Language Complexity and Speech Recognition Accuracy: Orthographic Complexity Hurts, Phonological Complexity Doesn't

Chihiro Taguchi
University of Notre Dame
ctaguchi@nd.edu

David Chiang
University of Notre Dame
dchiang@nd.edu

Abstract

We investigate what linguistic factors affect the performance of Automatic Speech Recognition (ASR) models. We hypothesize that orthographic and phonological complexities both degrade accuracy. To examine this, we fine-tune the multilingual self-supervised pretrained model Wav2Vec2-XLSR-53 on 25 languages with 15 writing systems, and we compare their ASR accuracy, number of graphemes, unigram grapheme entropy, logographicity (how much word/morpheme-level information is encoded in the writing system), and number of phonemes. The results demonstrate that a high logographicity correlates with low ASR accuracy, while phonological complexity has no strong correlation.

1 Introduction

When a human learns a second language, a complex writing system and a complex phonological system can both be obstacles to language learning. For example, learners of a language like Chinese may spend years learning thousands of characters, while Japanese learners of English commonly struggle with mastering the two liquid phonemes /r/ and /l/ that are not distinguished in Japanese. By analogy with these observations, we may ask: do computational models of language, like humans, also struggle with these linguistic complexities? In this paper, we investigate the relationship between the accuracy of Automatic Speech Recognition (ASR) and linguistic complexity, specifically orthographic and phonological complexity.

To answer this question, this study proposes three hypotheses about factors that may make learning ASR hard.

Hypothesis 1. If a language has more character (*grapheme*) types, then ASR accuracy gets lower. This idea corresponds to the example of Chinese mentioned above.

Hypothesis 2. The more a language's writing system encodes word- or morpheme-level information, the more ASR accuracy decreases. Sproat and Gutkin (2021) call this property *logographicity*, as opposed to *phonographicity*, which means that a language's written form is more predictable from its spoken form.

Hypothesis 3. If a language has more sound (*phoneme*) types, then ASR accuracy gets lower. This idea corresponds to the example of Japanese learners of English mentioned above.

To test these hypotheses, we fine-tune the same pre-trained ASR model (Wav2Vec2-XLSR-53) on 25 languages with 15 different orthographies.

The results demonstrate a significant correlation of ASR accuracy with measures related to orthographic complexity, while no significant correlation is observed with phonological complexity.

2 Related Work

Multilingual ASR. With the successful development of Transformer-based architectures, the field of multilingual ASR has also achieved drastic improvements. Wav2Vec 2.0 (Baevski et al., 2020) is a framework for learning speech representations with a self-supervised method like BERT (Devlin et al., 2019). For each span of speech with a fixed length, this architecture first obtains a latent representation $\boldsymbol{z}$ through a feature encoder with a CNN, and computes its discretized vector $\boldsymbol{q}$ via product quantization (Jégou et al., 2011). Next, it masks some $\boldsymbol{z}$ with a certain probability; the objective of this self-supervised training is then to predict the quantized vector $\boldsymbol{q}$ for the masked representation. The pretraining step is illustrated in Figure 1.

Wav2Vec 2.0 is known to perform well for various speech recognition tasks by fine-tuning the pre-trained model. Wav2Vec2-XLSR-53 (Conneau et al., 2021), which this study employs, is a 300M-parameter model pre-trained on speech samples

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15493–15503

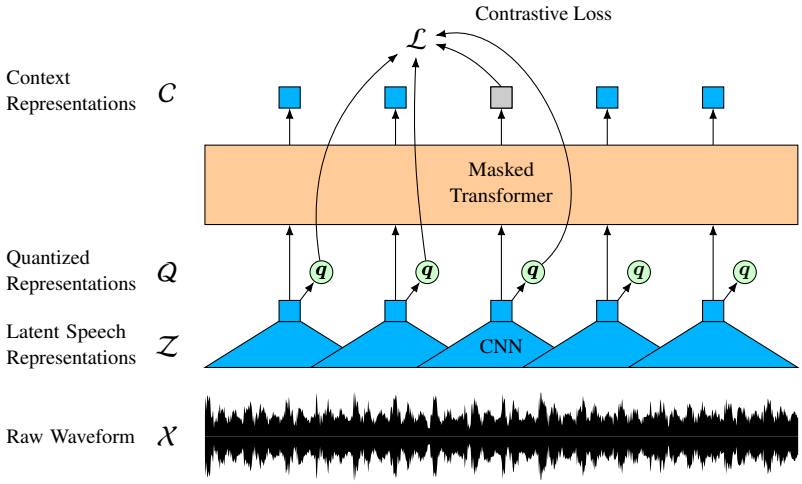

Figure 1: Visualization of the self-supervised pretraining step of Wav2Vec 2.0.

of 53 languages. It performs well for various languages, including those unseen in the pre-training step, by fine-tuning with a small amount of annotated data. Since the appearance of Wav2Vec2.0, other larger multilingual Wav2Vec 2.0 pretrained models have been developed, such as Wav2Vec2-XLS-R (Babu et al., 2021) which is pretrained on 0.5M hours from 128 languages and Wav2Vec2-BERT (Seamless Communication et al., 2023) pretrained on 4.5M hours from 143 languages.

