A Multi-task Approach to Predict Likability of Books

Suraj Maharjan
Dept. of Computer Science
University of Houston
Houston, TX, 77004
smaharjan2@uh.edu

John Arevalo and **Fabio A. González**
Computing Systems and
Industrial Engineering Dept.
Universidad Nacional de Colombia
Bogotá, Colombia
{jearevaloo, fagonzalezo}@unal.edu.co

Manuel Montes-y-Gómez
Instituto Nacional de Astrofisica
Optica y Electronica
Puebla, Mexico
mmontesg@ccc.inoep.mx

Thamar Solorio
Dept. of Computer Science
University of Houston
Houston, TX, 77004
solorio@cs.uh.edu

Abstract

We investigate the value of feature engineering and neural network models for predicting successful writing. Similar to previous work, we treat this as a binary classification task and explore new strategies to automatically learn representations from book contents. We evaluate our feature set on two different corpora created from Project Gutenberg books. The first presents a novel approach for generating the gold standard labels for the task and the other is based on prior research. Using a combination of hand-crafted and recurrent neural network learned representations in a dual learning setting, we obtain the best performance of 73.50% weighted F1-score.

1 Introduction

Every year millions of new books are published, but only a few of them turn into commercial successes, and even fewer achieve critical praise in the form of prestigious awards or meaningful sales. Editors have the difficult task of making the go/no-go decision for all manuscripts they receive, and the revenue for their publishing house depends on the accuracy of that judgment. The website www.litrejections.com documents some of the biggest mistakes in the history of the publishing industry, including Agatha Christie, J.K. Rowling, and Dr. Seuss, all of whom received many rejection letters before landing their first publishing deal.

Many factors contribute to the eventual success of a given book. Internal factors such as plot, story line, and character development all have a role in the likability of a book. External factors such as author reputation and marketing strategy are arguably equally relevant. Some factors might even be out of the control of an author or publishing house, such as the current trends, the competition from books released simultaneously, and the historical and contextual factors inherent to society.

Previous work by Ganjigunte Ashok et al. (2013) demonstrated relevant results using stylistic features to predict the success of books. Their definition of success was a function of the number of downloads from Project Gutenberg. However downloading a book is not by itself an indicator of a highly liked or a commercially successful book. We instead propose to use the rating from reviewers collected from Goodreads as a measure of success. We also propose features and deep learning techniques that have not been used before on this problem, and validate their usefulness in two different tasks: success prediction and genre classification. Our key contributions are the following:

- We provide a new benchmark dataset for predicting successful books in a more realistic class distribution. This data set is available to the community from this link[1].
- We show that sentiment analysis using SenticNet sentics is an accurate way to model emotion in books.

[1]The data can be downloaded from http://ritual.uh.edu/resources/ page.

- We provide the first results on using recurrent neural networks (RNN) to discover book content representations that are useful for classification tasks such as success prediction and genre detection.

- We show that the multitask approach, simultaneously evaluating success and genre prediction, benefits from its constituent tasks to obtain better performance than the single success prediction task approach.

2 Previous work

Predicting the success of books is a difficult task, even for an experienced editor. Researchers have studied related tasks, for example predicting the quality of text from lexical features, syntactic features and different measures of density. Pitler and Nenkova (2008) found a strong correlation between user-perceived text quality and the likelihood measures of the vocabulary as computed by a language model, as well as the likelihood measures of discourse relations, as determined by a language model trained on discourse relations. Louis and Nenkova (2013) proposed a combination of genre-specific and readability features with topic-interest metrics for the prediction of great writing in science articles. While some of the features in this prior work were relevant to our task, our goal is different and more aligned to Ganjigunte Ashok et al. (2013), since we aim to model success in books of different genres.

Ganjigunte Ashok et al. (2013) investigated the correlation between writing style and number of downloads. The authors analyzed lexical features, production rules, constituents, and sentiment features of books downloaded from Project Gutenberg[2]. They obtained an average accuracy of 70.38% using only unigram features with Support Vector Machines (SVM) as the classifier.

Deep learning representations have seen their share of successes in Natural Language Processing (NLP) tasks (Bahdanau et al., 2014; Zheng et al., 2013; Gao et al., 2014; Glorot et al., 2011; Samih et al., 2016). In particular, RNN models have been successfully applied in several scenarios where temporal dependencies provide relevant information (Ian Goodfellow and Courville, 2016; LeCun et al., 2015). Kiros et al. (2015) used RNN models to learn language models from books using an unsupervised approach. Also, word embedding (Mikolov et al., 2013) and Paragraph Vector (Le and Mikolov, 2014) have been shown to achieve state-of-the-art performance in several text classification and sentiment classification tasks. These techniques are able to learn distributed vector representations that capture semantic and syntactic relationships between words. Collobert and Weston (2008) trained jointly a single Convolutional Neural Network (CNN) architecture on different NLP tasks and showed that multitask learning increases the generalization of the shared tasks. Other researchers (Ian Goodfellow and Courville, 2016; Søgaard and Goldberg, 2016; Attia et al., 2016) have also reached to similar conclusions.

