Story Classification
A new approach for Movie, Anime, Novel and Series in Traditional Chinese

***Abstract*—Our goal is to inference which story it is when giving a sentences describing the plot of story. It is too difficult to answer without limitation, so we see this problem as a classification problem giving the predefined choices and the model will return the probability of each choice. To deal with such classification problem of NLP domain, we introduce the BERT model [1] which is a very famous pre-trained model . And fine tuning the bert-base-chinese model to handle this task. We use web crawler in ptt movie to obtain the posts which is related to target movie and performing some process to obtain corpus with label. To evaluate the performance, the model should correctly classify the plot in wiki pedia inside the range of choices and predict ”else” when giving the plot not in our choices. This may not be a good criterion because the posts written by other people might copy the plot in wiki pedia. To avoid the risk, when we pre-process the data we will mask out all the sentences which are entirely copied from the wiki pedia. In the end, we want to build a good model not only good at distinguish where the plots come from when describing it in different ways, but a model able to tell the sentences talking the same things. And also provide the other possible choices with given plots via probability.**

I. Introduction

Sometimes we will conjure some interesting plot or story, but we can’t recall where did we see or when did we read it. So we are seeking a stable method to find what story it is. Currently, we usually turn to the help of search engines for example: Google, Edge, Firefox or more fancy tool like the AI agent based on GPT-3. But these method have a big fallacy when encountering such problems. They only identify with exactly same words if the sentences don’t contain the ”keyword” they can’t perform well. However, the description of one story is very subjective which could vary by the choices of words decided by the writer. So, if your description is differ from other online resources, search engines are not able to return useful information. Thus, we want to develop a model/algorithm which can identify where this plot or story come from that not depending on the choices of words but the main idea or meaning behind sentences. In other word, current search engines are based on the ”keyword” of sentences. We want to build another model to search/classify with the ”content” of sentences.

(If you search the plot contain some keyword in English on Google, they may perform well. But it result not so well when search the plot in Traditional-Chinese without powerful keyword)

II. Related Knowledge

- 1.deep learning
- 2.neuron network
- 3.NLP
- 4.word embedding
- 5.self-attention layer
- 6.transformer
- 7.BERT
- 8.data augmentation
- 9.transfer learning

III. Related Work

This is basically a classification problem combined with some text understanding. Thus, other pre-trained model like ELMO [2], GPT [3], XLNet [4] can be used to solve this problem as well.Or maybe solving this problem in another angle.For example, Keyword extraction combined with search engine.If the keyword isn’t power enough, replacing the keyword to other keywords with same meaning.Then, using searching engine to acquire the answer.For example,currently, Google are trying to replace the keyword-search to semantic-search [5] [6] . When talking about semantic search, currently there are the models use similar idea which are the ”GPT models” [3] like ChatGPT. However, they are not specified for the story task. So, even they can make some useful responses or related answers, most of the time they just reply some plausible sentences.

IV. Plan and Schedule

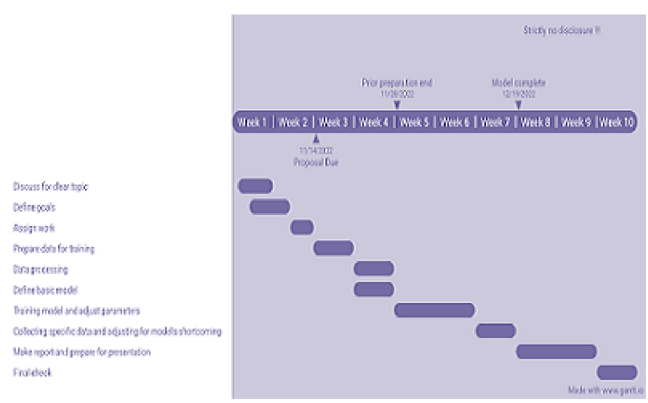

Fig. 1. gantt of schedule.

based model should be finished in 11/28 and complete finished in 12/19

V. Data used

1.Rough preprocess:just chunk the corpus when encounter / n, and clear all image(ex:http://imgur......) and constrain the sentence length to (6~80) Because our goal is to make the prediction model invariant to key word, so it's necessary to build a model that can output correct result without key-information(ex:name of character. So we try some tricks. See 4.).

