Clarifying orthography: Orthographic transparency as compressibility

Anonymous ACL submission

Abstract

Orthographic transparency—how directly spelling is related to sound—lacks a unified, script-agnostic metric. Using ideas from algorithmic information theory, we quantify orthographic transparency in terms of the mutual compressibility between orthographic and phonological strings. Our measure provides a principled way to combine two factors that decrease orthographic transparency, capturing both irregular spellings and rule complexity in one quantity. We estimate our transparency measure using prequential code-lengths derived from neural sequence models. Evaluating 22 languages across a broad range of script types (alphabetic, abjad, abugida, syllabic, logographic) confirms common intuitions about relative transparency of scripts. Mutual compressibility offers a simple, principled, and general yardstick for orthographic transparency.

1 Introduction

Orthographic transparency, conversely known as orthographic opacity or orthographic depth, refers to the extent that orthography encodes phonology in a simple and accessible way (Frost et al., 1987; Sproat, 2000). English readers will already be familiar with the sometimes poor correspondence English spelling has with its pronunciation, a fact that makes English orthography relatively opaque. Likewise, Chinese characters give very little—sometimes no—information about the pronunciation of their corresponding morphemes, being even more opaque than English. Spanish, on the other hand, is known for the extreme ease with which pronunciations can be extracted from spelling alone, making its orthography relatively transparent.

Despite the intuitive nature of the concept of orthographic transparency, it has no precise agreed-upon operational definition. Various measurements have attempted to associate orthographic transparency with numbers of grapheme-to-phoneme correspondence (GPC) rules, irregularity of the coverage of these rules (Ziegler et al., 2000), or a combination of the two (Van den Bosch et al., 1994; Protopapas and Vlahou, 2009), or proxies for these such as onset entropy (Borgwaldt et al., 2005) (which measures an unnormalized conditional Shannon entropy of the first phoneme given the first grapheme). Furthermore, existing measures rarely accommodate non-alphabetic systems (Borleffs et al., 2017), a striking limitation given the breadth of existing writing systems. In addition to its scientific interest, finding a transparency measurement has promise for real pedagogical applications, as it is commonly understood to impact the speed of reading acquisition (Aro and Wimmer, 2003; Seymour et al., 2003).

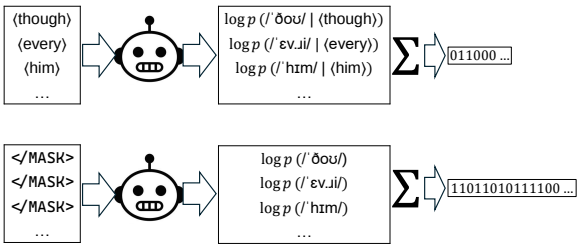

Figure 1: Phonological datasets are compressed both with **Top:** orthographic side-information and **Bottom:** no additional information. The difference between the lengths in the two compressed datasets yields the mutual algorithmic information between the dataset and the side information. We show that a normalized variant of this is a good approximation to orthographic transparency.

We offer a new method to quantify orthographic transparency (schematized in Figure 1) using a measure which combines different sources of opacity in a principled manner. Using ideas from algorithmic information theory (Li et al., 2008; Shen et al., 2017) we posit that orthographic transparency can

be measured as the shared structure between the orthography and phonology of a language. We show that this measure not only succeeds in unifying the aforementioned sources of opacity, but that it can generalize to any writing system (and, in fact, any two datatypes subject to possible transfer learning), and that it conforms to existing results and intuitions. In the remainder of the paper we will first provide background information on orthographic transparency and algorithmic information theory. After this background we will explain the methods we used to extract our measurements and present results of a multilingual survey, then compare these to some existing metrics.

2 Background

The concept of orthographic transparency imagines that languages' writing systems can be arranged along a continuum as in Figure 2 (Sproat, 2000, p. 6). Languages with writing systems that are highly informative are imagined at one end of the continuum, and language's with writing systems that are highly uninformative about their pronunciation are at the other end. However, this simple classification belies different sources that contribute to informativity.

