Classification of Movies and Books by Genre Using BERT

Moran Daori & Rotem Deutsch

1 Introduction

There has been a significant increase in popularity of streaming services in recent years. As part of their product, many of the companies that offer the above mentioned services also offer movie and TV recommendations to users. Similarly, there has been a significant increase in the online purchases of books. Many of the companies selling those recommend different books to their clients based on their previous purchases. Those recommendations rely, in many cases, on deep learning algorithms that identify the end-user's book, movie and TV preferences. Taking into account the magnitude of the movie, TV and book libraries that those companies offer, the need for automatic and smart classification algorithms is undeniable. In this paper we focused on classification by genre of movies and books and chose to examine the relation between a movie's or a book's plot summary and it's genre or genres. In our case we examine the compatibility of a plot summary and a genre's definition using BERT. Our results showed that BERT was able to identify the compatibility between the pair 76.29% of the time for movies and 66.07% for books.

2 BERT

Bidirectional Encoder Representations from Transformer, or in its shortcut BERT, is Google's neural network-based technique for natural language processing (NLP) pre-training. BERT's key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modeling. In the paper published by google [5], results show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models.

BERT is basically a trained Transformer Encoder stack, with 12 layers in the Base version, and 24 in the Large version, compared to 6 encoder layers in the original Transformer. BERT encoders have larger feedforward networks: 768 nodes in the Base version and 1024 in the Large one, 12 attention heads in the Base version and 16 in the Large one. BERT was trained on Wikipedia and Book Corpus, a dataset containing +10,000 books of different genres. BERT works similarly to the Transformer encoder stack, by taking a sequence of words as input which keep flowing up the stack from one encoder to the next, while new sequences are coming in. The final output for each sequence is a vector of 728 numbers in Base or 1024 in Large version.

In order to pre-train BERT the researches did not use traditional left-to-right or right-to-left language models, instead they pre-trained BERT using two unsupervised tasks: masked LM and next sentence prediction.

2.1 Masked LM

In order to train a deep bidirectional representation, 15% of the words in the input are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, Based on the context provided by the other, non-masked, words in the sequence. It is done by applying softmax to the final hidden vectors corresponding to the mask token, as in standard LM.

2.2 Next Sentence Prediction

During training process, the model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document or not. To do so, 50% of the inputs are a pair in which the second sentence is the subsequent sentence in the original document, while in the other 50% a random sentence from the corpus is chosen as the second sentence while assuming that the random sentence is disconnected

from the first sentence.

Practically, to help the model distinguish between the two sentences in training, A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence. Then, a sentence embedding indicating Sentence A or Sentence B is added to each token. Finally, a positional embedding is added to each token to indicate its position in the sequence. In order to predict if the second sentence is indeed connected to the first, the entire input sequence goes through the Transformer model. The output of the [CLS] token is transformed into a 2x1 shaped vector, calculating the probability of IsNextSequence with softmax.

3 Experiment Setup

Our approach was to try to fine tune BERT to determine a movie's or a book's genre by its plot summary and the genre's definition. In order to do so, we used binary classification: our inputs were pairs of a plot summary and a genre's definition, the model's task was to predict if the movie was in fact of that specific genre or not, by predicting 1 if they were a match and 0 otherwise.

3.1 Experiment Setup: Movies

For the movies part of the experiment, the model was given both positive and negative examples and the genres attached to the plots were out of a pool of 23 genres that were selected by us. The genres we chose to use are: Action, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film-Noir, History, Horror, Music, Musical, Mystery, Romance, Sci-Fi, Sport, Short, Thriller, War and Western.

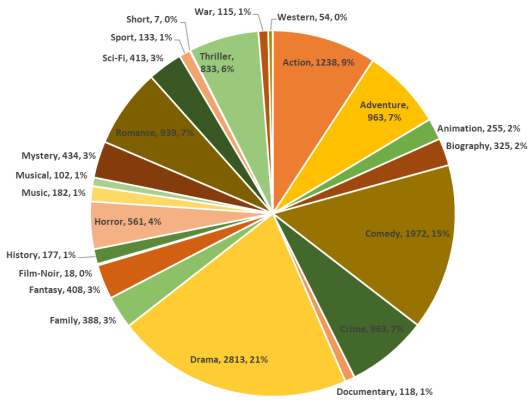

Figure 1: Distribution of Movies by Genre

The movie's data set was split into train and test data containing 80% (4316 movies) and 20% (1053 movies) of the data respectively. In order to further test the model, the data was divided into 5 folds. Each of them containing data excluding 4-6 genres. The goal was to see how training the model on slightly different examples (by having missing genres in the data), would affect the model's ability to classify correctly over the same test set, and examine how well the model would predict the missing genres.