2.Training Data: Use web crawler to collect the data variant to our predict-targets from PTT(https://www.ptt.cc/bbs/)So we just collect other people's posts with the title which mention our predict-targets.

3.Validation Data: Because we don't find systematical corpus to used, we Manually collect from many website and roughly scan the correctness while collecting the data.

(ex:https://zh.m.wikipedia.org/ https://baike.baidu.com/ and many other sites)

4.Mask Names:To mask character's name, which is related to POS (part-of-speach) and NER (named-entity-recognition) problems. We have tried two method. First,we use JIEBA model to do NER. Second, we try CKIP CoreNLP(which also based on BERT) provide by CKIP Lab resulting much better performance than JIEBA. (https://github.com/fxsjy/jieba) (https://github.com/ckiplab/ckipnlp)

VI. Experiment

0.Selection of Model:Pre-train model play an important role for NLP task. Currently, BERT, XLNet, ERNIE, PERT are some powerful and available choice. However, these models are usually available for simplified Chinese. The only choice for traditional Chinese we found, is the BERT model provided by CKIPLab

(https://huggingface.co/ckiplab/bert-base-chinese?)

Because BERT is a powerful pre-train model, which has large capacity to accomodate training data. It's easy to reach low loss and 100 percent accuracy for training process. So, it's more crucial to take care of the generalization of model. Despite the build-in regularize technique in the BERT model.Here we introduce 1.Early stop 2.Drop out 3.Data augmentation 4.SAM 5.Froze some layer from pre-trained

1.Early Stop: Maybe because the validation data isn't large enough, the results are a little bit noising, Here we just follow the basic algorithm, record the epoch with largest validation accuracy and for further training with validation data.

2.Drop Out: Drop out is a common technique to regularize model. Though it's usually recommend to set drop out to 0.5. Whereas, 0.5 might be too large for this problem that model cannot learn under this configuration. Through trial and error we think 0.1 to 0.4 might be an acceptable range for this problem and 0.25 usually perform relatively great.

3.Data Augmentation:Data augmentation usually used in image domain. Here we try to see if this technique can be applied to NLP domain. Because the name of character can be strongly connect to the target story, We try to block this information to train more robust model. Here we try two strategies see Fig.2. Fig.3. Fig.4. Strategy1: keep the character and plot information but anonymous, Strategy2: only keep plot information. Strategy2 performs slightly better, maybe because the strategy1 try to memorize the anonymous tokens which won't occur in validation and take relatively less attention to plot information.

4.SAM(Sharpness awareness minimization): This is a new technique introduced for regularization [7]. Based on the belief that convergence in flat area of training data will result in similar performance in testing data. We try to introduce this method into our project and increase around 3% accuracy for validation. And produce not-so-judgemental model which provides more information to the user.

5.Freeze particular layers: Pre-trained model have more generalized weight from the early unsupervised training stage. So, the weight itself hold some generalized information. However, this generalization may be corrupted during transfer learning, So it is important to keep some weight unchanged . Word Embedding is especially crucial for this project, which provide similar hidden input to next layer for synonyms. Whereas, the target specific tranfer learning might mess up the original semantic space from pre-train. So it's necessary to freeze the word embedding layer of pre-trained model, Which can improve 5% ~ 10% accuracy of validation through out our experiment. (Freeze other layers didn't see that much improvement)

Note:Although we use val data as an indicator of performance but in final work we will put val data into training because they contain more precise information compared to train data. We believe the model will become better after apply validation data, although we don't have data to prove it. So, the performance on validation data excluded might be seen as a lower bound of the final model

