FAIRNESS IN REPRESENTATION FOR MULTILINGUAL NLP: INSIGHTS FROM CONTROLLED EXPERIMENTS ON CONDITIONAL LANGUAGE MODELING

Ada Wan
University of Zurich
ada.wan@uzh.ch

ABSTRACT

We perform systematically and fairly controlled experiments with the 6-layer Transformer to investigate the hardness in conditional-language-modeling languages which have been traditionally considered morphologically rich (AR and RU) and poor (ZH). We evaluate through statistical comparisons across 30 possible language directions from the 6 languages of the United Nations Parallel Corpus across 5 data sizes on 3 representation levels — character, byte, and word. Results show that performance is relative to the representation granularity of each of the languages, not to the language as a whole. On the character and byte levels, we are able to eliminate statistically significant performance disparity, hence demonstrating that a language cannot be intrinsically hard. The disparity that mirrors the morphological complexity hierarchy is shown to be a byproduct of word segmentation. Evidence from data statistics, along with the fact that word segmentation is qualitatively indeterminate, renders a decades-long debate on morphological complexity (unless it is being intentionally modeled in a word-based, meaning-driven context) irrelevant in the context of computing. The intent of our work is to help effect more objectivity and adequacy in evaluation as well as fairness and inclusivity in experimental setup in the area of language and computing so to uphold diversity in Machine Learning and Artificial Intelligence research. Multilinguality is real and relevant in computing not due to canonical, structural linguistic concepts such as morphology or “words” in our minds, but rather standards related to internationalization and localization, such as character encoding — something which has thus far been sorely overlooked in our discourse and curricula.

1 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Most current work on fairness in Machine Learning (ML) and Natural Language Processing (NLP) focuses on the societal biases encoded in natural language data that are propagated and amplified when they are used at scale for/as Artificial Intelligence (AI) solutions. But little has been said or questioned about the bias, as in, the favoring of certain outcomes, implicit in our theoretical/scientific assumptions that results in the varying performance of different languages in computing.

Disparity in machine translation results For instance, results reported in Junczys-Dowmunt et al. (2016) for Phrase-Based Statistical Machine Translation (PBSMT) (Koehn et al., 2003) and Neural MT (Bahdanau et al., 2014) on the 6 official languages3 of the United Nations (UN) Parallel Corpus (Ziemski et al., 2016) indicate a disparity between EN/ES/FR and AR/RU/ZH in BLEU (Papineni et al., 2002) — translation performance in the latter group is generally worse, regardless of the MT algorithm used. AR and RU are traditionally considered morphologically complex (see e.g. Minkov et al., 2007, Seddah et al., 2010 and proceedings of related workshops in subsequent workshops and conferences on “Fairness, Accountability, and Transparency” (FAccT) on “Fairness, Accountability, and Transparency” (FAccT).

1 see e.g. work from conference (https://facctconference.org) and workshops in previous years on “Fairness, Accountability, and Transparency” (FAccT)

2 Arabic (AR), English (EN), Spanish (ES), French (FR), Russian (RU), and Chinese (ZH)
years for Statistical Parsing of Morphologically Rich Languages), and ZH morphologically frugal (for its lacking determiners and plural or tense markers) (Koehn, 2005). While Koehn (2005) found translating into EN to be easier than into morphologically rich languages based on BLEU scores of word tokens from PBSMT systems of 110 language directions from the 11 Europarl languages then, Bugliarello et al. (2020) found it is easier to translate out of EN than into it based on 21 Europarl languages in BPEs (Byte Pair Encodings) (Sennrich et al., 2016) with the Transformer (Vaswani et al., 2017) in a new metric, cross-mutual information.

Disparity in language modeling results Disparate performances across different languages seem to have been implicitly accepted in that it is often believed that some languages are harder to model than others. Bender (2009) advocated the relevance of linguistic typology for the design of language-universal NLP systems due to differences based on crosslinguistic structural notions, such as parts of speech and morphological complexity. Cotterell et al. (2018) studied (monolingual) language models (LMs) on the 21 Europarl languages using a word-level 7-gram standard Kneser & Ney (1995) model and LSTM-LMs (Sundermeyer et al., 2012) with characters and lemmatized forms in information-theoretic terms, and found morphological complexity to be the primary culprit for the differences in performance. Mielke et al. (2019) supplemented with the multilingual Bible corpus (Mayer & Cysouw, 2014) to extend the coverage to 69 languages, tested on RNN-LMs (an implementation of LSTM (Hochreiter & Schmidhuber, 1997)) with characters and BPEs, but concluded that basic data statistics in vocabulary size (|V|) and sequence length were the most predictive performance features.

We noticed, however, a discrepancy in the results from Mielke et al. (2019) for ZH — it came out as the least difficult for the character model, but it is the 6th most difficult language for the BPE model. As different input representations have been tested with different architectures with divergent results in different metrics in previous studies, each of them only testing with one data size, we decided to investigate the matter more systematically once again with statistical comparisons of score distributions between languages.

1.2 Research questions and contributions

Research questions Are there any statistically significant differences in hardness when it comes to Conditional-Language-Modeling (CLMing) languages which have been traditionally considered morphologically rich (AR and RU) and poor (ZH) with the 6-layer Transformer? Is morphological complexity inherent in language? When is the notion of morphological complexity relevant in computing?

Summary of findings and insights Based on our bilingual CLMing setup with the UN Parallel Corpus in the data size range from 10^2 to 10^6 lines on the character, byte, and word levels, we find:

1. Language has more finer-grained dimensions with different representations and learning patterns. Hardness in modeling is relative to its representation granularity (representation relativity).
2. There is neutralization of source language instances, i.e. there are no statistically significant differences between source language pairs. Only pairs of target languages differ significantly.
3. On the character and byte levels, hardness is correlated with statistical properties concerning sequence length and |V| of a language, regardless of its morphological profile. As it is possible to eliminate performance disparity by decomposing sequences into finer-grained units in characters and bytes, we show that morphological complexity is not an intrinsic property of language. Unless word-based methods are used, or unless we implement/model it explicitly, the notion of morphological complexity is irrelevant in computing.
4. On the word level, hardness is correlated with |V|, and a complexity hierarchy arises through the manual preprocessing step of word tokenization. This complexity/disparity effect by word segmentation can be improved by subword tokenization but cannot be eliminated due to the fundamental qualitative differences in the definition of a “word” being one that neither holds universally nor is suitable/consistent for fair crosslinguistic comparisons.
5. Representational units of finer granularity can help close the gap in performance disparity.

Orthogonal to our main research questions, we also observed 2 types of sample-wise non-monotonicity — Double Descent (Belkin et al., 2019; Nakkiran et al., 2020) and erraticity. For reasons due to length and scope for this paper, we will defer discussions and analyses of these beyond what is addressed in §5 to future work.
Outline of the paper  In §2 we define our method and experimental setup. We present our results and analysis on the primary representations in §3 and those from secondary set of controls in §4 in a progressive manner to ease understanding. Meta analysis on performance disparity and other discussions are in §5.