Consider the (in)famous example of the suffix `-ough` in English orthography. There is no consistent rule that can govern all the pronunciations found in `tough`, `through`, `cough`, and `dough` beyond memorizing the entire sequence of graphemes and how the entire corresponding word is pronounced. Even worse are homographs like `lead` for which contextual information must assist in determining the correct pronunciation. Both mappings are non-deterministic, and are characterized in the literature via irregularity measurements, by crafting rules—usually simple GPC rules—and measuring the coverage of those rules (Van den Bosch et al., 1994; Protopapas and Vlahou, 2009) or more recently via training sequence-to-sequence networks and measuring performance on holdout sets (Marjou, 2019). This measurement can operate in both the orthography-to-phonology and phonology-to-orthography direction.

Contrast this source of opacity with another. French is often considered severely to moderately opaque in the literature (Seymour et al., 2003; Goswami et al., 1998). This is due to the complexity of the mappings and unpredictability of the spellings. Consider the French word `oiseau` which is pronounced /wazo/. This pronunciation is derivable from the spelling via regular grapheme-to-phoneme correspondences. However, the correspondences are complex, mapping multiple graphemes to single phonemes. This kind of opacity can be captured by counting the number of rules which can cover a language, procedures which have also been done (Van den Bosch et al., 1994).

The existence of these two kinds of opacity (irregularity and complexity of mappings) have led some to outright reject the notion of orthographic opacity as a single concept at all (Schmalz et al., 2015). If the concept of opacity is accepted as meaningful, then any measurement of it must employ some estimate of the relative importance of both sources to the total measurement, or find some means of unifying these two sources of opacity. One measurement, the *onset entropy* (Borgwaldt et al., 2005), does offer such a unification, but here it encounters a different weakness and is only able to be used at the beginning of a word.

Yet it seems desirable to find a more general unification, as opacity is linked with reading proficiency during reading acquisition (Aro and Wimmer, 2003; Seymour et al., 2003). We show that such a unification is possible in a principled and practical way, and that it is natural: both of these sources of opacity contribute to making the orthography *mutually incompressible* with the phonology of the language (and vice versa). Below we review what we mean by mutual compressibility, and how this can be measured for orthographic systems. Then we will demonstrate results showing that this measure is a natural fit for existing data.

3 Algorithmic Information and Mutual Compressibility

We propose to measure orthographic transparency using complexity as defined in the algorithmic information theory literature (Li et al., 2008; Shen et al., 2017). Here we introduce and motivate our measure, and describe how it can be computed using neural sequence models.

Some definitions are in order before we introduce our measure. The first is the function $K(x)$. This measure, known as **algorithmic information** or Kolmogorov complexity, is the length of the shortest computer program which outputs the string x. Intuitively, this reflects the compressibility of the string x: a low value indicates that x is highly

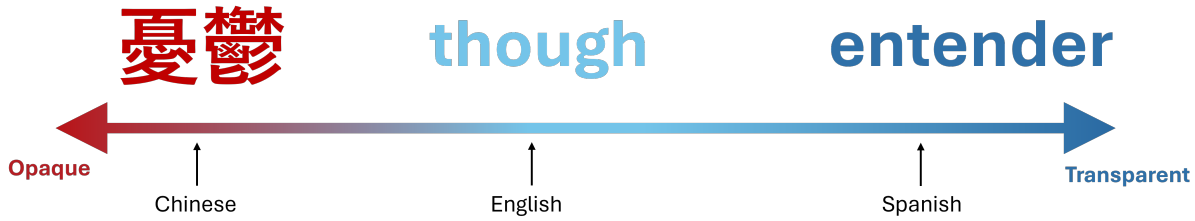

Figure 2: The hypothetical orthographic transparency continuum imagines languages and their writing systems organized on a one-dimensional line.

regular—containing some sort of pattern, which can be represented with a short program. For example, consider the strings below:

(1) $a = 101010101010101010101010101...$

(2) $b = 110000111010100110010010 1...$

Example (1) contains a regular pattern. The longer it becomes, the more it will benefit from this regularity under compression. In other words, the algorithmic information $K(a)$ is small. Example (2) is randomly sampled from a Bernoulli distribution, so the algorithmic information $K(b)$ should be large. There is no pattern, and a longer random sequence won't be any more compressible.