The current state-of-the-art in ASR for some languages has been achieved by Whisper (Radford et al., 2022), which is an encoder-decoder architecture based on next token prediction with weak supervision. However, unlike Wav2Vec 2.0, a pretrained Whisper model is weakly supervised; namely, the pretraining data contains some labeled data. Importantly, fine-tuning on the languages that have labeled data in Whisper's pretraining step is known to remarkably boost the performance, while there is less observed improvement between the languages included in Wav2Vec 2.0 pretraining and those not included (Rouditchenko et al., 2023). To avoid the performance difference biased by labeled pretraining data, our work conducts experiments on Wav2Vec2-XLSR-53 through fine-tuning.

Logographicity. In some languages, the spelling of a token encodes word- or morpheme-level information that is not predictable from the token's pronunciation alone. For instance, English /raɪt/ can be spelled as <write>, <right>, <rite>, or <wright>, which all have different meanings. Sproat and Gutkin (2021) calls this property *logographicity*. To measure logographicity, they train a phoneme-to-grapheme model and look at how widely dispersed its attention is to see how context-dependent the orthography is (see Section 3.4 for details).

3 Experimental Setup and Methods

This section describes the setup and the methods used in the experiments.[1]

3.1 Dataset

In this experiment, we use Common Voice 16.1 (Ardila et al., 2020) for every language examined except English and Korean. We use LibriSpeech (Panayotov et al., 2015) for English instead because Common Voice English contains a number of non-native speech samples and Zeroth-Korean[2] because the Korean subset in Common Voice 16.1 does not have enough samples. Since LibriSpeech and Zeroth-Korean have a longer maximum audio length than Common Voice 16.1, long audio samples are filtered out. For each setting, we keep extracting training data samples until the total sample length reaches 10,000 seconds. In doing so, we aim to standardize the training dataset size across the settings rather than to rely on number of samples that vary in their audio length.

3.2 Pre-trained model and fine-tuning

We use Wav2Vec2-XLSR-53[3] as the base pretrained model for every setting and fine-tune it for each target language and writing system. The

[1]The code used in the experiments is available at: https://github.com/ctaguchi/ASRcomplexity

[2]https://github.com/goodatlas/zeroth

[3]https://huggingface.co/facebook/wav2vec2-large-xlsr-53

fine-tuning step adds supervised training to the pretrained model with Connectionist Temporal Classification (CTC, Graves et al., 2006) as illustrated in Figure 1. In the experiment, the same hyperparameters were used for every fine-tuned model; among others, we ran 20 epochs and the learning rate was set to 0.0003. Each experiment took approximately two hours on two GPU cores (NVIDIA A10). Also, punctuation was removed at the preprocessing step.[4]

3.3 Graphemes

Our first hypothesis is that a language's higher number of graphemes worsens ASR accuracy. To test this in a controlled way, we include among our fine-tuning settings some languages that have multiple scripts, namely, Japanese, Korean, and Chinese.

For Japanese, we use the following three systems: a combination of *Kanji* (Chinese characters) and *Kana* (syllabary), which is the default orthography, *Kana*-only, and *Romaji*-only (romanized Kana). For example, the word for "the Japanese language" in Japanese is <日本語>, and can be transliterated as <ニホンゴ> in *Kana* and as <nihongo> in *Romaji*. *Kana* can be uniquely mapped to *Romaji* but not vice versa. The tokenization and conversion from the default orthography into *Kana*-only is done by SudachiPy (Takaoka et al., 2018). Then, these *Kana* are romanized with the `pykakasi` library.[5]

For Korean, we use *Hangul syllables*, a syllabary writing system where each syllable character is composed of phonemic components, and *Hangul Jamo*, which is decomposed Hangul so that each character represents a phoneme. For example, /han gɯl/ is written as 한글 in the default orthography (*Hangul syllables*) and can be decomposed into six *Hangul Jamo* letters: <ㅎ> /h/, <ㅏ> /a/, <ㄴ> /n/, <ㄱ> /g/, <ㅡ> /ɯ/, and <ㄹ> /l/. In the preprocessing step, *Hangul syllables* are converted to *Hangul Jamo* by the `g2pK` library (Park, 2019).

For Chinese, three writing systems are used: *Hanzi* (Chinese characters), *Zhuyin* (semi-syllabary), and *Pinyin* (romanized). For example, the Chinese word for "the Chinese language" is <漢語> in *Hanzi*, and can be expressed as <ㄏㄢˋ ㄩˇ> in *Zhuyin* and as <hànyǔ> in *Pinyin*. *Zhuyin* and *Pinyin* can be converted to each other by rules. In our implementation, we convert *Hanzi* into *Zhuyin* and *Pinyin* using the `dragonmapper` library.[6]

To measure the impact of the grapheme size to ASR accuracy, we employ two metrics. One is to naively count all the character (grapheme) types in the training data. The other is to calculate the unigram character entropy of the training data, to capture the fact that not all character types appear with the same probability. In fact, it is known that Chinese *Hanzi* have a Zipfian distribution (Deng et al., 2014). The unigram character entropy is computed as $H(C) = -\sum_{c \in C} p(c) \log p(c)$, where C is the set of character types in the corpus.

