Multiple Approaches to Multi-Label Genre Classification

Will Meyer
Summer 2024

Abstract

Multiple methods are explored in order to multi-label-classify books' genres based upon books' summaries. Building upon existing literature, two bag-of-words models (logistic regression and random forest), and one LLM model (decoder-only architecture) are employed. After simple adjustments, the logistic regression multi-label classifier had the greatest success. Although traditional metrics like accuracy, recall, precision, and F1-score were considered; the main metric of comparison was cosine similarity.

Introduction

Multi-label classification is an increasingly popular topic of machine learning research. This type of problem extends binary-classification and multi-classification problems by applying multiple labels to each record. A business application might be to generate relevant tags based upon media to improve search. It has also been applied to scientific settings where multiple conditions can coexist.[1] Beyond such practical applications, multi-label classification is interesting from a machine learning perspective because it is a less structured problem, with a large vocabulary of labels and no fixed number of labels to apply.

In this project, I explore different multi-label classification approaches to the problem of assigning genres to books. Specifically, a list of genres is predicted based upon a book's plot summary. The data is from a book summary database created by researchers at Carnegie Mellon.[2] The dataset contains plot summaries of 16,559 books, along with a title, some book metadata (IDs, author, publication date), and a list of genres. The goal was to predict a list of genres that described the book based upon its plot summary. In this research, the goal was to predict *similar* genre labels, *rather than the exact set of labels*. Unlike medical or scientific domains which may have clearly correct or incorrect labels, genre 'tagging' is highly subjective - especially when each book can have multiple genres. This subjective, similarity-based goal informed the key model comparison metric.

Existing Literature

Multi-label classification has become more popular because it aligns with many scientific and natural language situations. For instance, it has been used in microbiome disease research, since multiple diseases may be present at once.[1] Multi-label classification has been used to improve search for research articles, by generating more relevant tags than possible from simply using metadata.[3] In a similar vein, the technique has been explored using Ottoman and Russian literary and critical texts, which had meticulous labeling done by literary researchers.[4] This project builds most directly upon efforts in natural language multi-label classification.

---

[1] Wu, Shunyao et. al., https://doi.org/10.1016/j.csbj.2021.04.054
[2] CMU Book Summary Dataset, https://www.cs.cmu.edu/~dbamman/booksummaries.html
[3] Mustafa, Ghulam et. al., https://doi.org/10.1038/s41598-021-01460-7
[4] Gokceoglu, Gokcen et. al., http://arxiv.org/abs/2407.15136

There are many different approaches to multi-label classification. Bogatinovski et. al. describe several different types of random forest and decision tree methods in a comparative review.[5] Wu et. al. implemented random forest and decision trees.[1] Meanwhile, Gokceoglu et. al. compared a bag-of-words naive Bayes model with LLMs - finding that the bag-of-words models were *more* successful.[4] Another influential resource was Joshi Prateek's work using a logistic regression classifier[6]. Overall, the literature demonstrated that researchers are still identifying the best models for multi-label classification.

Data

The dataset contains plot summaries of 16,559 books, sourced from Wikipedia and Freebase. For each book, there is a title, book metadata (IDs, author, publication date), and a list of genres. After removing records missing either plot summaries or genre labels, there are 12,841 records. There are 227 unique genres, and books have up to 10 genres, with the average book having 2.34 genre labels.

The final method of comparison across models was the average cosine similarity between predicted genre-list vectors and the actual genre-list vectors. Cosine similarities were chosen because they approximate similarity in meaning, and the aim was not predicting the *exact* set of labels, but rather generating a *similar* set of labels. Again, this interest was rooted in the problem domain, since genre-tagging is a subjective consensus, unlike the discrete labeling that may occur in scientific applications. The cosine similarities were pairwise for each genre in the genre lists to account for the potential for varied lengths among genre lists. Because pairwise cosine similarities were computationally intensive, I limited the train and test data to a sample of 5,000 records, with an 80/20 train/test split. To compute the cosine similarities, predicted and actual genre vectors were encoded using the BERT-base-uncased tokenizer.

Methods

Akin to work of Gokceoglu et al. this research sought to compare bag-of-words models with an LLM. The exploratory phase of this research included trial and error fine-tuning T5 (an encoder-decoder architecture). Unfortunately this phase was more error than trial, as the model consistently output book summaries rather than genres. Ultimately, I chose OpenAI's gpt-3.5-turbo model (a decoder-only architecture) and prompted the model to list appropriate genres for the book summaries.