With the introduction of algorithmic information comes a second measure. **Conditional algorithmic information**, denoted $K(x \mid y)$, is the length of the shortest program that generates the string x when given y as input. Intuitively, conditional algorithmic information represents how compressible a string x is given another string y. To see how this can work, consider a third string:

(3) $c = 001111000101011001101 1010...$

This string is also random, so the algorithmic information $K(c)$ is large. However, c is the result of flipping each bit in b, and therefore if b is provided as input, then there *is* a simple program that will yield the string c as output, and thus the program length $K(c \mid b)$ will be short.

With those two definitions we can now define the **algorithmic mutual information** as the difference between the unconditional and conditional algorithmic information:

$$I_K(y; x) = K(x) - K(x|y). \quad (1)$$

Algorithmic mutual information measures shared structure between x and y. When $K(x)$ is large and $K(x \mid y)$ is small, then it can be said that the two strings share a lot of structural information, because having access to y enables compression of x. However, if $K(x)$ and $K(x \mid y)$ are similar or equal in size then we can say that y shares very little structural information with x, because knowing x does not contribute to making y more compressible. Note that unlike Shannon mutual information (Cover and Thomas, 2006), algorithmic mutual information is not commutative: in general, $I_K(x; y) \neq I_K(y; x)$.

We will not apply algorithmic mutual information directly in our measurements and comparisons of orthographic transparency. This reason is that it is difficult to use algorithmic mutual information when dealing with datasets of varying complexities. A low $K(x)$ places a low ceiling on mutual algorithmic information. In our study this would lead to unintuitive interactions between what we would like to measure—orthographic transparency—and extraneous factors including phonotactic and morphological complexity. For this reason we suggest a normalized measure which we dub **mutual compressibility**, or $C_K(y; x)$. We define mutual compressibility as the *proportion* of x explained by y, or

$$C_K(y; x) = \frac{I_K(y; x)}{K(x)} = 1 - \frac{K(x \mid y)}{K(x)}. \quad (2)$$

In order to apply these ideas to orthographic transparency, we will consider how much mutual compressibility exists between a language's orthography and its phonology. The motivation here is simple: we consider orthographic transparency as the complexity of the decoding process that converts phonology to orthography, or vice versa. Simple decoding processes are simple programs, but they also imply a shared structure in the data, a fact we will return to later.

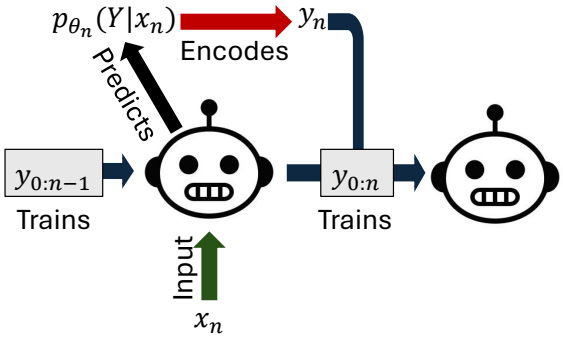

Figure 3: A depiction of prequential coding. Labels and inputs from previous samples at $t = 1, 2, ..., n-1$ are used to train the current model and yield parameterization θ_n. The current model is used to make a prediction given input x_n. The resulting distribution is used to encode label y_n.

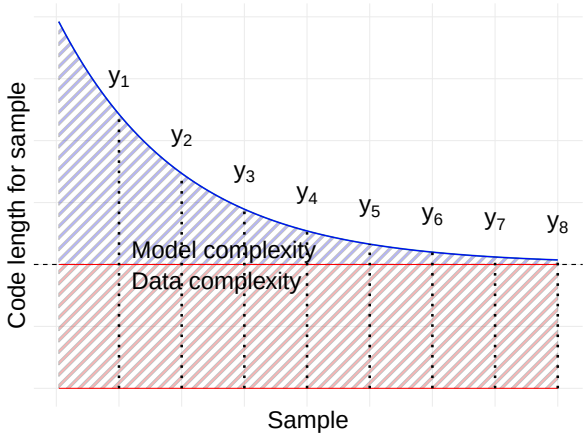

Figure 4: When predicting labels a model improves with successive samples. This codelength includes the complexity of the learning task and the inherent entropy of the data (Elmoznino et al., 2024).

