Genre Recommendation for books from their GoodReads description

Amritansh Gupta
University of California San Diego
amg080@ucsd.edu
and
Kailash Jayaram
University of California San Diego
kjayaram@ucsd.edu

In this paper we attempt to create predictive models for multi-tag genre classification of books from their GoodReads descriptions. We introduce a multi-layer neural network taking advantage of glove word embeddings fed into convolutional and LSTM layers in order to classify the data via a 8 binary Support Vector Machines. We show the advantages and disadvantages of this complex model over the simpler word-count based approaches and single layer perceptron classification. The model takes advantage of semantic structure learning on the description in order to gain more information about the content of the sentences than word counting based models could. It also uses a convolutional and max pooling layer to focus in on those parts of the description that are most influential on genre prediction. We show how this gained knowledge from use of the convolutional and LSTM layers might provide an advantage in a multitude of classification tasks and make marginal gains on the task of assigning genres to new books given only their GoodReads description.

Categories and Subject Descriptors:

Additional Key Words and Phrases: Text classification, Book Genre Prediction from Description, Convolutional Recurrent Neural Networks, Support Vector Machines, Logistic Regression, GoodReads

ACM Reference Format:

Kailash K. Jayaram and Amritansh Gupta. 2018. Models for genre classification of books from their GoodReads description.

1. DATA

The dataset we use is from the UCSD Book Graph website *Goodreads*[1]. The data originally contained 229,154,523 records collected from 876,145 users public book shelves and covers 2,360,655 books (with detailed meta-data including authors,series, editions, publishers, numbers of pages,languages of book contents, similar books and top user-generated shelf names for these books)[1]. We defined 8 genres for the books based on the separation done on the UCSD book graph website. These genres were collected by examining the names of the bookshelves users placed these books on. Because of this book genres are not unique, meaning that the same book could be listed under different genres. In order to obtain a usable working sample we took only 200,000 records (182K unique books with some having multiple genres) in total with exactly 25k of each genre.

The data has been processed in a way such that each record is a tuple of book data and category. Only the English books have been considered for the purpose of this task for the sake of simplicity. We separated out English books using the country code tag. The book data includes the popular shelves (with number of users) that the book was placed on by the users, average rating, description, authors, publishers, number of pages etc. The average description length over all the 182k unique books is $\sim$ 713 words with the largest being 2501 words.The 8 genres that we selected are Children, Comics and Graphic, Fantasy & Paranormal, History & Biography, Mystery Crime & Thriller, Poetry, Romance, Young Adult.

For the dataset we randomly sampled 150k unique examples for training and 16k examples for validation and the final 16k for testing. We felt that this dataset split left enough examples to train our models to a reasonable extent while still leaving room for a good validation set to analyze the comparative success of different models. While utilizing a larger portion of the 2.3 million examples would certainly have allowed for better results, we were limited in the capabilities of our hardware to use a dataset of that scale, both in processing time and memory limitations.

2. TASK

Due to the size and variance of the dataset, there were a plethora of tasks that we could think up. However the most obvious and useful seemed to us was category prediction for individual books from the given features. The problem of text classification has been studied at all levels and we found that a categorization model for book genres can be very useful in the real world for segmentation, management, sorting and even user recommendation based on previously read genres.

For a baseline comparison we consider a classifier that assumes that each book tends to belong to only one category. For each book in the training set, the classifier would use a simple bag of words representation of the book description run on a single layer logistic regression classifier to make a prediction. This classifier fails to take into account that some of the books have multiple genres assigned to them. However, despite this shortcoming it will still provide a useful baseline to beat. Another classier that can be deemed similar to this would be to train 8 binary classifiers based for each category using the same feature representation of the description and classifier model. This model will provide our main source of comparison for the accuracy of the proposed model.

We will evaluate our model based on how well it compares on each of the 8 predictive subtasks in comparison to the baseline model. This allows us to set a baseline of performance for the task basically allowing us to understand how difficult the task is. This will allow us to have a better understanding of the quality of the proposed models accuracy on this specific task. A good model

should hopefully be able to make at least marginal gains even on a simple task in comparison to a simple bag of words classifier.