3.4 Logographicity

Hypothesis 2 claims that the more logographic a language is (that is, the more irregular the mapping from pronunciation to orthography is in a language), the harder it is for an ASR model to transcribe the language. To measure logographicity, we follow Sproat and Gutkin (2021) and train a model that predicts a word's orthography given its phonemes and context. If a language is phonographic (*i.e.*, a word's orthography can be easily reconstructed by how it is pronounced), then the attention matrix of a learned model would only attend to a word's phonemes and would not attend its surrounding context. On the other hand, if a language is logographic and a word's pronunciation may depend on the context, then the model would attend to other surrounding words.

This attention-based metric of logographicity is calculated as follows. Given an attention matrix A and a mask matrix M, $M \circ A$ is their component-wise (Hadamard) product. The mask matrix M is a matrix of the same size as A whose entries i, j are 0 if $0 \leq i < k$ where k is the length of the target word's pronunciation and $m \leq j \leq n$ where m is the left edge of the target word in a text and n the right edge. Then, the attention spread S_w for a word w is:

$$S_w = \frac{\sum_{i,j} (M \circ A)_{i,j}}{\sum_{i,j} A_{i,j}}.$$

To apply this to a writing system of a language, one can compute the average attention spread of a word over a corpus $\mathcal{D}$:

$$S_{\text{token}} = \frac{\sum_{w \in \mathcal{D}} S_w}{|\mathcal{D}|}.$$

[4] The training code is available in the repository: https://github.com/ctaguchi/ASRcomplexity

[5] https://pykakasi.readthedocs.io

[6] https://github.com/tsroten/dragonmapper

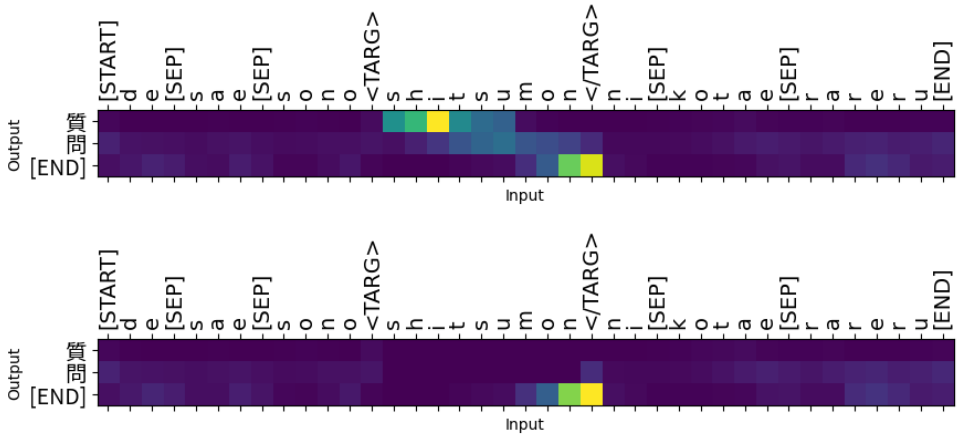

Figure 2: A visualization of attention masking. The top matrix shows the original distributions of attention scores with a Japanese phoneme input and an orthographic output of the target word. The bottom matrix has zeroed-out attention values for the cells corresponding to the target word. The logographicity score S_{token} measures how much information is retained after masking. Values near 1 are in yellow and those near 0 in dark purple.

A phonographic language would have lower S_{token} than a logographic language. See Figure 2 for an example of an attention matrix and a masked attention matrix.

In this experiment, we measure the logographicity of various languages with different types of writing systems. See Table 1 for the list of languages covered and the datasets used. For each language, we extracted 10k samples and used 80% as the training set and 20% as the validation set. Since our objective is not to measure the accuracy but to measure the attention spread, we do not prepare a test set. Punctuation is removed, and samples with numerical symbols are omitted. The phoneme string contains seven tokens, with the phonemized target token in the middle and three tokens on its left- and right-hand sides. In the phoneme string, the target word is enclosed by tag tokens `<TARG>` and `</TARG>`, and word boundaries are marked by `[SEP]`. Special characters `[START]` and `[END]` are added to the beginning and the end of both input and output, respectively.

