Fine-tuning techniques and data augmentation on transformer-based models for conversational texts and noisy user-generated content

Mike Tian-Jian Jiang
Zeals Co, Ltd.
Tokyo, Japan
tmjiang@gmail.com

Shih-Hung Wu
Department of CSIE
Chaoyang University of Technology
Taichung, Taiwan
shwu@cyut.edu.tw

Yi-Kun Chen
Department of CSIE
Chaoyang University of Technology
Taichung, Taiwan
kun26712930@gmail.com

Zhao-Xian Gu
Information Management
Tamkang University
New Taipei City, Taiwan
406630136@s06.tku.edu.tw

Cheng-Jhe Chiang
Information Management
Tamkang University
New Taipei City, Taiwan
406630649@s06.tku.edu.tw

Yueh-Chia Wu
Information Management
Tamkang University
New Taipei City, Taiwan
406630102@s06.tku.edu.tw

Yu-Chen Huang
Information Management
Tamkang University
New Taipei City, Taiwan
406630193@s06.tku.edu.tw

Cheng-Han Chiu
Information Management
Tamkang University
New Taipei City, Taiwan
406630284@s06.tku.edu.tw

Sheng-Ru Shaw
Information Management
Tamkang University
New Taipei City, Taiwan
poppydumb1220@gmail.com

Min-Yuh Day †
Information Management
National Taipei University
New Taipei City, Taiwan
myday@gm.ntpu.edu.tw

Abstract— **Transfer learning and Transformer-based language models play important roles in modern natural language processing research community. In this paper, we propose Transformer model's fine-tuning and data augmentation (TMFTDA) techniques for conversational texts and noisy user-generated content. We use two NTCIR-15 tasks, namely the first Dialogue Evaluation (DialEval-1) task and the second Numeral Attachment in Financial Tweets (FinNum-2) task, to evaluate the efficacy of TMFTDA. Experimental results show that TMFTDA substantially outperforms the baselines model of Bidirectional Long Short-Term Memory (Bi-LSTM) in multi-turn dialogue system evaluation at DialEval-1's Dialogue Quality (DQ) and Nugget Detection (ND) subtasks. Moreover, TMFTDA performs to a satisfactory level at FinNum-2 with a model of Cross-lingual Language Models using a Robustly Optimized BERT Pretraining Approach (XLM-RoBERTa). The research contribution of this paper is that, we help shed some light on the usefulness of TMFTDA, for conversational texts and noisy user-generated content in social media text analytics.**

Keywords*— *conversational texts, data augmentation, fine-tuning techniques, noisy user-generated content, transfer learning, Transformer-based models

I. Introduction

Because of recent advances in Transfer learning and Transformer-based language models, which play important roles in modern natural language processing research community, more and more researchers and engineers are developing task-oriented dialogue systems. Customer services may benefit from such a deep-learning-based chat-bot that responses to inquires 24/7. Assessing systems like that, however, often involves a labor-intensive hence costly annotation process that may defeat the purpose. The dilemma motivates the task organizers, of the third Short-Text Conversation (STC-3) [38] task at NTCIR-14 and the first Dialogue Evaluation (DialEval-1) [39] task at NTCIR-15, to examine automatic evaluation systems for helpdesk conversations, in either Chinese or English. Thus, they come up with Dialogue Quality (DQ) and Nugget Detection (ND) subtasks.

The DQ subtask uses subjective scales that quantify the quality of a whole dialogue. With 5-degree of rank each sorting from -2 to 2, the organizers define 3 score types:

1. A-score: Accomplishment

   —to what extent has an inquiry resolved;

2. S-score: Satisfaction

   —how assured a customer is with the conversation;

3. E-score: Effectiveness

   —how helpful and economical a dialogue is.

The ND subtask first defines what kind of dialogue turn is a nugget, determines whether it belongs to Customer side or Helpdesk side, and finally categorizes it into seven types of four groups:

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 © 2020 IEEE

1. CNaN / HNaN: Customer or Helpdesk's non-nuggets that are irrelevant to the problem-solving situation;
2. CNUG / HNUG: Customer or Helpdesk's regular nuggets that are relevant to the problem-solving situation;
3. CNUG* / HNUG*: Customer or Helpdesk's goal nuggets that confirm and provide solutions, respectively;
4. CNUG0: Customer's trigger nuggets that initiate a dialogue with certain problem descriptions.

