CYUT at the NTCIR-15 FinNum-2 Task: Tokenization and Fine-tuning Techniques for Numeral Attachment in Financial Tweets

MIKE TIAN-JIAN JIANG

ZEALS CO., LTD.

TOKYO, JAPAN

YI-KUN CHEN, SHIH-HUNG WU

CHAOYANG UNIVERSITY OF TECHNOLOGY

TAICHUNG, TAIWAN

Overview

- Run-1: BERT (uncased) with preprocessing
- Run-2: XLM-RoBERTa
  - Tokenization tricks
  - Fine-tuning techniques
- Run-2's performance (macro-F_1)
  - Dev: 95.99%
  - Test: 71.90%

Run-1

- Preprocessing
  - All cashtag instances
    ⟹ one representative tag
  - All numerals
    ⟹ one designated symbol
- Assumption: learn context for unseen attachments

target_cashtag ▁about target_num ▁million ▁more ▁share s ▁than ▁the ▁90 ▁day ▁average . …

~~$RAD~~ ~~9~~

Run-2: Tokenization Tricks

- XLM-RoBERTa’s special tokens
  - beginning of a sentence (<s>)
  - end of a sentence (</s>)
  - separator of sentences (</s> </s>)
- Customized tokens in the fastai convention of “xx” prefix
  - xxtag
  - xxnum

<s> _$ xxtag _RAD _about xxnum _9 _million _more _share s _than _the _90 _day _average . … </s>

Run-2:
Tokenization Tricks: a Side Note

- Not applying the default tokenizer of fastai
  - fastai default: SpaCy
    inserts special tokens before uncapitalized or originally repeated words/characters, e.g.:
    - $TSLA DHL ordered 10 Semis … at any moments $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

      the thirty-five characters of “$” ⟹ “xxrep 35 $”
    - $FOXA … bully bully BUY BUY … 20/20 lol

      “bully bully BUY BUY” ⟹ “xxwrep 2 bully xxwrep 2 xxup buy”

If the task were sentiment analysis of tweets, repetitions and capitalization could be important clues. However, even if the digits coming from those special context tokens won’t negatively impact the numeral attachment task, it is still hard to imagine that the lengths of word/character duplications can help semantically or syntactically, not to mention that XLM-RoBERTa already preserves letter cases of subword tokens.

Run-2: Fine-tuning Techniques

- Originally designed for AWD-LSTM and QRNN by ULMFiT
  - Must assess their usefulness for XLM-RoBERTa.
- Discriminative fine-tuning
  - Except graduate unfreezing
- One-cycle policy (fastai version)

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.

Run-2:
Discriminative Fine-tuning

- Each layer (group) with different learning rates.
- 4 groups (top-to-bottom)
  - classifier
  - pooling layer
  - Transformer layers
  - embeddings

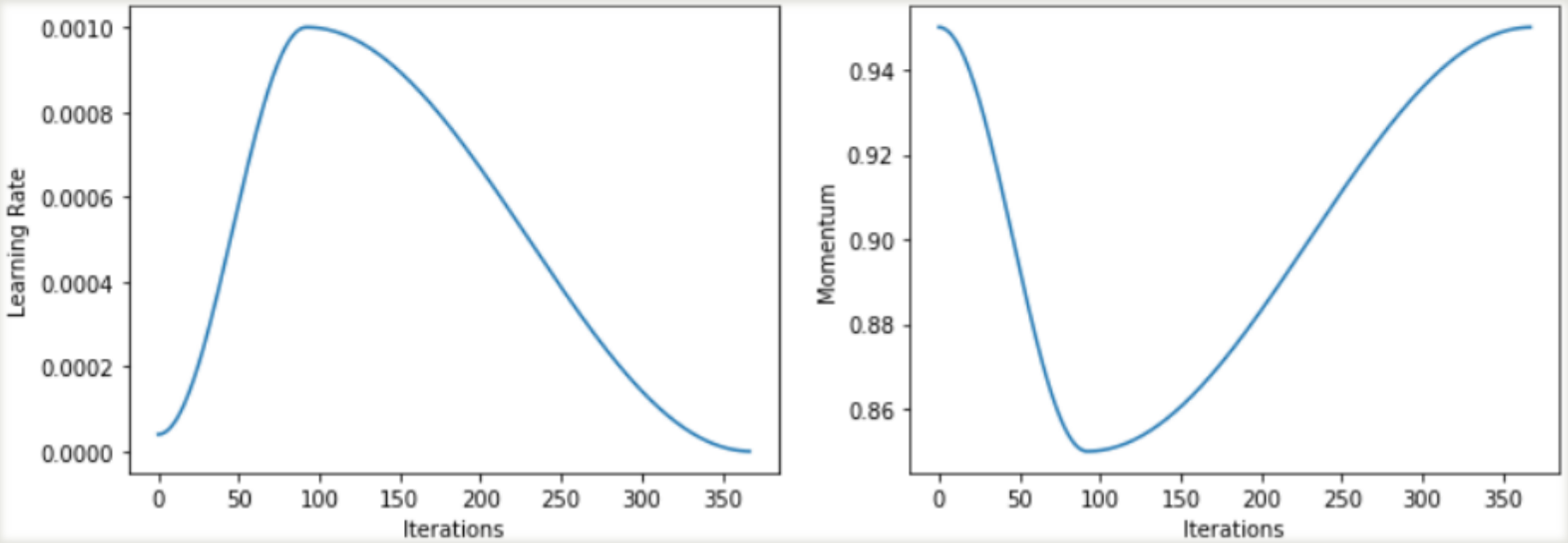
Intuitively, the lower the more general; the higher the more specific

$\Longrightarrow$ Base learning rate for the top, linearly decreased learning rates per lower groups.

