Movie Genre Prediction Using Text Embedded in Posters

백일웅
2018314951
alltun100@g.skku.edu
Software, SKKU

김우진
2017314712
Nicholasbear@naver.com
Software, SKKU

조현준
2017314593
triesin23@gmail.com
Mathematics, SKKU

손민혁
2018313925
cliff615@naver.com
Software, SKKU

ABSTRACT

When we choose the movie want to watch, we always find some information about that movie. We consider title, directors, synopsis, leading actors, etc. That's because we want to reduce our time-loss or money for downloading the movie or purchase the ticket. Then, many people might recognize about something that includes those information (ex. Title, actors, etc.) which we call poster. Many movie companies put a lot of efforts into it by making many versions for each country. Movie poster for advertisement is in every theater and close to customers. It contains most of the information about the movie so that people can decide to watch or not. That's why we call movie posters the faces of the movies. The goal of this paper is to show that by using the information from movie posters we can predict movie genres. While there are works that proved movie posters image give help to predict movie genre we also used text data extracted from the poster image to increases model accuracy about guessing movie genres. In this paper we will show some ways how to extract text data from images, and how can we make those text data contribute to improve model accuracy. Since movie is not a single-configuration of genre, the model needs method for calculating multi-labeled loss and method for multi-labeled accuracy. There are 3 architectures to merge CNN vector and CLS vector resulted from BERT.

CCS Concepts

• Computing methodologies → Image representations;

Keywords

Movie genre; Movie poster; Convolutional neural network; Bert;

1. INTRODUCTION

With the rapid development of Internet technology, people can watch movies in anywhere they want. Today, places with computers or smartphones or something else that can be connected with Internet space can be a movie theater. As time goes by movie genres have been increased and developed (by watching old movies people will say "Is that a movie?"). However, there is an immutable law that movies always have which we call the movie poster. The movie poster is the face of the movie. Movie posters contain title, directors, actors or actresses, and some provocative phrases. Then, is it impossible to guess movie genres with only using the data from movie poster?

We know some studies have already been conducted to predict the genre with part of the film. However, these studies predict genres with only image data from movie poster and the trailer of the movie. There are studies that use text data of movie summary and comments, but there was no research using text data in the movie poster. Since we think that text data in movie posters are important, we can improve the model by using text data and image data together.

The text data that exists in the movie poster is diverse. There are keywords that can predict specific genre, there are proper nouns such as specific local name or human names, there can be company names of movie productions. Of course, There are text data in the image data that do not improve model accuracy. Therefore, the pre-processing of text data is also important. We experimented with 3kinds of data. 1)Not using text data (only image data) 2) Raw text data (not preprocessed) 3) Preprocessed data.

To the best of our knowledge, it is the first study that using both image data and text data from movie poster. We constructed our experiment by these follow steps. First, extract text data from the movie poster image data. Second, pre-process that text data by remove some stopwords(ex. Proper noun in movie title, name of actors, etc..), and other things. Third, train model with convolutional neural network(CNN) for image data, and Bert for text data. Finally, merging vectors from step 3, and get accuracy with three architectures we made. To demonstrate the effectiveness of our approach, we set the baseline model that just trained by CNN. We show how much model accuracy has improved at the conclusion of this paper. Also, how can you utilize our resultant product to the other problem.

2. Related Work

There have been some works on movie genre classification. One of the earliest work is proposed by Zeeshan Rasheed using audio-visual features[1]. They tried to find trailer's distinguishing sound features for each movie genres.

Recently, many deep neural network models came out and used for classification problems. Especially we found out when classifying image data, CNN models are showing good performance. Since movie trailer is also sum of consecutive images, some research tried to use CNN models with movie trailers to classify movie genres [2]. To improve performance, new research on CNN models have been constantly being continued and driven to make models such as Resnet [3], VGGNet [4] and ViT [5]. Some works on classifying movie genre using these advanced CNN model came out [6]. In this paper modified ResNet-50, VGG-16, DenseNet-16 models and only posters for the data are used. Moreover, since object detecting technology has also advanced, there is work that collaborate two CNN models. One for original poster, and one for detected objects [7].

You can see many works are done for movie genre classification. However, there were no works about using text data with either poster or trailer. In this work, we propose new way to predict movie genre by adapting text data with image data.