3.1 Prequential Coding

The algorithmic complexity measures above are uncomputable in because finding such a measurement would require searching through all halting computer programs to find the shortest one that generates a certain output (Li et al., 2008). Therefore in practice we rely on approximations of these functions. We use **prequential coding**.

Prequential coding, discovered by Dawid (1984) and Rissanen (1989) independently, is a method of approximating the algorithmic complexity of a dataset using a statistical model. Imagine that you would like to communicate an unknown dataset to a partner (this dataset could be a sequence, or a set of labels for some shared inputs). Assuming you and your partner agree on a model class to encode the data, what is the optimal code for that data? If the data is known ahead of time, then the optimal codelength would be determined by the maximum likelihood estimate for the model class selected given the dataset. However the recipient doesn't have access to this, so instead you choose the next best thing: an incremental code which is determined by the maximum likelihood parameterization *given by the data seen so far*, as illustrated in Figure 3. This yields the following code length:

$$L_{p_\theta}(x) = -\sum_{i=1}^{N} \log p(x_i \mid \theta(x_{1:i-1})), \quad (3)$$

where N is the length of the dataset, p is the family of models chosen for the task, and $\theta(x_{1:i-1})$ is the maximum likelihood estimate for the model parameters θ given observations up to but not including the i'th. The logic is illustrated in Figure 4.

Prequential coding provides an estimate of the algorithmic information $K(x)$ given a statistical model class. However, we aim to use it in the estimation of algorithmic mutual information and mutual compressibility. This involves the calculation of two terms—$L_{p_\theta}(x)$ and $L_{p_\theta}(x \mid y)$—which we slot into Equation 1. That is, the estimate of mutual compressibility for output string x given input string y using prequential coding is

$$C_{p_\theta}(y; x) = 1 - \frac{L_{p_\theta}(x|y)}{L_{p_\theta}(x)}. \quad (4)$$

In all of our experiments below, p_θ will be a neural network parameterized by θ. We use the Mini-batch Incremental/Replay Streams approach to estimating the prequential code implemented in Bornschein et al. (2022), as standard prequential coding would involve training to convergence on every batch, a procedure which scales quadratically with the length of the data.

In addition to providing a convenient method to estimate mutual compressibility, prequential coding also measures the two sources of opacity discussed in Section 2: irregularity of spellings and complexity of rules. In particular, the 'data complexity' component in Figure 4 corresponds to irregularity, and the 'model complexity' component corresponds to rule complexity. For example, even if one has fully mastered all the regular rules of English spelling, the one-off irregular pronunciation of `colonel` as /kɚnəl/ will still be surprising if this word has never been seen before, contributing to data complexity. Conversely, while French spelling

is quite regular, its rules are complex, meaning that a learner will take many samples to figure out the rules, contributing to model complexity.

4 Methods and Dataset

4.1 Dataset Source

We use WikiPron (Lee et al., 2020),[1] a dataset of orthographic forms paired with IPA transcriptions scraped from Wiktionary,[2] a crowdsourced online dictionary. For Japanese phonetic information we supplemented WikiPron with an online IPA dictionary[3] which took its Japanese data from Breen (2013). This was done to provide broad transcriptions since Japanese was of particular interest to us. We reduce the WikiPron corpus for each language to the same sample size by taking the $N = 5,032$ most frequent forms per language,[4] with frequency ranks derived from the most recent 300k sentences collection from the Leipzig news corpus (Goldhahn et al., 2012).[5] The reason for taking the most frequent forms is that highly infrequent forms might be nonrepresentative in terms of orthography and phonotactics. Additionally, orthographic representations with multiple phonetic representations in Wikipron are reduced, and the first phonetic representation in the set was the only one preserved. Although this causes a loss of an important source of irregularity in our measurement (for example, a word like `lead` being restricted to only one pronunciation in our sets), it eliminates a major source of spurious irregularity (the representation of multiple dialectical pronunciations in the data).