3 Dataset

We experimented with two book collections: one prepared by Ganjigunte Ashok et al. (2013)[3] and the other constructed by us to evaluate a new definition of success. We refer to the first dataset as EMNLP13 and the second dataset as Goodreads.

The EMNLP13 collection contained Project Gutenberg books from eight different genres. The authors created a balanced dataset containing 100 books per genre, resulting in a total of 800 books. We manually reviewed the dataset and found missing or irrelevant content in 58 books: a total of 53 books contained Project Gutenberg license information repeated verbatim, and five books contained only the audio recording certificate in place of the actual book content. We removed the license-related text, since lexical features might be erroneously biased, and replaced the five files with the actual content of the books. Except for these corrections, the data we used is the same as that presented in Ganjigunte Ashok et al. (2013).

We also identified some odd adjudications. For example, 'The Prince And The Pauper' is a popular book by Mark Twain that was adapted into various films and stage plays. Also, 'The Adventures of Captain Horn' was the third best selling book of 1895 (Hackett, 1967). Both these books are labeled as unsuccessful due to their low download counts. We suspect as well that some of the counts are inflated by college students doing English or Literature assignments that may not be directly related to the potential commercial success

[2] https://www.gutenberg.org/

[3] The data can be downloaded from http://www3.cs.stonybrook.edu/~songfeng/success/

Genre	Unsuccessful	Successful	Total
Detective Mystery	60	46	106
Drama	29	70	99
Fiction	30	81	111
Historical Fiction	16	65	81
Love Stories	20	60	80
Poetry	23	158	181
Science Fiction	48	39	87
Short Stories	123	135	258
Total	349	654	1,003

Table 1: Goodreads Data Distribution

		EMNLP13 Success definition	
		Unsuccessful	Successful
Goodreads Success definition	Unsuccessful	73	32
	Successful	110	184

Table 2: Confusion matrix between two different definitions of success.

of a book.

To address these concerns, we propose a new approach to creating gold labels for successful books based on public reviews rather than download counts. We collected a new set of Project Gutenberg books for this benchmarking. We mapped the books to their review pages on Goodreads[4], a website where book lovers can search, review, and rate books. We consider only those books that have been rated by at least 10 people. We use the average star rating and total number of reviews for labeling each book. We then set an average rating of 3.5 as the threshold for success, such that books with average rating < 3.5 are classified as *Unsuccessful*. Table 1 shows the data distribution of our books. To our knowledge, we have one of the largest collection of books, as researchers generally work with a low number of books (Coll Ardanuy and Sporleder, 2014; Goyal et al., 2010; van Cranenburgh and Koolen, 2015).

Success Definitions Comparison: After compiling and labeling both the datasets, we drew a comparison between the two definitions of success. To do this, we downloaded the Project Gutenberg download counts for the books in Goodreads dataset and labeled them using the Ganjigunte Ashok et al. (2013) definition of success. Since they only considered books in the extremes of download counts, we could only label 399 books in the Goodreads dataset using their definition. We found that 142 books had different labels according to the two definitions. 19.7% of these mismatched books were labeled as unsuccessful despite having ratings ≥ 3.5 and being reviewed by more than 100 reviewers. Table 2 details the discrepancies between the two definitions.

[4] https://www.goodreads.com/

4 Methodology

We investigated a wide range of textual features in an attempt to capture the topic, sentiment, writing style, and readability for each book. This set included both new and previously used features. We also explored techniques for automatically learning representations from text using neural networks, which have been shown to be successful in various text classification tasks (Kiros et al., 2015; LeCun et al., 2015). These techniques include word embeddings, document embeddings, and recurrent neural networks.

4.1 Hand-crafted text features

Lexical: We used skip-grams, char n-grams, and typed char n-grams (Sapkota et al., 2015) with term frequency-inverse document frequency (TF-IDF) as the weighting scheme. Sapkota et al. (2015) showed that classical character n-grams lose some information in merging instances of n-grams like *the* which could be a prefix (***the**sis*), a suffix (*brea**the***), or a standalone word (***the***). They separated character n-grams into ten categories representing grammatical classes, like affixes, and stylistic classes, like beg-punct and mid-punct which reflect the position of punctuation marks in the n-gram. The purpose of these features is to correlate success with an author's word choice.

