Which book is the right one for me? Predicting tones of a book using its reviews

Jiaming Qu. Ximing Wen, Wan Zhang, Jiesong He

University of North Carolina at Chapel Hill
Chapel Hill, NC

1 Introduction

NoveList is a provider of research databases, e-books and discovery service to libraries of all kinds, which is established by UNC alumni. It defines the different aspects of a book that may catch readers' attention "story elements". The story elements include appeal terms, themes, and genres. Appeal terms, which is referred to as "tone" in our project, are a signature feature of NoveList [1]. NoveList imagine them as the secret language of books, all the ways a book speaks to a reader and lingers long after it's been returned to the library. And appeal terms help readers find exactly what kind of book they are in the mood for.

Currently librarians NoveList are tagging each book with appropriate tones manually, which is quite labor-intensive and time-consuming. Therefore, given a book, an algorithm which automatically predicts its tones would be much helpful. In this project, we are using reviews and genres of a book as its metadata, as we assume that these two fields have richer information than other fields like title and author. Since usually each book has more than one tones, our task becomes a multi-label classification problem.

2 Dataset Overview

The data for this project is provided by Novelist. This dataset consists of 28737 books yet not all the books have review and tone fields. For prediction, We use a subset with 7005 books having reviews recorded and tones labeled. Our analysis of the dataset reveals that the 59 tones are not evenly distributed. As is shown in Fig 1, top 3 most frequent tones used are "suspenseful" (12.1%), "atmospheric" (7.6%), and "moving" (6.8%). While the 3 tones being mentioned least are "sad" (0.1%), "liberal" (0.1%), and "patriotic" which is only for 3 books. On average, each book has 1.5 tones and each tone is assigned to 180 books.

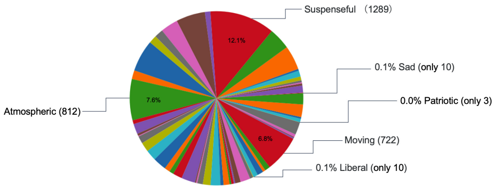


Fig. 1: Tone Distribution

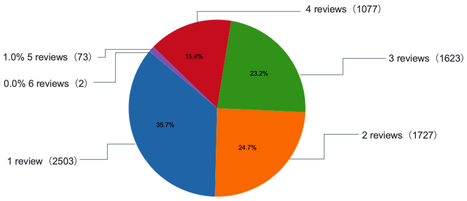


Fig. 2: Review Distribution

[1] https://www.ebscohost.com/novelist/idea-center/learn/learn-story-elements

Besides, each book has 1.5 reviews on average. Fig 2 shows that 86.3% books have 1 to 3 reviews. Almost 15.4% books are given 4 reviews. But the percentage drops dramatically when referring to books with 5 and 6 reviews. There are only 73 and 2 books receiving 5 and 6 reviews respectively.

3 Methodology

In this section, we show in detail how we build the model to predict tones of a book using its reviews. We first provide the preliminary knowledge of multi-label classification which is the foundation of our proposed model. Then we give a heuristic approach as our baseline, followed by two machine learning approaches, namely the *book-level* and *review-level* respectively.

3.1 Multi-label Classification

Multi-label classification is different from multi-class classification. Multi-class classification implies that the amount of unique labels is at least 2, and when the amount is exactly 2, it is binary classification. However, multi-label classification emphasizes that each instance has at least one label and some instances may have more than one. In this project, we are dealing with both multi-class (59 uniques tones) and multi-label (each book has 1.5 tones on average) classification.

There are a lot of methods which transform the multi-label classification problem to a simpler form. Binary Relevance (BR) proposes it could be L separate binary problems with one for each label, in which L is the amount of labels [1]. Then L one-vs-all classifiers could be trained. But BR does not model label dependency, as some labels could be correlated instead of independent. Label Powerset (LP) proposes it could be a single multi-class problem with 2^L possible class values, and any multi-class classifier could be trained [2]. But LP introduces too many class labels and much complexity. Pairwise Binary (PW) proposes it could be $L * \frac{(L-1)}{2}$ binary classifiers, which is known as all-vs-all [3]. Then each model is trained based on examples annotated by at least one of the labels, but not both. But the number of classifiers could be very high. Apart from these basic transformations, there are some advanced methods like classifier chains [4], ensemble [5] and etc. In this project, we use the BR method and we train 59 classifiers for 59 tones.

3.2 Heuristic Approach

Before we dive into machine learning techniques to tackle this problem, we start from a heuristic approach first. Since the two most frequent tones in the corpus are “suspenseful” and “atmospheric”, we do a majority guessing which assumes that all the books only have these two tones. This approach gives us an interpretable lower bound when the learning process is purely based on counting.

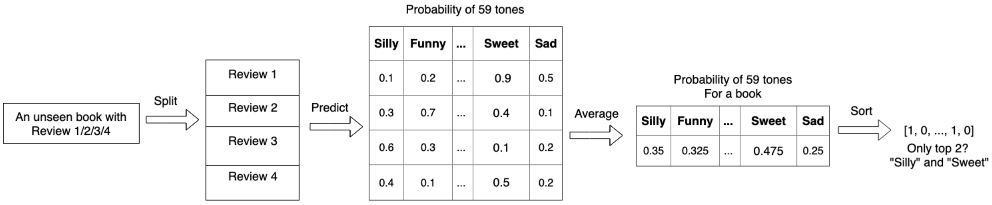
3.3 Machine Learning Approach

It is a common practice to use machine learning techniques to deal with text classification problems. As is discussed above, the Binary Relevance transformation is used, therefore we need to train 59 binary classifiers for all the tones, and the Logistic Regression model is used for binary classification. These independent classifiers predict how much likely an instance has such a tone.