4.2 Broad and narrow transcriptions

When assembling our datasets we used broad transcriptions where possible. Where this was not possible we used the narrow transcriptions provided. We made one exception for Japanese, as mentioned earlier, where a different transcription was sourced in order to use a broader transcription type than Wiktionary could provide. Breadth of transcription is a source of additional—and difficult to control—complexity, so we did our best to keep them consistent across languages to the extent our data source allowed.

WikiPron—a scrape of Wiktionary—contains many different crowdsourced phonetic transcriptions. Although it has broad linguistic coverage, there can be high variance in the specificity of the transcriptions. As an example, consider the following two transcriptions for the same English word `winter` (both found in the WikiPron set).

(4) /ˈwɪntɚ/

(5) [ˈwɪɾ̃ɚˠ]

Example 4 transcribes `winter` broadly, without representing additional phonological processes that may be operating when pronouncing the word. Example 5 on the other hand includes symbols which denote gestures a speaker makes in producing this word that are phonologically conditioned and predictable. For example, the narrow transcription uses the symbol ɾ to denote the rapid flapping motion an English speaker will use to produce a /t/ in that phonetic environment. While technically a spectrum (as one can transcribe movements with increasing levels of detail) Wiktionary makes a two-way distinction between these trascription types reflecting the broader tendency in linguistics to denote phonemic and phonological levels of detail.

4.3 Treatment of sub-character structure

We preprocessed our dataset to decompose characters where they contained subcomponents that would normally be accessible to a human reader. To this end we performed decompositions on Japanese and Chinese characters using Ideographic Description Sequences (Kishore, 2016; The Unicode Consortium, 2023),[6] jamo decompositions on Korean Hangul (Dong, 2018),[7] and NFD normalization on all characters (The Unicode Consortium, 2023). To further explain why, consider first Chinese, Japanese, and Korean. Chinese characters, kanji, and Hangul all have compositional structure which is not reflected in their UTF-8 encodings (The Unicode Consortium, 2023). Many Chinese characters (and kanji) contain phonetic elements which are partially informative about pronunciation in Chinese. The case of Hangul is even more important, as Hangul are composed of component jamo, which are not usually represented in Unicode formatted text (including our datasets), yet are meant

[1] https://github.com/CUNY-CL/wikipron

[2] Available under a CC-BY-SA license.

[3] https://github.com/open-dict-data/ipa-dict

[4] $5,032$ was the smallest number of types found in any dataset with a number of types $\geq 5,000$, our chosen cutoff.

[5] https://wortschatz.uni-leipzig.de/en/download, available under a CC-BY license.

[6] https://github.com/skishore/makemeahanzi, available under LGPL.

[7] https://github.com/JDongian/python-jamo, available under Apache License 2.0.

to represent particular phonemes. These processing techniques are our best attempts at preserving the internal structure of characters that is accessible to humans, in order to get a human-applicable measure from neural networks.

4.4 Implementation of Prequential Coding

We now turn to the estimation of mutual compressibility in the dataset using prequential coding. We use a sequence-to-sequence CNN as the underlying statistical model (Gehring et al., 2017). The CNN has a single encoding and decoding layer, and all hidden layers are sized to 100 dimensions with a kernel window of 5. These hyperparameters were chosen after a grid search as detailed in Appendix A. CNNs were chosen principally to facilitate speed of computation to allow us to expand the number of languages analyzed over preliminary work where we had used sequence-to-sequence LSTMs. However, we also found that we achieved lower-variance results over these preliminary investigations, with similar levels of accuracy.