The baseline model is a simple bag-of-words approach. Specifically, I use sklearn libraries for multi-label binarizing and for the model, which is a OneVsRestClassifier evaluated using logistic regression. The multilabel binarizer was fit on the genres in the data. This converts the series of genre vectors into a matrix of one-hot encodings which can be input into the OneVsRestClassifier multi-label extension of the logistic-regression classifier. This approach is considered a 'bag-of-words' approach because the classifier is agnostic to the overall context of the words.

Models 1.2 - 1.5 tweak the baseline bag-of-words approach. First, models 1.2 and 1.5 drop 'stopwords', which are words like "the" "a", etc. as these convey little meaning in a statistical model. This technique

---

[5] Bogatinovski, Jasmin et. al., https://doi.org/10.1016/j.eswa.2022.117215
[6] Joshi, Prateek. https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/

decreases noise in the data, while reducing the size of the dataset. Next, building on the work of Mustafa et. al., models 1.3 - 1.5 adjust classification thresholds to improve their multi-label classification. I discuss why this works in the results section.

Model 2.1 employs a random forest classifier in place of the logistic regression classifier. Bogtinovski et al.'s literature review of various methods for multi-label classification included a couple papers using random forest approaches. Although random forests are robust and can have multiple outputs, I did not attempt to fine-tune this random forest model because its baseline was far less effective than the baseline logistic regression model.

Finally, the research extends beyond bag-of-words approaches, and applies a LLM. Specifically, I use OpenAI's GPT-3.5 Turbo model. This is a decoder framework that is able to retain context over long sequences like those found in the book summary dataset. In theory, the language model's understanding of the summary text could better-inform the final output labeling relative to the initial bag-of-words statistical models. An advantage of OpenAI's GPT-3.5 model vs. an alternative LLM like Google's T5 was that it was easier to tune to multi-label generation task. I did not train this model, but gave the following prompt:

"Given the following book summary, list the appropriate genres from this list: {','.join(genre_list)}. Summary: {summary}")"

This is 0-shot learning as I did not provide any examples. Without a training phase, only the test data was fed into the OpenAI API. This prompt worked quite well, in contrast to the difficulty encountered using T5, where the model output summaries rather than genre lists.

Results and Discussion

The results below show each model's performance relative to the test data's genre lists.

Model #	Model	Accuracy	Precision	Recall	F1	Cosine Similarity	Unlabled
1.1	Baseline (Logistic Regression)	0.04	0.72	0.14	0.23	0.26	692
1.2	Logistic without stopwords	0.04	0.77	0.12	0.21	0.22	749
1.3	Logistic, threshold = 0.3	0.08	0.53	0.43	0.47	0.77	72
1.4	Logistic, threshold = 0.2	0.04	0.42	0.59	0.49	0.82	0
1.5	***Logistic without stopwords, threshold = 0.2***	***0.04***	***0.43***	***0.58***	***0.50***	***0.82***	0
2.1	Random Forest	0.02	0.76	0.05	0.10	0.10	889
3.1	Open AI, GPT 3.5-Turbo	*N/A**	*N/A**	*N/A**	*N/A**	0.72	0

The first takeaway is that model accuracy is extremely low. This reflects the fact that multi-label classification is very difficult to perfect. In addition to predicting genres that do appear in the actual data's genre list, the model needs to determine the number of genres worth predicting, in other words, the size of the set of genres for that book's summary. Although this study did not employ a human reference example, multi-label genre-classification is highly subjective, such that I would not expect any model or human to reach 100% accuracy.

Next, observe the interplay between model precision and recall, and how this works out to the F1-score (F1-score is the harmonic mean of precision and recall). The three bag-of-word models without probability thresholds had higher precision, but very low recall (1.1, 1.2, 2.1). These models made strong predictions when they did predict a genre, but also assigned 0 genres to many books. Note column "Unlabeled" in the

results table, where you can see that of the 1,000 records classified, models 1.1, 1.2, and 2.1 assigned 0 genres to a majority.

In response to this non-labeling, models 1.3, 1.4, and 1.5 lowered the default threshold value from 0.5. No adjustments were made to the random forest model because it performed far worse than the logistic regression model prior to applying a threshold. A threshold of 0.2 led to all records being labeled. By lowering the models' threshold for confidence needed to predict a genre, the precision declined while the recall increased. Note how the key measure, cosine similarity, dramatically improved. This was primarily because all genre vectors were populated, but it also meant that lowering the prediction threshold did not lead to dramatically-worse predictions.

While eliminating stopwords had a negative impact without a threshold, the more focused bag-of-words model which included both a threshold of 0.2 and dropping stopwords performed best.

Next, I briefly explored a Random Forest model. This was also a bag-of-words approach, but was far less successful than the baseline logistic regression. The performance with the default 0.5 probability threshold was so poor that I did not explore other variations and focused on the LLM alternative.