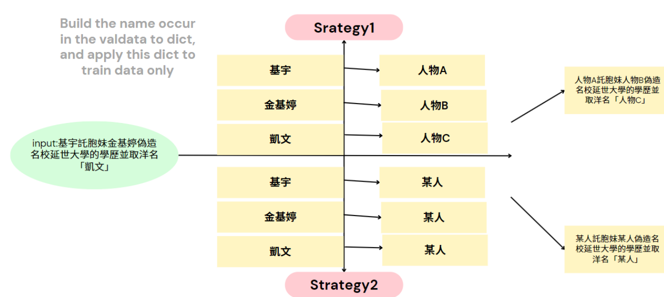

Fig. 2. replace strategy 1

VII. Results

We train five model for different task. Each of first four train with 8 different target, while last one is the combination of first three targets as illustrate in Fig.5. (Novel model use novel itself as training data and PTT posts as validation data, it's quite different from other so we don't include in total model) In the begin, we were wondering whether it work. Because the posts contain many unrelated content to the target and sometimes only a little part mention about our target. Surprisingly, in

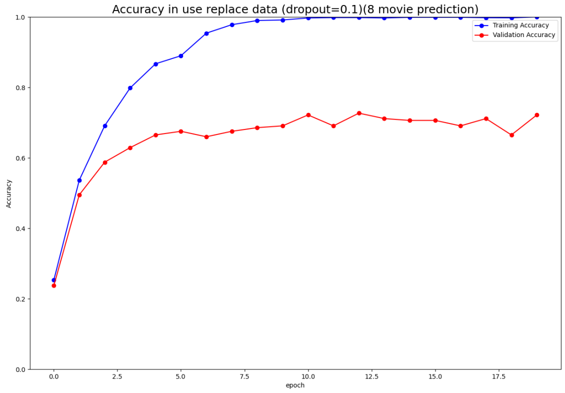

Fig. 3. replace strategy 1

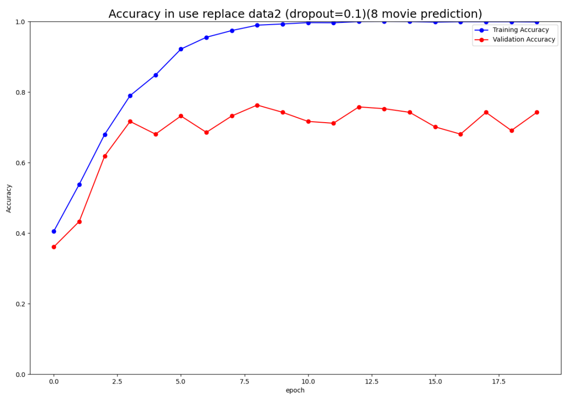

Fig. 4. replace strategy 2

our first try, the movie task, it easily reach 70% accuracy of validation. Thus, we know this architecture might work. So we can proceed and dig into more detail to improve this model.

We use some manually create testing sentences to compared total model with search engine look Fig.6. We can see that although our model may fail sometime but it can still provide some information by giving possible choice.

However, we still face some problem. First, some targets are rarely pop out even we use some related test sentence. Maybe caused by the problem of unbalancing data. And this problem become more severe when training with different type

MODEL TYPE	TARGETS' NUM	TRAIN ACC	VAL ACC
MOVIE	8	0.854	0.802
SERIES	8	0.812	0.442
NOVEL	8	0.957	0.751
ANIME	8	0.955	0.500
MOVIE+SERIES+ANIME	24	0.912	0.307