2 Method and definitions

Conditional language modeling  CLMing is the modeling of the probability of the next token, given the history of the preceding tokens and conditioning context. In our case, such conditioning context is a line from the source language. To explicitly focus on modeling the complexities that may or may not be intrinsic to the languages, we study the more fundamental process of CLMing without performing any translation. This allows us to eliminate confounds associated with generation and other evaluation metrics. One could think of our setup as estimating conditional probabilities with the Transformer, with a bilingual (one-to-one) setup where the perplexity (PP) of one target language (l_{trg}) is estimated given the parallel data in one source language (l_{src}), where l_{src} \neq l_{trg}. We focus on the very basics and examine the first step in our pipeline — input representations, holding everything else constant. Instead of measuring absolute cross-entropy scores at one data size, we evaluate the relative differences between development (dev) set score distributions between languages.

Controlled experiments as basic research for scientific understanding of language data  Using the UN Parallel Corpus, the data from which the MT results in Junczys-Dowmunt et al. (2016) stem, we perform a series of controlled experiments with the Transformer, holding the hyperparameter settings for all 30 one-to-one language directions from the 6 languages constant. We control for size (from 10^2 to 10^6 lines) and language with respect to representational granularity. We examine 3 primary representation types/levels — character, byte (UTF-8), and word, and upon encountering some unusual phenomena, we perform a secondary set of controls with 5 alternate representations — on the character level: Pinyin and Wubi (ASCII representations for ZH phones and character strokes, respectively), on the byte level: code page 1256 (for AR) and code page 1251 (for RU), and on the word level: BPE. These symbolic variants allow us to manipulate the statistical properties of the representations, while staying as “faithful” to the language as possible. We adopt this symbolic data-centric approach because we would like to more directly interpret the confounds, if any, that make language data different from other data types. We operate on a smaller data size range as this is more common in traditional language sciences and one of our higher goals is to bridge an understanding between language sciences and engineering (the latter being the dominant focus in NLP), and between traditional symbolic sciences and ML. We run statistical tests to identify the strongest correlates of performance and to assess whether the differences between the mean performance of different groups are indeed significant. We are concerned not with the absolute scores, but with the differences between score distributions from different languages.

Fair evaluation with multitexts  Multitexts are multiway parallel corpora. The UN Parallel Corpus is a 6-way parallel corpus consisting of manually translated UN documents from the 25-year period between 1990 and 2014. We use the UN Parallel Corpus because it contains languages conventionally regarded as morphologically rich and poor, has quality and size sufficient for evaluation, and more importantly, it comes as raw texts (untokenized), unlike both of the corpora that Mielke et al. (2019) used. Detokenization (esp. the evaluation thereof) is not a trivial task.

Fair information-theoretic evaluation metric  Most sequence-to-sequence models are optimized using a cross-entropy loss, defined as:

$$H(t, s) = - \sum_{i=1}^{N} \log_2 p(t_i \mid t < i, s)$$

where t is the sequence of tokens to be predicted, t_i refers to the i^{th} token in that sequence, s is the sequence of tokens conditioned on, and N = |t|. It is customary to report scores as PP, which is $2^{-\frac{1}{N}H(t, s)}$, i.e. 2 to the power of the cross-entropy averaged by the number of tokens in the dev

---

Footnote: Two testing/evaluation approaches — data-centric: hold the algorithm constant and tweak data, vs. algorithm-centric: hold data constant and tweak the algorithm.
data. Cotterell et al. (2018) proposed to use “renormalized” PP to evaluate LMs tokenwise fairly by dividing the overall bits per utterance/sequence by one constant token count in any one arbitrary language (e.g. so to arrive at “bits per character” in one language to evaluate all languages). But we find that it is not necessary to assign a perspective that is centered on any one particular language, when we can evaluate simply by the total number of bits for a larger portion of texts/sequences. This can be a fairer, more general and flexible way of evaluating data that has not been or cannot be perfectly segmented or aligned line by line. We hence used instead unnormalized PP, i.e. the total number of bits needed to encode the dev set (3,077 lines per language, after length filtering, in our case). As the implementation we used only reports PP, we transformed it back to entropy as defined above via $H(t, s) = \log_2 PP(t|s) \times N$.

**Disparity/Inequality** In the context of our CLMing experiments, we consider there to be “disparity” or “inequality” between languages $l_1$ and $l_2$ if there are significant differences between the performance distributions of these two languages with respect to each representation. Here, by performance we mean the number of bits required to encode the held-out data using a trained CLM. With 30 directions, there are 15 pairs of source languages ($l_{src1}, l_{src2}$) and 15 pairs of target languages ($l_{ trg1}, l_{ trg2}$) possible. We compare the source languages among each other, and the target languages among each other. Each $l_{src}$ or each $l_{ trg}$ consists of scores from all models trained across various sizes and directions. To assess whether the differences are significant, we perform unpaired two-sided significance tests with the null hypothesis that the score distributions for the two languages are not different. Upon testing for normality with the Shapiro-Wilk test (Shapiro & Wilk, 1965; Royston, 1995), we use the parametric unpaired two-sample Welch’s t-test (Welch, 1947) (when normal) or the non-parametric unpaired Wilcoxon test (Wilcoxon, 1945) (when not normal) for the comparisons. We use the implementation in R [R Core Team, 2014] for these 3 tests. To account for the multiple comparisons we are performing, we correct all p-values using Bonferroni’s correction (Benjamini & Heller, 2008; Dror et al., 2017) and follow Holm’s procedure (Holm, 1979; Dror et al., 2017) to identify the pairs of $l_{src}$ and $l_{ trg}$ with significant differences after correction. We report all 3 levels of significance ($\alpha \leq 0.05, 0.01, 0.001$) for a more comprehensive overview. In contrast to Dror et al. (2017), which aimed to compare the performance of different algorithms, we compare languages (in the context of computing).

**Experimental setup** The systematic, identical treatment we give to our data is described as follows with further preprocessing and hyperparameter details in Appendices A and B respectively.

After filtering length to 300 characters maximum per line in parallel for the 6 languages, we made 3 subsets of the data with 1 million lines each — one having lines in the order of the original corpus (dataset A) and two other randomly sampled (without replacement) from the full corpus (datasets B & C). Lines in all datasets are extracted in parallel and remain fully aligned for the 6 languages. For each run and each representation, there are 30 pairwise directions (i.e. one $l_{src}$ to one $l_{ trg}$) that result from the 6 languages. We trained all 150 (for 5 sizes) 6-layer Transformer models for each run using the SOCKEYE Toolkit (Hieber et al., 2018). We optimize using PP and use early stopping if no PP improvement occurs after 3 checkpoints up to 50 epochs maximum, taking the best checkpoint. Characters and bytes are supposed to mitigate the out-of-vocabulary (OOV) problem on the word level. In order to assess the effect of modeling with finer granularity more precisely, all vocabulary items appearing once in the train set are accounted for (i.e. full vocabulary on train, as in Gerz et al. (2018a,b)). But we allow our system to categorize all unknown items in the dev set to be unknown (UNK) so to measure OOVs (open vocabulary on dev (Jurafsky & Martin, 2009)). To identify correlates of performance, we perform Spearman’s correlation (Spearman, 1904) with some basic statistical properties of the data (e.g. length, $|V|$, type-token-ratio, OOV rate) as metrics — a complete list is provided in App. C. See App. D for sample construction for statistical comparisons.