The last model used Open AI's GPT 3.5-Turbo as a classifier. Note that all 1,000 book summaries received a label vector - unlike some of the initial logistic regression approaches. Even still, the OpenAI predicted genre labels were less similar to the actual genre labels than the improved Logistic models. Further, although the prompt specified a list of the 227 genres from the data, the OpenAI model output genres outside this set. This made metrics like accuracy, precision, recall, and F1-score less relevant because the 227-genre one-hot vector for each book was not appropriate (with some labels outside of this 227-genre set).

For instance, in the first record in the test set, (*Daniel's Story* by Carol Matas), the actual genre label was ["Children's literature"]. The best logistic regression model (model 1.5) predicted ["Fiction", "Speculative fiction"], while GPT predicted: ["Historical fiction", "War novel", "Holocaust literature"]. "Holocaust literature" did not appear as a genre in the original data, and so did not appear in the 227-genre one-hot encoding of potential genre lists. Reading the plot summary, the book is about the Holocaust, so GPT's prediction is valid. Each model's predictions had similar cosine similarities for this record (0.82 for GPT, and 0.83 for the logistic model). This example re-affirms the value of using cosine similarity as the key metric for model classification.

Conclusion

Multiple machine learning approaches were applied to multi-genre classification of book summaries. Both bag-of-words and decoder strategies were employed. The baseline logistic and random forest bag-of-words had the lowest performance - primarily driven by predicting empty genre lists a majority of the time. Logistic regression performance was dramatically improved by adjusting the classification threshold from 0.5 to 0.3 and 0.2. Ultimately, the best-performing model, as measured using cosine similarity to the test data's genre vectors, was the logistic regression without stopwords and using a 0.2 threshold. An LLM, OpenAI's GTP-3.5 Turbo, was a decent alternative but still scored lower than the best logistic regression models.

Code

Section / Model	Path (Root: https://github.com/WMeyer98/266Final_WMeyer)
EDA & 1.1	Book Multi-Label Classification Baseline - 1.0.ipynb
1.2	Book Multi-Label Classification Model 2.0.ipynb
1.3	Book Multi-Label Classification Model 3.0.ipynb
1.4	Book Multi-Label Classification Model 4.0.ipynb
1.5	Book Multi-Label Classification Model 5.0.ipynb
2.1	Book Multi-Label Classification Random Forest.ipynb
3.1	Book_Summary_Multi_Label_Classification_OpenAI.ipynb

Acknowledgements

Bogatinovski, Jasmin, Ljupčo Todorovski, Sašo Džeroski, and Dragi Kocev. “Comprehensive Comparative Study of Multi-Label Classification Methods.” *Expert Systems with Applications* 203 (October 1, 2022): 117215. https://doi.org/10.1016/j.eswa.2022.117215.

“CMU Book Summary Dataset.” Accessed July 30, 2024. https://www.cs.cmu.edu/~dbamman/booksummaries.html.

Gokceoglu, Gokcen, Devrim Cavusoglu, Emre Akbas, and Özen Nergis Dolcerocca. “A Multi-Level Multi-Label Text Classification Dataset of 19th Century Ottoman and Russian Literary and Critical Texts,” July 21, 2024. http://arxiv.org/abs/2407.15136.

Joshi, Prateek. “Predicting Movie Genres Using NLP - An Awesome Introduction to Multi-Label Classification.” *Analytics Vidhya* (blog), April 22, 2019. https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/.

MMA. “Metrics for Multilabel Classification.” Mustafa Murat ARAT, January 25, 2020. https://mmuratarat.github.io//2020-01-25/multilabel_classification_metrics.

Mustafa, Ghulam, Muhammad Usman, Lisu Yu, Muhammad Tanvir Afzal, Muhammad Sulaiman, and Abdul Shahid. “Multi-Label Classification of Research Articles Using Word2Vec and Identification of Similarity Threshold.” *Scientific Reports* 11, no. 1 (November 9, 2021): 21900. https://doi.org/10.1038/s41598-021-01460-7.

OpenAI Platform. “GPT-3.5 Turbo.” Accessed August 3, 2024. https://platform.openai.com/docs/models/gpt-3-5-turbo.

Wu, Shunyao, Yuzhu Chen, Zhiruo Li, Jian Li, Fengyang Zhao, and Xiaoquan Su. “Towards Multi-Label Classification: Next Step of Machine Learning for Microbiome Research.” *Computational and Structural Biotechnology Journal* 19 (April 28, 2021): 2742–49. https://doi.org/10.1016/j.csbj.2021.04.054.