual Behavior. *Media psychology*, 11(4):443–461, 2008.

[2] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] D. Chakraborty and R. Nikhil. A Neuro-Fuzzy Scheme for Simultaneous Feature Selection and Fuzzy Rule-Based Classification. *IEEE Transactions on Neural Networks*, 15(1):110–123, January 2004.

[4] Y. Chen, H. Xu, Y. Zhou, and S. Zhu. Is This App Safe for Children?: A Comparison Study of Maturity Ratings on Android and iOS Applications. In *WWW*, pages 201–212, 2013.

[5] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[6] T. Cover and P. Hart. Nearest Neighbor Pattern Classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[7] S. Coyne, L. Stockdale, D. Nelson, and A. Fraser. Profanity in Media Associated with Attitudes and Behavior Regarding Profanity Use and Aggression. *Pediatrics*, 128(5):867–872, 2011.

[8] E. Frank and M. Hall. A Simple Approach to Ordinal Classification. In *European Conference on Machine Learning*, pages 145–156, 2001.

[9] D. Gentile, S. Coyne, and D. Walsh. Media Violence, Physical Aggression, and Relational Aggression in School Age Children: A Short-term Longitudinal Study. *Aggressive behavior*, 37(2):193–206, 2011.

[10] B. Hu, B. Liu, N. Gong, D. Kong, and H. Jin. Protecting Your Children from Inappropriate Content in Mobile Apps: An Automatic Maturity Rating Framework. In *ACM CIKM*, pages 1111–1120, 2015.

[11] Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence*, pages 488–499. Springer, 2004.

[12] D. Kleinbaum and M. Klein. *Logistic Regression: A Self-Learning Text, 3rd Edition*. Springer, 2010.

[13] G. Liang and A. Cohn. An Effective Approach for Imbalanced Classification: Unevenly Balanced Bagging. In *AAAI-13*, pages 1633–1634, 2013.

[14] S. Merayo-Alba, E. Fidalgo, V. Gonzalez-Castro, R. Alaiz-Rodriguez, and J. Velasco-Mata. Use of Natural Language Processing to Identify Inappropriate Content in Text. In *HAIS*, pages 254–263, 2019.

[15] G. Miller. *WordNet: An Electronic Lexical Database*. MIT press, 1998.

[16] A. Razavi, D. Inkpen, S. Uritsky, and S. Matwin. Offensive Language Detection Using Multi-Level Classification. In *Canadian AI*, pages 16–27, 2010.

[17] F. Rosenblatt. The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain. *Psychological review*, 65(6):386, 1958.

[18] J. Sargent, J. Gibson, and T. Heatherton. Comparing the Effects of Entertainment Media and Tobacco Marketing on Youth Smoking. *Tobacco Control*, 18(1):47–53, 2009.

[19] M. Schuster and K. Paliwal. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[20] L. Titus-Ernstoff, M. Dalton, A. Adachi-Mejia, M. Longacre, and M. Beach. Longitudinal Study of Viewing Smoking in Movies and Initiation of Smoking by Children. *Pediatrics*, 121(1):15–21, 2008.

[21] F. Wanner, J. Fuchs, D. Oelke, and D. Keim. Are My Children Old Enough to Read These Books? Age Suitability Analysis. *POLIBITS*, 43:93–100, Jan 2011.

[22] H. Yenala, A. Jhanwar, M. Chinnakotla, and J. Goyal. Deep Learning for Detecting Inappropriate Content in Text. *Data Sci Anal*, 6:273–286, 2018.