3 Experimental results of primary representations

Subfigures 1a, 1b, and 1c present the mean results across 12 runs of the 3 primary representations — character, byte, and word, respectively. The x-axis represents data size in number of lines and y-axis the total conditional cross-entropy, measured in bits (Eq. 1). Each line connects 5 data points corresponding to the number of bits the CLMs (trained with training data of $10^2$, $10^3$, $10^4$, $10^5$, and $10^6$)
Number of bits (the lower the better) across data size from \(10^2\) to \(10^6\) lines plotted for all 30 directions. Subfigures (a), (b), and (c) show mean scores across 12 runs. Subfigures (d), (e), and (f) depict the corresponding information respectively sorted in 6 facets by target language and with error bars. Legend in Subfigure (g) shows the correspondence between colors and source languages, in Subfigure (h) between line types and target languages. (These figures are also shown enlarged in Appendix F. Please note that results pertinent to our first research question in this paper are summarized in Table 1.) Figures are a visual aid only. We are not concerned with the absolute scores but the distances between scores (i.e. spaces between the set of lines by \(l_{log}\)). The point here is to show the differences in Transformer’s overall learning pattern relative to the representational granularity. Language has more finer-grained dimensions with different representations and learning patterns. Hardness in modeling is relative to its representation granularity (representation relativity).

Subfigures (d), (e), and (f) display the corresponding information sorted into facets by target language, source languages represented as line types. Through these we see more clearly that results can be grouped rather neatly by target language — as implicit in the Transformer’s architecture, the decoder is unaware of the source language in the encoder. As shown in Table 1 in § 5 summarizing the number of source and target language pairs with significant differences, there are no significant differences across any source language pairs. The Transformer neutralizes source language instances. This could explain why transfer learning or multilingual/zero-shot translation (Johnson et al. 2017) is possible at all on a conceptual level.
In general, for character and byte models, most language directions do seem to converge at \(10^4\) lines to similar values across all target languages, with few notable exceptions. There are some fluctuations past \(10^4\), indicating further tuning of hyperparameters would be beneficial due to our present setting possibly working most favorably at \(10^5\). On the character level, target language \(\text{ZH}_\text{trg}\) shows a different learning pattern throughout. And on the byte level, \(\text{AR}_\text{trg}\) and \(\text{RU}_\text{trg}\) display highly unstable behavior, which we refer to as erratic. Word models exhibit Double Descent across the board (note the spike at \(10^3\)), but overall, difficult/easy languages stay consistent, with \(\text{AR}\) and \(\text{RU}\) being the hardest, followed by \(\text{ES}\) and \(\text{FR}\), then \(\text{EN}\) and \(\text{ZH}\). A practical takeaway from this set of experiments: in order to obtain more robust training results, use bytes for \(\text{ZH}\) (as suggested in \(\text{Li et al.}\) [2019a]) and characters for \(\text{AR}\) and \(\text{RU}\) (e.g. \(\text{Lee et al.}\) [2017]) — also if one wanted to avoid any “class” problems in performance disparity with words. Performance disparity for these representations is reported in Table 1 under “CHAR”, “BYTE”, and “WORD”. Do note, however, that the intrinsic performance of \(\text{ZH}\) with word segmentation is not particularly subpar. But this often does not correlate with its poorer downstream tasks results (recall results from \(\text{Junczys-Dowmunt et al.}\) [2016]). Since the notion of word in \(\text{ZH}\) is highly contested and ambiguous — i) it is often aimed to align with that in other languages so to accommodate academic theories and manual feature engineering, ii) there is great variation among different conventions, and iii) native \(\text{ZH}\) speakers identify characters as words — there are reasons to rethink this procedure now that fairer and language-independent processing in finer granularity is possible. \(\text{Li et al.}\) [2019b] questioned the necessity of CWS in Deep Learning (DL)-based \(\text{ZH NLP}\) and presented evidence in favor of character-based processing, including results from downstream NLP tasks. In Linguistics, \(\text{Duanmu}\) [2017] presented a summary on the contested nature of wordhood in (Mandarin) \(\text{ZH}\) in relation to \(\text{EN}\). A more native account of \(\text{ZH}\), however, despite a couple of dialects/varieties of it being considered a high-resource language, has not yet been fully recognized and accepted in NLP.

4 Understanding the Phenomena with Alternate Representations

To understand why some languages show different results than others, we carried out a secondary set of controlled experiments with representations targeting the problematic statistical properties of the corresponding target languages.

![Character-level remedies for ZH: Wubi vs. Pinyin.](image)

Figure 2: Character-level remedies for ZH: Wubi vs. Pinyin.

It is a “legacy interpretation” stemmed from a practical compromise from the early days in \(\text{ZH NLP}\) when the goal was to align with \(\text{EN}\) words for \(\text{MT}\). Chinese word segmentation (CWS) has been a decades-long issue in text processing. But even in \(\text{EN}\), for computing, the variability in “word” counts (from the trivial convention of whitespace tokenization) results in different bit counts, affecting file sizes. In NLP, such method of “word” counting brings about a high \(|V|\), hence different tokenization schemes have been designed to mitigate this problem. For humans, there is no consensus about the definition of “words”. Even for an academic account external of CWS, it is held to be indeterminate (see \(\text{Haspelmath}\) [2011] and references therein from the past century). \(\text{Kilgarriff}\) [1997, 2014] pointed out that “words” and “word senses” and the number thereof, in terms of lexical entries for dictionaries, are contextual and arbitrary.
We reduced the high $|V|$ in ZH with representations in ASCII characters — Pinyin and Wubi. We replaced the ZH data in these formats only on the target side and reran the experiments involving ZH$_{trg}$ on the character level. Results in Figure 2 and Table 1 show that the elimination of disparity on character level is possible if ZH is represented through Pinyin (transliteration), as in Subfigure 2c, though at the cost of the native script information. Models with ZH logographic scripts display a behavioral tendency unlike those with other (phonetic) alphabetic scripts in (Subfigure 2a), suggesting that this script pattern decomposes differently.

**Character level**  We reduced the high $|V|$ in ZH with representations in ASCII characters — Pinyin and Wubi. We replaced the ZH data in these formats only on the target side and reran the experiments involving ZH$_{trg}$ on the character level. Results in Figure 2 and Table 1 show that the elimination of disparity on character level is possible if ZH is represented through Pinyin (transliteration), as in Subfigure 2c, though at the cost of the native script information. Models with ZH logographic scripts display a behavioral tendency unlike those with other (phonetic) alphabetic scripts in (Subfigure 2a), suggesting that this script pattern decomposes differently.