We will compare the accuracy scores on the individual tasks rather than trying to define a multi-tag classifier score for the models. A model that performs better on each of the individual tasks will perform better on the multi-genre prediction task as we will be assigning genres to each book based on which of the 8 models predicted the book description to be a member of its assigned genre. Because of this we feel that the binary class prediction accuracy is an adequate substitute for any form of accuracy definition for multi-tag classification problems. We will use the accuracy of the 8 binary classifiers on the validation set in order to evaluate and choose the best models for the task.

In our model we use the book description text as the predictive feature to train on. In order to standardize the input, and to minimize the memory usage of the model, we set the maximum description length (in number of words) to a value and truncate/pad all of the descriptions to this length. We then chose to represent each description text via a vector of word embeddings. We map the description to a sequence of words that are defined after stripping punctuation and mapping the description text to lower case. Then we represent each word as a fixed length vector embedding In order to obtain the word embeddings we used the Stanford pre-trained Wikipedia glove model (https://nlp.stanford.edu/projects/glove/). The glove model is a way of learning the meaning of words and attempting to map words into a vector space such that similar words are closer to each other in this vector space. It has found many uses in NLP for a multitude of text related tasks. Each vector can be thought of as a representation of the word, and words with similar vectors can be thought of to have similar meanings. We opted to use pre-trained glove vector models instead of training our own due to the size of the dataset the Stanford models were trained on. The Stanford model contains over 400k words all learned from Wikipedia articles. It had a much larger input dataset than we would have been able to train locally on our computers. On top of this we felt that our dataset was similar enough to the Wikipedia dataset. While something like a social media or review dataset that has a lot of unedited content that may lead to an abundance of misspellings and such, the good reads descriptions appear to be well edited so as not to contain such mistakes. Since the Wikipedia corpus is similarly well maintained we concluded that the majority of words in our corpus would be in the glove model and that the word meanings would be roughly the same. Because of this we felt that the massive data advantage of the pre-trained vectors made them much preferable to training our own.

3. MODEL

Because the dataset was structured in such a manner that it was possible for books to be listed under multiple genres, and the fact that in general many genres overlap (like young adult and fantasy for example), we chose to construct 8 binary classifiers. Then when evaluating a give book, we can run its description through each of the classifiers and determine which of the genres that book belongs too. The model of classifier we propose to solve this problem is a 3 layer network. The network takes as input a vector of word embedding vectors. Once we constructed the vector of word embedding vectors for each review we then passed this into the first layer of the model. The first layer of the model is a convolutional network with a max pooling layer that helps the model to select which words play the largest role in text classification. Justification for this usage of CNN in text classification tasks was inspired by Siwei Lais paper on usage of Recurrent Convolutional Neural Networks for Text Classification. We then use the output of the pooling layer into a LSTM network that will capture the recurrent nature of the text. Since the order of words in a sentence has significant impact on text classification tasks we implement this RNN to capture this ordering. We specifically choose to use a LSTM in order to avoid the exploding gradient problem that was encountered in our first attempts. By using a LSTM with a probabilistic dropout of 0.2 we avoid the exploding/vanishing gradient issue. The final layer of the network takes the LSTM output and runs it through a linear layer that can then be sent to the SVM to make a binary classification as the final output of the network. We choose to use an SVM (hinge loss) over a logistic regression (binary cross entropy loss) classification layer as the SVM had better performance on the validation set. We trained each of the models on the same dataset (adjusting the target vectors to each of the 8 binary classification tasks).

In order to optimize the model we took a subset of the training data (about 10,000 books) in order to choose hyper-parameters for each model in a reasonable time period. The main hyper parameters to select were the glove vector / LSTM vector length, the regularization constant on the SVM and the number of epochs to train the network for. We ran into hardware limitations on utilizing the larger glove/LSTM vectors. While it quickly became clear that the larger vectors performed better we were unable to use a vector length of greater than 100 due to memory constraints. The models appeared to have similar performance in relation to the number of epochs we ran the network. It appeared that the model made significant gains up until the third epoch, after which it began to quickly suffer from over-fitting. We fit the regularization constant independently to each model. The model also slightly suffered from scalability issues as its training time and memory requirements were significantly larger than simpler models. However, on the sampled dataset of 182k unique books the model was able to train fast enough even on a laptop.