For phonemization, we used the same converters for Japanese, Korean, and Chinese. For Thai romanization, we used PyThaiNLP's transliteration module (Phatthiyaphaibun et al., 2023). For English phonemization, we used the CMU Pronouncing Dictionary[7] via `eng-to-ipa` package.[8] Other alphabetic languages and Arabic were converted into IPA by `gruut` library.[9]

[7] http://www.speech.cs.cmu.edu/cgi-bin/cmudict
[8] https://pypi.org/project/eng-to-ipa
[9] https://github.com/rhasspy/gruut

The architecture is an attention-based sequence-to-sequence encoder–decoder model. The encoder consists of an embedding layer and a bidirectional GRU layer. The decoder consists of an embedding layer and a unidirectional GRU layer, followed by a single cross-attention layer that receives context vectors from the encoder. The attention matrix that we focus on here is the result of multi-head attention before residual connection and layer normalization. The hyperparameters are the same as the implementation by Sproat and Gutkin (2021), except that we run 10 epochs with a batch size of 64.

3.5 ASR Metrics

To measure ASR accuracy, we employ two approaches in this experiment: Character Error Rate (CER) and Calibrated Errors Per Second (CEPS). The following describes the details of these metrics.

3.5.1 Character Error Rate

CER is calculated as:

$$\text{CER} = \frac{S + D + I}{N}$$

where S, D, I are the numbers of substitution, deletion, and insertion errors, respectively, and N is the length of the reference text. Though Word Error Rate (WER) is also a widely used metric in the ASR literature, we do not use it in this experiment because the notion of *word* or *token* varies across languages. For example, Tatar, an agglutinative language, can take multiple inflectional suffixes to form one token as in (1), where the English translation is expressed with seven tokens.

Language	Script type	Dataset
Chinese	logographic	LCCC (Wang et al., 2020)
Japanese	logographic/syllabary	SNOW (Maruyama and Yamamoto, 2018)
Korean	syllabary	Korean Parallel Corpora (Park et al., 2016)
Thai	abugida	ThaiGov V2 Corpus (Phatthiyaphaibun et al., 2023)
Arabic	abjad	Rasaif Classical-Arabic–English Parallel Texts
English	alphabetic	Europarl (Koehn, 2005)
French	alphabetic	Europarl
Italian	alphabetic	Europarl
Czech	alphabetic	Europarl
Swedish	alphabetic	Europarl
Dutch	alphabetic	Europarl
German	alphabetic	Europarl

Table 1: Details of the datasets for measuring logographicity. The original Arabic data were published on https://rasaif.com.

(1) *tanı-ş-qan-ıbız-ğa*
get.to.know-each.other-participle-our-to
'to our getting to know each other'

The use of WER can unfairly harm the evaluation of such agglutinative languages compared to analytic languages. In addition, some languages and writing systems are less clear as to what a word is, particularly those written without a whitespace character (Japanese, Chinese, and Thai in our experiments). In contrast, the notion of a (Unicode) character is always clear, so we use CER using Unicode characters.

3.5.2 Errors Per Second

A problem with ASR evaluation metrics that are solely based on text like WER and CER is that they are not comparable across languages or scripts. Consider, for example, evaluating the same system on the same data, but represented in two ways: one where the characters are *Hangul Jamo* and one where the characters are *Hangul syllables*. The two settings should have the same accuracy, but will have different CERs.

We make the following simplifying assumptions. First, all languages communicate the same amount of information per second (Coupé et al., 2019) and therefore times are comparable across languages. Second, speech is divided into equal-length slices of τ seconds each. Third, an ASR error is an event that occurs at a single point in time, and errors are Poisson-distributed, so the probability that a given slice has k errors is

$$P(k;\lambda) = \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!}.$$

The parameter λ can be thought of as the number of errors per second, which we propose to use as an error rate that is comparable across different segmentations, writing systems, and languages. We now describe how to estimate λ from a run of an ASR system on test data.

First, define a slice to be a character, a word, or something else, and consider the ASR output and reference transcription as strings of slices. Let τ be the average length of a slice, in seconds, and let n be the number of slices in the reference. Compute the Levenshtein distance d between the output and the reference, and let $p = d/n$ be the usual normalized Levenshtein distance.

Recall that we assumed that an error occurs at a single point in time, so a single slice could contain more than one error. However, we can only detect whether a slice has at least one error; we can't distinguish between a slice with one error versus a slice with two errors. The probabilities of a slice having no errors and at least one error are

$$P_\lambda(k=0) = e^{-\lambda\tau}$$
$$P_\lambda(k>0) = 1 - e^{-\lambda\tau}$$

so the log-likelihood is

$$L(\lambda) = pn\log(1-e^{-\lambda\tau}) - (1-p)n\lambda\tau$$

and the maximum-likelihood estimate of λ is

$$\lambda = \frac{1}{\tau}\log\frac{1}{1-p}. \quad (2)$$

We call this the *calibrated errors per second (CEPS)*. Note that for $p \ll 1$, this reduces to $\lambda \approx \frac{p}{\tau}$,

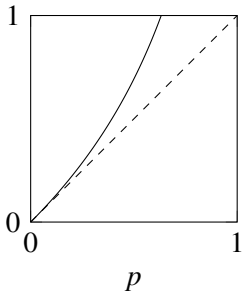

Figure 3: Calibrated errors per second (solid) compared with raw errors per second (dashed), assuming $\tau = 1$.

the raw number of errors per second, but for larger p, the CEPS grows faster than the raw errors per second, accounting for the fact that slices may have more than one error.