The data folds:

- All movie genres
- All movie genres excluding Action, Documentary , Sci-Fi and Music.
- All movie genres excluding Adventure, Biography, Short, Sport, History and Film-Noir.
- All movie genres excluding Animation, Crime, Romance and Horror.
- All movie genres excluding Comedy, Family, Fantasy, Mystery and Western.
- All movie genres excluding Drama, War, Thriller and Musical.

More so, after being trained on the movies data the model was tested on the books test set.

3.2 Experiment Setup: Books

For the Books part of the experiment, the model was given both positive and negative examples and the genres attached to the plots were out of a pool of 19 genres that were selected by us. The genres we chose to use are: Adventure, Biography/ Memoir, Funny, Crime, Fantasy, Historical, Horror, Mystery, Romance, Sci-Fi, Short Stories, Thriller, War, Young Adult, Children's, Fiction, Graphic Novels, Holocaust and True Story.

The books data set was split into train and test sets, containing 80% (3641 books) and 20% (891 books) of the data respectively. Additionally, after being trained on the books data the model was tested on the movies test set.

3.3 Datasets

For the movies' data we used the following dataset, containing 5368 movies with up to 3 genres per each movie: https://data.world/studentoflife/imdb-top-250-lists-and-5000-or-so-data-records/

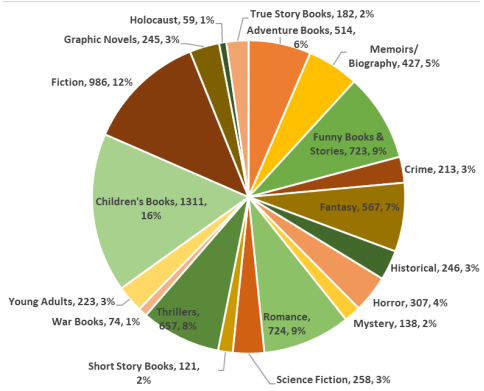

Figure 2: Distribution of Books by Genre

workspace/file?filename=IMDBdata_MainData2.csv

For the books' data we used the following dataset, containing 4532 movies with up to 5 genres per each book: https://www.kaggle.com/sp1thas/book-depository-dataset

For the genres definitions we created two files: one for movies containing 23 movie genres and their definitions, and one for books containing 19 book genres and their definitions. The definitions were mostly taken from their respective Wikipedia values.

The mentioned above were merely the raw data, For our model we had to generate positive and negative examples. For each movie/book in the data set, we created positive examples matching their given genres (each example was built from the movie's/book's plot summary and one of its true genre's definition). We created negative examples by going over each genre in the data set and matching it with movies/books that don't belong to this genre. Since the amount of negative examples far out-way the amount of positive ones we limited the amount of negative examples to be up to 15% larger than the positive ones per genre.

3.4 Model's Parameters

We ran the BERT-Adam optimizer and the following parameters:

Parameters	movies	books
Batch size	16	16
Number of Epochs	8	8
Learning Rate	1e-5	2e-5

Table 1: Model's Parameters

4 Results

4.1 Movies and Books Cross

Table 2 presents the accuracy in which the model predicted after being trained and tested on movies or books.

Trained/ Tested	Movies	Books
Movies	0.76	0.48
Books	0.48	0.66

Table 2: Percent Accuracy of Books and Movies

4.2 Movies

Table 3 presents the accuracy in which the model predicted after being trained and tested on movies alone, by genre.