**Byte level**  Length is the most salient statistical attribute that makes AR and RU outliers. To shorten their sequence lengths, we tested with alternate encodings on AR$_{trg}$ and RU$_{trg}$ — code page 1256 and 1251, which provide 1-byte encodings specific to AR and RU, respectively. Results are shown in Subfigures 3a and 3b. Not only is erraticity resolved, the number of 15 possible target language pairs with significant differences reduces from 8 with the UTF-8 byte representation to 0 (Table 1 under “ARRU”), indicating that we eliminated disparity with this optimization heuristic. Since our heuristic is a lossless and reversible transform, it shows that a complexity that is intrinsic and necessary in language does not exist in computing, however diverse they may be, as our 6 are, from the conventional linguistic typological, phylogenetic, historical, or geographical perspectives.

**Word level**  The main difference between word and character/byte models is length not being a top contributing factor correlating with performance, but instead $|V|$ is. This is understandable as word segmentation neutralizes sequence lengths. To remedy the OOV problem, we use BPE, which learns a fixed vocabulary of variable-length character sequences (on word level, as it presupposes word segmentation) from the training data. It is more fine-grained than word segmentation and is known to better model subword units for morphologically complex languages. We use the same vocabulary of 30,000 as specified in Junczys-Dowmunt et al. (2016). This reduced our averaged OOV token rate by 89-100% across the 5 sizes. The number of language pairs with significant differences reduced to 7 from 8 for word models. While BPEs are still not as effective as our character/byte variants, their results show how more fine-grained modeling contributes positively to closing the disparity gap.

## 5 Meta results, analyses, and discussion

**Performance disparity**  Table 1 lists the number of language pairs with significant differences under the representations studied. Since it is possible for our character and byte models to effect no performance disparity for the same languages on the same data, a complexity intrinsic to language does not exist. In fact, the customary expectation that languages ought to perform differently is created through our word segmentation practice. Furthermore, the order of AR/RU > ES/FR > EN/ZH

---

6 which are sequences with an implicit/explicit pattern made up of consonants and vowels

7 aside from its statistical properties related to length and vocabulary. To show something is not necessarily true, only 1 counter observation is needed.
Table 1: **Disparity Table** Number of language pairs out of 15 with significant differences, with respective p-values. ARRU\(_t\) refers to AR & RU being optimized only on the target side; whereas ARRU\(_{s,t}\) denotes optimization on both source and target sides (relevant for directions AR-RU and RU-AR).

<table>
<thead>
<tr>
<th>p-value</th>
<th>CHAR</th>
<th>Pinyin</th>
<th>Wubi</th>
<th>BYTE</th>
<th>ARRU(_t)</th>
<th>ARRU(_{s,t})</th>
<th>WORD</th>
<th>BPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>0.01</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>0.001</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2: Target language pairs with significant differences indicate that the 2 languages are not equally/similarly good or equally/similarly bad. 15 (non-directional) language pairs total possible from 30 language directions, p=0.001.

<table>
<thead>
<tr>
<th>LANG(_{trg}) PAIR</th>
<th>CHAR</th>
<th>Pinyin</th>
<th>Wubi</th>
<th>BYTE</th>
<th>ARRU(_t)</th>
<th>ARRU(_{s,t})</th>
<th>WORD</th>
<th>BPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR-EN</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>AR-ES</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EN-ES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>AR-FR</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>EN-FR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ES-FR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>AR-RU</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>EN-RU</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ES-RU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FR-RU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>AR-ZH</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>EN-ZH</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ES-ZH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FR-ZH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RU-ZH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

(Figure 1c) resembles the idea of morphological complexity. Considering there are character-internal meaningful units in languages with logographic script such as ZH (cf. Zhang & Komachi (2018)) that are rarely captured, studied, or referred to as “morphemes”, this goes to show that linguistic morphology, along with its complexity, as it is practiced today and that which has occurred in the NLP discourse thus far, has only been relevant on the “word” level, conceptually constrained by unstandardizable units such as “words” (and “sentences”). The definition of word, however, has been recognized as problematic for a very long time in the language sciences (cf. Footnote 5).

While the lack of significant differences between pairs of source languages would signify neutralization of source language instances, it does not mean that source languages have no effect on target. For our byte solutions with code pages, we experimented also with source side optimization in the directions that involve AR/RU as source. This affected the distribution of the disparity results for that representation — with 2 pairs being significantly different (see Table 1 under “ARRU\(_{s,t}\)”). We defer further investigation on the nature of source language neutralization to future work.

Target language pairs with significant differences are summarized in Table 2. We show that morphological complexity can be empirically eliminated in this one-setting-for-all configuration with a 6-layer network, no hyperparameter tuning, and a maximum line length of 300 characters (and its corresponding equivalence in other representations) as constrained by our hardware and compute time listed in App. A and current data availability. A more analytical solution can be obtained through data statistics (see App. E). A conceptual solution lies in the definition of “words” and morphology.

**Sample-wise Double Descent (DD)** Sample-wise non-monotonicity/DD (Nakkiran et al., 2020) denotes a degradation followed by an improvement in performance with increasing data size. We

---

8 But there are no reasons why we cannot adopt a statistical science of language in finer granularities beyond/without “words”, with standardized units (characters/bytes) and/or continuous representations. Resources, e.g. quality parallel data or contrast sets, can serve both data science and ML interpretation and evaluation well.
notice word models and character models with ZH
trg, i.e. models with high target |V|, are prone
to exhibit a spike at $10^3$. A common pattern for these is the ratio of target training token count to
number of parameters falls into $O(10^{-4})$ for $10^2$ lines, $O(10^{-3})$ at $10^3$, $O(10^{-2})$ at $10^4$, and $O(10^{-1})$
for $10^5$ lines and so on. But for more atomic units such as alphabetic (not logographic) characters
(may it be Latin, Cyrillic, or Abjad) and for bytes, this progression instead begins at $O(10^{-3})$ at $10^2$
lines. Instead of thinking this spike of $10^3$ as irregular, we may instead want to think of this learning
curve as shifted by 1 order of magnitude to the right for characters and bytes and/or the performance
at $10^2$ lines for words and ZH-characters due to being overparameterized and hence abnormal. This
would also fit in with the findings by Belkin et al. (2019) and Nakkiran et al. (2020) attributing DD to
morphology or “words” in our minds, but rather standards related to internationalization and
localization, such as character encoding — something which has thus far been sorely overlooked in
our discourse and curricula. We also believe that a more fine-grained statistical data science can well
complement algorithmic analyses with a view that is more empirically robust (i.e. experimentally
verifiable) and more relevant to machine processing, contributing to a more generalizable and inter-
pretable pool of knowledge for ML/NNs/DL. A more comprehensive study can lead us not only to
new scientific frontiers, but also to better designs and evaluation, benefitting the development of a
more general, diverse and inclusive AI.