4 Results

This section reports the results of the experiments to test the three hypotheses. Table 2 summarizes the results across languages and metrics. Comparisons with different writing systems within Japanese, Korean, and Chinese clearly demonstrate that logographic writing systems like Japanese *Kanji* and Chinese *Hanzi*, as well as writing systems that are phonographic but have hundreds of syllable characters like Korean *Hangul syllables*, make ASR models harder to learn to transcribe correctly. Indeed, there is a significant correlation ($p < 0.05$) between CER and orthography-related variables (the number of graphemes, unigram entropy, and logographicity score), as shown in the correlation matrix in Table 3. On the other hand, there is no significant correlation ($p > 0.05$) in Table 3 between ASR accuracy and the number of phoneme types.

We can also observe that CEPS mitigates the score deterioration coming from purely orthographic differences. For instance, though *Hangul syllables* and *Hangul Jamo* only differ in how phoneme characters are encoded, we see a large difference in CER (28.21 and 16.72, respectively). However, in CEPS, *Hangul syllables* have a slightly higher score than *Hangul Jamo* (2.63 and 3.23, respectively). In addition, as Table 3 demonstrates, CEPS has less significant correlation with the orthographic factors ($|C|$: 0.49, $H(C)$: 0.41, S_{token}: 0.61) than CER does ($|C|$: 0.85, $H(C)$: 0.81, S_{token}: 0.76).

We also ran experiments on additional languages with phonographic scripts (i.e., alphabetic or abugida writing systems), for which we were unable to measure logographicity due to lack of reliable grapheme-to-phoneme tools. These results are summarized in Table 4. In this case, we see less but moderate positive correlations of CER with orthographic complexities ($|C|$, $H(C)$, and S_{token}) and no correlation with phonological complexity ($|\Phi|$), as summarized in Table 5. All of the correlation coefficients with CER were small ($\leq \pm 0.20$), and there is no significant correlation ($p > 0.05$) with respect to CER.

Thus, there is a clear positive correlation of ASR accuracy of fine-tuned Wav2Vec 2.0 models with orthographic complexities but not with phonological complexities. In other words, logographic writing systems and large character inventories can harm ASR performance, supporting the first and second hypotheses, but the self-supervised pretrained model is robust to different phonological complexities, rejecting the third hypothesis.

In addition to the numerical results, there were also differences in the learning curve of fine-tuning to different writing systems. As Figure 5 shows for the three writing systems of Japanese, the model struggles to learn to transcribe in a writing system with a larger inventory of graphemes. For the mixed orthography of *Kanji* and *kana*, the validation CER never goes under 100% until 5800 steps, while the *Kana*-only and *Romaji*-only models start to grasp transcription at much earlier steps (1800 steps and 1300 steps, respectively). Furthermore, the curves of validation CERs over the steps is less smooth in complex orthographies (*Kanji*, *Hangul syllables*, and *Hanzi*) than the phonographic scripts (*Kana*, *Romaji*, *Hangul Jamo*, *Zhuyin*, and *Pinyin*). This demonstrates that complex orthographies result not only in poorer ASR performance but also slower learning speed and more required training data to achieve the desired performance.

5 Discussion

In this section, we discuss the implications of the main results from the previous section.

5.1 Low CER in English

In Table 2, one can notice that the CER of the English fine-tuned model is markedly lower than those of other languages. There are two possible reasons for this tendency. One is that Wav2Vec2-

Language	Writing system	CER (%)↓	CEPS	$\lvert C\rvert$	$H(C)$	S_{token} (%)	$\lvert\Phi\rvert$
Japanese	*Kanji + Kana*	58.12	7.21	1702	7.74	44.98	27.00
	Kana	29.71	3.48	92	5.63	41.22	27.00
	Romaji	17.09	2.91	27	3.52	29.46	27.00
Korean	*Hangul syllables*	28.21	2.63	965	7.98	25.27	39.50
	Hangul Jamo	16.72	3.23	62	4.90	15.99	39.50
Chinese	*Hanzi*	62.81	2.65	2155	9.47	39.46	42.50
	Zhuyin	9.71	1.04	49	4.81	24.32	42.50
	Pinyin	9.17	1.01	56	5.02	22.50	42.50
Thai	*Thai*	19.77	1.80	67	5.24	20.55	40.67
Arabic	*Perso–Arabic*	40.59	4.78	53	4.77	21.57	37.00
English	*Latin*	3.17	0.58	27	4.17	19.17	41.22
French	*Latin*	19.64	2.79	69	4.42	20.37	36.75
Italian	*Latin*	14.81	1.84	48	4.27	21.28	43.33
Czech	*Latin*	16.89	1.86	46	4.92	20.57	39.00
Swedish	*Latin*	20.31	2.71	34	4.52	19.81	35.00
Dutch	*Latin*	12.35	1.77	36	4.20	19.67	49.38
German	*Latin*	7.61	1.03	48	4.18	18.03	40.00

Table 2: A summary of the experimental results. C and Φ are the sets of grapheme types and phoneme types, respectively, that appeared in the training data. Thus, $\lvert C\rvert$ is the number of grapheme types, $H(C)$ is the unigram character entropy, S_{token} is logographicity, and $\lvert\Phi\rvert$ is the number of phoneme types. The number of grapheme types and the unigram entropy $H(C)$ were calculated from the ASR training data. The number of phoneme types was retrieved from Phoible 2.0 (Moran and McCloy, 2019); when there is more than one total number of phoneme types reported, we use the averaged number.