Constituents: We computed the normalized counts of *'SBAR', 'SQ', 'SBARQ', 'SINV',* and *'S'* syntactic tag sets from the parse tree of each sentence in each book, following the method of Ganjigunte Ashok et al. (2013) to determine the syntactic style of the authors.

Sentiment: We computed sentence neutrality, positive and negative, using SentiWordNet (Baccianella et al., 2010) along with the counts of nouns, verbs, adverbs, and adjectives. We averaged these scores for every 50 consecutive sentences in order to evaluate change in sentiment throughout the course of each book, because we anticipate emotions, like suspense, anger, and happiness to contribute to the success of the book.

SenticNet Concepts: We extracted sentiment concepts from the books using the Sentic Concept

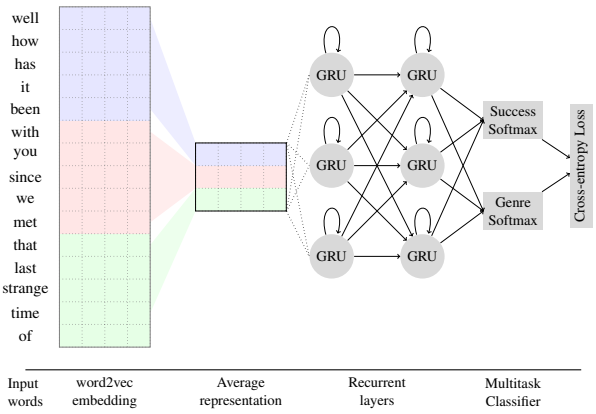

Figure 1: Multitask method. Words are represented in the Word2Vec space. Such representations are averaged per window. Sequences are feed to GRU network. Finally, the features are feed to two softmax components to predict genre and success simultaneously.

Parser[5]. The parser chunks a sentence into noun and verb clauses, and extracts concepts from them using Part Of Speech (POS) bigram rules. We modeled these as binary bag-of-concepts (BoC) features. We also extracted average polarity, sensitivity, attention, pleasantness, and aptitude scores for the concepts defined in the SenticNet-3.0 knowledgebase, which contains semantics and sentics associated with 30,000 common-sense concepts (Cambria and Hussain, 2015).

Writing density: We computed the number of words, characters, uppercase words, exclamations, question marks, as well as the average word length, sentence length, words per sentence, and lexical diversity of each book, with the expectation that successful and unsuccessful writings will have dissimilar distributions of these density metrics.

Readability: We computed multiple readability measures including Gunning Fog Index (Gunning, 1952), Flesch Reading Ease (Flesch, 1948), Flesch Kincaid Grade Level (Kincaid et al., 1975), RIX, LIX (Anderson, 1983), ARI (Senter and Smith, 1967), and Smog Index (Mc Laughlin, 1969) and used their mean normalized values for training. Intuitively, the use of simple language will resonate with a larger audience and contribute to book success.

4.2 Neural network learned representations

Representation learning techniques are able to learn a set of features automatically from the raw data. Our hypothesis is that the learned representation can capture the complex factors that influence the success of a book.

Word embeddings with Book2Vec: In contrast with Word2Vec, which learns a representation for individual words, Doc2Vec learns a representation for text fragments or even for full documents. We trained the Doc2Vec module of the Gensim (Řehůřek and Sojka, 2010) Python library, on all the books in the Goodreads dataset to obtain a 500 dimensional dense vector representation for each book. Using Doc2Vec, we first trained a distributional memory (DM) model with two approaches: concatenation of context vectors (DMC) and sum of context word vectors (DMM). Then we trained a distributional bag of words (DBoW) model and combined it with the DMC and the DMM for a total of five different models. We set the number of iterations to 50 epochs and shuffled the training data in each pass. We called these book vectors *Book2Vec*. Furthermore, we created two 300 dimensional vector representations for each book by averaging the vectors of each word in the book using pre-trained Word2Vec vectors from the Google News dataset[6] and our own Word2Vec trained with ∼350M words from 5,000 random books crawled from Project Gutenberg.