6 Conclusion

Summary We investigate whether the performance disparity between languages which have been
traditionally considered morphologically rich (AR and RU) and poor (ZH) in the 6-layer Transformer
CLM is justified due to morphological complexity and find that it is not. Performance disparity can
be explained by data statistics and in the context of computing, it can be eliminated by optimization
on length and |V| through character/byte representations. In fact, morphological complexity is not
a necessary concept in computing because “word” is not a necessary concept in computing, unless
we make it so through word segmentation. A morphological complexity hierarchy can result simply
through word segmentation. Furthermore, there are many possible interpretations to “words” for
humans and since morphology is defined with the concept of “word”, there is no stable ground for
assessing this complexity. Representational units of finer granularity were shown to help eliminate
performance disparity though at the cost of longer sequence length, which can have a negative impact
on robustness. In addition, we found all word models and character models with ZH
trg to behave similarly in their being prone to exhibit a peak (as sample-wise DD) around $10^3$ lines in our setting.

Outlook ML has enabled greater diversity in NLP [Joshi et al. 2020]. Fairness, in the elimination
of disparity, does not require big data. This paper made a pioneering attempt to bridge research in
NNs/DL, language sciences, and language engineering through a data-centric perspective. Multilin-
guiality is real and relevant in computing not due to canonical, structural linguistic concepts such
as morphology or “words” in our minds, but rather standards related to internationalization and
localization, such as character encoding — something which has thus far been sorely overlooked in
our discourse and curricula. We also believe that a more fine-grained statistical data science can well
complement algorithmic analyses with a view that is more empirically robust (i.e. experimentally
verifiable) and more relevant to machine processing, contributing to a more generalizable and inter-
pretable pool of knowledge for ML/NNs/DL. A more comprehensive study can lead us not only to
new scientific frontiers, but also to better designs and evaluation, benefitting the development of a
more general, diverse and inclusive AI.
ETHICS STATEMENT: FAIRNESS CONCERNS FOR MULTILINGUALITY

Clearer nomenclature  If/When the intent is not to explicitly model linguistic morphology in computing, one can simply describe languages and their statistical profiles with respect to their representational granularity in characters or bytes (which are and can be exhaustively standardized in computing), or refer to sequences as longer/shorter or having a higher/lower vocabulary size when comparing them with each other, rather than “richer”/“poorer” based on concepts (e.g. “words”, “sentences”) that can be ambiguous, contested, and inaccessible to many.

Accessibility  Language communities who are unfamiliar with languages similar to dominant languages or those who are reluctant to conform to one structurally similar interpretation should not have to feel inadequate in processing their own language “from scratch”, if they so choose to. As technologists, we can help take equitable measures to make fairer data representations and infrastructure available. A “word”-free view of language preempts linguistic/cultural hegemony and such an interpretation would also help make the analyses of language data more objective and clearer.

Scarcity of quality and/or multiway data for science, evaluation, and documentation  With the rise of multilingual models comes an alleged decrease in reliance on parallel corpora for MT. But data, esp. high-quality, textual and multimodal multiway parallel data, play not only an important role in scientific research, but also one for historical/cultural documentation. And as this paper shows, they can also serve as evaluation data for ML models for better understanding and interpretation. As challenge sets, data for machine processing would need to be statistically diverse and challenging. Parallel data from previous years have often come in the form of bitexts (2-way parallel text data), usually “word”-tokenized where real length information has been compromised. (In fact, the Bible data from [Mayer & Cysouw](2014) came with another confound — the ZH numeral ‘’一’’ (“one”) is recognized as a dash (punctuation) and hence tokenized with surrounding whitespaces.) At the time of our present study (2019-present), the UN Parallel Corpus was the only unperturbed, fully multiway parallel data sufficient for reliable evaluation for our size range. Data for science, evaluation, and documentation require long-term, stable platform(s) and support. There are many forms of data science. But in terms of having a sustainable practice to collect and curate good data, and exercising enough of a science with that data so to improve collective intelligence and mutual understanding, instead of looking at data from an utilitarian point of view only for consumer application purposes, there seems to be room for improvement still. We hope our work could help effect a positive change in this direction.

ACKNOWLEDGMENTS

The author thanks all anonymous reviewers and PCs as well as the following individuals for their informative comments on previous version(s) of this paper (names in alphabetic order by last name): Benjamin Börschinger, Jordan Boyd-Graber, Christian Buck, Kyunghyun Cho, Kenneth Church, Miryam de Lhoneux, Rotem Dror, San Duanmu, Chris Dyer, Roman Flury, John Goldsmith, Yannic Kilcher, Sandra Kübler, Thomas McCollan, Jason Naradowsky, Ryokan Ri, Anna Rogers, Mark Rowan, Nikunji Saunshi, Rico Sennrich, Ekaterina Vylomova, Jason Weston, and Kie Zuraw.

We also thank Timothy Baldwin, Kevin Duh, Daniela Gerz, Kenneth Heafield, Felix Hieber, Marcin Junczys-Dowmunt, Gal Kaplan, Dan Klein, Sabrina Mielke, Mathias Müller, Nakikiran Preetum, and Annette Rios for their kind responses in correspondence concerning related work and software. We thank Ismail Moukadiri for his help with the Arabic language.

Some initial experiments were supported by the computing resources at the Department of Informatics and Computational Linguistics at UZH and the Institute of Neuroinformatics at UZH and ETH Zurich (for which we especially thank Richard H.R. Hahnloser, Nikola I. Nikolov, Yuhuang Hu, and Pawel Pyk).

---

Please see further discussion in [Wan](2022).
REFERENCES


Lin Chen, YiFei Min, Mikhail Belkin, and Amin Karbasi. Multiple descent: Design your own generalization curve, 2020.


Martin Haspelmath. The indeterminacy of word segmentation and the nature of morphology and syntax. Folia Linguistica, 2011.


APPENDICES

A DATA SELECTION AND PREPROCESSING DETAILS 17
B HYPERPARAMETER SETTING 18
C CORRELATION STATISTICS 19
D STATISTICAL COMPARISONS 20
E DATA STATISTICS 21
F ENLARGED FIGURES FOR ALL 30 LANGUAGE DIRECTIONS (AGGREGATE RESULTS FROM ALL RUNS) 25
A DATA SELECTION AND PREPROCESSING DETAILS