	CER	CEPS	$\lvert C\rvert$	$H(C)$	S_{token}	$\lvert\Phi\rvert$
CER	1.00	0.77	0.85	0.81	0.76	−0.37
CEPS		1.00	0.49	0.41	0.61	−0.66
$\lvert C\rvert$			1.00	0.93	0.72	−0.14
$H(C)$				1.00	0.67	−0.08
S_{token}					1.00	−0.60

Table 3: Correlation matrix of CER and other variables based on the results in Table 2.

XLSR presumably has more English pretraining data than other languages. The experimental results of fine-tuning Wav2Vec2-XLS-R 300M by Rouditchenko et al. (2023) also show lower CER in English models than other languages. The second possible factor is the nature of the fine-tuning dataset used in our experiment. Due to the uncertain quality of Common Voice English, we instead used LibriSpeech, which is a carefully read speech from book texts. Since it has been empirically known in ASR evaluation that audio samples of noisy speech can degrade performance (Babu et al., 2021), the clear articulation and recording of LibriSpeech could help the model achieve better results.

5.2 Revisiting logographicity

The concept of logographicity used in this study measures the degree of one-to-many mapping between phonemes (pronunciation) and graphemes (orthography), following Sproat and Gutkin (2021). Since the task of ASR is to map pronounced words into written words, this measurement of logographicity is valid. However, we can also think of the many-to-one mapping relationship between phonemes and graphemes as an orthographic complexity. For example, English <gh> can be phonologically realized as either /f/ (as in <tough>), /g/ (as in <ghost>), or silent (as in <though>). Because the attention-based metric S_{token} used in our study only considers how much the model looks outside the target word, it is unable to look at how much the attention is spread across the target characters, failing to take this type of complexity into account.

Language	Writing system	CER (%)$^{\downarrow}$	CEPS	$\lvert C\rvert$	$H(C)$	$\lvert\Phi\rvert$
Lithuanian	*Latin*	19.20	2.60	39	4.55	52.50
Polish	*Latin*	12.58	1.63	40	4.56	36.00
Basque	*Latin*	6.28	0.78	27	3.89	30.71
Indonesian	*Latin*	24.01	3.36	35	4.04	31.00
Kabyle	*Latin*	31.59	3.02	46	4.30	30.00
Swahili	*Latin*	17.83	2.14	33	4.00	36.50
Hungarian	*Latin*	15.41	1.98	37	4.52	52.00
Russian	*Cyrillic*	14.44	1.99	40	4.65	39.33
Tatar	*Cyrillic*	21.43	3.27	43	4.72	43.00
Abkhaz	*Cyrillic*	15.09	1.66	41	4.60	66.00
Georgian	*Georgian*	14.69	1.78	37	4.29	33.75
Armenian	*Armenian*	10.87	1.45	49	4.57	36.50
Hindi	*Devanagari*	21.81	2.44	119	5.10	68.40

Table 4: Additional experimental results on languages with phonographic writing systems.

	CER	CEPS	$\lvert C\rvert$	$H(C)$	$\lvert\Phi\rvert$
CER	1.00	0.89	0.20	0.16	−0.18
CEPS		1.00	0.18	0.17	0.02
$\lvert C\rvert$			1.00	0.72	0.58
$H(C)$				1.00	0.61

Table 5: Correlation matrix of CER and other variables based on the results of phonographic languages.

5.3 Broader impacts

Understanding the factors that affect fine-tuned accuracy of the self-supervised pretrained model provides benefits to broader applications. In an extremely low-resource setting like this study, a fine-tuned model would learn faster and better with a phonographic (*i.e.*, spelled as is pronounced) writing system or phonemic transcription than with a logographic writing system. This strategy can apply to low-resource languages with a larger number of graphemes such as Yi syllabary of the Nuosu language, the Cherokee syllabary of Cherokee, and Canadian Aboriginal syllabics of Canadian indigenous languages, providing a path to inclusion of these languages in language technologies.

Furthermore, understanding the model's flexibility of adapting to different phonological systems and struggle in learning complex writing systems confirms the strength and weakness of the self-supervised approach of pretraining. Namely, a multilingual Wav2Vec 2.0 model is good at learning phonology of any language but not so good at writing. This finding possibly sheds light on computational modeling of phonology and written language acquisition.