Based on the above specifications, we formulate the DQ and the ND subtasks as a multilabel classification problem and a multiclass classification problem, respectively. Since STC-3 participants didn't outperform the baselines model of Bidirectional Long Short-Term Memory (Bi-LSTM) [3,4,14,34], we take on the challenge to discover another strong baseline. To alleviate the high cost of architecture engineering and model training, our study pays more attention to tokenization and optimization for transfer learning. We apply well-established techniques of tokenization and fine-tuning to pretrained Transformer models. We find that some specific combinations of techniques work well with Cross-lingual Language Models using a Robustly Optimized BERT Pretraining Approach (XLM-RoBERTa) [5] and certain variations of Bidirectional Encoder Representations from Transformers (BERT) [7], for English and Chinese, respectively.

For DQ, the task organizers measure performance by Normalised Match Distance (NMD) and Root Symmetric Normalised, Order-aware Divergence (RSNOD). For ND, the metrics are Root Normalised Sum of Squares (RNSS) and Jensen-Shannon Divergence (JSD). In terms of NMD, our run2 for Chinese DQ subtask substantially outperforms the baselines. According to RSNOD, our run0 for English DQ subtask also achieve a significant difference of S-score statistically. Almost all of our runs for ND tasks reach the first places at DialEval-1. Those results suggest that one can easily optimize Transformers for DQ and ND subtasks.

The second Numeral Attachment in Financial Tweets (FinNum-2) [42] task, on the other hand, is also a shared task hold in NTCIR-15 conference. Its goal is to analyze the association between numbers and stock names in financial tweets [41]. Domain experts annotate these tweets for stock names, numbers, and their relevance. In FinNum-2 dataset, a tweet must have at least one pair of a target numeral and a cashtag of stock name, and each data instance represents a different pair. So, it naturally becomes a binary classification problem on whether the target numeral applies to the given cashtag or not.

II. Related Works

A. Dialogue system evaluation (DialEval-1)

In the past, researchers have relied on human to judge the quality of a dialogue system [1]. To overcome the inefficiency and the inconsistency of manual assessments for spoken dialogue agents, PARADISE, one of the earliest works on learning an automatic evaluation function, isolates task requirements from an agent's conversational behavior, at the cost of measurable completeness and complexity of the task [31]. Since the measurement is not always available, a recent model called ADEM seeks to learn and predict the appropriateness of utterances [25]. ADEM and its successors keep evolving to adopt one new model by another: Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) [9], and now BERT. It is then conceivable that many STC-3 participants have used LSTM or BERT. As one may argue that Bi-LSTM usually outperforms other architectures [8], STC-3 outcomes also suggest the bar set by a model of Bi-LSTM and GloVe [26] is uneasy to meet.

Despite the architecture differences, almost all of the participants have modeled the ND and DQ subtasks as classification problems. We adopt the same tactic for DialEval-1, such that our efforts may focus on developing a recipe of transfer learning that comprises the state-of-the-art ingredients. For that matter, we look into various works of transfer learning, especially on optimization algorithms and loss functions. Layer-wise Adaptive Rate Scaling (LARS) [36] aims to implicitly adapt various learning rates for different layers of convolutional networks with large batches, and soon spawns a version called LAMB [37] for BERT training. As the name suggests, however, they are designed for relatively big batch-sizes for the efficiency of pretraining, and we fail to find significant improvements using them for fine-tuning. The fact that we're already using discriminative fine-tuning, which we will describe in a latter section, that also sets various learning rates, may further complicate the behavior of convergence.

Another perspective on taming the behavior of convergence is about stabilizing gradient updates. Lookahead [40], Rectified Adam [20], and Gradient Centralization [35] fall into this category. Ranger further combines them together as one optimizer. Again, based on our pre-trials for the DQ and ND subtasks, they are neither faster nor stabler.

Last but not least, if we see the tokenization tricks as feature engineering for deep neural networks, whilst being seldom used for text classification and fine-tuning, it is a common approach for text generation and pretraining. CTRL [15] and GPT-3 [2] have many designated "prompts" that enable conditioned generations. Feature engineering done in such a preprocessing manner may be easier for adapting different tasks or pretrained models than specialized embeddings.