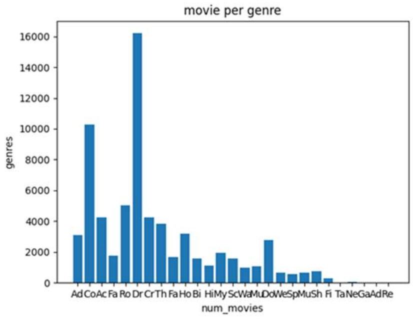


Figure 1

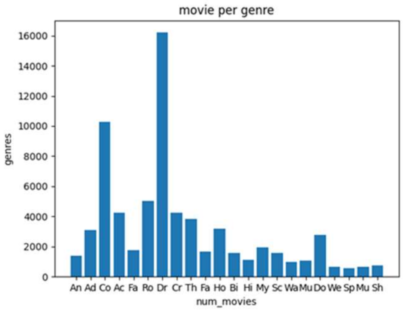


Figure 2

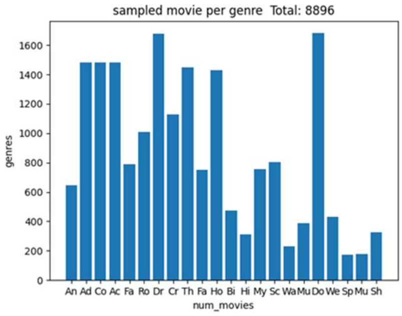
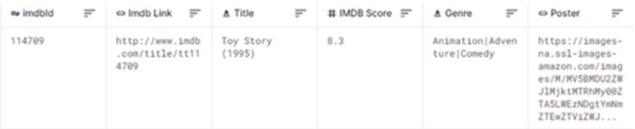


Figure 3

3. DATA

3.1 Source

We got data from Kaggle with almost 40000 images with IMDB ID, IMDB link, title, IMDB score, genre, and lastly the URL of the images. We used only the URLs of the images and the genre data.



imdbId	Imdb Link	Title	IMDB Score	Genre	Poster
114709	http://www.imdb.com/title/tt114709	Toy Story (1995)	8.3	Animation|Adventure|Comedy	https://images-na.ssl-images-amazon.com/images/M/MV5BMDU2ZWJlMjktMTRhMy00ZTA5LWEzNDgtYmNmZTEwZTViZWJ...

Kaggle Data

3.2 Analysis

There are 28 Genres. Animation, Adventure, Comedy, Action, Family, Romance, Drama, Crime, Thriller, Fantasy, Horror, Biography, History, Mystery, Sci-Fi, War, Music, Documentary, Western, Sport, Musical, Short, Film-Noir, Talk-Show, News, Adult, Reality-TV, Game-Show. There were many problems in the poster URL so we reduced the data to 31741. (**Figure 1**)

3.3 Subsampling

1) Getting rid of minor Genres

While we had ten thousand of data in Comedy, Drama for genres in Film-Noir, Talk-Show, News, Adult, Reality-TV, Game-Show they had tens of data which is a big difference and this made a disadvantage in training data so we got rid of these 6 genres. (**Figure 2**)

2) Balance

Still, we reduced 6genres there are still imbalance in data. When we trained with the above data because there were too much data in Comedy and Drama it predicted with almost Comedy and Drama. Therefore, we needed to make a balance with the amount of data. The result is the graph beneath. (**Figure 3**)

3.4 Result

We used 8896 images and multi-labeled by 22genres. We made a multi-hot vector to use the data.

Ex) [Action, Adventure, Music] = [0, 1 …,1,0]

4. Text Detection

To get text from images we needed to use a technology called OCR. OCR (optical character recognition) is the use of technology to distinguish printed or handwritten text characters inside digital images of physical documents, such as a scanned paper document. The basic process of OCR involves examining the text of a document and translating the characters into code that can be used for data processing. OCR is sometimes also referred to as text recognition.

4.1 Tesseract, EasyOcr

1) Definition

Tesseract and EasyOCR are open-source OCR engine that has gained popularity among OCR developers. They are used in python and have access to over 70+ languages.

2) Problem

Figure 4 extracted by Tesseract: WGseg TOY StoRy

Figure 4 extracted by EasyOCR: 7Diat ToY Stont

By Looking at the result we can know that there are many problems.