6 Conclusion

This study investigated what linguistic factors can confuse ASR performance of fine-tuned self-supervised models (Wav2Vec2-XLSR-53) focusing on orthographic and phonological complexities. The experiments trained a fine-tuned model for each language and writing system, covering 25 languages and 11 writing systems in total. The results demonstrated that speech recognition accuracy, in particular CER, strongly correlates with orthographic complexities, that is, with the size of the grapheme inventory and the degree of logographicity of a language's writing system. On the other hand, the results showed that CER has no significant correlation with the size of the phonological inventory of the target language. In addition, more complex orthographies turned out to make the model learn less accurately, more slowly, and less stably than phonographic writing systems. These results confirm the robustness of the self-supervised pretrained model fine-tuned on languages with unseen phonology and the negative effect of orthographic complexities on ASR performance.

7 Limitations

It is worth mentioning that there are still several methodological limitations in this study.

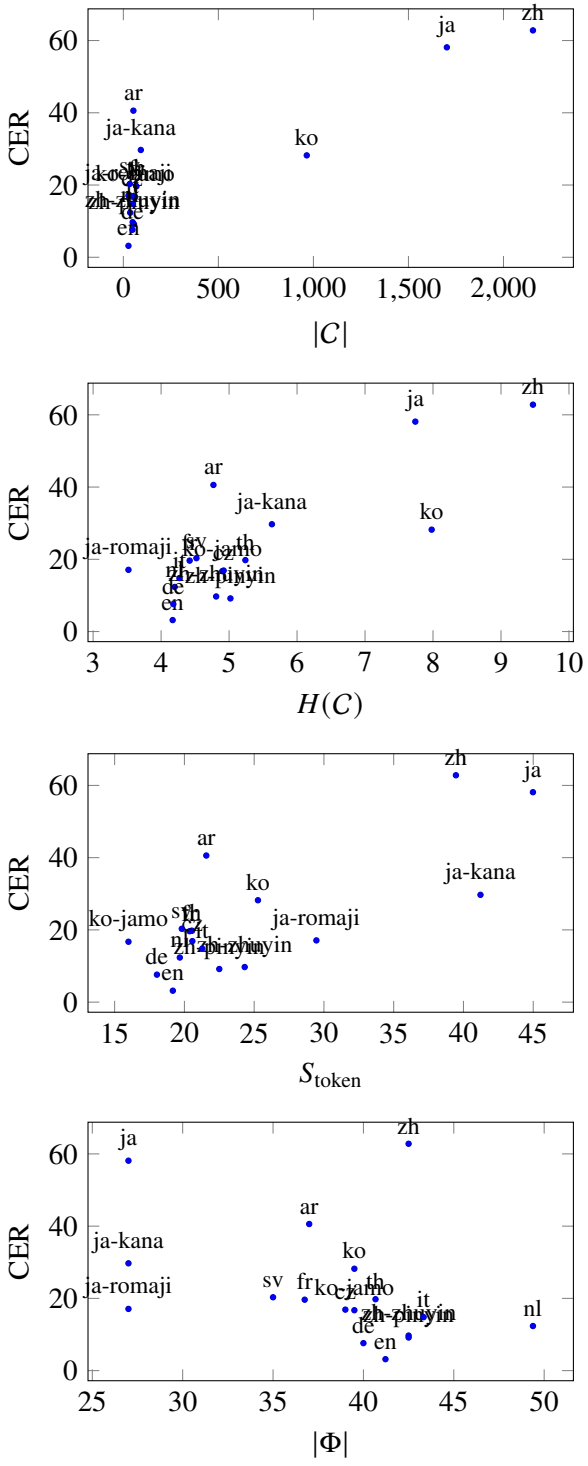

Figure 4: CER versus various measures of linguistic complexity.

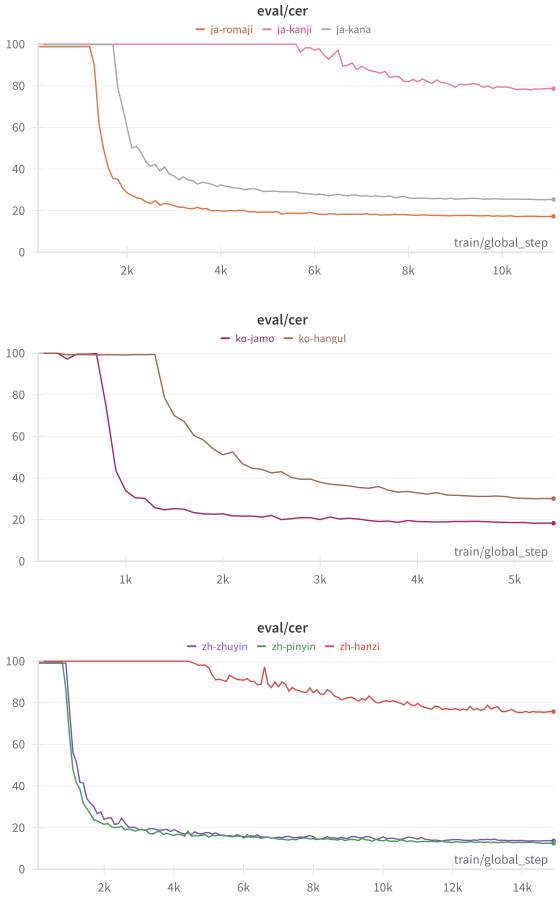

Figure 5: Comparison of validation CERs during the training with different writing systems for Japanese (top), Korean (middle), and Chinese (bottom).