The actual prequential coding procedure we use—Mini-batch Incremental with Replay Streams (Bornschein et al., 2022, MI/RS)—is an approximate way to find the prequential codelength. True prequential coding requires training to convergence on all prefixes of the data to obtain maximum likelihood estimates (Rissanen, 1989); however, the procedure's time complexity scales quadratically in the length of the dataset to compress. MI/RS, on the other hand, has a time complexity of $O(kn)$. This approximation works by keeping a series of k replay streams running on previously seen data with a probability p of resetting each stream after each training iteration. Details of the implementation can be found in Bornschein et al. (2022), but for all of our procedures we run $k = 50$ replay streams with a probability $p = \frac{1}{i+1}$ of resetting at each iteration i (a setting which ensures uniform sampling over all data as i increases when indexed at 1).

To obtain compressibility measurements, MI/RS must be run once with side information (the conditioning variable y, either phonology or orthography) and once without. What this means in practice is running the sequence-to-sequence CNN with and without input information masked. This procedure yields the two terms in Equation 4. In both runs, we initialize the models with the same seeds. We repeat this procedure 20 times (with 20 different seeds) for each language, and for each direction (phonology-to-orthography and orthography-to-phonology). The distributions predicted by our models allow us to calculate the length (in bits) of the ideal encoding given by those distributions via Equation 3.

5 Results

5.1 Main Results

We present average mutual compressibility for orthography from phonology, and vice versa, for 22 languages in our WikiPron dataset.

Figure 5 collects main results. At a high level, we observe some encouraging patterns. For example, as a sanity check, we should expect phonological systems to be more compressible than logographic ones. This is the case, with Chinese and Japanese both scoring extremely low in the compressibility metric.

We can also observe cross-directional effects with some languages showing significant differences between the orthography-to-phonology mapping and the phonology-to-orthography mapping. Although some past work has made the claim that spelling (that is, phonology-to-orthography) is more difficult than reading in general (Bosman and Van Orden, 1997), we can see this doesn't always hold in terms of mutual compressibility. While some languages like French or Greek exhibit the expected effect, there are many languages which exhibit the reverse phenomenon. In many cases this appears to be due to the transcription form in the dataset—with narrow transcriptions making the "reading" mapping more difficult—thus making the results a likely artifact of the data. But this is not always the case. For example, German exhibits this phenomenon and its transcription is broad. More interestingly, for some languages this directional difficulty may be inherent to the writing system. Hindi and Arabic both show this effect, and both have at least occasional optional vowel writing, with Hindi having an implicit central vowel when unmarked (Snell and Weightman, 2016), and with Arabic leaving short vowels unwritten in most contexts (including in our dataset). We provide a table comparing the languages in Appendix B for those interested in more detail.

5.2 The Case of Japanese

We want to highlight the case of Japanese, in particular, to show that our approach delivers sensible results. With our method we can both get a measure