3) Image Preprocessing

To solve the problems, we tried several image preprocessing. 1. Making image black-and-white 2. Resize 3. Apply Kernel 4. Image contrast 5. Image Binarization 6. Getting rid of noises.

Figure 5 result: Dialtp Atrurei ToY Story

Figure 6 result: No Text Detected

As we see the problem became more serious, so we had to find a new OCR engine.

4.2 Google Cloud Vision Api

The Google Cloud Vision API uses machine learning to identify images from pre-trained models on huge datasets of images. Google Cloud Vision can also automatically identify a broad range of different languages. By giving language-hints to English because our dataset is English posters, we got text data.

Figure 4 with Google Cloud Vision Api: Wer Disney Pictures presents TOY STORY

Figure 4

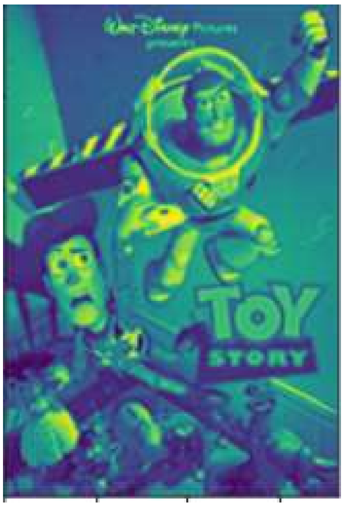


Figure 5

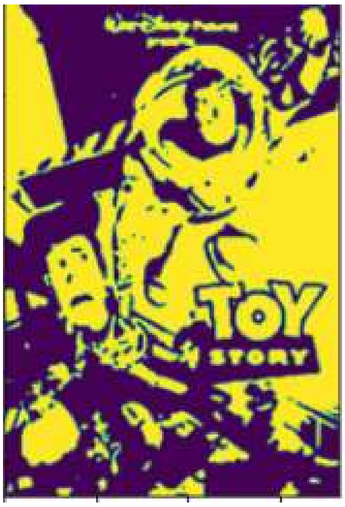


Figure 6

As we can see we can know the performance of google cloud vision is outstanding.

4.3 Text Data Preprocessing

Despite the high performance of Google Cloud Vision Api there were several problems. 1) No text in image 2) Image that is hard to get text data (bad image, text and image overlapped) 3) Text that is not related to movie ex) text in characters clothes. To avoid these problems, we needed to preprocess text data. We used the NLTK library from python for preprocessing.

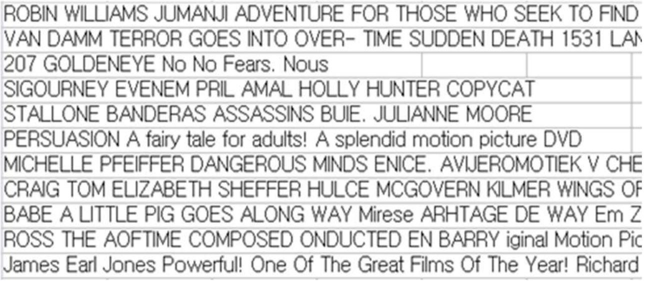
1) Remove special characters and numbers

2) Remove stopwords

3) Change plural words to singular

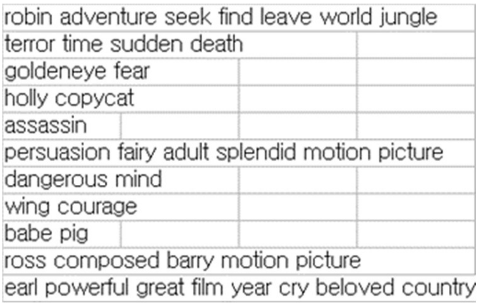
4) Remove meaningless words

If the word is not in the NLTK package we decided meaningless. Since there are only singular words in the NLTK package. We made plural words to singular.