Multitask RNN method: When dealing with variable length data such as time series or plain text, traditional approaches like feed-forward neural networks are not easily adapted since they expect fixed-size input to model sequential data. One limitation of RNNs is that it has problems dealing with long sequences (Pascanu et al., 2013). We propose a strategy to represent large documents, such as books, with an aggregated representation. Figure 1 depicts the proposed multitask method. The overall strategy uses a RNN to learn a model of sequences of sentences. Each sentence is represented by the average of the Word2Vec representation of its constituent words. The RNN is composed of 2 hidden layers with 32 hidden gated recurrent units (GRU) (Cho et al., 2014) each, and the output is a softmax layer. We train the RNN

[5] https://github.com/pbhuss/Sentimental/blob/master/parser/SenticParser.py

[6] The pre-trained Word2Vec was downloaded from https://code.google.com/p/word2vec/

in a supervised fashion using the success categorization and the book genre as labels. The RNN serves a feature extractor and the last hidden states for each sequence acts as its representation. At training time, all sentences from one book are extracted and divided in chunks of 128 sentences. The book's success/genre labels are assigned to each sequence. A sentence is then represented as the average of its constituent word vectors. To make the book label assignment at testing time, we average the predictions of all sequences extracted from each book. Using 128 sentences has threefold a motivation: (1) mitigate vanishing gradient problem (Pascanu et al., 2013), (2) obtain more examples from one book, and (c) be a power of 2 to efficiently use the GPU.

An interesting property of neural networks is that the same learning approach, i.e stochastic gradient descent, still holds for more complex architectures as long as the objective cost function is differentiable. We take advantage of this property to build a unified neural network that addresses both genre and success prediction using a single model. These kinds of multitask architectures are also useful as regularizers (Ian Goodfellow and Courville, 2016). In particular, our cost function $J(X, Y)$ is defined as follows:

$$h_i = rnn(x_i)$$

$$\hat{y}_i^{succ} = \frac{e^{z_i^{succ}}}{\sum_k e^{z_k^{succ}}}$$

$$\hat{y}_i^{gen} = \frac{e^{z_i^{gen}}}{\sum_l e^{z_l^{gen}}}$$

$$J(X, Y) = -\sum_i (y_i^{succ} \ln \hat{y}_i^{succ} + y_i^{gen} \ln \hat{y}_i^{gen})$$

where x_i represents the i-th sample and y^{succ} and y^{gen} are success and genre labels respectively. The $rnn(\cdot)$ function represents the forward propagation over the recurrent neural network and h represents the last hidden state. $\hat{y}^{succ}$ and $\hat{y}^{gen}$ represent predictions for the two labels. Notice that both of them are computed using the same unified representation h. z^{succ} and z^{gen} represent two different linear transformations over h that map to the number of classes.

5 Experiments and Results

5.1 Experiments on Goodreads dataset

We merged books from different genres, and then randomly divided the data into a 70:30 training/test ratio, while maintaining the distribution of *Successful* and *Unsuccessful* classes per genre. As a preprocessing step we converted all words to lowercase and removed infrequent tokens having document frequency ≤ 2. For our tagging and parsing needs, we used the Stanford parser (Socher et al., 2013). We then trained a LibLinear Support Vector Machine (SVM)[7] classifier with L2 regularization using the hand-crafted features described in Section 4. We tuned the C parameter in the training set with 3-fold grid search cross-validation over different values of 1e{-4,...,4}.

With the features used by Ganjigunte Ashok et al. (2013), we obtained the highest weighted F1-score of 0.659 with word bigram features. We set this value as our baseline. In order to study the effect of the multitask approach, we devised analogous experiments to our proposed multitask RNN method and predicted both genre and success together for the features described in Section 4. Hence we have two settings for the classification experiments, Single task (ST) and Multitask (MT).

Since we had average rating information, we also modeled the problem as a regression problem and predicted the average rating using only the content of the books. Our work differs from other researchers in this aspect, as most of them (Lei et al., 2016; Li et al., 2011; Mudambi et al., 2014) use review content instead of the actual book content to predict the average rating. We used the Elastic Net regression algorithm with $l1\_ratio$ tuned over range {0.01, 0.05, 0.25, 0.5, 0.75, 0.95, 0.99} with 3-fold grid search cross-validation of the training data.

Parameter tuning for RNN: We trained 25 models with random hyper-parameter initialization for learning rate, weights initialization ranges and regularization parameters. We chose the best validation performance model. This is preferable over grid search when training deep models (Bergstra and Bengio, 2012). We used the ADAM algorithm (Kingma and Ba, 2014) to update the gradients. Since these models are prone to overfitting because of the high number of parameters, we applied clip gradient, max-norm weights, early stopping and dropout regularization strategies.