B. Numeral attachment in financial Tweets (FinNum-2)

In the first Numeral Attachment in Financial Tweets (FinNum-1) task at NTCIR-14, most works use word/character embeddings to represent token information of tweets [41], such as Skip-grams, GloVe [26], ELMo, and BERT [7]. One BERT model pretrained with Microsoft Research Paraphrase Corpus (MRPC) has obtained the best performance. Consequently, the result of FinNum-1 inspires us to further explore recent Transformer models pretrained with different datasets and tokenization schemes.

III. Proposed Methods

Figure 1 shows our research framework of Transformer model's fine-tuning and data augmentation (TMFTDA) techniques for conversational texts and noisy user-generated content. Firstly, we establish our tool-chain. To go through the

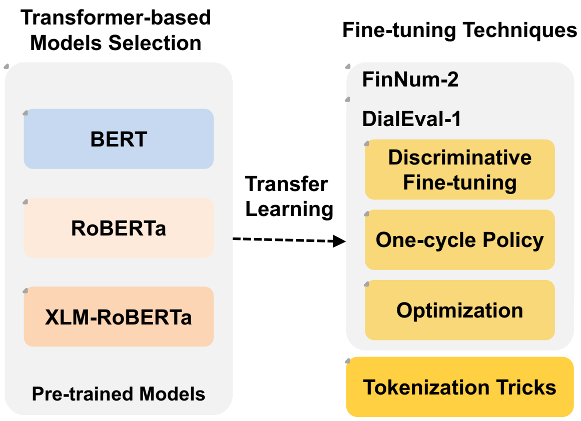


Fig 1. Research framework of Transformer model's fine-tuning and data augmentation (TMFTDA) techniques for conversational texts and noisy user-generated content

trial-and-error phase as quick as possible, we only try pretrained models available on HuggingFace's Transformers [32], and use fastai [10,11] to control the quality and the speed of transfer learning. We introduce model specifications and training procedures that are conceptually related to multilabel and multiclass classifications for DialEval-1's DQ and ND subtasks, as well as binary classification for FinNum-2 task.

A. Selection of Transformer models for transfer learning

We conduct transfer learning by fine-tuning pretrained BERT, RoBERTa, and XLM-RoBERTa models for text sequence classification. To meet our goal of rapid experimentations, all pretrained models are the base versions. For Chinese DQ and ND subtasks, we test the official one (denoted as bert-chinese when necessary) and a whole-word masking version (bert-chinese-wwm) [6] of BERT. The official XLM-RoBERTa model (xlm-roberta) runs for both Chinese and English. Finally, the runs of the official RoBERTa model (roberta) [21] and the case-reserved BERT (bert-cased), are merely control groups for the English ND subtask. The principle behind the choices is simple: they cover representative differences of the pretraining scheme and the token specification.

BERT by default tokenizes each input sequence using WordPiece [33]. Its pretraining typically relies on two objectives: masked language modeling (MLM) and next sentence prediction (NSP). The former requires the model to predict tokens that have been randomly masked in a 15% chance per input sentence, and the latter demands the model to predict whether two randomly concatenated sentences are actually adjacent to each other or not. XLM-RoBERTa, on the other hand, combines and revises techniques of cross-lingual language model (a.k.a. XLM) pretraining schemes [19] and a robustly optimized BERT pretraining approach (a.k.a. RoBERTa). In terms of optimization, RoBERTa builds on BERT and modifies key hyperparameters such as the MLM objectives, removing the NSP objective and training with much larger mini-batches and learning rates. As for tokenization, it differs from BERT by using a byte-level Byte Pair Encoding (BPE) [28] as a tokenizer, and dynamically changing the masking pattern applied to the training data. XLM-RoBERTa follows most of XLM approaches, except it removes language embeddings for a better code-switching ability. It also differs from RoBERTa by tokenizing with unigram-level sentencepiece [17,18] instead of BPE.