ROBIN WILLIAMS JUMANJI ADVENTURE FOR THOSE WHO SEEK TO FIND
VAN DAMM TERROR GOES INTO OVER- TIME SUDDEN DEATH 1531 LAN
207 GOLDENEYE No No Fears. Nous
SIGOURNEY EVENEM PRIL AMAL HOLLY HUNTER COPYCAT
STALLONE BANDERAS ASSASSINS BUIE. JULIANNE MOORE
PERSUASION A fairy tale for adults! A splendid motion picture DVD
MICHELLE PFEIFFER DANGEROUS MINDS ENICE. AVIJEROMOTIEK V CHE
CRAIG TOM ELIZABETH SHEFFER HULCE MCGOVERN KILMER WINGS OF
BABE A LITTLE PIG GOES ALONG WAY Mirese ARHTAGE DE WAY Em Z
ROSS THE AOFTIME COMPOSED ONDUCTED EN BARRY iginal Motion Pic
James Earl Jones Powerful! One Of The Great Films Of The Year! Richard

(Text before preprocessing)



robin adventure seek find leave world jungle
terror time sudden death
goldeneye fear
holly copycat
assassin
persuasion fairy adult splendid motion picture
dangerous mind
wing courage
babe pig
ross composed barry motion picture
earl powerful great film year cry beloved country

(Text after preprocessing)

4.4 Text Embedding

For text embedding we used a Bert-Base uncased model. Bert-Base uncased model is a pretrained model on English language using a masked language modeling (MLM) objective. This model has 12-layer, 768-hidden dimension, 12-heads, 110M parameters. After tokenizing the text data, we put it into the model and get the CLS-Vector and we will call it $v_{cls} \in R^{768}$.

5. PROPOSED METHOD

Now we got CLS vector v_{cls} from text data. We get image vector $v_{img} \in R^{512}$ by passing poster image to Resnet-18's residual block. We thought both text and image in same genre have similar features in high dimensional area, so we decided to combine v_{cls} and v_{img} to get output vector $V_{out} \in R^{22}$ which the dimension size is same as number of genres.

$$\boldsymbol{V_{out}} = [\boldsymbol{l_0}, \boldsymbol{l_1} \dots, \boldsymbol{l_{21}}],$$

where l_i is logit value of each 22 genre, l_0 stands for the predicted logit value of 0^{th} genre which is 'Animation'.

Therefore, we designed 3 architectures to get output vector V_{out}.

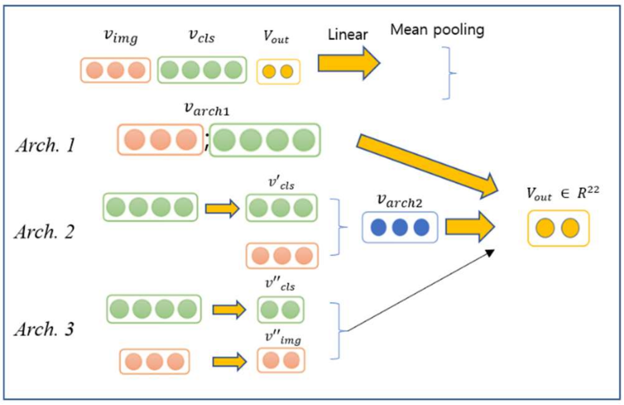


Figure 7: Brief explanation about overall 3 architectures

5.1 Three Architectures

We differentiated 3 architectures according to the way how v_{cls} and v_{img} are combined.

$$v_{img} \in R^{d_i}, \qquad v_{cls} \in R^{d_c}, \qquad V_{out} \in R^{d}$$

For convenience, we denote each vectors dimension as d_c and d_i and the output dimension as d

7.1.1 Architecture 1. Merging. First way to combine is concatenating two vectors

$$v_{arch1} = [v_{cls}; v_{img}] \in R^{d_c+d_i},$$

where [;] denotes concatenation operation

We will pass v_{arch1} through 2-fc layers to get V_{out}

$$Arch1 \; V_{out} = W_2(W_1 \cdot v_{arch1} + b_1) + b_2, \qquad (1)$$

where $W_1 \in R^{512 \times (d_c+d_i)}, W_2 \in R^{d \, x \, 512}$

7.1.2 Architecture 2. Element wise mean pooling (middle layer). Second way to combine is averaging element wise sum of v_{cls} and v_{img} at the middle of layer. However, the dimensions of two vectors do not match, so we pass v_{cls} through 1-fc layer to get $v'_{cls} \in R^{d_i}$ and average the element wise sum of v_{img} and v'_{cls}.

