Fast and Accurate Text Classification: Skimming, Rereading and Early Stopping

Keyi Yu[1 2], Yang Liu[2], Alexander G. Schwing[2] and Jian Peng[2]

[1]School of Software, Tsinghua University

[2]Department of Computer Science, University of Illinois, Urbana-Champaign

Introduction

Motivations: For text classification, reading the entire input is not always necessary in practice & we do not have to treat each individual word equally.

Goal: Augment existing RNN models to realize efficient classification, while maintaining a higher or comparable accuracy compared to reading the full text.

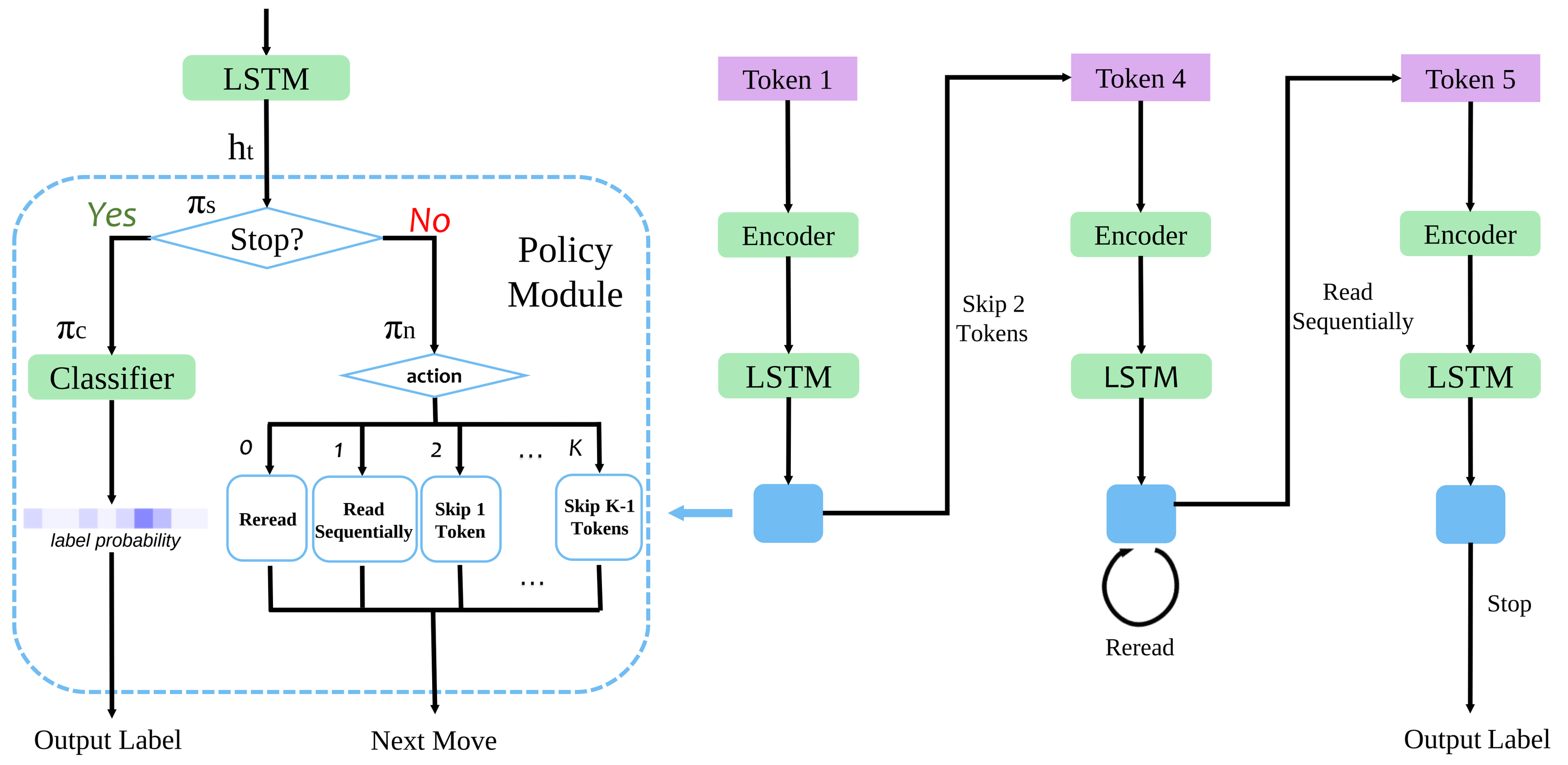
Contributions:

- Proposed an end-to-end trainable approach for skimming, rereading and early stopping mimicking human fast reading, which is applicable to classification tasks.
- Realized the control of trade-off between accuracy and energy cost with single parameter.
- Proved the the effectiveness of each component in our approach and their combinations.

Model Architecture

Model Overview

- Given an input sequence $x_{1:T}$ with length T, our model aims to predict a single label y for the entire sequence.
- Develop a technique for skimming, re-reading, early stopping and prediction, with the goal of skipping irrelevant information and reinforcing the important parts.



