IMTKU Multi-Turn Dialogue System Evaluation at the NTCIR-15 DialEval-1 Dialogue Quality and Nugget Detection

Mike Tian-Jian Jiang
Zeals Co., Ltd.
Tokyo, Japan
tmjiang@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
406630136@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

Following the third Short-Text Conversation (STC-3) task at NTCIR-14, the first Dialogue Evaluation (DialEval-1) task continue examining, for Chinese and English, how well each participant's system can tackle the two subtasks of Dialogue Quality (DQ) and Nugget Detection (ND). The former estimates the three quality scores of a dialogue, namely Accomplishment (A-score), Satisfaction (S-score), and Effectiveness (E-score), using integer ranks ranging from -2 to 2 each. The latter categorizes dialogue turns by seven nugget types. For DQ subtask, the task organizers measure performance by Normalised Match Distance (NMD) and Root Symmetric Normalised, Order-aware Divergence (RSNOD). For ND subtask, the metrics are Root Normalised Sum of Squares (RNSS) and Jensen-Shannon Divergence (JSD). We consider both subtasks classification problems and tackle them with several models of Transformer, to create a reliable and efficient process using the most recent advances of transfer learning. Our approaches involve various techniques of tokenization and fine-tuning for those Transformers. This paper describes their usages and usefulness of our official runs. 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. NTCIR-15 DialEval-1 task. Those results suggest that one can easily optimize Transformers for DQ and ND subtasks.

†Corresponding author. He has conducted this work partly at Tamkang University.

CCS CONCEPTS

• Information systems ~ Information retrieval ~ Retrieval tasks and goals ~ Question answering

KEYWORDS

Tokenization, Fine-tuning, Transformers, Dialogue evaluation, Dialogue quality.

TEAM NAME

IMTKU

SUBTASKS

Nugget Detection (Chinese, English),
Dialogue Quality (Chinese, English).

1 Introduction

Because of recent advances in natural language processing, more and more researchers and engineers are developing task-oriented dialogue systems. Customer services may benefit from such a chat-bot that responses to inquires 24/7. However, assessing such systems often involves a costly and labor-intensive annotation process that defeats the purpose. The dilemma motivates the task organizers of NTCIR-14 STC-3 [38] and NTCIR-15 DialEval-1 [39] to come up with Dialogue Quality (DQ) and Nugget Detection

(ND) subtasks that examine automatic evaluation systems for helpdesk conversations in Chinese or English.

The DQ subtask uses subjective scales to 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 a nugget as a dialogue turn, determines whether it belongs to Customer side or Helpdesk side, and finally categorizes it into seven types of four groups:

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. $CNUG_0$: 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 XLM-RoBERTa [5] and certain variations of the Bidirectional Encoder Representations from Transformers (BERT) [7], for English and Chinese, respectively.

The next section describes what tricks of the we use or not, together with tech/model specifications that concern our goals. Subsequently, reports and discussions about our official runs follows. Section 4 will then provide complementary summaries of related works. Finally, the paper concludes and presents promising directions of future works.

2 Proposed Approaches

Firstly, we establish our tool-chain. To go through the 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. This section will only introduce model specifications and training procedures that are conceptually related to multilabel and multiclass classifications of the DQ and ND subtasks. Please kindly refer to the original papers of those models and techniques for further details.

2.1 Selected Models

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.

2.2 Tokenization Tricks

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[1] that provides context. 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,