Genre	Percent Accuracy
Action	0.74
Adventure	0.76
Animation	0.71
Biography	0.79
Comedy	0.74
Crime	0.80
Documentary	0.76
Drama	0.72
Family	0.76
Fantasy	0.76
Film-Noir	0.83
History	0.85
Horror	0.85
Music	0.78
Musical	0.83
Mystery	0.83
Romance	0.80
Sci-Fi	0.82
Short	0.75
Sport	0.85
Thriller	0.75
War	0.85
Western	0.93

Table 3: Percent Accuracy by Genre - Movies

4.2.1 Folds

Table 4 presents the accuracy in which the model predicted after being trained on the different folds (each having 4-6 genres missing) and tested on the same test set.

Genre/ Fold	Action, Documentary, Sci-Fi and Music	Drama, War, Thriller and Musical	Crime, Animation, Romance and Horror	Short, Adventure, Biography, Sport, History and Film-Noir	Comedy, Family, Fantasy, Mystery and Western
Action	0.66	0.72	0.72	0.71	0.75
Adventure	0.71	0.74	0.74	0.65	0.76
Animation	0.76	0.64	0.65	0.70	0.76
Biography	0.81	0.55	0.79	0.69	0.83
Comedy	0.74	0.69	0.74	0.74	0.60
Crime	0.79	0.69	0.72	0.79	0.81
Documentary	0.69	0.55	0.85	0.83	0.71
Drama	0.71	0.54	0.72	0.73	0.70
Family	0.80	0.60	0.77	0.78	0.73
Fantasy	0.75	0.52	0.79	0.79	0.78
Film-Noir	0.83	0.67	0.83	0.67	0.83
History	0.79	0.60	0.84	0.72	0.85
Horror	0.84	0.69	0.68	0.88	0.86
Music	0.82	0.62	0.90	0.93	0.79
Musical	0.85	0.53	0.82	0.82	0.75
Mystery	0.81	0.53	0.77	0.81	0.68
Romance	0.78	0.69	0.73	0.80	0.78
Sci-fi	0.79	0.63	0.85	0.84	0.85
Short	0.50	0.75	0.75	0.75	0.75
Sport	0.92	0.81	0.94	0.94	0.92
Thriller	0.71	0.56	0.68	0.73	0.67
War	0.92	0.64	0.85	0.92	0.89
Western	0.94	0.53	0.94	0.94	0.53
Total	0.75	0.64	0.74	0.75	0.73

Table 4: Percent Accuracy by Genre (each fold is represented by the missing genres)

4.3 Books

Table 5 presents the percent accuracy of the model, trained and tested on books data, by genre.

Genre	Percent Accuracy
Adventure	0.76
Children's	0.71
Crime	0.44
Fantasy	0.64
Fiction	0.74
Funny	0.69
Graphic Novels	0.79
Historical	0.52
Holocaust	0.83
Horror	0.77
Memoir/ Biography	0.46
Mystery	0.78
Romance	0.71
Sci-Fi	0.77
Short Stories	0.36
Thriller	0.51
True Story	0.39
War	0.42
Young Adult	0.79

Table 5: Percent Accuracy by Genre - Books

**All result tables can be seen in a graph form in the appendix section.

5 Analysis

5.1 Books and Movies Overall and Cross

It can clearly be seen in Table 2, that the prediction on the movies' end is better than the books'. That can be due to a number of reasons. Firstly, the amount of data used to train and test the books' model was about 20% smaller than the movies' one. Secondly, the books' plot summaries were taken from what is written on the back cover of the book. The back cover text is meant to appeal to the reader and to be enticing, it is not necessarily informative of a book's plot, which is unlike a movie's plot summary which is usually quite informative.

Furthermore, it is clear that both models (the one trained on books and the one trained on movies) failed to classify correctly samples of the other kind. This can be again due to the fact that the books' summaries were not as informative as the movies' and therefore the model was trained to notice different things though a lot of the genres overlap.

5.2 Movies

As mentioned above, overall the model trained by movies was able to classify the compatibility of a pair of a movie's plot summary and a genre's definition correctly 76.29% of the time (As seen in Table 2). We argue that this result is a degree to how informative the plot summaries of movies are.

5.2.1 Folds

As seen in table 4 and table 3 for most genres the percent accuracy was best when the model was trained on a fold containing them but still having some missing genres. A possible explanation is that the specific genre was of greater percentage in the data set overall and thus the model was trained better on that genre for that fold. For most genres the percent accuracy was worst when either they were missing from the train data set or the train data set was missing the genres: Drama, War, Thriller and Musical. That might be due to the fact that the Drama genre was over 20% of the overall data, which meant that the model was trained on a smaller data set.