Fig. 5. model accuracy

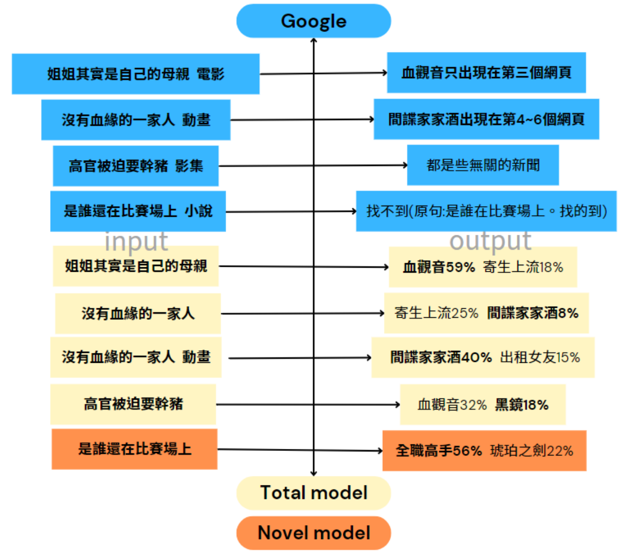

Fig. 6. input and output(SPOILER ALERT) (bold for ground Truth)

of story. For example, series have times of content more than movies(10 ˜ 50 hours compare to 2 ˜ 3 hours). However, due to the lack of indicator or proper criterion, we haven't find a good approach to handle this problem but only try to keep the training data balance by random delete. Second, synonymous problem. Although the model we use is an traditional-Chinese version of BERT. The word with same meaning still lead to different result(ex:"older sister" in Chinese there are two way to write)Maybe using BERT-wmm [8] can provide better result. But the current provided pre-train model is based on simplified Chinese.

VIII. Summary

1.Foresight: In the past time, we use the prediction of word to train the word embedding layer, and use the relation of sentences to train the BERT model. We hope this project can achieve the similar effect like training the text features extraction of NLP model

2.SAM For Smoothing: In the begin, we didn't include this method in our design. However, the model train with SAM resulting some smoothing effect.The model would provide more information by not-so-judgemental predictions (kind like the result of label smoothing). The convergence in flat area will reduce the probability of True answer but increase the probability of related answer. So, even the prediction fail, user still can get insight by the high-possibility choices.

3.Advantage: Although this architecture don't have dominant accuracy compared to the current search engine. It still have some advantage. Fist, it can provide more information to user. While giving rough description which appears in many story to the model, it will return some possible choices based on the plot. However, search engine usually give the most possible answer or sometimes most popular answer with the website related to it. Second, memory-oriented. Human memory usually based on the Episodic memory which contain the full plot and scene. However, we often forced to subtract only some keyword of your memory to fit in search engines, many information loss in this stage. So, our models are able to facilitate full human memory

4.Ability to Combine: Because the emerge of ChatGPT, We think the GPT model have the ability to roughly create the testing data for our task or generate the full plot based on the given input and predicted answer. However, most of our targets are not included in ChatGPT, because it only train with the data before 2020 while most of our targets are new movie.

5.Future(Expanding Size): Because not all the target story have sufficient source of text, maybe we can introduce the method like [9]for converting video source to it's story text or [10] provide caption of video which may perform similar effect as text of story.

Here is our github repo(doesn't contain the .pt file,but with the data and ipynb to train from scratch): https://github.com/BlueDyee/Story-classification

IX. References

References

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

[2] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[4] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019.

[5] Ramanathan Guha, Rob McCool, and Eric Miller. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, pages 700–709, 2003.

[6] Hannah Bast, Björn Buchhold, Elmar Haussmann, et al. Semantic search on text and knowledge bases. *Foundations and Trends® in Information Retrieval*, 10(2-3):119–271, 2016.

[7] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization, 2020.

[8] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514, 2021.

[9] Jesus Perez-Martin, Benjamin Bustos, Silvio Jamil F. Guimarães, Ivan Sipiran, Jorge Pérez, and Grethel Coello Said. A comprehensive review of the video-to-text problem, 2021.

[10] Mingkang Tang, Zhanyu Wang, Zhenhua Liu, Fengyun Rao, Dian Li, and Xiu Li. Clip4caption: Clip for video caption, 2021.

X. Members

Hsiao Teng-Fang: Architecture design, Training model.

Ni Zi-Xiang: Date crawler, Data pre-process, Data augmentation