Weighted Sampling For Masked Language Modeling

Linhan Zhang\textsuperscript{1} Qian Chen\textsuperscript{2} Wen Wang\textsuperscript{2} Chong Deng\textsuperscript{2} Yuxin Jiang\textsuperscript{1} Kongzhang Hao\textsuperscript{1} Wei Wang\textsuperscript{3} Xin Cao\textsuperscript{3}

\textsuperscript{1}University of New South Wales, School of Computer and Engineering
\textsuperscript{2}Speech Lab of DAMO Academy, Alibaba Group, China
\textsuperscript{3}Hong Kong University of Science and Technology (Guangzhou), China
Masked Language Model

Given a sentence $S = \{t_1, t_2, \ldots, t_n\}$

- **Standard Masking Strategy**
  - Randomly chooses 15% of tokens to mask
  - 10% of the time replaced by a random token from corpus
  - 10% of the time remains unchanged
  - 80% of the time replaced by a special token [MASK]

- **Objective**
  - The language model must learn to predict the masked tokens with bidirectional context

- **Use Cases**
  - Helps understand the contextual relationships between words
  - Can be used for various natural language processing tasks such as text classification, question answering, and named entity recognition
Motivation - Addressing the Frequency Bias Issue

- Frequency Bias in the Standard Masking Strategy
  - High-frequency tokens are masked frequently
  - More informative tokens with lower frequencies are masked much less frequently during pre-training
  - This greatly harms the efficiency of pre-training

![Input Text:](image)

Fig. 1. An example from WikiText. Randomly selected tokens are in blue while Frequency Weighted Sampled tokens are in pink.
Proposed Method

Weighted Sampling: masking tokens based on (1) token frequency or (2) training loss

![Diagram of Weighted Sampling](image.png)

Fig. 2. Illustration of the proposed Dynamic Weighted Sampling for mask language modeling (MLM). The sampling weight of choosing a token to mask is computed based on the prediction loss of this token by the current PLM. We store the sampling weights of each token in the weight dictionary.
Weighted Sampling Strategy

• Method 1: Frequency Weighted Sampling

  • Step 1: Remove the influences of extremely rare tokens

  \[
  \text{freq}^*(w) = \begin{cases} \text{freq}(w), & \text{if } \text{freq}(w) > \theta \\ \theta, & \text{otherwise.} \end{cases}
  \] (1)

  • Step 2: Compute Sample Weight \( wt(w) \) for \( w \)

  \[
  wt(w) = \left( \text{freq}^*(w) \right)^{-\alpha}
  \] (2)

  • Step 3: Compute Sample Probability \( p(t_i) \) for token \( t_i \) in sentence \( S \)

  \[
  p(t_i) = \frac{wt(t_i)}{\sum_{j=1}^{n} wt(t_j)}
  \] (3)
Weighted Sampling Strategy

• Method 2: Dynamic Weighted Sampling

• Step 1: Initialize Sampling Weight
  • \( wt(t_\_i) = 1 \) for each token \( t_\_i \in T \) in the weight dictionary
  • \( T \) denotes all tokens in the pre-training dataset

• Step 2: Compute Total Cross-Entropy Loss for token \( t_\_i \)
  \[
  L_{t_i} = -\log P(t_i | x, \theta)
  \]
  (4)

• Step 3: Compute Sampling Weight \( wt(t_\_i) \)
  • Compute sampling weight for each token based on its prediction loss by the current pre-trained language model
  • Store these sampling weights in the weight dictionary
  \[
  wt(t_i) = \exp\left(\frac{L_{t_i}}{\tau}\right)
  \]
  (5)

• Step 4: Compute Sampling Probability \( p(t_\_i) \)
  • Normalize \( wt(t_\_i) \) to obtain the sampling probability for each token \( t_\_i \)
Experiments - Semantic Textual Similarity

- **Objective:** To evaluate unsupervised sentence representation on STS tasks
- **Evaluation Metric:** Spearman’s correlation coefficient between the predicted similarity and the gold standard similarity scores
- **Baselines**
  - BERT: bert-base-uncased
  - BERT-CP: continue pre-training on BERT with random sampling on the Wiki-Text
- **Proposed Method**
  - WSBERT_Freq: continue pre-training on BERT with Frequency Weighted Sampling on the Wiki-Text
  - WSBERT.Dynamic: continue pre-training on BERT with Dynamic Weighted Sampling on the Wiki-Text
### Experiments - Semantic Textual Similarity