The UN Parallel Corpus v1.0 (Ziemski et al., 2016) consists of manually translated UN documents from 1990 to 2014 in the 6 official UN languages. Therein is a subcorpus that is fully aligned by line, comprising the 6-way parallel corpus we use. We tried to have as little preprocessing or filtering as necessary to eliminate possible confounds. But as the initial runs of our experiment failed due to insufficient memory on a single GPU with 12 GB VRAM\footnote{GPUs used for experiments in this paper range from a NVIDIA TITAN RTX (24 GB), NVIDIA GeForce RTX 2080 Ti (11 GB), a GTX Titan X (12 GB), to a GTX 1080 (8 GB). All jobs were run on a single GPU setting. Some word-level experiments involving AR\textsubscript{en} or RU\textsubscript{en} at 10\textsuperscript{6} had to be run on a CPU as 24 GB VRAM were not sufficient. Models with higher maximum sequence lengths (e.g. byte models) were trained with 24 GB VRAM. Difference in equipment does not necessarily lead to degradation/improvement in scores.}, we filtered out lines with more than 300 characters in any language in lockstep with one another for all the 6 languages such that the subcorpora would remain parallel, thereby keeping the material of each language semantically equivalent to one another. 8,944,859 lines for each language were retained as our training data which cover up to the 75\textsuperscript{th} percentile in line length for all 6 languages. In order to monitor the effect of data size, we made subcorpora of each language in 5 sizes by heading the first $10^2$, $10^3$, $10^4$, $10^5$, $10^6$ lines\footnote{The terms “line” and “sentence” have been used interchangeably in the NLP literature. We use “line” to denote a sequence that ends with a newline character and “sentence” as one with an ending punctuation. Most parallel corpora, such as ours, are aligned by line, as a line may be part of a sentence or without an ending punctuation (e.g. a header/title). Using a standardized unit such as “line” would also be a fairer measure to language/scriptonies continuae (languages/scripts with no explicit punctuation).}. We refer to this as dataset A. In addition, to better understand and verify the consistency of the phenomena observed, we made 2 supplemental datasets by shuffling the 8,944,859 lines two different times randomly and heading the number of lines in our 5 sizes for each language, again in lockstep with one another (datasets B and C).

The systematic training regime that we gave to our language directions is identical for all and we controlled also for seeds. For each of the 3 primary representations — character, byte, and word, we performed:

- 5 runs in 5 sizes ($10^2$ – $10^6$ lines): A0 (seed=13), B0 (13), C0 (9948), A1 (9948), A2 (265), and
- 7 more runs in 4 sizes ($10^2$ – $10^5$ lines): A3 (777), A4 (42), A5 (340589), A6 (1000), A7 (83146), B1 (9948), & C1 (13).

Figure 1 shows results from all 12 runs in all sizes for the primary representations.

For the alternate/secondary representations, we performed 3 runs each in 5 sizes ($10^2$-$10^6$ lines) (A0, B0, & C0). Due to limitations in computing resources, we were not able to perform as many runs as the primary representations. But important for our statistical comparisons is that we evaluate based on an equal number of runs and on the same data for all candidates. Tables 1 & 2 are the results.

For each run and each size, there are 30 pairwise directions (i.e. 1 source language to 1 target language, e.g. AR-EN for Arabic to English) that result from the 6 languages. We trained all 150 jobs (30 directions x 5 sizes) for each run and representation using the Transformer model\footnote{https://github.com/lxyu/pinyin} as supported by the SOCKEYE Toolkit\footnote{https://github.com/fxsjy/jieba} (Hieber et al., 2018) (version 1.18.85), based on MXNet (Chen et al., 2015). A detailed description of the architecture of the Transformer can be found in (Vaswani et al., 2017). The same set of hyperparameters applies to all and its values are listed in Appendix B.

For character modeling, we used a dummy symbol to denote each whitespace. For byte, we turned each UTF-8-encoded character into a byte string in decimal value, such that each token is a number between 0 and 255, inclusive. For word, we followed (Junczys-Dowmunt et al., 2016) and used the Moses tokenizer\footnote{https://github.com/fxsjy/jieba} (Koehn et al., 2007) as is standard in NMT practice when word tokenization is applied and Jieba\footnote{https://github.com/lxyu/pinyin} for segmentation in ZH.

Pinyin is a romanization of ZH characters based on their pronunciations and Wubi an input algorithm that decomposes character-internal information into stroke shape and ordering and matches these to 5 classes of radicals (Lunde, 2008). For Pinyin, we used the implementation from https://github.com/lxyu/pinyin in the numerical format such that each character/syllable
followed by a single digit indicating its lexical tone in Mandarin. For Wubi, we used the dictionary from the implementation from https://github.com/arcsecw/wubi

We have implemented all representations such that they would be reversible even when the sequence contains code-mixing. Additional code will be made available at https://github.com/dadasci

We used the official dev set as provided in (Ziemski et al., 2016), 3,077 lines per language remained from 4,000 after filtering line length to 300 characters. Data statistics is provided in Appendix E for reference.

Notes on training time Each run of 30 directions in 5 sizes took approximately 8-12 days for character and byte models. Byte models generally took longer — hence training time is positively correlated with length (concurring with observations by Cherry et al. (2018) as they compared character with BPE models). A maximum length of 300 characters entails a maximum length of at least 300 bytes in UTF-8. Each run of word models (30 directions, 5 sizes) took about 6 days (excluding the training of some 7-9 directions out of 30 per run involving AR trg or RU trg at 10^6 on word level which took about 12-18 hours each direction to train on a CPU as these required more space and would run out of memory (OOM) on our GPUs otherwise). These figures do not include the additional probing experiments described in §4.

B HYPERPARAMETER SETTING

- encoder transformer;
- decoder transformer;
- num-layers 6:6;
- num-embed 512:512;
- transformer-model-size 512;
- transformer-attention-heads 8;
- transformer-feed-forward-num-hidden 2048;
- transformer-activation-type relu;
- transformer-positional-embedding-type fixed;
- transformer-preprocess d; transformer-postprocess drn;
- transformer-dropout-attention 0.1;
- transformer-dropout-act 0.1;
- transformer-dropout-prepost 0.1;
- batch-size 15;
- batch-type sentence;
- max-num-checkpoint-not-improved 3;
- max-num-epochs 50;
- optimizer adam;
- optimized-metric perplexity;
- optimizer-params epsilon: 0.000000001, beta1: 0.9, beta2: 0.98;
- label-smoothing 0.0;
- learning-rate-reduce-num-not-improved 4;
- learning-rate-reduce-factor 0.001;
- loss-normalization-type valid;
- max-seq-len 300 for character, word, and BPE, 672 for all bytes, 688 for Wubi, 680 for Pinyin;
- checkpoint-frequency/interval 4000.

(For smaller datasets, the end of 50 epochs is often reached before the first checkpoint. Since SOCKEYE only outputs scores at checkpoints, we adjusted the checkpoint frequency as follows to get a score outputted by the end of 50 epochs: 1000 for 100 lines for all character & byte instances, 400 for 100 lines for word and 500 for 100 lines BPE, 3450 for 1000 lines for word & BPE. For the very few cases that this default does not suffice due to bucketing of similar length sequences, we manually set the checkpoint frequency to the last batch.)
C  CORRELATION STATISTICS

Best correlating metrics, i.e. the union of top 3 metrics for all representations. For each representation, the top 3 metrics are boldfaced. All correlations are highly significant \((p < 10^{-30})\), except for min source length for word \((p \approx 0.0001)\) and min target length for word \((p \approx 0.3861)\).