B. Tokenization tricks for Transformer models

To better represent the structure of a dialogue, using XLM-RoBERTa's markups as example, we not only utilize special tokens for the beginning of a sentence (`<s>`), the end of a sentence (`</s>`), and the separator of sentences (`</s> </s>`), but also customize a couple of tokens in the fastai convention of "`xx`" prefix that provides context, which is probably one of the simplest form of data augmentation, dubbed as Tokenization Tricks hereafter. For example, consider a tokenized turn below:

```
xxlen _3 <s> xxtrn _1 xxsdr _customer _@
_China _Uni com _Customer _Service _in
_Gu ang dong ⋯ _Middle _Road . </s>
```

The special tokens `xxlen` and `xxtrn` stand for length of the dialogue in turns and the position of each turn of the dialogue, respectively. The numbers right next to them provide certain features of turns. The same trick goes with `xxsdr` that differentiates whether the sender is Customer or Helpdesk. When a turn's context says "`xxtrn _1 xxsdr _customer`", the nugget type is almost definitely CNUG0. As for DQ, a whole dialogue can be tokenized in a similar fashion, where `xxlen` could be useful for certain quality scores that may implicitly involve the time/turns spent on resolving an inquiry:

```
xxlen _3 <s> xxtrn _1 xxsdr _customer _@
_China _Uni com _Customer _Service _in
_Gu ang dong ⋯ _Middle _Road . </s> </s>
xxtrn _2 xxsdr _help desk _Hello ! ⋯
_Thank _you ! </s> </s> xxtrn _3 xxsdr
_customer _The _Uni com ⋯ _No _phone
_call _is _answered ! </s>
```

Although we don't apply the default tokenizer of fastai, it might be worthwhile to explain what it is and why we don't use it. The fastai convention of "xx" prefix denotes special context tokens. By default, fastai tokenizes English texts using SpaCy and inserts special tokens before uncapitalized or originally repeated words/characters . For instance, consider the following utterance from the test set:

```
… Beijing Unicom Unicom still …
```

If we apply fastai's default tokenization to it, the outcome will have "`Unicom Unicom`" converted into "`xxwrep 2 xxmaj unicom`" for title case and word duplication simultaneously. As lossless as the conversion may be, since pretrained Transformer models are unaware of those special context tokens, we must ask whether they can still help fine-tuning for a specific task or not. In our opinions, if the task were sentiment analysis of utterance, repetitions and capitalization could be important clues. However, it is hard to imagine that the recurring word/character can help semantically or syntactically, not to mention that XLM-RoBERTa already preserves letter cases of subword tokens. Based on the above observations, we don't apply them for the DialEval-1 subtasks.

C. Fine-tuning techniques on downstream tasks

We adopt recently advanced fine-tuning techniques as much as possible. Some of them are originally designed for AWD-LSTM and QRNN by ULMFiT [22,23], such that we must assess their usefulness for XLM-RoBERTa. Based on our preliminary tests, discriminative fine-tuning and fastai's version of one-cycle policy work well, but graduate unfreezing produces little effect, which is consistent with the findings of similar studies [13,27]. Techniques other than the above mainly involve choosing the most promising combination of optimization algorithms and loss functions. For the FinNum-2 task in a binary classification setting, we find none of more recent optimizers and loss functions work better than Adam optimizer with class weights. We will list configuration values of finally used techniques in the next section of experiments.

1) Discriminative fine-tuning

As different layers may capture various types of information, we shall fine-tune them to different extents. Instead of using the same learning rate for all layers of a model, discriminative fine-tuning enables us to tune each layer with different learning rates. We use blurr to split the model layers into groups automatically corresponding to different model architectures. For both BERT and XLM-RoBERTa, it results in four groups: the top layer of classifier, the pooling layer, the Transformer layers, and the bottom layer of embeddings. Intuitively, the lower groups may contain more general information while the higher ones contain more specific information. Therefore, we set a base learning rate for the top group and then assign linearly decreased learning rates per lower groups.

2) One-cycle policy

A cycle wraps an arbitrary number of epochs for sharing the same policy of hyperparameters, especially for learning rates and momentums. For training a deep neural network with stochastic gradient decent or similar algorithms, a policy of cyclical learning rates, meaning it periodically increases for a step size and then decreases the learning rates, may converge faster and better [29,30]. In addition, the fastai version of the One-cycle Policy comprises three complementary techniques that balance the trade-off between fast convergence and overshooting. The Slanted Triangular Learning Rates (STLR) [12] and the Cyclical Momentum (CM) [29,30] allow us to micro-manage iterations/updates within a cycle, whereas changing the maximum learning rate (max_lr) per cycle let us control the quality of each. Empirically, STLR and CM together work best when they simultaneously change in a reversed direction. In other words, SLTR uses a warm-up and annealing for the learning rate while CM does the opposite. As for the macro-management per cycle, we apply a simply decay on their max_lr's.