Model Specification

- At each time step t, policy module Π takes hidden state h_t of an encoder, which summarizes the text read before and the current token x_{i_t}. Outputs a probability distribution π_t defined over actions.
- A sequence of actions are generated by first sampling a stopping decision in the form of a binary variable s from a Bernoulli distribution $\pi_S(\cdot|h_t)$.
  - If $s=1$, the model stops and draws a label $\hat{y}$ from a conditional multinomial distribution specified by a classifier $\pi_C(\cdot|h_t, s=1)$
  - Otherwise, the model draws a step size $k \in \{0, \dots, K\}$ from another conditional multinominal distribution $\pi_N(\cdot|h_t, s=0)$ to jump to the token $x_{i_{t+1}=i_t+k}$.

Joint Training Method

Joint Distribution:

- We are modeling the possibility of given label $\hat{y}$ as:

$$\Pi(X_{i_1:i_t}, \hat{y}) = \pi_S(s=1|h_t)\pi_C(\hat{y}|h_t, s=1)\prod_{j=1}^{t-1}\pi_S(s=0|h_j)\pi_N(k_j=i_{j+1}-i_j|h_j, s=0), \quad (1)$$

- It could be simplified as:

$$\Pi(X_{i_1:i_t}, \hat{y}) = \pi_S(1|h_t)\pi_C(\hat{y}|h_t, 1)\prod_{j=1}^{t-1}\pi_S(0|h_j)\pi_N(k_j|h_j, 0). \quad (2)$$

Reward Design:

We want to combine the accuracy between predicted label $\hat{y}$ and true label y as well as computational cost $\mathcal{F}$, with single trade-off parameter α

$$r_j = \begin{cases} -\mathcal{L}(\hat{y}, y) - \alpha\mathcal{F}_t & \text{if } j = t \text{ is the final time step} \\ -\alpha\mathcal{F} & \text{otherwise} \end{cases}, \quad (3)$$

Joint Training:

- Our final goal is to find the optimal $\theta = \{\theta^{\pi_S}, \theta^{\pi_C}, \theta^{N}, \theta^{RNN}\}$, which maximize the expected return defined by:

$$J(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sum_t \mathbb{E}_{(X_{i_1:i_t},\hat{y})\sim\Pi}\sum_{j=1}^{t}\gamma^{j-1}r_j\right], \quad (4)$$

- The REINFORCE policy gradient of the objective on data (x, y) can be derived as follows:

$$\widehat{\nabla_\theta J} = \nabla_\theta[\log\pi_S(1|h_t) + \log\pi_C(\hat{y}|h_t, 1) + \sum_{j=1}^{t-1}(\log\pi_S(0|h_j) + \log\pi_N(k_j|h_j, 0))]\sum_{j=1}^{t}\gamma^{j-1}r_j. \quad (5)$$

- Fit a value function as the baseline for accumulative reward to handle large variance.

Ablation Analysis

- **Target:** We aim to demonstrate the effectiveness of each action mechanism in our method: skimming, rereading and early-stopping.

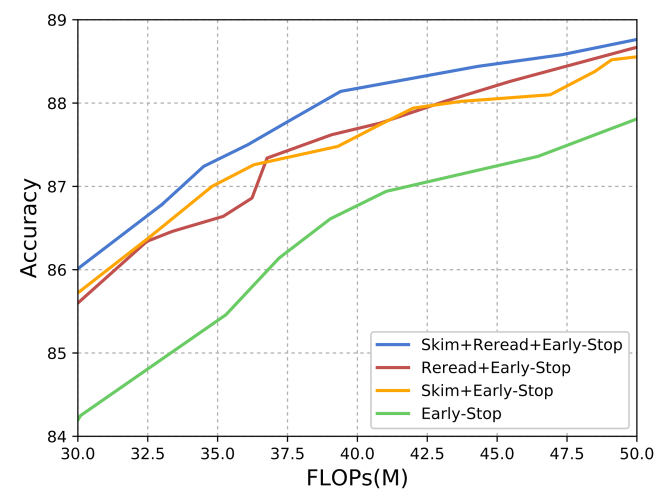


Figure 1: Comparison between different action combination settings

- **Notation:**
  - Blue: Our model (all actions)
  - Orange: Skimming and early-stopping (No rereading)
  - Red: Rereading and early-stopping (No Skimming)
  - Green: Only an early-stopping module
- **Analysis & Conclusion:**
  - Performance of green curve is the worst, indicating that rereading and skimming mechanisms are useful.
  - Performance of blue curve is better than all other ones, indicating that combining skimming and rereading together can further improve performance.

Experiment Results

- **Datasets:** IMDB(Word level), AG news/DBpedia(Character level), Yelp(Sentence level)
- **Baselines:**
  - Whole Reading: A classifier use whole corpus as training data.
  - Partial Reading: Only a stopping module to decide when to terminate reading the paragraph.
  - Early Stopping: Whole Reading model trained on the truncated sentences decided by the stopping model.