$$v_{arch2} = \frac{1}{2}(v_{img} + v'_{cls}) \in R^{d_i},$$

then we get V_{out}

$$Arch2 \; V_{out} = W \cdot v_{arch2} + b, \qquad (2)$$

where $W \in R^{d \times d_i}$

7.1.3 Architecture 3. Element wise mean pooling (end layer). For the last architecture, first by passing v_{img} and v_{cls} through 2-fc layers we get $v''_{img} \in R^{d}, v''_{cls} \in R^{d}$, where dimension is same as output vector. Now we can get V_{out} by averaging the element wise sum of v''_{img} and v''_{cls}.

$$Arch3 \; V_{out} = \frac{1}{2}(v_{img} + v'_{cls}). \qquad (3)$$

Figure 7 is the brief image explanation of our architectures

5.2 Multi-Label Loss

Since our data is multi-labeled, we need appropriate loss function. Therefore, we use loss function that can consider multi-label as much as possible and it is the multi-label soft margin loss.

$$loss(x,y) = -\frac{1}{C} * \sum_{i=0}^{21} y[i] * \log((1 + \exp(-x[i])^{-1})) + (1 - y[i]) * \log\left(\frac{\exp(-x[i])}{1 + \exp(-x[i])}\right),$$

i is each label of movie genre, calculate all labels probability by sigmoid the logit values in V_{out} and get entropy value by computing entropy with real value. Finally, we average all the entropy value from 22 labels and use it for each movie's loss.

6. EXPERIMENT

In this section we would like to evaluate the performance of the proposed xxxx model. Experiments conducted on movie posters are aimed to answer the following questions:

- **RQ1:** Does our model using text data outperform the genre classifier using only movie poster image?
- **RQ2:** Does text preprocessing show performance increase in classifying genre?

In what follows, we first set up the experiments, and then present and analyze the results. Finally, represent how the proposed method performs in a specified case.

6.1 EXPERIMENT SETUP

Dataset. We evaluate our model on datasets provided by Kaggle data with IMDB. The data sets were constructed from subsampling the original 31741 images to 8896 images, due to biased distribution of genre labels.

Baseline Models. We compare our method with the following baseline models:

Resnet18 optimized by Adam via minimizing multilabel soft margin loss, contains multi-layer skips to avoid vanishing gradient problem.

Alexnet optimized by Adam

Baseline models use only the image data in prediction.

Our models. Each three architectures are evaluated:

Arch. 1 Resnet18 merged with text data

Arch. 2 Resnet18 with mean pooling in the middle

Arch. 3 Resnet18 with mean pooling in the end.

Each of these architectures were trained with both raw text data and preprocessed genres.

Implementation Details and Evaluation. Training was done by 3-fold validation with 10% test, 60% train, 30% valid. We assume prediction is correct if movie's top-1 predicted genre is involved in that movie's real genre. We divide all the correct ones by total, and we call it as accuracy.

