Applying the Transformer to Character-level Transduction

Shijie Wu\(^3\) Ryan Cotterell\(^5,9\) Mans Hulden\(^2\)
\(^3\)Johns Hopkins University \(^9\)University of Cambridge
\(^5\)ETH Zürich \(^2\)University of Colorado Boulder
shijie.wu@jhu.edu ryan.cotterell@inf.ethz.ch mans.hulden@colorado.edu

Abstract

The transformer (Vaswani et al., 2017) has been shown to outperform recurrent neural network-based sequence-to-sequence models in various word-level NLP tasks. Yet for character-level transduction tasks, e.g. morphological inflection generation and historical text normalization, there are few works that outperform recurrent models using the transformer. In an empirical study, we uncover that, in contrast to recurrent sequence-to-sequence models, the batch size plays a crucial role in the performance of the transformer on character-level tasks, and we show that with a large enough batch size, the transformer does indeed outperform recurrent models. We also introduce a simple technique to handle feature-guided character-level transduction that further improves performance. With these insights, we achieve state-of-the-art performance on morphological inflection and historical text normalization. We also show that the transformer outperforms a strong baseline on two other character-level transduction tasks: grapheme-to-phoneme conversion and transliteration.

1 Introduction

The transformer (Vaswani et al., 2017) has become a popular architecture for sequence-to-sequence transduction in NLP. It has achieved state-of-the-art performance on a range of common word-level transduction tasks: neural machine translation (Barrault et al., 2019), question answering (Devlin et al., 2019) and abstractive summarization (Dong et al., 2019). In addition, the transformer forms the backbone of the widely-used BERT (Devlin et al., 2019).

Yet for character-level transduction tasks like morphological inflection, the dominant model has remained a recurrent neural network-based sequence-to-sequence model with attention (Cotterell et al., 2018). This is not for lack of effort—but rather, it is the case that the transformer has consistently underperformed in experiments on average (Tang et al., 2018b).\(^1\) As anecdotal evidence of this, we note that in the 2019 SIGMORPHON shared task on cross-lingual transfer for morphological inflection, no participating system was based on the transformer (McCarthy et al., 2019).

Character-level transduction models are often trained with less data than their word-level counterparts: In contrast to machine translation, where millions of training samples are available, the 2018 SIGMORPHON shared task (Cotterell et al., 2018) high-resource setting only provides \(\approx 10k\) training examples per language. It is also not obvious that non-recurrent architectures such as the transformer

\(^1\)This claim is also based on the authors’ personal communication with other researchers in morphology in the corridors of conferences and through email.

![Figure 1: Development set accuracy for 5 languages on morphological inflection with different batch sizes. We evince our two primary contributions: (1) we set the new state of the art.](https://example.com/figure1.png)
should provide an advantage at many character-level tasks: For instance, Gehring et al. (2017) and Vaswani et al. (2017) suggest that transformers (and convolutional models in general) should be better at remembering long-range dependencies. In the case of morphology, none of these considerations seem relevant: inflecting a word (a) requires little capacity to model long-distance dependencies and is largely a monotonic transduction; (b) it involves no semantic disambiguation, the tokens in question being letters; (c) it is not a task for which parallelization during training appears to help, since training time has never been an issue in morphology tasks.\(^2\)

In this work, we provide state-of-the-art numbers for morphological inflection and historical text normalization, a novel result in the literature. We also show the transformer outperforms a strong recurrent baseline on two other character-level tasks: grapheme-to-phoneme (g2p) conversion and transliteration. We find that a single hyperparameter, batch size, is largely responsible for the previous poor results. Despite having fewer parameters, the transformer outperforms the recurrent sequence-to-sequence baselines on all four tasks. We conduct a short error analysis on the task of morphological inflection to round out the paper.

### 2 The Transformer for Characters

**The Transformer.** The transformer, originally described by Vaswani et al. (2017), is a self-attention-based encoder-decoder model. The encoder has \(N\) layers, consisting of a multi-head self-attention layer and a two-layer feed-forward layer with ReLU activation, both equipped with a skip connection. The decoder has a similar structure as the encoder except that, in each decoder layer between the self-attention layer and feed-forward layer, a multi-head attention layer attends to the output of the encoder. Layer normalization (Ba et al., 2016) is applied to the output of each skip connection. Sinusoidal positional embeddings are used to incorporate positional information without the need for recurrence or convolution. Here, we describe two modifications we make to the transformer for character-level tasks.

**A Smaller Transformer.** As the dataset sizes in character-level transduction tasks are significantly smaller than in machine translation, we employ a smaller transformer with \(N = 4\) encoder-decoder layers. We use 4 self-attention heads. The embedding size is \(d_{\text{model}} = 256\) and the hidden size of the feed-forward layer is \(d_{\text{FF}} = 1024\). In the preliminary experiments, we found that using layer normalization before self-attention and the feed-forward layer performed slightly better than the original model. It is also the default setting of a popular implementation of the transformer (Vaswani et al., 2018). The transformer alone has around 7.37M parameters, excluding character embeddings and the linear mapping before the softmax layer. We decode the model left to right in a greedy fashion.