On the way to building our final model we tried a few different variations. Our first attempts were to simply construct a multi-class SVM and logistic regressor. However it quickly became apparent that these models would not perform well enough because a large enough portion of our dataset contained books that had more than one genre, so we wanted to capture a way of doing multi-tag classification. We trained the logistic regressor on each of the binary tasks while fitting its regularization constant to each task as a baseline of comparison the our network. We inputted a bag of words representation of the top 2000 words into the regressor. We used logistic regression as opposed to an SVM because the regressor outperformed the SVM on the validation set for the bag of words model.

The strength of the regressor was mostly in its rapid training time. It took only a few minutes to train this model while training the network took up to an hour. However its weakness lies in its inability to use semantic structure in its classification decision due to the nature of bag of words models. The network on the other hand uses multiple layers to hone in on the important details of the description text and takes into account semantic structure through the LSTM's recurrent network. This allows the model to theoretically learn much more about what makes a given description a member of a genre then the bag of words model can, but unfortunately this greatly increases the number of parameters that need to be trained resulting in much longer training times and much higher memory usage.

4. RELATED LITERATURE

The dataset we use was prepared by Julian McAuley and Mengting Wan for their for their study of Item recommendation on monotonic behavior chains. They used the dataset to define a model that combines data on both implicit and explicit user feedback in order to create a more successful recommender system then those that focus on the use of only one or the other. Their use of the dataset took advantage of the review portion of the GoodReads records which is slightly different from the individual book data that we leveraged. Since our task was only focused on the categorization of books by descriptions, the only valuable user data we leveraged was in using the separate genre data files that were created in McAuley and Wans study. They used the user defined bookshelf's to map books to genres and we used this data as the target of our predictive task. While their use of the dataset was for a different purpose than ours their exploratory analysis and preprocessing of the data was extremely useful to our task of genre prediction.

When comparing our work to more similar predictive tasks on text classification we found that traditional text categorization models rely on standard machine learning techniques like Naive Bayes, maximum entropy classification, and support vector machines[2]. The research by Pang et al use a dataset of movie reviews. These review datasets are especially popular due to the large online collections and the rating associated with the reviews for judgment such as number of stars[2]. There are some areas of research that concentrate on the source or the style of the source to classify the texts with statistically-detected stylistic variation (Biber,1988) serving as an important feature[2]. An example if this research is by Douglas Biber's Variation across Speech and Writing, 1988.

Peter D. Turney and Michael L. Littman's Unsupervised learning of semantic orientation from a hundred-billion-word corpus is a relevant work on classification of reviews. They applied a specific unsupervised learning technique based on the mutual information between document phrases and the words excellent and poor, where the mutual information is computed using statistics gathered by a search engine. In contrast to them, we built a prior-knowledge-free supervised machine learning model.[4]

The inspiration for our work and the techniques uses for text classification tasks were from Siwei Lais paper on usage of Recurrent Convolutional Neural Networks for Text Classification. Lai's model, applies a recurrent structure to capture contextual information which is supposed to introduce considerably less noise compared to traditional window-based neural networks. They also employ a max-pooling layer that automatically judges which words play key roles in text classification to capture the key components in texts.[3] The datasets used by Lai consisted of messages from twenty newsgroups - choosing major categories (comp, politics, rec, and religion); ACL Anthology Network3 which is a dataset containing scientific documents published by the ACL and by related organizations; as well as Stanford Sentiment Treebank4 dataset which contains movie reviews. The results that we obtain are very much similar to those obtained by Lai's paper. To the traditional baseline solutions mentioned above, our results show that the neural network approaches outperform the traditional methods. Lai also found that "Neural networks can capture more contextual information of features compared with traditional methods, and may suffer from the data sparsity problem less."[3] His convolutional recurrent network also made marginal gains on existing models for many of the text classification datasets he applied it to. We saw very similar gains in our model over the baseline approaches we took.

5. RESULTS

The models results can be found in the table below. The table lists each of the models, the hyper parameters of the network, the accuracy of the network and then the accuracy of the baseline bag of words model. As you can see in the table both models had very high performances on the data. All of the predictive tasks achieved classification accuracies above 90% on the specific category. The network accuracy was better in 6 of the 8 tasks while the baseline model outperformed the network on both the thriller and romance categories. The results show that the GoodReads book descriptions are highly indicative of novel genre. The model performed very well on all 8 categories so it was able to create very good mappings of the description into separable vectors for the SVM to properly classify each book genre.