[7] We use LibLinear SVM wrapper from `http://scikit-learn.org/stable/`

Features	ST (F1)	MT (F1)	MSE
Word Bigram	0.659	0.685	0.152
2 Skip 2 gram	0.645	0.688	0.156
2 Skip 3 gram	0.506	0.680	0.156
Char 3 gram	0.669	0.700	0.155
Char 4 gram	0.676	0.689	0.155
Char 5 gram	0.683	0.699	0.154
Typed beg_punct 3 gram	0.621	0.672	0.151
Typed mid_punct 3 gram	0.598	0.641	0.151
Typed end_punct 3 gram	0.626	0.677	0.151
Typed mid_word 3 gram	0.653	0.687	0.156
Typed whole_word 3 gram	0.658	0.666	0.154
Typed multi_word 3 gram	0.607	0.657	0.154
Typed prefix 3 gram	0.624	0.624	0.154
Typed space_prefix 3 gram	0.589	0.646	0.155
Typed suffix 3 gram	0.624	0.637	0.154
Typed space_suffix 3 gram	0.626	0.664	0.154
Clausal	0.506	0.558	0.156
Writing Density (WR)	0.605	0.640	0.156
Readability (R)	0.506	0.634	0.144
SentiWordNet Sentiments(SWN)	0.582	0.610	0.156
Sentic Concepts and Scores (SCS)	0.657	0.670	0.155
GoogleNews Word2Vec	0.669	0.692	0.156
Gutenberg Word2Vec	0.672	0.673	0.140
Book2Vec (DBoW)	0.643	0.654	0.130
Book2Vec (DMM)	0.686	**0.731**	0.142
Book2Vec (DMC)	0.640	0.674	0.131
Book2Vec (DBoW+DMC)	0.647	0.677	0.131
Book2Vec (DBoW+DMM)	**0.695**	0.729	0.142
RNN	0.529	0.686	**0.125**

Table 3: Results for classification (ST = Single task setting, MT = Multi-task setting) and regression tasks on Goodreads dataset. MSE = Mean Square Error, F1 score is weighted F1 scores across *Successful* and *Unsuccessful* classes.

Features	ST (F1)	MT (F1)	MSE
Unigram+Bigram	0.660	0.691	0.15
Unigram+Bigram+Trigram	0.660	0.700	0.149
Char 3,4,5 gram	0.682	0.689	0.153
All Typed ngram	0.663	0.691	0.144
SCS+WR+Typed mid word	**0.720**	0.710	0.155
SCS+Book2Vec	0.695	0.731	0.139
R+Book2Vec	0.695	0.729	0.139
WR+Book2Vec	0.693	0.726	0.139
Word Ngram+ RNN	0.691	0.688	0.125
Skip gram + RNN	0.689	0.683	0.125
Typed char ngram+ RNN	0.689	0.702	0.125
Char 3 gram + RNN	0.689	0.688	0.125
Clausal+ RNN	0.689	0.688	0.125
SCS + RNN	0.691	0.688	0.125
WR+Book2Vec+ RNN	0.701	**0.735**	0.129
SCS+WR+RNN	0.675	0.696	**0.123**
All hand-crafted	0.670	0.689	0.148
All hand-crafted+neural	0.667	0.712	0.129

Table 4: Feature Combination Results for Goodreads dataset. (ST = Single Task, MT =Multi-task, SCS = Sentic concept+average scores of sensitivity, attention, pleasantness, aptitude, polarity, WR = Writing Density, R = Readability)

5.2 Results on Goodreads dataset

Table 3 shows the results with our new proposed feature sets for the classification and regression tasks. In the ST setting, except for the character n-gram features, all proposed hand-crafted features individually had a weighted F1-score less than the word bigram baseline. On the other hand, the neural network methods obtained better results than the baseline. We obtained the highest weighted F1-score of 0.695 and 0.731 with the *Book2Vec* method in the ST and MT settings, respectively. The results show that the MT approach is better than the ST approach. The genre prediction task must have acted as a regularizer for the success prediction task. Also, we found that modeling the entire book as a vector, rather than modeling it as the average of word vectors, gave better performance. Although the ST *Book2Vec* performs better than the MT RNN method, the difference is very small. We performed McNemar's test on these methods and found that the results were not statistically significant, with *p=0.5*. The MT RNN method had the lowest mean square error (MSE) for the regression task, at 0.125.

The character $n$gram proved to be one of the most important hand-crafted features, whereas clausal feature was the least important one. Individually, writing density and readability features seemed to be weak features. We assumed that the sentiment changes in books would be an important characteristic for the task. However, the results in Table 3 show an unimpressive F1-score of 0.610 for sentiment features. On the other hand, the bag of sentic concepts model with average scores for sensitivity, attention, pleasantness, aptitude, and polarity gave a more impressive F1-score of 0.670, much higher than the baseline. This result points to the relevance of performing a more nuanced sentiment analysis beyond lexical statistics for this task.