3) Other optimization schemes

We test several optimizers and find none of them improve the convergence stability significantly than Adam [16]. For the choice of loss function, we realize that the label smoothing function [24] suits multilabel/multiclass classification better than typical cross-entropy one.

Table 1. Configurations of DialEval-1 Official Runs

Task	Lang.	Run	Model	B.	Recipe
DQ	en	0	xlm-roberta	12	a
	zh	0	xlm-roberta		
		1	bert-chinese-wwm		
		2	bert-chinese		
ND	en	0	xlm-roberta	12	b
		1	bert-cased	8	c
		2	roberta	24	d
	zh	0	xlm-roberta	12	e
		1	bert-chinese-wwm	8	f
		2	bert-chinese	16	d

IV. Experimental Results and Discussions

We present our experimental results on DialEval-1's DQ and ND subtasks that examine automatic evaluation systems for helpdesk conversations in both Chinese and English, along with FinNum-2) task for numeral attachment in financial tweets.

A. DialEval-1

Table 1 shows the mapping between our official runs, the designated models, the batch sizes (B), and the recipes of hyperparameters. Important hyperparameters include the cycle schemes and their max_lr's of discriminative learning rates. Every cycle contains just one epoch. Discriminative learning rates share the same reduction rate: the lower bound is always max_lr/1000.

a. 2e-3 for three cycles, 1e-3, 5e-4, 1e-4;

b. 3e-4;

c. 1e-3;

d. 1e-4, 1e-5, 1e-6;

e. 3e-4, 1e-4;

f. 6e-4.

The factor of 1000 hints that we hope the four layer-groups may roughly have the rates distributed evenly. However, it comes to our attention that, after the timing of the official runs, the version 3.3.0 and above of HuggingFace's Transformers has removed the pooling layer from classification-oriented models, because in theory they are unrelated. Should any reader want to reproduce the outcome, please be advised that it will definitely vary if using different versions.

Table 2 shows the Chinese ND results of our official runs. Table 3 shows the English ND results of our official runs.

Table 2. Chinese Nugget Detection Results

Run	JSD	Run	RNSS
IMTKU-run0	0.0674	IMTKU-run0	0.1636
BL-lstm	0.0709	BL-lstm	0.1673
IMTKU-run1	0.0726	IMTKU-run1	0.1700
IMTKU-run2	0.0752	IMTKU-run2	0.1754

Table 3. English Nugget Detection Results

Run	JSD	Run	RNSS
IMTKU-run0	0.0707	IMTKU-run0	0.1699
IMTKU-run2	0.0757	IMTKU-run2	0.1753
BL-lstm	0.0762	BL-lstm	0.1781
IMTKU-run1	0.0789	IMTKU-run1	0.1804

In ND for both Chinese and English, corresponding run0 results of XLM-RoBERTa are only slightly better than the LSTM baselines. For that matter, we closely examine the outcomes and then notice intriguing phenomenon, such as

“Are you from a security software manufacturer?”

and

“Do you think if it would be better for me to complain to the Ministry of Industry and Information Technology?”

of IDs 4245108926487325 and 4392549047578258, respectively. The turn types like the above examples are mostly CNaN, but the models predict them as CNUG. We anticipate that the word "you" has caused confusions. The models might have taken it literally for Customer replying to Helpdesk, but the turns and similar are likely sarcasm hence unrelated to the problem-solving situation.

For DQ, we manually compare the differences among models for different runs. Table 4, 5, and 6 present the A-score, E-score, and S-score of English DQ results of IMTKU official runs. Table 7, 8, and 9 present the A-score, E-score, and S-score of Chinese DQ results of IMTKU official runs. Although the Chinese versions of BERT outperform XLM-RoBERTa, they all share the same recipe of cycle schemes. In addition, since we know that the English datasets are translations of the Chinese ones, it is as expected that XLM-RoBERTa appears equally competitive for both languages.

