Robust Text Classifier on Test-Time Budgets

Md Rizwan Parvez
University of California Los Angeles
rizwan@cs.ucla.edu

Tolga Bolukbasi
Boston University
tolgab@bu.edu

Kai-Wei Chang
University of California Los Angeles
kwchang@cs.ucla.edu

Venkatesh Saligrama
Boston University
srv@bu.edu

A Stop-words Removing:

Our preliminary experiments show that although Stop-words achieves notable speedup, it sometimes comes with a significant performance drop. For example, removing Stop-words from SST-2 dataset, the test-time is 2x faster but the accuracy drops from 85.5 to 82.2. This is due to the stop-words used for filtering text are not learned with the class labels; therefore, some meaningful words (e.g., "but", "not") are filtered out even if they play a very significant role in determining the polarity of the full sentence (e.g., "cheap but not healthy"). Besides, we cannot control the budget in the Stop-words approach.

B Hyperparameter Tuning:

As the performance is proportionate to the text selected, controlling the selection budget we indeed control the performance. In this section we discuss how to control the selection budget by tuning the hyperparameters.

B.1 Tuning the Bag-of-Words *selector*:

As an example, the following is the regularization hyper-parameter C[1] and corresponding selection rate by the bag-of-words *selector* on IMDB.

C	Selection rate (%)
0.01	27
0.05	37
0.1	53
0.11	63
0.15	66
0.25	73
0.7785	79
1.5	82
2.5	88

[1] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

B.2 Tuning skim-RNN:

We re-implement the skim-RNN model as the same baseline as ours with large RNN size $d = 300$, and small RNN sizes $d' \in \{5, 10, 15, 20\}$, and $\gamma \in \{1e^{-9}, 1e^{-10}, 1e^{-11}\}$. For results in Table 2 (in main paper), we compare our model with the best results found from the skim-RNN models with different d', and γ. For IMDB, we found the best speedup and accuracy with $d' = 10$ and hence for Figure 2 (in main paper), we consider this model with $d' = 10$ and vary the selection threshold θ at inference time as described in Seo et al. (2018) for getting different selection of words. We report the accuracy and the test-time for each setting and plot it in Figure 2 (in main paper). The following is the selection thresholds for IMDB.

θ	skimmed(%)
0.45	99
0.48	97
0.47	93
0.505	63
0.51	54
0.52	34
0.53	20

B.3 Tuning the WE *selector*:

For the WE *selector*, we vary the selection budget by tuning the two hyperparameters sparsity (λ_1), and coherent (λ_2) of Lei et al. (2016). In the table below we provide an example settings for corresponding fraction of text to select.

Sparsity (λ_1)	Continuity (λ_2)	Selection rate (%)
8.5e-05	2.0	2.0
8.5e-05	1.0	3.0
9.5e-05	2.0	5.0
9.5e-05	1.0	6.0
0.0001	2.0	9.0
0.0001	1.0	12.0
0.000105	2.0	13.0
0.000105	1.0	15.0
0.00011	2.0	16.0
0.00011	1.0	22.0
0.000115	2.0	23.0
0.000115	1.0	24.0
0.00012	2.0	28.0
0.00012	1.0	64.0

C Machine Specification:

```
Architecture:           x86_64
CPU op-mode(s):
32-bit, 64-bit
Byte Order:
Little Endian
CPU(s):                 12
On-line CPU(s) list:    0-11
Thread(s) per core:     2
Core(s) per socket:     6
Socket(s):              1
NUMA node(s):           1
Vendor ID:
GenuineIntel
CPU family:             6
Model:                  63
Stepping:               2
CPU MHz:                1200.890
BogoMIPS:               6596.22
Virtualization:         VT-x
L1d cache:              32K
L1i cache:              32K
L2 cache:               256K
L3 cache:               15360K
NUMA node0 CPU(s):      0-11
```

References

Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2016. Rationalizing neural predictions. In *EMNLP*.

Min Joon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Neural speed reading via skim-rnn. *ICLR*.