Our next set of experiments included the combinations of hand-crafted and neural network representations. Some of the best combination results are shown in Table 4. Out of the different possible feature combinations, we obtained the highest weighted F1 score of 0.735 by combining hand-crafted and learned representations in the MT setting. We observed that combining low performing hand-crafted features like readability, syntactic clauses, and skip grams with neural representation boosted their performance. Likewise for the regression task, the MT RNN representation proved to be a better choice, as its combination with other features generally lowered the MSE. The best combinations for the regression task lowered the MSE to 0.123. Deep learning and hand-crafted methods may capture complementary sources of information, which upon combination boost performance.

5.3 Results on EMNLP13 dataset

We tried to reproduce the results reported in Ganjigunte Ashok et al. (2013) by re-implementing their system. Unlike our setup, they performed experiments on individual genres and reported average accuracy across all genres. We obtained similar results, but not as close as we expected, even after extensive experimentation, and extending the search for parameter optimization. For most of their features we obtained a lower accuracy [8]. The differences may be due to a combination of the curating process we described in Section 3 that corrected content in the books used, as well as the different set of parameter values we explored for tuning the classifier. As pointed out by Fokkens et al. (2013), even seemingly small differences in preprocessing can prevent reproducibility. *Hence, we consider our best accuracy so far (71.25%) to be the state-of-the-art performance on this data set.*

Table 5 shows the results from some of our best feature sets. The features that worked best for the Goodreads data also worked best for the EMNLP13 data. Significantly, with the combination of the sentic concepts and scores, typed $n$grams, and writing density, we obtained an average accuracy of 73.00%, much higher than the baseline score of 71.25% for this dataset.

The RNN performance was very low in comparison with the handcrafted features. We relate this behavior to the small size of this particular training dataset and evaluation setup. Notice that Ganjigunte Ashok et al. (2013) experimented per genre, i.e. trained a single classifier per genre. Thus, in a 5-fold approach we only have 80 samples to train and 20 to test. Additionally, we must take out some samples from the training data for validation. It has been empirically shown that one of the key elements in the success of representation learning strategies is a large amount of data, on the order of tens of thousands of samples at least. Moreover, in the EMNLP13 dataset, it is not possible to take advantage of the multitask approach because there is only one target genre in each experiment.

6 Discriminative features

Table 6 lists some of the features that were highly-weighted by the classifier. For the sentic concepts, salient features included important adjectives, verbs and relations; all objects that might trigger a crucial event. Similarly, for the character n-gram features, honorific titles, stop words, common word endings, and especially n-grams with quotation marks were highly weighted. Quotation marks indicate the exchange of dialogues between characters. This suggests that dialogue is an important aspect of novels. Word n-gram features also support this suggestion. Features like *s/he said, said Person_Name* were also highly weighted. Moreover, pronouns and titles related to male gender also had high weights. Features like *␣ i was, . i, ␣ i am* also had high weights. This might be an indication that books with first person narration tend to be more successful. Another interesting observation was that the number of question marks in a book was also consistently positively correlated with success. This might suggest that readers enjoy books consisting of dialogue or interaction between the characters. We also calculated the maximal information criterion (MIC) and correlation coefficient (CC) for the writing density as well as the readability features against the average rating. Generally, readers prefer books with high writing density (0.19 MIC, 0.25 CC) and somewhat complex writing (0.17 MIC, 0.21 CC).

[8] There was a maximum 4% difference for some features.

Features	**Avg Accuracy(%)**
Word Bigram	71.25
Char 3 grams	71.00
Typed mid_word 3-gram	70.25
Writing Density (WR)	68.38
Readability	61.38
Sentic concepts & scores(SCS)	72.38
GoogleNews Word2Vec	69.88
Gutenberg Word2Vec	64.25
Book2Vec	72.38
RNN	55.80
Unigram+Bigram+Trigram	72.75
Book2Vec+SCS	64.75
Book2Vec+WR	66.38
SCS+WR+Typed char ngrams	**73.00**

Table 5: Average accuracy results with new feature and their combinations on EMNLP13 dataset.