<table>
<thead>
<tr>
<th>Metric</th>
<th>CHAR</th>
<th>Pinyin</th>
<th>Wubi</th>
<th>BYTE</th>
<th>ARRU</th>
<th>ARRU_{int}</th>
<th>WORD</th>
<th>BPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum length (target)</td>
<td>0.84</td>
<td>0.85</td>
<td>0.86</td>
<td>0.60</td>
<td>0.84</td>
<td>0.84</td>
<td>-0.02</td>
<td>0.65</td>
</tr>
<tr>
<td>minimum length (source)</td>
<td>0.82</td>
<td>0.84</td>
<td>0.85</td>
<td>0.57</td>
<td>0.84</td>
<td>0.84</td>
<td>0.10</td>
<td>0.64</td>
</tr>
<tr>
<td>number of tokens (source)</td>
<td>-0.78</td>
<td>-0.81</td>
<td>-0.82</td>
<td>-0.60</td>
<td>-0.81</td>
<td>-0.81</td>
<td>-0.59</td>
<td>-0.83</td>
</tr>
<tr>
<td>TTR (target)</td>
<td>0.83</td>
<td>0.83</td>
<td>0.84</td>
<td>0.48</td>
<td>0.81</td>
<td>0.81</td>
<td>0.61</td>
<td>0.83</td>
</tr>
<tr>
<td>(</td>
<td>V</td>
<td>) (source)</td>
<td>-0.54</td>
<td>-0.51</td>
<td>-0.51</td>
<td>-0.50</td>
<td>-0.67</td>
<td>-0.68</td>
</tr>
<tr>
<td>data size in lines</td>
<td>-0.80</td>
<td>-0.83</td>
<td>-0.83</td>
<td>-0.59</td>
<td>-0.81</td>
<td>-0.81</td>
<td>-0.62</td>
<td>-0.86</td>
</tr>
<tr>
<td>OOV token rate (target)</td>
<td>0.69</td>
<td>0.66</td>
<td>0.66</td>
<td>0.47</td>
<td>0.67</td>
<td>0.68</td>
<td>0.66</td>
<td>0.62</td>
</tr>
<tr>
<td>OOV type rate (target)</td>
<td>0.70</td>
<td>0.71</td>
<td>0.72</td>
<td>0.47</td>
<td>0.69</td>
<td>0.70</td>
<td>0.65</td>
<td>0.62</td>
</tr>
<tr>
<td>TTR (source)</td>
<td>0.67</td>
<td>0.71</td>
<td>0.71</td>
<td>0.60</td>
<td>0.81</td>
<td>0.81</td>
<td>0.56</td>
<td>0.82</td>
</tr>
</tbody>
</table>

The full list of metrics used for the correlation analysis is:

1. minimum length (source),
2. minimum length (target),
3. maximum length (source),
4. maximum length (target),
5. median length (source),
6. median length (target),
7. mean length (source),
8. mean length (target),
9. length std (source),
10. length std (target),
11. data size in lines,
12. number of parameters,
13. number of types (\(|V|\)) (source),
14. number of types (\(|V|\)) (target),
15. number of tokens (source),
16. number of tokens (target),
17. type-token-ratio (TTR) (source),
18. type-token-ratio (TTR) (target),
19. OOV type rate (source),
20. OOV type rate (target),
21. OOV token rate (source),
22. OOV token rate (target),
23. token ratio,
24. target type-to-parameter ratio,
25. target token-to-parameter ratio,
26. distance between the TTRs of source and target \(= (1 - \text{TTR}_{src} / \text{TTR}_{trg})^2\),
27. token-to-parameter ratio (i) = (median length source * median length target * num_lines) / num_parameters,
28. token-to-parameter ratio (ii) = (num_source_tokens * num_target_tokens) / num_parameters.
D  STATISTICAL COMPARISONS

Recall the definition and method for our Disparity/Inequality assessment from §2.

In the context of our CLM-ing experiments, we consider there to be “disparity” or “inequality” between languages \(l_1\) and \(l_2\) if there are significant differences between the performance distributions of these two languages with respect to each representation. Here, by performance we mean the number of bits required to encode the held-out data using a trained CLM. With 30 directions, there are 15 pairs of source languages \((l_{src1}, l_{src2})\) and 15 pairs of target languages \((l_{trg1}, l_{trg2})\) possible. We compare the source languages among each other, and the target languages among each other. Each \(l_{src}\) or each \(l_{trg}\) consists of scores from all models trained across various sizes and directions. To assess whether the differences are significant, we perform unpaired two-sided significance tests with the null hypothesis that the score distributions for the two languages are not different. Upon testing for normality with the Shapiro-Wilk test (Shapiro & Wilk, 1965; Royston, 1995), we use the parametric unpaired two-sample Welch’s t-test (Welch, 1947) (when normal) or the non-parametric unpaired Wilcoxon test (Wilcoxon, 1945) (when not normal) for the comparisons. We use the implementation in R (R Core Team, 2014) for these 3 tests. To account for the multiple comparisons we are performing, we correct all p-values using Bonferroni’s correction (Benjamini & Heller, 2008; Dror et al., 2017) and follow Holm’s procedure (Holm, 1979; Dror et al., 2017) to identify the pairs of \(l_1\) and \(l_2\) with significant differences after correction. We report all 3 levels of significance (\(\alpha \leq 0.05, 0.01, 0.001\)) for a more comprehensive overview. In contrast to Dror et al. (2017), which aimed to compare the performance of different algorithms, we compare languages (in the context of computing).

To get samples for the statistical comparison results for the Disparity Table (Table 1):

For each representation, we used 3 runs (A0, B0, C0) in 5 sizes (\(10^2-10^6\) lines) for each \(l_{src}\) and each \(l_{trg}\). There are:

6 \(l_{src}\) (AR\(_{src}\), EN\(_{src}\), ES\(_{src}\), FR\(_{src}\), RU\(_{src}\), ZH\(_{src}\)) and
6 \(l_{trg}\) (AR\(_{trg}\), EN\(_{trg}\), ES\(_{trg}\), FR\(_{trg}\), RU\(_{trg}\), ZH\(_{trg}\)).

We compare pairwise among the \(l_{src}\). The 15 pairs are:

AR\(_{src}\)-EN\(_{src}\), AR\(_{src}\)-ES\(_{src}\), AR\(_{src}\)-FR\(_{src}\), AR\(_{src}\)-RU\(_{src}\), AR\(_{src}\)-ZH\(_{src}\),
EN\(_{src}\)-ES\(_{src}\), EN\(_{src}\)-FR\(_{src}\), EN\(_{src}\)-RU\(_{src}\), EN\(_{src}\)-ZH\(_{src}\),
ES\(_{src}\)-FR\(_{src}\), ES\(_{src}\)-RU\(_{src}\), ES\(_{src}\)-ZH\(_{src}\),
FR\(_{src}\)-RU\(_{src}\), FR\(_{src}\)-ZH\(_{src}\),
RU\(_{src}\)-ZH\(_{src}\).