Table 4. English Dialogue Quality (A-score) Results

Run	RSNOD	Run	NMD
IMTKU-run0	**0.2197**	IMTKU-run0	**0.1437**
BL-lstm	0.2271	BL-lstm	0.1591

Table 5. English Dialogue Quality (E-score) Results

Run	RSNOD	Run	NMD
IMTKU-run0	**0.1657**	IMTKU-run0	**0.1221**
BL-lstm	0.1687	BL-lstm	0.1248

Table 6. English Dialogue Quality (S-score) Results

Run	RSNOD	Run	NMD
IMTKU-run0	**0.1892**	IMTKU-run0	**0.1250**
BL-lstm	0.2111	BL-lstm	0.1413

Table 7. Chinese Dialogue Quality (A-score) Results

Run	RSNOD	Run	NMD
IMTKU-run2	**0.2130**	IMTKU-run2	**0.1392**
IMTKU-run0	0.2165	IMTKU-run0	0.1406
IMTKU-run1	0.2204	IMTKU-run1	0.1442
BL-lstm	0.2305	BL-lstm	0.1598

Table 8. Chinese Dialogue Quality (E-score) Results

Run	RSNOD	Run	NMD
IMTKU-run1	**0.1631**	IMTKU-run1	**0.1165**
IMTKU-run0	0.1648	IMTKU-run0	0.1181
IMTKU-run2	0.1655	IMTKU-run2	0.1194
BL-lstm	0.1782	BL-lstm	0.1386

Table 9. Chinese Dialogue Quality (S-score) Results

Run	RSNOD	Run	NMD
IMTKU-run2	**0.1918**	IMTKU-run2	**0.1254**
IMTKU-run1	0.1964	IMTKU-run1	0.1284
IMTKU-run0	0.1977	IMTKU-run0	0.1290
BL-lstm	0.2088	BL-popularity	0.1442

B. FinNum-2

For financial tweets analysis, we propose BERT-FN-PS, which uses a BERT model with a preprocessing strategy. We also propose XLM-RoBERTa-FN-FTT, which facilitates an XLM-RoBERTa model with TMFTDA techniques.

For BERT-FN-PS, the preprocessing strategy is normalizing all cashtags as one representative tag and all numerals as one designated symbol. The strategy is based on an assumption that, the exact same cashtags or numerals, along with their attachments, might be absent in the test set, so we treat them as identical ones, and then expect the model to be more focused on learning the patterns of the context.

For XLM-RoBERTa-FN-FTT, We use Tokenization Tricks instead. For example, consider a tokenized tweet below:

```
<s> ▁$ xxtag ▁RAD ▁about xxnum ▁9 ▁million
▁more  ▁share  s  ▁than  ▁the  ▁90  ▁day
▁average . … </s>
```

The special tokens `xxnum` and `xxtag` annotate the numeral (`▁9` but not `▁90`) and the cashtag (`▁RAD`) in question, respectively. Combining with the actual subwords of number/cashtag right next to `xxnum/xxtag`, the annotated tokens provide certain features of the token sequence.

For BERT-FN-PS, we run 10 epochs using a batch-size of 32, with the learning rate being 1e-7. Most of Adam optimizer related hyperparameters remain default. For XLM-RoBERTa-FN-FTT, we also apply Mixed Precision to the optimizer, and assign a class weight ratio of 4.28:1 to the loss function. The ratio is simply the inverse of the class distributions. As for the One-cycle scheme specific to XLM-RoBERTa-FN-FTT, every cycle runs one epoch in a batch-size of 8. All cycles share the same range of CM, which uses the default of fastai. For the step size of STLR, we also simply let fastai decide it. Finally, we list the ranges of the learning rates and their decays among cycles:

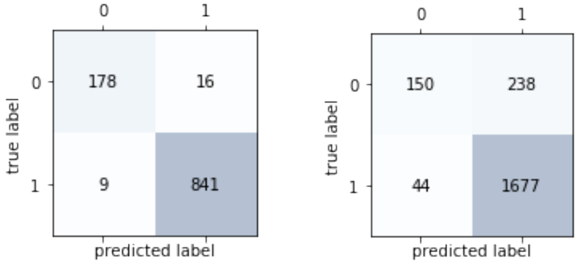
1. 3 cycles: 5e-4 – 5e-7
2. 1 cycle: 5e-5 – 5e-8
3. 1 cycle: 1e-8 – 1e-5

Table 10 shows the result. The macro-F$_1$ of the proposed XLM-RoBERTa-FN-FTT model is 95.99% on the development set, and 71.90% on the test set, which ranks the second best in FinNum-2, using merely 5 cycles of single epoch.

Figure 2 shows the confusion matrix of XLM-RoBERTa-FN-FTT model. On the development set, the model performs well and shows no tendency to classify data into the majority. The recall of the minority is still very high, about 92%. On the test set, however, numbers of both error types greatly increase.