**Feature Invariance.** Some character-level transduction is guided by features. For example, in the case of morphological reinflection, the task requires a set of morphological attributes that control what form a citation form is inflected into (see Fig. 2 for an example). However, the order of the features is irrelevant. In a recurrent neural network, features are input in some predefined order as special characters and pre- or postpended to the input character sequence representing the citation form. The same is true for a vanilla transformer model, as shown on the left-hand side of Fig. 2. This leads to
different relative distances between a character and a set of features. To avoid such an inconsistency, we propose a simple remedy: We set the positional encoding of features to 0 and only start counting the positions for characters. Additionally, we add a special token to indicate whether a symbol is a word character or a feature. The right-hand side of Fig. 2 evinces how we have the same relative distance between characters and features.

### 3 Empirical Findings

#### Tasks
We consider four character-level transliteration tasks: morphological inflection, grapheme-to-phoneme conversion, transliteration, and historical text normalization. For morphological inflection, we use the 2017 SIGMORPHON shared task data (Cotterell et al., 2017) with 52 languages. The performance is evaluated by accuracy (ACC) and edit distance (Dist). For the g2p task, we use the unstressed CMUDict (Weide, 1998) and NETtalk (Sejnowski and Rosenberg, 1987) resources. We use the splits from Wu et al. (2018). We evaluate under word error rate (WER) and phoneme error rate (PER). For transliteration, we use the NEWS 2015 shared task data (Zhang et al., 2015).

#### Optimization
We use Adam (Kingma and Ba, 2014) with a learning rate of 0.001 and an inverse square root learning rate scheduler (Vaswani et al., 2017) with 4k steps during the warm-up. We train the model for 20k gradient updates and save and evaluate the model every 400 gradient updates. We select the best model out of 50 checkpoints based on development set accuracy. The number of gradient updates and checkpoints are roughly the same as Wu and Cotterell (2019), the single model state of the art on the 2017 SIGMORPHON dataset. We use their model as a baseline model. For all experiments, we use a single predefined random seed.

### 3.1 A Controlled Hyperparameter Study
To demonstrate the importance of hyperparameter tuning for the transformer on character-level tasks, we perform a small controlled hyperparameter study. This is important since researchers had previously failed to achieve high-performing results with the transformer on character-level tasks. Here, we look at morphological inflection on the five languages in the 2017 SIGMORPHON dataset where submitted systems performed the worst: Latin, Faroese, French, Hungarian, and Norwegian (Nynorsk). We set the dropout to 0.3, $\beta_2$ of Adam to 0.999 (the default value), and do not use label smoothing. We do not tune any other hyperparameter except the following three hyperparameters.

#### The Importance of Batch Size.
While recurrent models like Wu and Cotterell use a batch size of 20, halving the learning rate when stuck and employing early stopping, we find that a less aggressive learning rate scheduler, allowing the model to train longer, outperforms these hyperparameters. Fig. 1 shows that the significant impact of batch size on the transformer. The transformer performance in-
Table 2: Average test performance on morphological inflection of Transformer against models from the literature. * denotes model ensembling.

<table>
<thead>
<tr>
<th>Model</th>
<th>ACC</th>
<th>Dist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silfverberg et al. (2017)*</td>
<td>92.97</td>
<td>0.170</td>
</tr>
<tr>
<td>Wu et al. (2018)</td>
<td>93.60</td>
<td>0.128</td>
</tr>
<tr>
<td>Wu and Cotterell (2019)</td>
<td>94.40</td>
<td>0.113</td>
</tr>
<tr>
<td>Wu and Cotterell (2019) (Our eval)</td>
<td>94.81</td>
<td>0.123</td>
</tr>
<tr>
<td>Makarov et al. (2017)*</td>
<td>95.12</td>
<td>0.100</td>
</tr>
<tr>
<td>Bergmanis et al. (2017)*</td>
<td>95.32</td>
<td>0.100</td>
</tr>
<tr>
<td>Transformer (Dropout = 0.3)</td>
<td>95.59</td>
<td>0.088</td>
</tr>
<tr>
<td>Transformer (Dropout = 0.1)</td>
<td>95.56</td>
<td>0.090</td>
</tr>
</tbody>
</table>

Table 3: Average test performance on historical text normalization of Transformer against models from the literature. * denote subset of dataset as Flachs et al. (2019) only experiment with subset of languages.