Model	Description Length	LSTM Vector Length	Glove Vector Length	L2 regularization parameter	Network Accuracy	Base Accuracy (bag of words baseline model)
0, child	200	200	200	.03	92.9%	92.4%
1, graphic	200	200	200	.025	92.3%	90.8%
2, fantasy	200	200	200	.03	91.5%	91.4%
3, history	200	200	200	.035	91.9%	91.7%
4, poetry	200	200	200	.0325	94.5%	94.3%
5, thriller	200	200	200	.03	91.5%	92.4%
6, romance	200	200	200	.03	92.9%	93.9%
7, ya	200	200	200	.025	90.8%	90.3%

The glove feature representation passed through the CNN and LSTM seems to be the most effective feature representation for the model. This representation out performed the bag of words approach in most of the tasks. The advantages of this representation is that it takes into account the position of each words as the recurrent network is trained over a time series of data. So while bag of words discards any notion of semantic structure the LSTM representation of the description does not. The LSTM representation is also being trained as part of the model so it can be adjusted to better fit the classification task while the bag of words approach is not as flexible as the features stay the same. The network also has the advantage of the convolutional layer which helps the model to construct feature representations that focus more on the important details of the vector to the classification task. And this CNN is also trained as part of the model which allows it to increase its ability to do this throughout the training process. We also attempted a TF-IDF implementation to similarly help the bag of words model focus in on words that might be important to classification however its accuracy was not an improvement on the bag of words model. This model suffers from the same issue of bag of words since it also discards any notion of sentence structure in favor of just taking word counts. In comparing the pre-trained glove vectors to training on the dataset we found that the pre-trained vectors out performed the vectors we attempted to train. This is most likely due to the lack of data and other reasons we listed in the task section of this paper. We also attempted to add an embedding layer to the model to be trained however this extra layer of complexity made the model pro-

hibitively difficult to train due to the massive increase in parameters that needed to be fitted.

The parameters of the network can be examined by layer. The first network parameters that need to be trained are the CNN and pooling layer parameters. These weights define how the model chooses to extract information from the word embedding vectors. This layer helps us to focus in on the important features of the embedding vectors to allow the LSTM to create a description representation more tailored to the task of genre classification. The weights of these parameters determine the input of the LSTM. The next layer of parameters to be trained were the LSTM network weights. These weights add another layer of complexity in determining the important features of the words and the semantic structure and combining them into one single output vector that can be classified. The LSTM acts as a state machine and so the weights of the machine determine how each new word seen affects the final output vector. These weights are critical to mapping the word embedding vectors into a more easily separable subspace. The final layer of the network is just a linear mapping of the output of the LSTM to determine which parts of the final feature representation of the text are most important. Running this final output through the SVM classifier produces the output of the model.

Overall both the baseline model and our network were highly successful at categorizing the book descriptions by genre of book. The main failures of the original models we tried were in trying to assign single labels to data we later realized was multi-label by nature. The main reason I believe our model slightly out-performed the bag of words approach was due to the previously explained RNN portion that mapped the descriptions to vectors that took semantic structure into account. I think that on more and more difficult tasks the gain you would see from not throwing out this information would become larger, however in the case of book genre classification, the models were very close. I think this was largely due to the fact that there are a lot of key words that are dead giveaways for certain genres that both models can make use of since singular keywords dont require semantic structure to learn on. Words like child for children's books or magic for fantasy books can be assigned heavy weights in both models. So while the network only made marginal gains on the classification task, we think overall it is a much stronger model for mapping text sequences to a vector space over the bag of words model.

6. BIBLIOGRAPHY

Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. (2018). http://cseweb.ucsd.edu/ jmcauley/pdfs/recsys18b.pdf

Bp Pand, Lillian Lee and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. (2002). EMNLP '02 Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10 Pages 79-86. https://dl.acm.org/citation.cfm?id=1118704

Siwei Lai, Liheng Xu, Kang Liu and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/download/9745/9552

Peter D. Turney and Michael L. Littman. 2002. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report EGB-1094, National Research Council Canada.

Douglas Biber. 1988. Variation across Speech and Writing. Cambridge University Press.