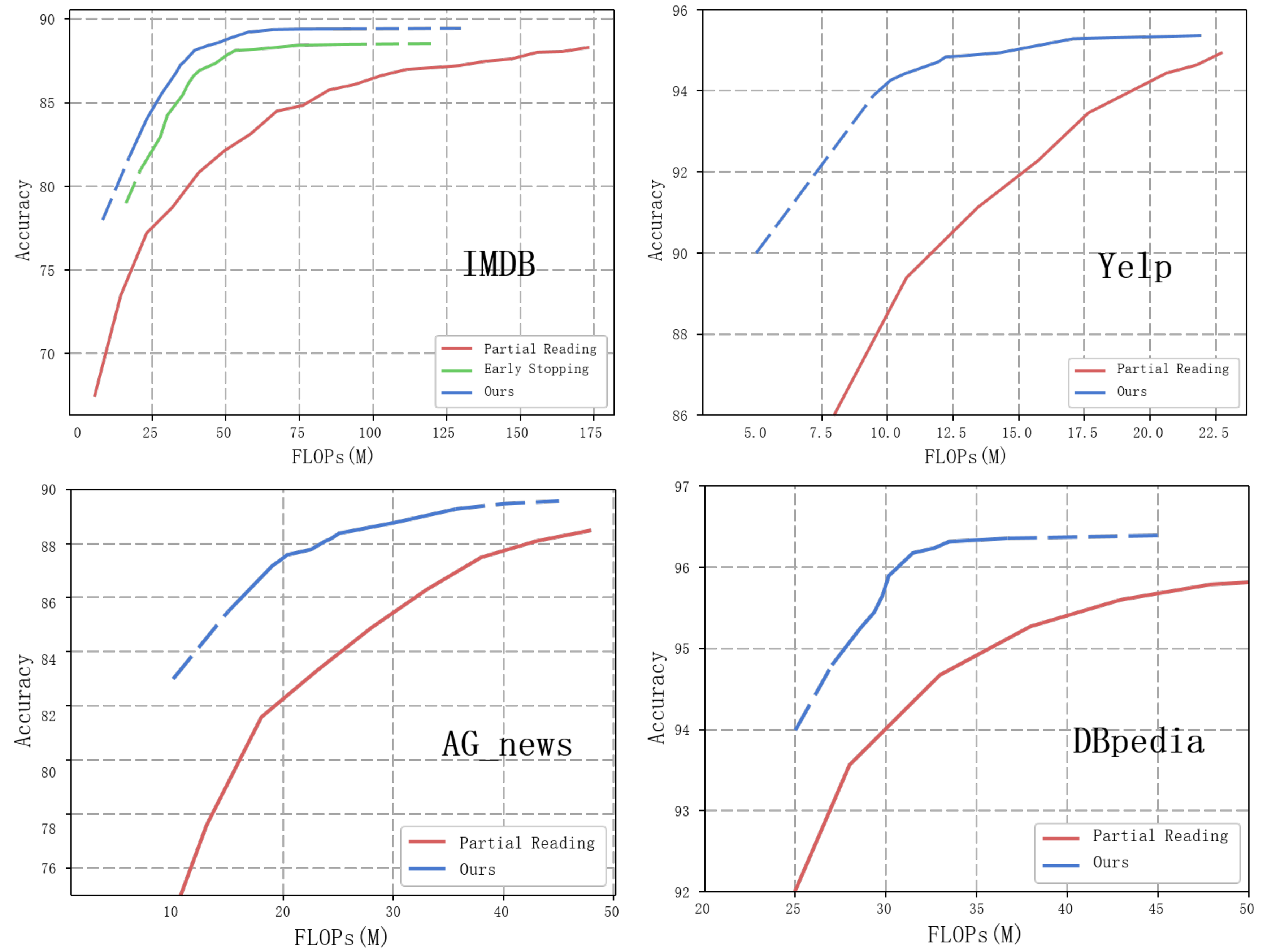


Figure 2: The x-axis and y-axis are representing FLOPs and accuracy, respectively. Curves are obtained by changing the computational budget for each method. Especially, for Ours and Early Stopping model, we adjust the parameter α.

- **Metrics:**
  - Accuracy: Obtained by whole reading model.
  - Speedup: Speedups of our model compared to whole-reading baseline at the same accuracy level.
  - Relative PR Accuracy: Relative performance of the partial reading baseline with the same computational cost as our model.

Dataset	Speedup	Accuracy	Relative PR Accuracy
IMDB	**4.11x**	88.32%	-7.19%
AG news	**1.85x**	88.50%	-4.42%
DBpedia	**2.42x**	95.99%	-1.94%
Yelp	**1.58x**	94.95%	-3.38%

Table 1: Summary of our results on four datasets. Training the classifier jointly with the policy model improves both computational efficiency and accuracy.

- **Conclusion:** We observe that our proposed model can achieve superior performance while being significantly faster.

yu-ky14@mails.tsinghua.edu.cn

May 1st, 2018

jianpeng@illinois.edu