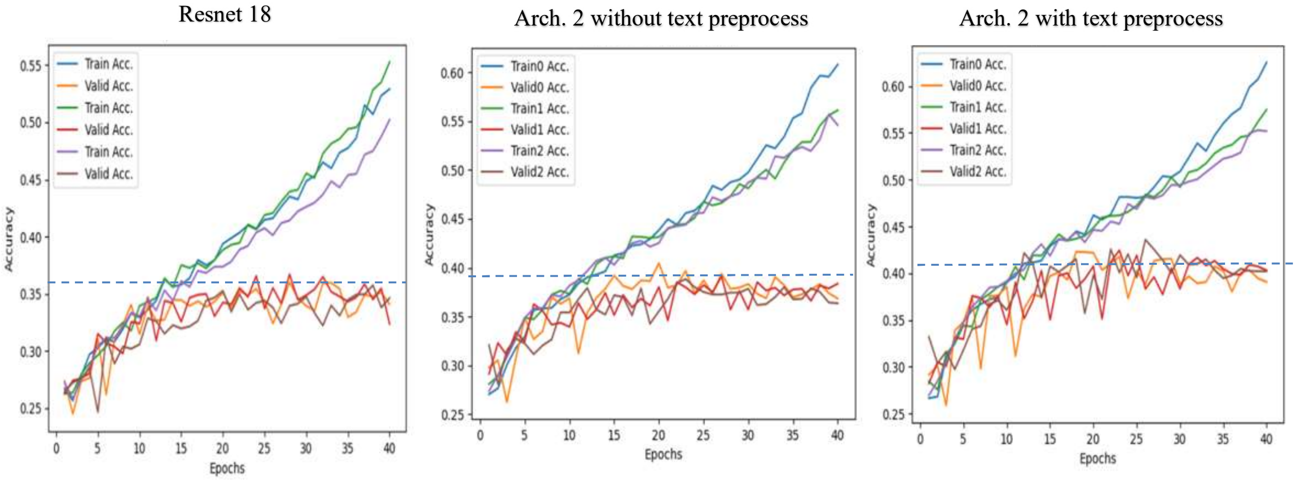


Figure 8

6.2 Experiment Result

Above **Figure 8.** is the comparison of Resnet18 using only image data, Resnet18 using image data and raw text data, Resnet18 using image and preprocessed text data respectively. The blue borderline represents the approximate best value of the valid accuracy.

Performance between image data only and image data with text data (RQ1)

Table 1 presents the accuracy comparison between the base line model Resnet18 using only image data and models using both image data and raw text data extracted from the image. Between the five models, Resnet18 augmented with text data via mean pooling on middle achieved the best performance. Performance increase using text data improved the accuracy on average of 3.8%.

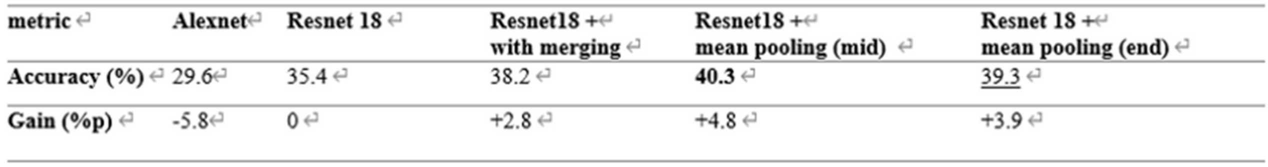
Possible reason for performance increase is that the movie from similar genres tend to contain similar adjectives. Similar adjectives are added to the image data which results in a similar vector overall.

Performance between models using raw text data and preprocessed text data (RQ2)

Table 2 presents the accuracy performance comparison between the three models using raw text data and the same models using preprocessed text data. Resnet18 with mean pooling on middle layer using preprocessed text data showed the best performance in predicting the correct genre. Preprocessing the text data showed a performance increase of 3.7% on average compared to the model.

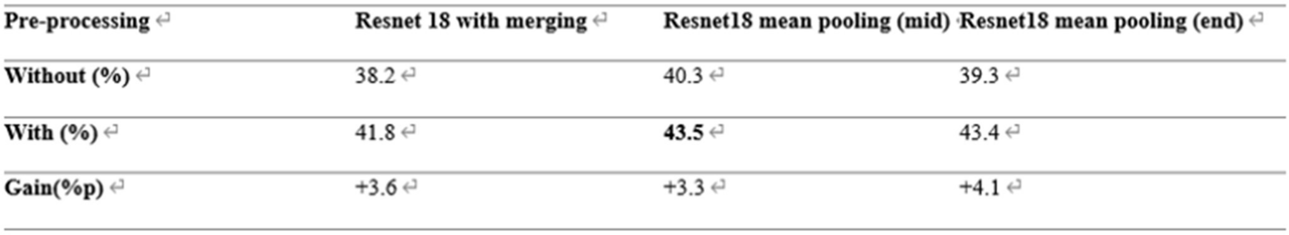
Possible reason for the performance increase could be from adjusting the noisy text data. Same words in the image can be interpreted into different words due to invalid OCR. After preprocessing, these different interpretations of the same words can be corrected back to the original word, resulting in similar vocabularies between same movie genres.

Table 1: Performance comparison of all methods with test data. The best performance and the second-best performance methods are denoted in bold and the underlined fonts respectively↵



metric ↵	**Alexnet**↵	**Resnet 18** ↵	**Resnet18 +**↵ **with merging** ↵	**Resnet18 +**↵ **mean pooling (mid)** ↵	**Resnet 18 +**↵ **mean pooling (end)** ↵
Accuracy (%) ↵	29.6↵	35.4 ↵	38.2 ↵	**40.3** ↵	39.3 ↵
Gain (%p) ↵	-5.8↵	0 ↵	+2.8 ↵	+4.8 ↵	+3.9 ↵

Table 2: Performance comparison of methods between data with/without text data. The best performance is denoted in bold. ↵



