Judge a Book by its Cover

Farah Shehab
fas41@mail.aub.edu

Haya Charafeddine
hnc03@mail.aub.edu

Lara Hammoud
lah30@mail.aub.edu

Tamara Samaha
tas18@mail.aub.edu

Abstract

The project we're working on aims to identify the genre(s) of a certain book simply by conducting an analysis on its front cover. This could be of great help in big libraries, where instead of a having a person look up the genre of a book manually, the librarian would just scan the cover of the book and let the program do the job on his/her behalf. The program would make the process go faster, smoother and less prone to mistakes - such as typing in the wrong genre for a book - mainly caused by human inattention.

1 Introduction

Classifying books based solely on their covers is surely not an easy task, and there haven't been many experiments conducted on the matter. For this task we would need to analyze the components of the book covers: text and visual. So, the broad idea was to use Natural Language Processing for the former, and Convolutional Neural Networks for the latter.

We found a few papers and articles written about the genre classification of books based on their cover; notably, Chiang et al. [2] experimented with several approaches including the use of Convolutional Neural Networks, Natural Language Processing and Color features, each individually, and then in combinations. Separately, both NLP and CNN scored approximately 0.6 on accuracy. And according to their findings, the best approach which increased accuracy to 0.62, consisted of combining the CNN (also incorporates color) and NLP features using a multi-class soft margin SVM, since some categories are not very separable.

It was also mentioned that they extracted features from a fully connected layer of a neural network that was pre-trained on ImageNet, since it covers a variety of images from a large database. And most papers we encountered in our research mentioned something similar to the technique above in terms of combining visual and textual features, using CNNs and RNNs, since visual features in book covers can sometimes be ambiguous, causing incorrect predictions (Iwana et al.) [3]. The best model used by Iwana et al. was AlexNet which scored a 24.7% top-1 accuracy and a 40.3% top-3 accuracy.

We found some other approaches to a similar problem, that could apply to ours, which is the classification of movies based on their posters After searching through multiple techniques such as using Naïve-Bayes, K-Nearest Neighbors, etc.

We decided that we will be using this combination of CNN and NLP. As for using pre-trained CNNs such as ResNet and VGG16, versus a custom CNN, we decided have to experiment with both options.

2 Dataset

For this phase of the project, we looked for datasets that were large enough and that consisted of a collection of books, their cover page and their genre(s). The one we agreed belong to user uchidalab, and can be found on github.com [1], it's the one used in the 'Judging a Book by its cover' paper by Iwana et al.mentioned above; it consists of two .csv files: one for training and one for testing; and a folder all the images of book covers, each belonging to one of 30 genres.

The .csv files contain information such as: book Id, path, title, URL, author(s), category and category Id.

The train dataset contains around 51 thousand books, with over a thousand book in each category. And the test dataset contains a total of 5,700 books. The images are all already resized to 224x224x3.

3 Model

As stated earlier in this paper, we decided to use both textual and visual features in our approach.

(I) For the textual features (i.e. the title), we experimented with two models: a bi-directional LSTM - since we are dealing with sequential data, and a Bag of Words model - which is a much simpler model than the previous one, but nonetheless effective.

a. For the former, we used stanford glove embeddings to create an embedding matrix; and the best LSTM model we got had the following architecture:
An input layer, followed by an embedding layer, a bi-directional LSTM layer with 100 units and dropout rate of 0.3 to avoid over-fitting, then a global max-pooling layer, a dense layer with 50 units and 'relu' activation, followed by a 0.3 dropout, and finally a softmax output layer with 30 units since we are doing multi-class classification.

b. As for the Bag of Words, we used Term Frequency-Inverse Document Frequency approach to pre-process the data, and a rather simple architecture consisting of three consecutive Dense layers, each followed by a dropout layer of 0.5, activation set to 'relu' and with the number of units respectively: 512, 128, 64; followed by the final softmax output layer.

(II) As for visual features, we tried a few other pre-trained networks that were pre-trained on imagenet, such as Resnet and InceptionV3, but ended up using VGGnet-16 model as it

gave us better results. And considering the size of our dataset is rather decent, we froze all the layers of the model except for the fully connected layers.

(III) In order to combine both visual and textual features, we created a multi-modal neural network, that takes in two inputs: title and image, and gives out one output.
The first branch consisted of the VGG-16 model one mentioned above - without the output layer, and the second is the LSTM model without its output layer. Both branches were concatenated, followed by a dense layer, then the final softmax layer.

4 Results

For the experimentation phase, the train file was split into 80% train and 20% validation, and the test file was kept as is.

Before starting with the training, we pre-processed all our data in two steps:

1. Pre-processing titles:

   For the text data, we removed all stopwords and used a lemmatizer n, then created an embedding matrix for the LSTM model and applied a tf-idf vectorizer on the data for the BoW model, each separately.

2. Pre-processing images:

   Pre-process the images by normalizing them and applying some augmentation.

In order to get to the models mentioned above (LSTM, BoW and CNN), we chose a set of hyper-parameters for each and performed a random search on each one of them, and we also experimented with two optimizers: Adam and RMSprop. After the three random searches were done, we chose the best for each and ended up with 3 models: one LSTM, one BoW and a VGG-16.

After comparing the LSTM and the BoW, we decided to go with a combination of the LSTM and the VGG-16 for the multi-modal approach.
The optimzer used for all the models is RMSprop with a learning rate of 0.001, and the loss function is categorical crossentropy since we are dealing with multiple classes. We also implemented early stopping so we could get the best epoch for each model with the least validation loss.

The final experiment we conducted was trying ensemble learning on the three models we got earlier. We used max voting, averaging and weighted averaging techniques to see if we could improve our prediction accuracy. But it was unsuccessful as the accuracy ended up being lowered significantly.

Table 1: Results of the experiments conducted on the test set

	LSTM	**BoW**	**VGG**	**Multi-modal**
Top-1 Accuracy	0.55	0.52	0.23	0.55
Top-3 Accuracy	0.75	0.69	0.4	0.74

As we can observe from the table above, models with text inputs have a much higher accuracy then models with image inputs, and that is due a number of factors such as the fact that book covers can sometimes have ambiguous features, and that CNNs might depend too much on color since there aren't many other objects present in the cover. So, therefore it has a higher chance of wrongly classifying a book, than an LSTM does.

Another problem with book classification, is the similarity between some of the classes such as 'Children's Books' and 'Comics & Graphic novels', 'Science & Math' and 'Medical Books' etc.

References

[1] https://github.com/uchidalab/book-dataset/tree/master/Task1

[2] Chiang, Holly, et al. "Classification of Book Genres By Cover and Title." Retrieved from https://www.semanticscholar.org/paper/Classification-of-Book-Genres-By-Cover-and-Title-Chiang-Ge/d0d0096d307a6da1332153b9cb8a72c29df38f87

[3] Uchida, Seiichi, et al. "Judging a Book by its Cover" 13 October 2017 Retrieved from https://arxiv.org/pdf/1610.09204v3.pdf

[4] https://nlp.stanford.edu/projects/glove/