Likewise 15 pairs among the \(l_{trg}\).

For example, for the character (primary) representation to compare between AR\(_{src}\) and EN\(_{src}\), we construct the sample for AR\(_{src}\) \(\text{sample}_{AR_{src}}\) and the sample for EN\(_{src}\) \(\text{sample}_{EN_{src}}\) as follows:

Out of the 30 CHAR directions, there are 5 directions involving AR\(_{src}\) trained for each run and data size (i.e., the directions: AR-EN, AR-ES, AR-FR, AR-RU, AR-ZH).

For each direction, there are 15 models trained (3 runs x 5 directions). We take all 75 CHAR models (15 models x 5 directions) involving AR\(_{src}\) as \(\text{sample}_{AR_{src}}\). That’s a sample of size 75.

Likewise, for EN\(_{src}\) (5 directions: EN-AR, EN-ES, EN-FR, EN-RU, EN-ZH), we also have 75 data points for \(\text{sample}_{EN_{src}}\). (Likewise also for all 6 \(l_{src}\) and all 6 \(l_{trg}\).)

For the comparisons, we compare pairwise, i.e., with two samples each time, but with unpaired two-sample Welch’s t-test (when normal) or the non-parametric unpaired Wilcoxon test (when not normal) because \(\text{sample}_{AR_{src}}\) and \(\text{sample}_{EN_{src}}\) have one direction that is not paired: AR-EN and EN-AR. Other directions can be seen as paired, e.g., AR-ES and EN-ES as both having the same \(l_{trg}\).

\(\text{https://github.com/rtmdrr/replicability-analysis-NLP}\)
E  DATA STATISTICS

- Number of types, i.e. vocabulary size ($V$). Note that Sockeye adds for its calculations 4 additional types: <pad>, <s>, </s>, <unk>.
- Number of tokens. This excludes the 1 EOS/BOS (end-beginning of sentence) marker added by Sockeye to each line.
- Out-of-vocabulary (OOV) type rate (in %), i.e. the fraction of the types in the dev data that is not covered by the types in the training data.
- OOV token rate (in %), i.e. the fraction of tokens in the dev data that is not covered by the types in the training data.
- Type-token-ratio (in %), i.e. the ratio between the number of types and tokens in the data. This is a rough proxy for lexical diversity in that a value of 1 would indicate that no type is ever seen twice, and a value very close to 0 would indicate that very few distinct types account for almost all of the data.
- Line length (excl. EOS/BOS markers): mean±standard deviation, and the 0.25/0.75/1/10/100th percentile.

<table>
<thead>
<tr>
<th></th>
<th>100</th>
<th>1,000</th>
<th>10,000</th>
<th>100,000</th>
<th>1,000,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type stats</td>
<td>52.0 ± 1.2</td>
<td>72.0 ± 1.2</td>
<td>100.0 ± 1.0</td>
<td>52.0 ± 1.2</td>
<td>72.0 ± 1.2</td>
</tr>
<tr>
<td>Dev ratio</td>
<td>50.0 ± 1.0</td>
<td>72.0 ± 1.2</td>
<td>100.0 ± 1.0</td>
<td>52.0 ± 1.2</td>
<td>72.0 ± 1.2</td>
</tr>
<tr>
<td>Tokens</td>
<td>1,000 ± 100</td>
<td>10,000 ± 1,000</td>
<td>100,000 ± 10,000</td>
<td>1,000,000 ± 100,000</td>
<td></td>
</tr>
<tr>
<td>Type stats</td>
<td>52.0 ± 1.2</td>
<td>72.0 ± 1.2</td>
<td>100.0 ± 1.0</td>
<td>52.0 ± 1.2</td>
<td>72.0 ± 1.2</td>
</tr>
<tr>
<td>Dev ratio</td>
<td>50.0 ± 1.0</td>
<td>72.0 ± 1.2</td>
<td>100.0 ± 1.0</td>
<td>52.0 ± 1.2</td>
<td>72.0 ± 1.2</td>
</tr>
</tbody>
</table>

Statistics for dataset A

| Type stats | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 |
| Dev ratio | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 |
| Tokens   | 1,000 ± 100 | 10,000 ± 1,000 | 100,000 ± 10,000 | 1,000,000 ± 100,000 |
| Type stats | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 |
| Dev ratio | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 | 10.0 ± 1.0 |
| Tokens   | 1,000 ± 100 | 10,000 ± 1,000 | 100,000 ± 10,000 | 1,000,000 ± 100,000 |

Published as a conference paper at ICLR 2022
## Statistics for dataset C

<table>
<thead>
<tr>
<th>Language</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>0.12</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.12</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.09</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Additional Table

<table>
<thead>
<tr>
<th>Language</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.09</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.04</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Additional Table

<table>
<thead>
<tr>
<th>Language</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
<th>µ (±σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>AR</td>
<td>0.04</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
As a different set of vocabulary is learned from each training dataset and data size, BPE has a distinct dev set for each.

<table>
<thead>
<tr>
<th>Performance in TTR (%)</th>
<th>100</th>
<th>1,000</th>
<th>5,000</th>
<th>10,000</th>
<th>50,000</th>
<th>150,000</th>
<th>500,000</th>
<th>1,500,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EN</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>DE</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>RO</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>ZH</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

As a different set of vocabulary is learned from each training dataset and data size, BPE has a distinct dev set for each.
F  ENLARGED FIGURES FOR ALL 30 LANGUAGE DIRECTIONS (AGGREGATE RESULTS FROM ALL RUNS)

Figure 4: CHAR: character models
Figure 4: CHAR: character models (target language as facet)
Figure 5: CHAR with Pinyin for ZH_{\text{\tiny trg}}
Figure 5: CHAR with Pinyin for ZH_{trg} (target language as facet)
Figure 6: CHAR with Wubi for ZH_{trg}
Figure 6: CHAR with Wubi for ZH$_{trg}$ (target language as facet)
Figure 7: BYTE models with UTF-8 encoding
Figure 7: BYTE models with UTF-8 encoding (target language as facet)
Figure 8: BYTE with AR\textsubscript{trg} & RU\textsubscript{trg} optimized with code pages 1256 & 1251 (ARRU\textsubscript{trg})
Figure 8: BYTE with $\text{AR}_{\text{trg}}$ & $\text{RU}_{\text{trg}}$ optimized with code pages 1256 & 1251 (target language as facet)
Figure 9: BYTE with directions AR-RU & RU-AR optimized on both source and target sides (ARRU_{src,trg})
Figure 9: BYTE with directions AR-RU & RU-AR optimized on both source and target sides (target language as facet)
Figure 10: WORD models
Figure 10: WORD models (target language as facet)
Figure 11: BPE models
Figure 11: BPE models (target language as facet)
Fairness in Representation for Multilingual NLP: Insights from Controlled Experiments on Conditional Language Modeling

Version 1.0

ICLR 2022 camera-ready copy (20220315)