Run-2:
One-cycle Policy

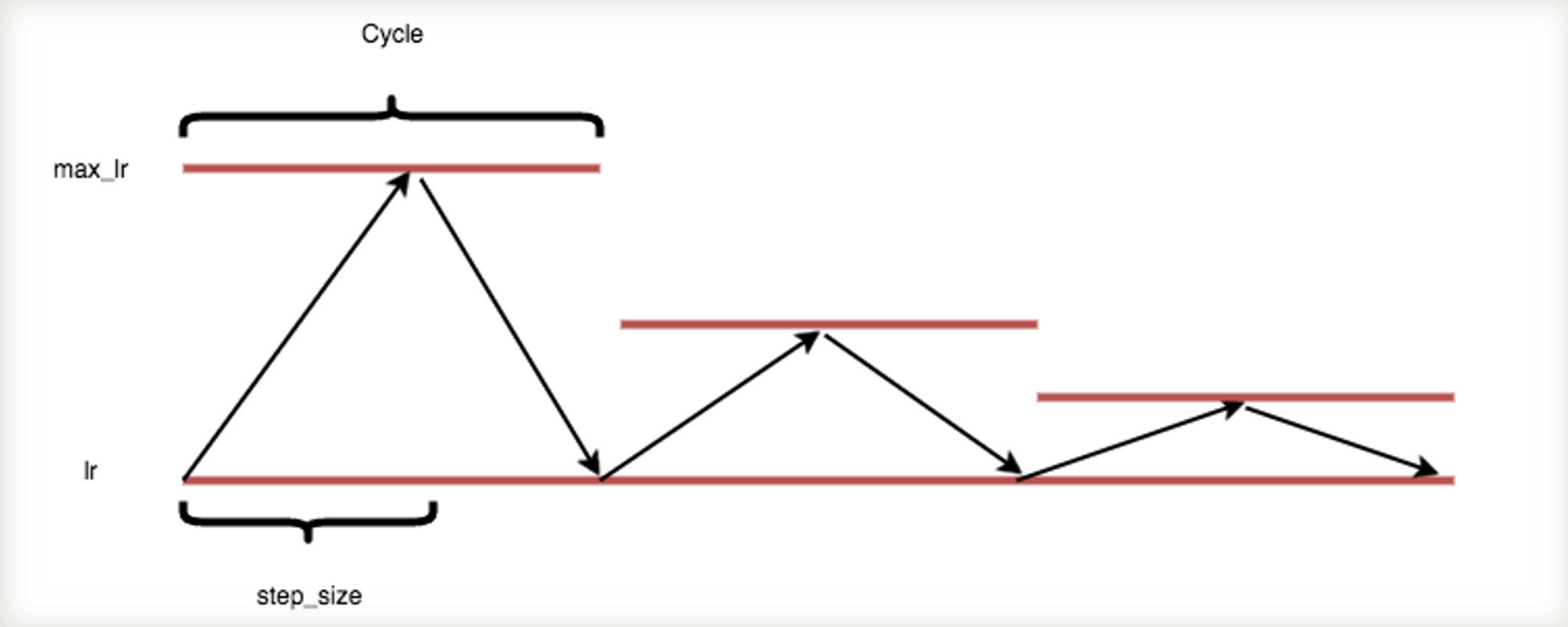
- Cycle:
  an arbitrary number of epochs sharing the same policy of hyperparameters
  - Especially for learning rates and momentums.
- The fastai version
  - The Slanted Triangular Learning
  - Cyclical Momentum
  - Changing maximum learning rate (max_lr) per cycle

Run-2:
One-cycle Policy – warm-up and annealing



STLR and Cyclical Momentum. Image Credit: https://arxiv.org/abs/2002.04688

Run-2:
One-cycle Policy – max_lr decay



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

Run-2:
Other Optimization Schemes

- We test several optimizers and find none of them improve the convergence stability significantly than Adam.
- For the choice of loss function, we realize that there’s no need to use the label smoothing function since the FinNum-2 task is in a typical binary classification setting.

Run-1 Configurations

- Pretrained BERT model: “bert-base-uncased”
- Fine-tuning with the tweets in training set
- Hyper-parameters:
  - Optimizer: Adam
  - Learning-rate: 1e-6
  - Epoch: 30
  - Batch size: 8

Run-2 Configurations

- Mixed Precision
- Class weight: 4.28:1
- Batch size: 8
- Cycles:
  - 3 cycles: 5e-4 – 5e-7
  - 1 cycle: 5e-5 – 5e-8
  - 1 cycle: 1e-8 – 1e-5

The bottom line: just 5 epochs!

Official Runs and Additional Runs Results

(macro F_1 in %)

	Development	Test
Majority	44.88	44.93
BERT*	49.9	49.2
BERT + preprocessing*	*86.6*	*62.7*
CYUT-2	**95.99**	**71.90**
Average of 17 runs	88.18	64.11

* Additional Runs

Error Analysis

- #(false negatives) > #(false positives)
- An intriguing case of a numeral "2C."
  - Test set: link between global warming and the stock price of Tesla.
  - Training/development sets: "2C" / "2c" $\Longrightarrow$ "to see."
- Both informal usages of tweet and the domain knowledge of stocks can use some more efforts.

Conclusion & Future Works

- BERT and XLM-RoBERTa models.
- XLM-RoBERTa's F_1 scores:
  - Dev: 95.99%
  - Test: 71.90%
    - The second best
- User-generated (noisy) data
  - More annotations
  - Data augmentation

Thank You

ANY QUESTION?