Table 10. Official Results of FinNum-2 (macro-F$_1$ in %)

Model	Development	Test
Majority [42]	44.88	44.93
BERT-FN-PS	86.60	62.70
XLM-RoBERTa-FN-FTT	95.99	71.90



(a) (b)

Fig 2. The confusion matrix of XLM-RoBERTa-FN-FTT model on (a) the development set, and on (b) the test set

There are 388 tweets belong to class-0 in the test set, and the model can only recognize 150 of them. For the performance gap, since the false positives are more frequent than the false negatives, we wonder whether it means the model over-fits because of the class weights, or the model simply doesn't find any clue for the true negatives. We skim read some false positives and find an intriguing yet probably representative case of a numeral "2C." In the test set, a tweet uses it to refer the link between global warming and the stock price of Tesla. In the training and the development sets, however, all the "2C" and "2c" stand for "to see" yet with inconsistent classes. This case probably also indicates that both informal usages of tweet and the domain knowledge of stocks can use some more efforts.

The reasons why the model makes mistake could still include the class imbalance of the datasets, which is a common issue of classification problems. Based on the training set, since the class-1's instances outnumber class-0's 4.4 times to 1, the model may tend to classify a test instance into class-1. Although the class imbalance of the development set is about the same of the training set, of the test set it is worse. The test set's class-1 instances are 5.4 times more than class-0's. This phenomenon likely causes a low recall of class-0, which is only 39%. Such a disappointing outcome might be due to a distribution shift of the test data, as the organizers have mentioned in the overview report [42].

V. Conclusion

In this paper, we have proposed Transformer model's fine-tuning and data augmentation (TMFTDA) techniques for conversational texts and noisy user-generated content. To evaluate the efficacy of TMFTDA, we have fine-tuned various Transformer models for two NTCIR-15 tasks, DialEval-1 FinNum-2.

Experimental results show that the proposed TMFTDA approaches substantially outperform the baselines model of Bi-LSTM for DialEval-1's DQ and ND subtasks. Moreover, TMFTDA performs to a satisfactory level for FinNum-2 task with an XLM-RoBERTa model.

The research contribution of this paper is that, we help shed some light on the usefulness of TMFTDA, for conversational texts and noisy user-generated content in social media text analytics.

Acknowledgment

This research was supported in part of Tamkang University (TKU) research grant, National Taipei University (NTPU), Zeals Co., Ltd., and Ministry of Science and Technology (MOST) under the grant number MOST 109-2221-E-324-024.

Reference

[1] Hua Ai and Diane J. Litman. 2008. Assessing Dialog System User Simulation Evaluation Measures Using Human Judges. In Proceedings of ACL-08: HLT, Association for Computational Linguistics, Columbus, Ohio, 622–629. Retrieved October 29, 2020 from https://www.aclweb.org/anthology/P08-1071

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and others. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020).

[3] Hsiang-En Cherng and Chia-Hui Chang. 2019. Dialogue quality and nugget detection for short text conversation (STC-3) based on hierarchical multi-stack model with memory enhance structure. In Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies.

[4] Kai Cong and Wai Lam. 2019. CUIS at the NTCIR-14 STC-3 DQ Subtask. In Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies.

[5] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 8440–8451. DOI:https://doi.org/10.18653/v1/2020.acl-main.747

[6] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. arXiv preprint arXiv:1906.08101 (2019).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. DOI:https://doi.org/10.18653/v1/N19-1423

[8] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 18, 5 (July 2005), 602–610. DOI:https://doi.org/10.1016/j.neunet.2005.06.042

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. Neural computation 9, (December 1997), 1735–80. DOI:https://doi.org/10.1162/neco.1997.9.8.1735

[10] Jeremy Howard and Sylvain Gugger. 2020. Fastai: A Layered API for Deep Learning. Information 11, 2 (February 2020), 108. DOI:https://doi.org/10.3390/info11020108

[11] Jeremy Howard, Sylvain Gugger, Soumith Chintala, and an O'Reilly Media Company Safari. 2020. Deep learning for coders with fastai and PyTorch: AI applications without a PhD.

[12] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 328–339. DOI:https://doi.org/10.18653/v1/P18-1031

[13] Hairong Huo and Mizuho Iwaihara. 2020. Utilizing BERT Pretrained Models with Various Fine-Tune Methods for Subjectivity Detection. In Web and Big Data, Springer International Publishing, Cham, 270–284.