Pre-processing ↵	**Resnet 18 with merging** ↵	**Resnet18 mean pooling (mid)**	**Resnet18 mean pooling (end)** ↵
Without (%) ↵	38.2 ↵	40.3 ↵	39.3 ↵
With (%) ↵	41.8 ↵	**43.5** ↵	43.4 ↵
Gain(%p) ↵	+3.6 ↵	+3.3 ↵	+4.1 ↵

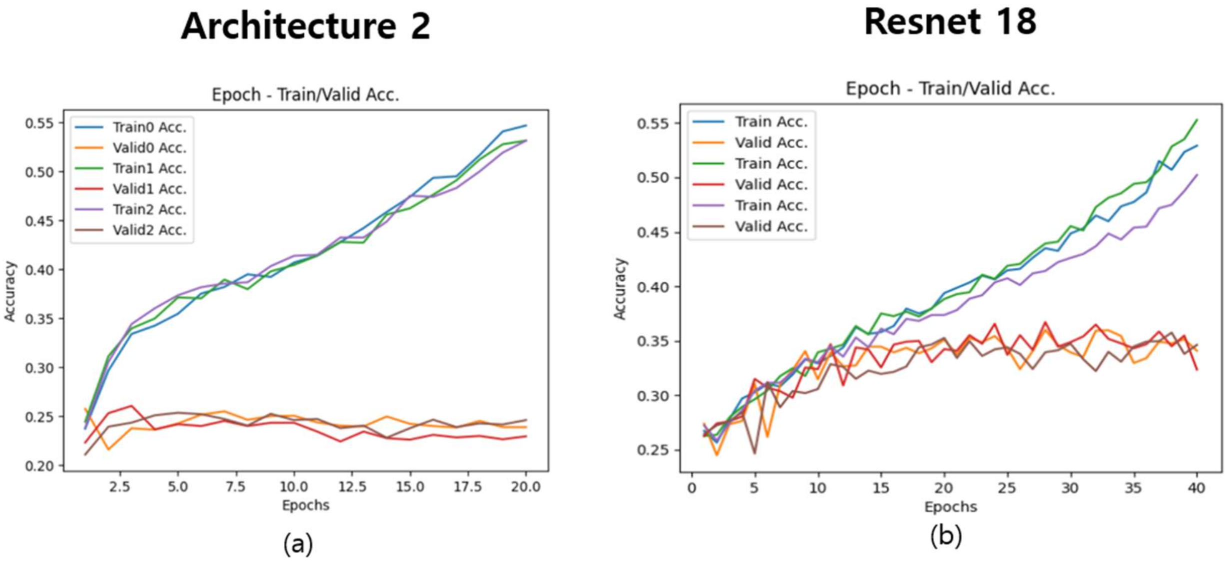
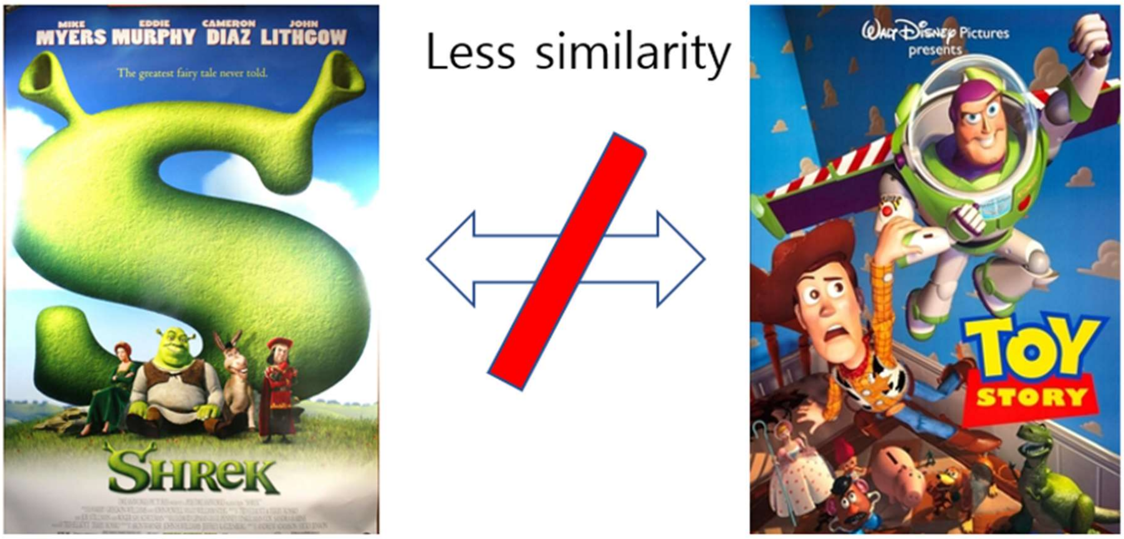