References

[1] Jay Alammar. The illustrated bert, elmo, and co.(how nlp cracked transfer learning). *Dec*, 3:1–18, 2018.

[2] Terra Blevins and Luke Zettlemoyer. Moving down the long tail of word sense disambiguation with gloss-informed biencoders. *arXiv preprint arXiv:2005.02590*, 2020.

[3] Jason Brownlee. What is the difference between a batch and an epoch in a neural network? *Machine Learning Mastery*, 2018.

[4] Jacob Devlin and Ming-Wei Chang. Open sourcing bert: State-of-the-art pre-training for natural language processing. *Google AI Blog. Weblog.[Online] Available from: https://ai. googleblog. com/2018/11/open-sourcing-bertstate-of-art-pre. html [Accessed 4 December 2019]*, 2018.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Rani Horev. Bert explained: State of the art language model for nlp. *URL: https://towardsdatascience. com/bert-explained-state-of-the-artlanguage-model-for-nlp-f8b21a9b6270*, 2018.

[7] David Mack. How to pick the best learning rate for your machine learning project, 2018.

[8] Pandu Nayak. Understanding searches better than ever before. *Google Blog, October*, 25, 2019.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[10] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

A Appendices

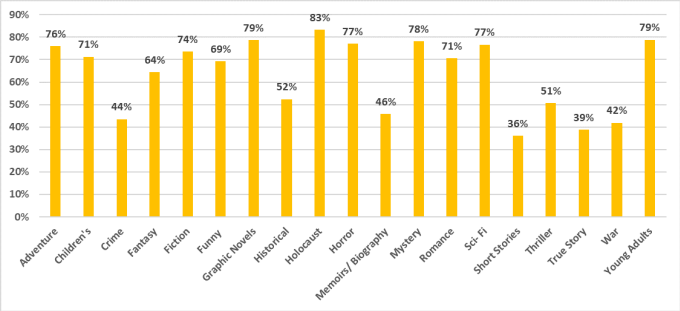

Figure 3: Books' Results by Genre- Graph

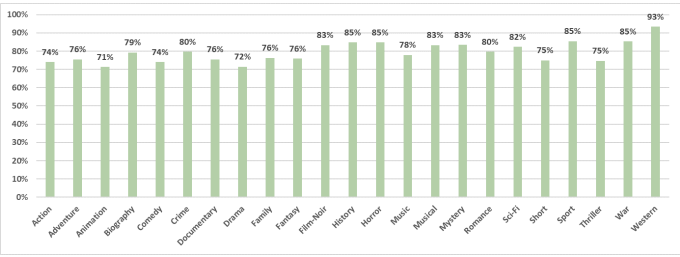

Figure 4: Movies' Results by Genre- Graph

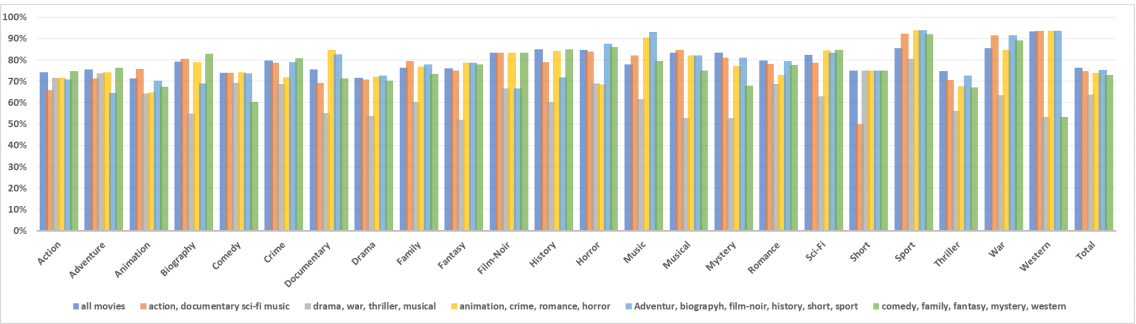

Figure 5: Results by Genre- Folds