[14] Sosuke Kato, Rikiya Suzuki, Zhaohao Zeng, and Tetsuya Sakai. 2019. SLSTC at the NTCIR-14 STC-3 dialogue quality and nugget detection subtasks. In Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies.

[15] Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL - A Conditional Transformer Language Model for Controllable Generation. arXiv preprint arXiv:1909.05858 (2019).

[16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. Retrieved from http://arxiv.org/abs/1412.6980

[17] Taku Kudo. 2018. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 66–75. Retrieved from http://aclweb.org/anthology/P18-1007

[18] Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 66–71.

[19] Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. Advances in Neural Information Processing Systems (NeurIPS) (2019).

[20] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the Variance of the Adaptive Learning Rate and Beyond. In International Conference on Learning Representations. Retrieved from https://openreview.net/forum?id=rkgz2aEKDr

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR abs/1907.11692, (2019). Retrieved from http://arxiv.org/abs/1907.11692

[22] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and Optimizing LSTM Language Models. In International Conference on Learning Representations. Retrieved from https://openreview.net/forum?id=SyyGPP0TZ

[23] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. An Analysis of Neural Language Modeling at Multiple Scales. CoRR abs/1803.08240, (2018). Retrieved from http://arxiv.org/abs/1803.08240

[24] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In Advances in Neural Information Processing Systems, 4694–4703.

[25] Michael Noseworthy, Ryan Lowe, Iulian V Serban, Yoshua Bengio, Iulian Vlad Serban, Nicolas Angelard-Gontier, and Joelle Pineau. 2017. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Stroudsburg, PA, USA, 1116–1126. DOI:https://doi.org/10.18653/v1/P17-1103

[26] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 1532–1543. DOI:https://doi.org/10.3115/v1/D14-1162

[27] Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), Association for Computational Linguistics, Florence, Italy, 7–14. DOI:https://doi.org/10.18653/v1/W19-4302

[28] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 1715–1725. DOI:https://doi.org/10.18653/v1/P16-1162

[29] Leslie N. Smith. 2017. Cyclical Learning Rates for Training Neural Networks. In 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 464–472. DOI:https://doi.org/10.1109/WACV.2017.58

[30] Leslie N Smith. 2018. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. arXiv.org (2018). Retrieved from http://arxiv.org/abs/1803.09820

[31] Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. In 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Madrid, Spain, 271–280. DOI:https://doi.org/10.3115/976909.979652

[32] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. CoRR abs/1910.03771, (2019). Retrieved from http://arxiv.org/abs/1910.03771

[33] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. CoRR abs/1609.08144, (2016). Retrieved from http://arxiv.org/abs/1609.08144

[34] Ming Yan, Maofu Liu, and Junyi Xiang. 2019. WUST at the NTCIR-14 STC-3 Dialogue Quality and Nugget Detection Subtask. In Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies.

[35] Hongwei Yong, Jianqiang Huang, Xiansheng Hua, and Lei Zhang. 2020. Gradient Centralization: A New Optimization Technique for Deep Neural Networks. arXiv preprint arXiv:2004.01461 (2020).

[36] Yang You, Igor Gitman, and Boris Ginsburg. 2017. Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017).

[37] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. In International Conference on Learning Representations. Retrieved from https://openreview.net/forum?id=Syx4wnEtvH

[38] Zhaohao Zeng, Sosuke Kato, and Tetsuya Sakai. 2019. Overview of the NTCIR-14 short text conversation task: Dialogue quality and nugget detection subtasks. In Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies.

[39] Zhaohao Zeng, Sosuke Kato, Tetsuya Sakai, and Inho Kang. 2020. Overview of the NTCIR-15 Dialogue Evaluation (DialEval-1) Task. In Proceedings of the 15th NTCIR Conference on Evaluation of Information Access Technologies.

[40] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. 2019. Lookahead optimizer: k steps forward, 1 step back. In Advances in Neural Information Processing Systems, 9597–9608.

[41] Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Final Report of the NTCIR-14 FinNum Task: Challenges and Current Status of Fine-Grained Numeral Understanding in Financial Social Media Data. In NII Conference on Testbeds and Community for Information Access Research, Springer, 183–192.

[42] Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2020. Overview of the NTCIR-15 FinNum-2 Task: Numeral Attachment in Financial Tweets. In Proceedings of the 15th NTCIR Conference on Evaluation of Information Access Technologies.