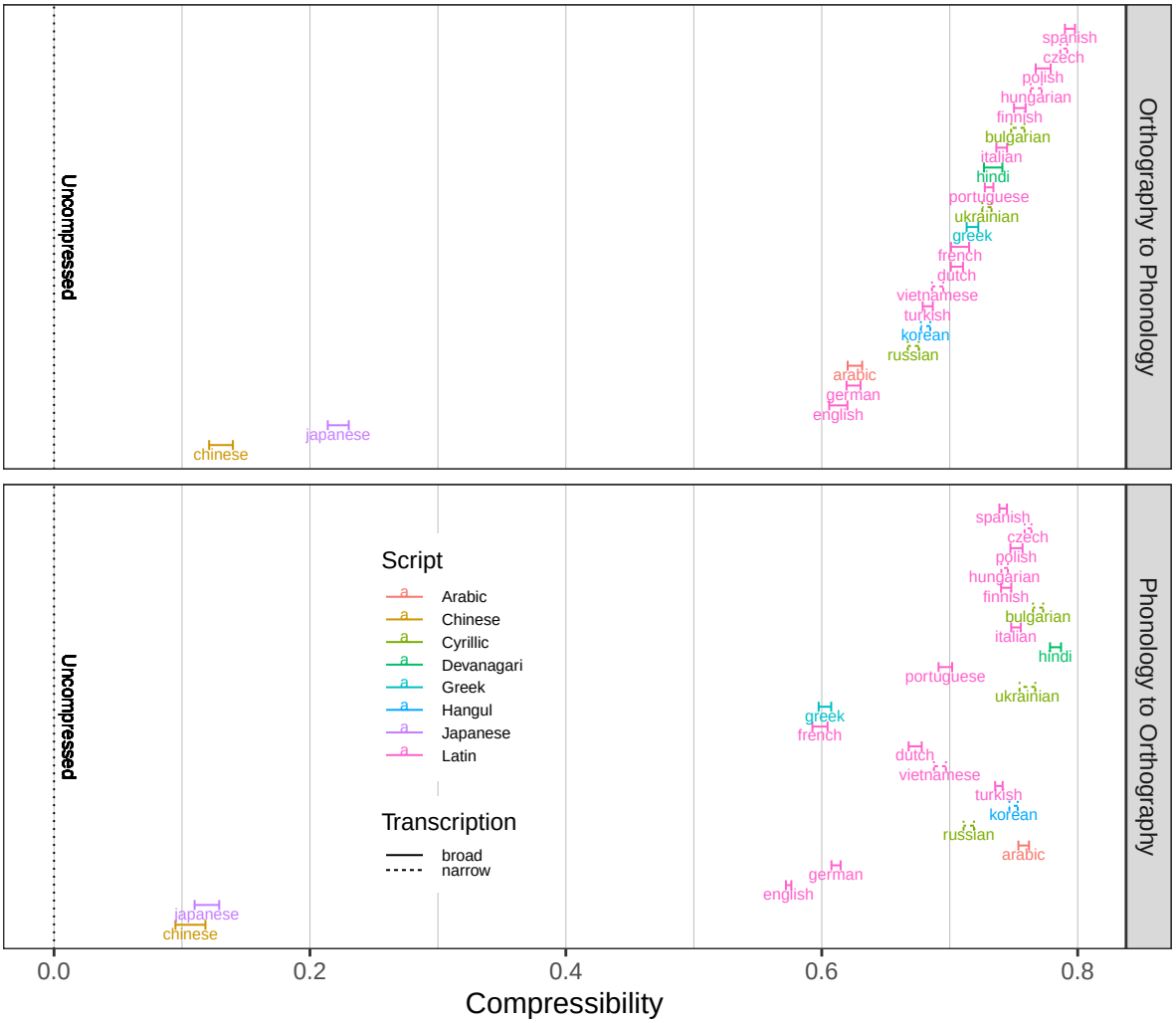

Figure 5: Confidence intervals for mutual compressibility of orthography given phonology, and vice versa, in 22 languages. Error bars represent 95% confidence intervals as determined over all 20 runs.

of the unified transparency of the whole system, and by filtering by character type, a measure of the transparency of individual scripts. This makes Japanese an interesting case. The system is a hybrid of three scripts: kanji, katakana, and hiragana. Two scripts, katakana and hiragana, are syllabic scripts, meaning single characters stand for whole syllables. One script, kanji, is composed of Chinese character borrowings and is thus logographic. What is the transparency of individual component scripts, and how does it affect the overall unified system? To investigate this question we analyzed katakana, kanji, and the union of all systems.[8]

The plots of this split can be seen in Figure 6. As one would hope, the syllabic script katakana shows a much higher transparency in both directions. Kanji are less compressible, with the overall orthography intermediate between the two. This is not too surprising, but interestingly we can also see directional effects. Katakana is relatively transparent in both directions, with an effect perhaps favoring the phonology-to-orthography direction. Kanji, however, shows ease in the orthography-to-phonology direction. We hypothesize that some kanji may have transparency if pronounced with the Chinese readings. Since our method only preserved a single pronunciation per kanji, it is possible that this effect could be exaggerated by our preprocessing.

5.3 Comparison with existing expectations

Results from our analysis largely conform to existing beliefs about how these languages would rank in terms of transparency. For example, there is relative agreement about the high transparency of

[8] Hiragana was dropped as an individual system because not enough stand-alone hiragana-only words were present in our dataset matching our $N \geq 5000$ cutoff.