Quality of the dataset. Since Common Voice is a dataset with speech samples collected through crowdsourcing, the quality of data may vary. While there is a validation system to filter out poor samples, there are often speakers' mistakes and other errors in the dataset.

Evaluation metrics. Though CER is one of the most commonly used metrics in ASR, its cross-lingual applicability is questionable. Since a letter in different orthographies can encode different lengths of phonemes, an orthography that represents multiple phonemes like Chinese characters might be more prone to errors than alphabetic orthographies that have one-to-one mapping to pronunciation. In addition, as mentioned in Section 5, the attention-based logographicity measure S_{token} captures the one-to-many mapping between pronunciation and writing but not the many-to-one mapping.

Other Wav2Vec models. As mentioned in Section 2, other multilingual pre-trained Wav2Vec2 models have been developed with more pretraining data and with various model parameter sizes. We have not shown the results from these models here, and the findings in this paper do not necessarily promise reproducibility with these pretrained models.

Low-resource setting. For the sake of controlled experiments, this study limited the amount of training data for each language to be 10k seconds in total. As described in Section 4, different languages and writing systems can exhibit different learning curves, and training on a larger dataset or with more epochs might yield different performance results.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. BCS-2109709.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common Voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC)*, pages 4218–4222.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. XLS-R: Self-supervised cross-lingual speech representation learning at scale.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 12449–12460.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised cross-lingual representation learning for speech recognition. In *Proceedings of INTERSPEECH*, pages 2426–2430.

Christophe Coupé, Yoon Mi Oh, Dan Dediu, and François Pellegrino. 2019. Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche. *Science Advances*, 5(9):eaaw2594.

Weibing Deng, Armen E. Allahverdyan, Bo Li, and Qiuping A. Wang. 2014. Rank-frequency relation for Chinese characters. *The European Physical Journal B*, 87(47).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, pages 4171–4186.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 369–376.

Herve Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X*, pages 79–86.

Takumi Maruyama and Kazuhide Yamamoto. 2018. Simplified corpus with core vocabulary. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*.

Steven Moran and Daniel McCloy. 2019. PHOIBLE 2.0. `http://phoible.org`.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Jungyeul Park, Jeen-Pyo Hong, and Jeong-Won Cha. 2016. Korean language resources for everyone. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, pages 49–58.

Kyubyong Park. 2019. g2pK. https://github.com/Kyubyong/g2pk.

Wannaphong Phatthiyaphaibun, Korakot Chaovavanich, Charin Polpanumas, Arthit Suriyawongkul, Lalita Lowphansirikul, Pattarawat Chormai, Peerat Limkonchotiwat, Thanathip Suntorntip, and Can Udomcharoenchaikit. 2023. PyThaiNLP: Thai natural language processing in Python. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 25–36.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

Andrew Rouditchenko, Sameer Khurana, Samuel Thomas, Rogerio Feris, Leonid Karlinsky, Hilde Kuehne, David Harwath, Brian Kingsbury, and James Glass. 2023. Comparison of Multilingual Self-Supervised and Weakly-Supervised Speech Pre-Training for Adaptation to Unseen Languages. In *Proceedings of INTERSPEECH*, pages 2268–2272.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, Ruslan Mavlyutov, Alice Rakotoarison, Kaushik Ram Sadagopan, Abinesh Ramakrishnan, Tuan Tran, Guillaume Wenzek, Yilin Yang, Ethan Ye, Ivan Evtimov, Pierre Fernandez, Cynthia Gao, Prangthip Hansanti, Elahe Kalbassi, Amanda Kallet, Artyom Kozhevnikov, Gabriel Mejia Gonzalez, Robin San Roman, Christophe Touret, Corinne Wong, Carleigh Wood, Bokai Yu, Pierre Andrews, Can Balioglu, Peng-Jen Chen, Marta R. Costa-jussà, Maha Elbayad, Hongyu Gong, Francisco Guzmán, Kevin Heffernan, Somya Jain, Justine Kao, Ann Lee, Xutai Ma, Alex Mourachko, Benjamin Peloquin, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Anna Sun, Paden Tomasello, Changhan Wang, Jeff Wang, Skyler Wang, and Mary Williamson. 2023. Seamless: Multilingual expressive and streaming speech translation.

Richard Sproat and Alexander Gutkin. 2021. The taxonomy of writing systems: How to measure how logographic a system is. *Computational Linguistics*, 47(3):477–528.

Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. 2018. Sudachi: a Japanese tokenizer for business. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*.

Yida Wang, Pei Ke, Yinhe Zheng, Kaili Huang, Yong Jiang, Xiaoyan Zhu, and Minlie Huang. 2020. A large-scale Chinese short-text conversation dataset. In *Proceedings of the 9th CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC)*.