Type	Features
ngrams	. “, . ”, said :, young man, very young man, the young man, boys, . i, father, his father, mother, he said, she said, said NE, princess, lord, colonel, captain, doctor, tour, mr, miss
Sentic concepts	conceive, grieve, zealous, emptiness, bitterness, corpse, hypothesis, irony, theory_of_the, wagon,deep_blue, scarred, screaming, grudging, vigil, vein, beautiful_place, rural, marriage, friendship, cats, 911 avg aptitude, polarity, pleasantness, attention scores
Character and typed character ngrams	mr., mrs., john, thou, amor, pen, his, and,the, ing, n’s,ed, gg’, pt’, d’a, t”, i-t, ␣ ,”i ,” ”, ” say,” s,” she

Table 6: Discriminative Features

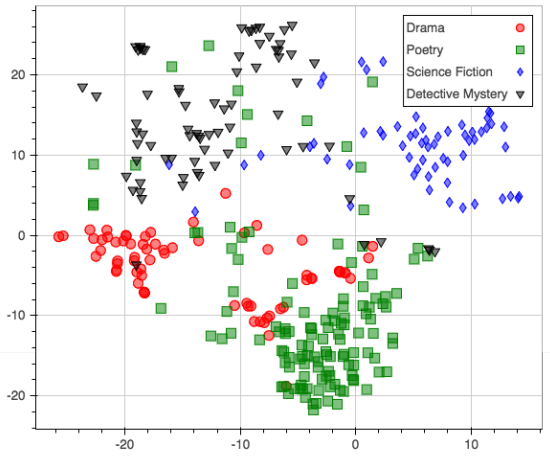

Figure 2: Projection of Book2Vec from four different genres into 2D space for the Goodreads dataset.

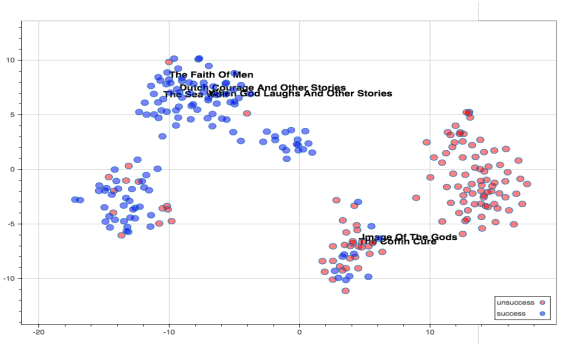

Figure 3: Projection of successful and unsuccessful books using representation learned with the RNN model.

7 Analysis of learned representations

In order to investigate deep vectors, we projected them onto 2-dimensional space using t-SNE. Figure 2 suggests that the vectors successfully capture genre-related concepts, as books from the same genre are close to each other in the 2D space. We then performed 8-way genre classification experiment using random stratified division of the data into 70:30 training/test ratio. We obtained an accuracy of 62.50% and F1 score of 69.30% for the EMNLP13 and Goodreads datasets, respectively. These scores were well above the random baseline of 12.50% accuracy and 15.23% F1-score for the EMNLP13 and Goodreads datasets, respectively. We further found that Poetry and Drama were the most accurately classified genres, whereas Fiction was the most difficult to classify.

In order to further investigate the representations learned by RNN for successful and unsuccessful books, we plotted the 2D t-SNE projection of the book representations. Figure 3 shows the projection of vectors for the Short stories genre. The visualization shows that the RNN is able to cluster the book vectors into two separate regions. Furthermore, to investigate what else the RNN might be learning, we plotted some books by the same authors. Figure 3 also shows books from authors Jack London and Alan E. Nourse. The four books by Jack London and the two books by Alan E. Nourse are very close to each other. We thus infer that along with learning peculiarities of successful and unsuccessful classes, the RNN was able to capture features related to the style of authors.

8 How much content is needed for success prediction?

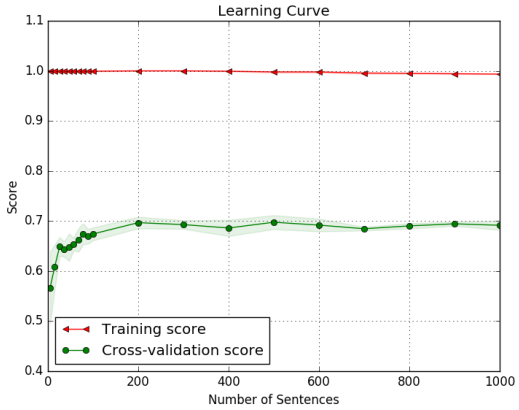

Figure 4: Weighted F1 score for training and validation data for varying number of sentences with char 3 gram feature.

Humans are good at detecting poor writing after reading just a few pages. We wanted to investigate if it is the same for machines. We devised stratified 3-fold cross-validation exploratory experiments on training data by gradually increasing the content of the books in the training fold. The results are shown in Figure 4. We see that the cross-validation score gradually increases until we reach 200 sentences. After this point, it plateaued out. Hence, we conclude that 200 sentences is the minimum threshold for the classifier.