Figure 9

7.1 Conclusion

We presented architectures for classifying movies by genre with movie poster image data and embedded text data in image data. Text data extracted from image not always can be used meaningful for model training. So, we pre-process text data that was meaningless for model training. We construct three architectures that merge two vectors for considering both image vector and text vector. Also, we construct the way to calculate multi-labeled loss and accuracy for many diverse movie genre labels. Compared to the baseline model that used only image data, the model accuracy was increased, and when we used pre-processed text data, the model accuracy was increased additionally. Especially, architecture 2 showed the best performance and it means training factor from text data have a great importance. We can conclude that applying both the image data and the embedded text data to training model is meaningful for the classification model. Also, meaningful word embedding works positively to classification model.

For the bad side, despite of our evaluation metric is top-1 accuracy, our best test accuracy doesn't get over 45%. It seems low considering it is top-1 accuracy. There may be the several reasons for it such as data was too little or etc. We thought the most essential reason is irrelevance with movies in same genre. As you can see in **Figure 10** movie Shrek and Toy Story are in same genre 'Animation' but have less similarity in both image features and text data. This irrelevance seems to make hard for train data to get correct prediction for valid or test data.



MIKE MYERS EDDIE MURPHY CAMERON DIAZ JOHN LITHGOW
The greatest fairy tale never told.
SHREK
Less similarity
Walt Disney Pictures presents
TOY STORY

Figure 10↵

7.2 Case Study

Figure 9-(b) is performance of baseline model (only use CNN) and its average accuracy is 35.4%. However, **Figure 9** - (a) is performance of architecture with randomly generated text data and its average accuracy is 23.5%. It shows us meaningful embedded text data increases model performance while meaningless embedded text data decreases model performance.

7.3 Future Works

Our model related with classification movie genres by movie posters, and it proves it is effect that if image contains meaningful embedded text, it can help improve model classify accuracy. Not only classification of movie genres, but our model can show better performance of any case for classify problem with valid image and valid embedded text.

[image from 'https://en.wikipedia.org/wiki/Aposematism']

Especially, in the case of classification of animals with protective colors as in the above example, it is difficult to predict the correct answer only with image data. The reason why it is hard to find features from image data. However, if image data have embedded text data, our system can improve their classification model by applying valid text data.

We still have some points that should be improved. Our architectures just merge vectors and mean pooling them. In fact, it shows better model performance, however we think there are more effective way to merge vectors (like max pooling or something additional post-processing). We hope someone will reinforce our experiment by that way.

. REFERENCES

[1] Zeeshan Rasheed and Mubarak Shah. 2002. Movie Genre Classification by Exploiting Audio-Visual Features of Previews. In Proceedings of International Conference on Pattern Recognition.

[2] Simos, G. S., Wehrmann, J., Barros, R. C., & Ruiz, D. D. (2016). *Movie genre classification with convolutional neural networks*. In 2016 International Joint Conference on Neural Networks (IJCNN) (pp. 259-266). IEEE.

[3] He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*.

[4] Karen Simonyan, Andrew Zisserman. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition.*

[5] Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[6] Gabriel Barney and Kris Kaya. (2019). Predicting genre from movie posters. Stanford CS 229: Machine Learning

[7] Wei-Ta Chu, Hung-Jui Guo.(2017) *Movie Genre Classification based on Poster Images with Deep Neural Netwo*rks

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

Contribution of Each team member

백일웅: image data fitting, CNN model implementation, propose methods (3 architectures), week presentations, ppt, final report

김우진: Getting text data from OCR, Text data preprocessing, final report and ppt related to text data, 마지막 발표

손민혁: proposal, weight visualization, ppt, pre-processing imdb dataset

조현준: image data preprocessing, ppt, text data preprocessing