<table>
<thead>
<tr>
<th>Method</th>
<th>STS12</th>
<th>STS13</th>
<th>STS14</th>
<th>STS15</th>
<th>STS16</th>
<th>STS-B</th>
<th>SICK-R</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>39.70</td>
<td>59.38</td>
<td>49.67</td>
<td>66.03</td>
<td>66.19</td>
<td>53.87</td>
<td>62.06</td>
<td>56.70</td>
</tr>
<tr>
<td>BERT-CP</td>
<td>41.00</td>
<td>60.02</td>
<td>51.11</td>
<td>68.43</td>
<td>64.59</td>
<td>56.32</td>
<td>62.07</td>
<td>57.65</td>
</tr>
<tr>
<td>WSBERT_Freq</td>
<td>42.60</td>
<td>61.32</td>
<td>52.04</td>
<td>69.84</td>
<td>66.61</td>
<td>59.89</td>
<td>61.94</td>
<td>59.18</td>
</tr>
<tr>
<td>WSBERT_Dynamic</td>
<td>47.80</td>
<td>67.28</td>
<td>57.13</td>
<td>71.41</td>
<td>68.87</td>
<td>65.28</td>
<td>64.90</td>
<td>63.24</td>
</tr>
<tr>
<td>BERT-Whitening</td>
<td>54.28</td>
<td>78.07</td>
<td>65.44</td>
<td>64.83</td>
<td>70.16</td>
<td>71.43</td>
<td>62.23</td>
<td>66.43</td>
</tr>
<tr>
<td>WSBERT-Whitening</td>
<td>55.14</td>
<td>78.45</td>
<td>66.13</td>
<td>65.47</td>
<td>70.68</td>
<td>71.98</td>
<td>61.91</td>
<td>67.10</td>
</tr>
<tr>
<td>BERT + Prompt†</td>
<td>60.96</td>
<td>73.83</td>
<td>62.18</td>
<td>71.54</td>
<td>68.68</td>
<td>70.60</td>
<td>67.16</td>
<td>67.85</td>
</tr>
<tr>
<td>WSBERT + Prompt</td>
<td><strong>63.03</strong></td>
<td>71.66</td>
<td>63.80</td>
<td><strong>75.32</strong></td>
<td><strong>76.67</strong></td>
<td><strong>74.79</strong></td>
<td>65.32</td>
<td><strong>70.08</strong></td>
</tr>
</tbody>
</table>

#### Findings
- Weighted sampling methods, WSBERT_Freq and WSBERT_Dynamic, outperform the baselines (BERT and BERT-CP).
- For instance, WSBERT_Dynamic outperforms BERT and BERT-CP by 6.54 and 5.59 absolute points respectively.
- WSBERT_Dynamic can be effectively combined with Whitening and Prompt to further improve performance.
Experiments - GLUE Evaluation

- **Purpose**: to evaluate transfer learning capability

- **Findings**
  - WSBERT achieves the best average GLUE score compared to BERT and BERT-CP, outperforming BERT by 0.52 absolute
  - BERT-CP degrades GLUE AVG by 0.35 absolute compared to BERT
  - WSBERT outperforms BERT-CP by 0.87 absolute
  - The gain of WSBERT over BERT is from continual pre-training with Dynamic Weighted Sampling, not from continual pre-training

STS and GLUE evaluations demonstrate that Dynamic Weighted Sampling improves the transfer learning capability while enhancing sentence representations.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BERT</th>
<th>BERT-CP</th>
<th>WSBERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNLI</td>
<td>84.30±0.26</td>
<td>84.26±0.19</td>
<td>84.42±0.35</td>
</tr>
<tr>
<td>QQP</td>
<td>91.31±0.04</td>
<td>90.94±0.59</td>
<td>91.43±0.05</td>
</tr>
<tr>
<td>QNLI</td>
<td>91.47±0.01</td>
<td>91.32±0.17</td>
<td>91.14±0.17</td>
</tr>
<tr>
<td>SST-2</td>
<td>92.86±0.13</td>
<td>92.78±0.43</td>
<td>91.35±0.47</td>
</tr>
<tr>
<td>CoLa</td>
<td>56.47±0.65</td>
<td>57.44±0.95</td>
<td>58.29±0.33</td>
</tr>
<tr>
<td>STS-B</td>
<td>89.68±0.26</td>
<td>89.52±0.37</td>
<td>89.86±0.18</td>
</tr>
<tr>
<td>MRPC</td>
<td>86.13±1.63</td>
<td>85.13±0.53</td>
<td>88.20±2.39</td>
</tr>
<tr>
<td>RTE</td>
<td>69.23±0.4</td>
<td>67.25±1.84</td>
<td>70.89±0.17</td>
</tr>
<tr>
<td><strong>AVG</strong></td>
<td>82.68±0.33</td>
<td>82.33±0.32</td>
<td>83.20±0.10</td>
</tr>
</tbody>
</table>

Table 2. GLUE Validation results from BERT-base-uncased (BERT-base), BERT-base-uncased continually pre-trained (BERT-CP), and Weighted-Sampled BERT (WSBERT). BERT-CP and WSBERT both continually train on BERT with the same training settings. WSBERT refers to WSBERT_Dynamic. The best results for each dataset and AVG are in bold.
Takeaway and Future work

- Proposed two Weighted Sampling methods to **address the frequency bias issue** in conventional masked language modeling.
- Developed a new PLM, **WSBERT**, by applying Weighted Sampling to BERT.
- WSBERT outperforms BERT in both **sentence representation quality** and **transfer learning capability**.
- Future work includes investigating other dynamic sampling methods and exploring training objectives with a penalty for frequency bias.