9 Conclusions

In this paper we propose new features for predicting the success of books. We used two main feature categories: hand-crafted and RNN-learned features. Hand-crafted features included typed character n-grams and sentic concepts. For the

learned features we proposed two different strategies based on neural networks. The first extends Word2Vec-type representations to work in large documents such as books, and the second one uses an RNN to capture sequential patterns in large texts. We evaluated our methods on our Goodreads dataset, whose classes are not based on download counts, but rather are a function of average star ratings and number of reviewers. Our results outperform state-of-the-art methods. We conclude that instead of having either deep-learning or hand-crafted features outperform the other, both methods capture complementary information, which upon combination gives better performance. Also, the multitask setting is preferable to the single task setting, as the multitask approach helps the classifier better generalize during learning by letting constituent tasks act as regularizers. As our next steps, we plan to investigate features that capture plot-related aspects, such as character profiles and interaction through social network analysis, historical setting, and other feature-learning strategies.

Acknowledgments

We would like to thank the National Science Foundation for partially funding this work under awards 1462141 and 1549549. We also thank Simon Tice and the three anonymous reviewers for reviewing the paper and providing helpful comments and suggestions.

References

Jonathan Anderson. 1983. Lix and rix: Variations on a little-known readability index. *Journal of Reading*, 26(6):490–496.

Mohammed Attia, Suraj Maharjan, Younes Samih, Laura Kallmeyer, and Thamar Solorio. 2016. Cogalex-v shared task: Ghhh - detecting semantic relations via word embeddings. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pages 86–91, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Erik Cambria and Amir Hussain. 2015. *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*, volume 1. Springer.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October. Association for Computational Linguistics.

Mariona Coll Ardanuy and Caroline Sporleder. 2014. Structure-based clustering of novels. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 31–39, Gothenburg, Sweden, April. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–223.

Antske Fokkens, Marieke van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. 2013. Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1691–1701, Sofia, Bulgaria, August. Association for Computational Linguistics.

Vikas Ganjigunte Ashok, Song Feng, and Yejin Choi. 2013. Success with style: Using writing style to predict the success of novels. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1764, Seattle, Washington, USA, October. Association for Computational Linguistics.

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2014. Learning continuous phrase representations for translation modeling. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 699–709, Baltimore, Maryland, June. Association for Computational Linguistics.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520.

Amit Goyal, Ellen Riloff, and Hal Daume III. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 77–86, Cambridge, MA, October. Association for Computational Linguistics.

Robert Gunning. 1952. The technique of clear writing. McGraw-Hill.

Alice Payne Hackett. 1967. *Seventy years of best sellers, 1895-1965*. RR Bowker Co.

Yoshua Bengio Ian Goodfellow and Aaron Courville. 2016. Deep learning. Book in preparation for MIT Press.

J. Peter Kincaid, Robert P. Fishburne Jr, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Xiaojiang Lei, Xueming Qian, and Guoshuai Zhao. 2016. Rating prediction based on social sentiment from textual reviews. *IEEE Transactions on Multimedia*, 18(9):1910–1921.

Fangtao Li, Nathan Liu, Hongwei Jin, Kai Zhao, Qiang Yang, and Xiaoyan Zhu. 2011. Incorporating reviewer and product information for review rating prediction. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 1820–1825. AAAI Press.

Annie Louis and Ani Nenkova. 2013. What makes writing great? first experiments on article quality prediction in the science journalism domain. *Transactions of the Association for Computational Linguistics*, 1:341–352.

G. Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *International Conference on Learning Representations (ICLR), Workshop*.

Susan M. Mudambi, David Schuff, and Zhewei Zhang. 2014. Why aren't the stars aligned? an analysis of online review content and star ratings. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 3139–3147. IEEE.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1310–1318.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii, October. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. `http://is.muni.cz/publication/884893/en`.

Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual code-switching identification via lstm recurrent neural networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59, Austin, Texas, November. Association for Computational Linguistics.

Upendra Sapkota, Steven Bethard, Manuel Montes, and Thamar Solorio. 2015. Not all character n-grams are created equal: A study in authorship attribution. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–102, Denver, Colorado, May–June. Association for Computational Linguistics.

R.J. Senter and E.A. Smith. 1967. Automated readability index. Technical report, DTIC Document.

Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria, August. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235,

Berlin, Germany, August. Association for Computational Linguistics.

Andreas van Cranenburgh and Corina Koolen. 2015. Identifying literary texts with bigrams. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 58–67, Denver, Colorado, USA, June. Association for Computational Linguistics.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for Chinese word segmentation and POS tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657, Seattle, Washington, USA, October. Association for Computational Linguistics.