[1] https://fastai1.fast.ai/text.transform.html#Tokenizer

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 $CNUG_0$. As for DQ, a whole dialogue can be tokenized in a similar fashion, where `xxlen` could be useful for certain quality scores, should it be about the time/turns spent on resolving a problem:

```
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[2]. 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 task.

2.3 Fine-tuning Techniques

We adopt recently advanced fine-tuning techniques as much as possible. Some of them are originally designed for AWD-LSTM and QRNN [22,23] by ULMFiT, 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. The section of related works will briefly describe what optimizers and loss functions we have evaluated.

2.3.1 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 the model, discriminative fine-tuning enables us to tune each layer with different learning rates. We use blurr[3] to split the model layers into groups automatically corresponding to 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.

2.3.2 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 [29,30] allow us to micro-manage iterations/updates within a cycle, whereas changing maximum learning rate (`max_lr`) per cycle let us control the quality of each.

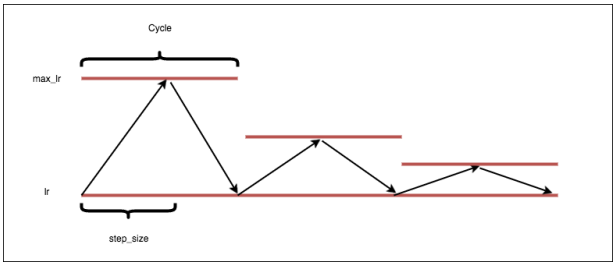

Figure 1: One-cycle Policy with a Max-learning-rate Decay. Image credit: https://github.com/bckenstler/CLR

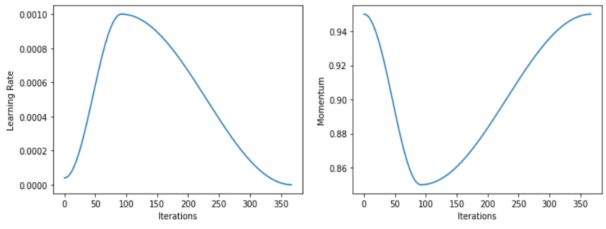

Figure 2: The Slanted Triangular Learning Rates (STLR) and Cyclical Momentum. Image Credit: [7]

[2] https://fastai1.fast.ai/text.transform.html#SpacyTokenizer

[3] https://ohmeow.github.io/blurr/modeling-core/#hf_splitter

Table 1. Configurations of Our 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

Empirically, STLR and cyclical momentum together work best when they simultaneously change in a reversed direction. As Figure 2 shows, it uses a warm-up and annealing for the learning rate while doing the opposite with the momentum. Figure 2, on the other hand, indicates that we apply a simply decay on `max_lr` per cycle.

2.3.3 Other Optimization Schemes. We test several optimizers and find none of them improve the convergence stability significantly than Adam [16]. The section of related works will list those tested optimizers. For the choice of loss function, we realize that the label smoothing function [24] suits our multilabel/multiclass classification better than typical cross-entropy one.

3 Official Run Results and Discussions

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, while they share the same reduction rate: the lower bound is always `max_lr`/1000, and every cycle contains just one epoch.

a. 2e-3 * 3 times, 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, because in theory they are unrelated to classification. Should any reader want to reproduce the outcome, please be advised that it will definitely vary if using different versions.

The rest of tables compare our runs with the best baseline per task-language-metrics. Table 2 shows the Chinese Nugget Detection Results of IMTKU official runs. Table 3 shows the

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

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

English Nugget Detection Results of IMTKU official runs. In the ND subtask 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 types of turns 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 the DQ subtask, 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 Dialogue Quality results of IMTKU official runs. Table 7, 8, and 9 present the A-score, E-score, and S-score of Chinese Dialogue Quality 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 seems equally competitive for both languages.

4 Related Works

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 man-made assessments for spoken dialogue agents, one of the earliest works on learning an automatic evaluation function called PARADISE isolates task requirements from an agent's conversational behavior, at the cost of measurable completeness and complexity of the task [31]. Since the measurement are not always available, instead 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 them 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 size of batches for the efficiency of pretraining, we fail to find significant improvements using them for fine-tuning. The fact that we're already using discriminative fine-tuning 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[4] 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.

5 Conclusion and Future Works

We have taken part in the DialEval-1 DQ and ND subtasks and submitted ten runs. Most of our runs outperform the baselines. We demonstrate that XLM-RoBERTa performs relatively well for both Chinese and English data sets. The Chinese versions of BERT show even better results for the Chinese DQ subtask.

The major contribution of our work is that we have proposed two important ingredients, namely tokenization tricks and fine-tuning techniques, for improving dialogue quality and nugget detection subtasks in dialogue evaluation.

Model choices aside, a particularly more important treatment may be the cycle scheme. A reasonably good recipe of cycle scheme may reduce some burden of hyperparameter tuning, such that we can further explore more research directions in the future. For example, data augmentation may help generalize the patterns of the training sets externally, to compensate the issues of both data sparseness and overfitting. For the sake of generalization, the effect of batch size can also be a sensible perspective to study.

ACKNOWLEDGMENTS

Our thanks to NTCIR-15 task organizers for their great efforts on organizing NTCIR-15 DialEval-1 subtask. 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).

REFERENCES

[1] Hua Ai and Diane J. Litman. 2008. Assessing Dialog System User Simulation Evaluation Measures Using Human Judges. In *Proceedings of ACL-08: HLT*,

[4] https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer

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.