<table>
<thead>
<tr>
<th>Model</th>
<th>WER</th>
<th>PER</th>
<th>ACC</th>
<th>MFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wu et al. (2018)</td>
<td>28.20</td>
<td>0.068</td>
<td>41.10</td>
<td>0.894</td>
</tr>
<tr>
<td>Wu and Cotterell (2019)</td>
<td>28.20</td>
<td>0.069</td>
<td>41.20</td>
<td>0.895</td>
</tr>
<tr>
<td>Transformer (Dropout = 0.3)</td>
<td>28.08</td>
<td>0.070</td>
<td>43.39</td>
<td>0.897</td>
</tr>
<tr>
<td>Transformer (Dropout = 0.1)</td>
<td>27.63</td>
<td>0.069</td>
<td>41.35</td>
<td>0.891</td>
</tr>
</tbody>
</table>

Table 4: Average test performance on Grapheme-to-Phoneme and dev performance on Transliteration of Transformer against models from the literature.

<table>
<thead>
<tr>
<th>Model</th>
<th>WER</th>
<th>PER</th>
<th>ACC</th>
<th>MFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer (Dropout = 0.3)</td>
<td>91.78</td>
<td>0.392</td>
<td>90.37</td>
<td>0.360</td>
</tr>
<tr>
<td>Transformer (Dropout = 0.1)</td>
<td>91.56</td>
<td>0.399</td>
<td>89.93</td>
<td>0.368</td>
</tr>
<tr>
<td>Transformer (Dropout = 0.3)</td>
<td>91.27</td>
<td>0.381</td>
<td>89.73</td>
<td>0.350</td>
</tr>
<tr>
<td>Transformer (Dropout = 0.1)</td>
<td>91.67</td>
<td>0.389</td>
<td>90.32</td>
<td>0.358</td>
</tr>
</tbody>
</table>

**Morphological Inflection.** As shown in Tab. 2, the feature invariant transformer produces state-of-the-art results on the 2017 SIGMORPHON shared tasks, improving upon ensemble-based systems by 0.27 points. We observe that as the dataset decreases in size, a model with a larger dropout value performs slightly better. A brief tally of phenomena that are difficult to learn for many machine learning models, categorized along typical linguistic dimensions (such as word-internal sound changes, vowel harmony, circumfixation, ablaut, and umlaut phenomena) fail to reveal any consistent pattern of advantage to the transformer model. In fact, errors seem to be randomly distributed with an overall advantage of the transformer model. Curiously, errors grouped along the dimension of word length reveal that as word forms grow longer, the transformer advantage shrinks (Fig. 3).

**Historical Text Normalization.** Tab. 3 shows that the transformer model with dropout of 0.1, as in the case of morphological inflection, improves upon the previous state of the art, although the model with a dropout of 0.3 yields a slightly better CER$_i$.

**G2P and Transliteration.** Tab. 4 shows that the transformer outperforms previously published strong recurrent models on two tasks despite having fewer parameters. A dropout rate of 0.3 yields

---

6It is also large in the context of character-level tasks, which typically have around 10k training examples. Batch size of 400 would imply approximately 4% of training data in a single gradient update.

---

**Other Hyperparameters.** Vaswani et al. (2017) applies label smoothing (Szegedy et al., 2016) of 0.1 to the transformer model and shows that it hurts perplexity, but improves BLEU scores for machine translation. Instead of the default $0.999 \beta_2$ for Adam, Vaswani et al. (2017) uses 0.98 and we find that both choices benefit character-level transduction tasks as well (see Tab. 1).

### 3.2 New State-of-the-Art Results

We train our feature invariant transformer on the four character-level tasks, exhibiting state-of-the-art results on morphological inflection and historical text normalization.
significantly better performance on the transliteration task while a dropout rate of 0.1 is stronger on the g2p task. This shows that transformers can and do outperform recurrent transducers on common character-level tasks when properly tuned.

4 Related Work

Character-level transduction is largely dominated by attention-based LSTM sequence-to-sequence (Luong et al., 2015) models (Cotterell et al., 2018). Character-level transduction tasks usually involve input-output pairs that share large substrings and alignments between these are often monotonic. Models that address the task tend to focus on exploiting such structural bias. Instead of learning the alignments, Aharoni and Goldberg (2017) use external monotonic alignments from the SIGMORPHON 2016 shared task baseline Cotterell et al. (2016). Makarov et al. (2017) use this approach to win the CoNLL-SIGMORPHON 2017 shared task on morphological inflection (Cotterell et al., 2017). Wu et al. (2018) shows that explicitly modeling alignment (hard attention) between source and target characters outperforms soft attention. Wu and Cotterell (2019) further shows that enforcing monotonicity in a hard attention model improves performance.

5 Conclusion

Using a large batch size and feature invariant input allows the transformer to achieve strong performance on character-level tasks. However, it is unclear what linguistic errors the transformer makes compared to recurrent models on these tasks. Future work should analyze the errors in detail as Gorman et al. (2019) does for recurrent models. While Wu and Cotterell shows that the monotonicity bias benefits character-level tasks, it is not evident how to enforce monotonicity on multi-headed self-attention. Future work should consider how to best incorporate monotonicity into the model, either by enforcing it strictly (Wu and Cotterell, 2019) or by pretraining the model to copy (Anastasopoulos and Neubig, 2019).

References