In terms of feature selection, we first only use text features which are unigrams after text pre-processing including converting to lower case, removing punctuations, removing stop words and stemming. We then add genres as a high-level feature into the model.

Book-level Approach As is shown in the dataset overview section, there are a lot of books with more than 1 reviews, and on average each book has 1.5 reviews. To deal with this issue, we first simply concatenate all the reviews for each book, then we predict its tones based on the merged review. Despite the simplicity and convenience, some books may have an extremely long review after concatenating, thus the features could be very sparse in the bag-of-words model.

Review-level Approach An alternative approach is to predict tones of a book based on all its reviews, and we train binary classifiers using all the reviews as the corpus, which means we predict tones from reviews. As is shown in Fig 3, instead of directly predicting labels, for each book we first split its reviews and then use each piece of review to predict the probabilities of tones. Then we take the average of these probability vectors as the probabilities of tones assigned to the book. Since each book has 1.5 tones on average, we then sort the probability from high to low and select two tones with the highest probability as the predicted label.



Silly	Funny	...	Sweet	Sad
0.1	0.2	...	0.9	0.5
0.3	0.7	...	0.4	0.1
0.6	0.3	...	0.1	0.2
0.4	0.1	...	0.5	0.2

Silly	Funny	...	Sweet	Sad
0.35	0.325	...	0.475	0.25

Fig. 3: Work flow of our proposed model

4 Result and Comparison

We first use *Hamming Loss* which is calculated as:

$$\frac{1}{NL} * \sum_{i=1}^{N} \sum_{i=1}^{L} I[\hat{y}_j^{(i)} \neq y_j^{(i)}] \tag{1}$$

and *Zero-one Loss* which is calculated as:

$$\frac{1}{N} * \sum_{i=1}^{N} I[\hat{y}^{(i)} \neq y^{(i)}] \tag{2}$$

to compare the heuristic approach and machine learning approach, in which L is the number of labels and N is the number of instances, and $I[c]$ returns 1 if condition c holds. *Hamming Loss* could be understood as how many mistakes the prediction for an instance makes for each label, while *Zero-one Loss* consider all the labels, i.e., predictions for all the labels should be exactly the same with true labels.

Table 1: Comparison of Hamming Loss and Zero-one Loss

	Heuristic Approach	**Book-level**	**Review-level**
Hamming Loss	0.0495	**0.026**	0.040
Zero-one Loss	0.983	0.995	**0.966**

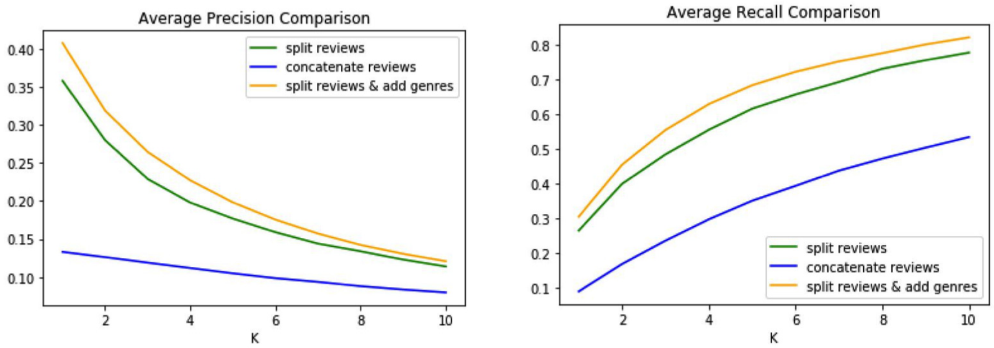


Fig. 4: Comparison of Average Precision and Recall @K

Table 1 summarizes performances of the three approaches mentioned above. It is surprising that concatenating all the reviews for a book outperforms splitting reviews. However, as we pry into the result, we found that most labels predicted by the book-level approach are all 0, and since there are a few 1's in the true labels, its *Hamming Loss* is much more lower. These two metrics are too vague and hard to interpret. Therefore, we change these two evaluation metrics to another two metrics which are more interpretable. In real work, librarians who use the model may care about not only what the predicted labels are, but also what are the most possible ones. Thus, instead of predicting labels, we predict probabilities of 59 tones, sort from high to low, and return a ranked list of 59 tones, which is transformed into a ranking problem. We use Precision and Recall @K to evaluate how these approaches perform.

As is shown in Fig 4, the review-level approach which predicts on reviews and then take the average has a significant improvement over the book-level approach which predicts on all the reviews of a book. Also, adding genres as high-level features further enhances this approach.

5 Conclusions

In this project, we use reviews and genres of a book to predict its tones. The feature of a book is represented in TF-IDF vectors from text and count vectors from genres, and 59 Logistic Regression models are trained. The results prove that splitting reviews, predicting probabilities and taking average outperforms concatenating reviews, and adding genres as a high-level feature further improves the performance.

6 Acknowledgement

We appreciate Novelist for sharing the data and Dr. Yue Wang from the School of Information and Library Science for supervising the whole project.

References

1. Oscar Luaces, Jorge Díez, José Barranquero, Juan José del Coz, and Antonio Bahamonde. Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence*, 1(4):303–313, 2012.
2. Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089, 2010.
3. Mikhail Petrovskiy. Paired comparisons method for solving multi-label learning problem. In *2006 Sixth International Conference on Hybrid Intelligent Systems (HIS'06)*, pages 42–42. IEEE, 2006.
4. Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer, 2009.
5. Jesse Read, Bernhard Pfahringer, and Geoffrey Holmes. Multi-label classification using ensembles of pruned sets. In *8th IEEE international conference on data mining*, pages 995–1000. IEEE, 2008.