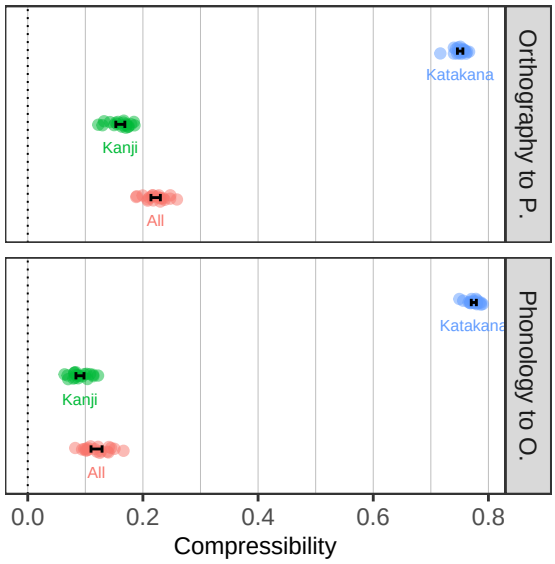

Figure 6: Within languages the measure also can provide distinctions between script types. Here we compare katakana, kanji, and the full language including all three scripts. Hiragana was excluded because there were not enough stand-alone words made exclusively of hiragana characters to analyze.

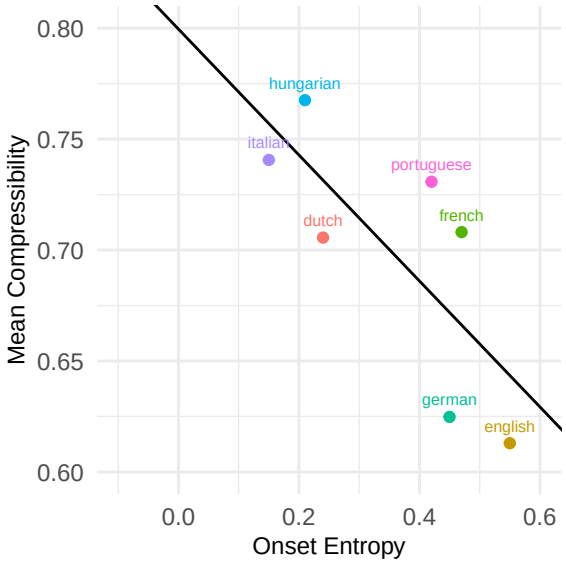

Figure 7: Our measure plotted against the onset entropy values reported in Borgwaldt et al. (2005). Onset entropy is the unweighted mean of the Shannon entropy of the first phoneme in a word given the first grapheme.

Italian, Finnish, and Hungarian (Seymour et al., 2003). There is also agreement that English is relatively opaque among the alphabetic systems, and among these systems English does rank lowest in our metric. We have also reproduced expected directional effects. For example, we observe the strong directionality effects in Greek reported by Protopapas and Vlahou (2009) whereby the orthography-to-phonology mappings are significantly more transparent than the phonology-to-orthography mappings. The same goes for the directional effects observed in French as in (Ziegler et al., 1996). Against onset entropy (Borgwaldt et al., 2005), our measure seems to obtain similar rankings as plotted in Figure 7, showing high compressibility when there is low entropy as in Italian and Hungarian, and vice versa as in English, though the relationship is not significant given the small number of datapoints.

We note that this is not always the case. Korean, often considered quite transparent, ranks as middling-to-low among the alphabetically based writing systems in Figure 5. This is plausibly an issue with the dataset, as the phonological dataset for Korean is extremely narrowly transcribed. This means that the model, while learning the Korean mappings, mustn't just recover the grapheme-to-phoneme mappings, but the phoneme-to-allophone mappings as well. Equally surprising to us was the case of German, as in the literature it is often contrasted with English as relatively transparent. Yet our results show it to be opaque in the orthography-to-phonology direction. We suspect this reflects loanwords which retain the source language orthography.

6 Conclusion

We have demonstrated that mutual compressibility provides a sensible measure of orthographic transparency that is more general than previous proposals, conferring the ability to analyze multiple kinds of systems (logographic, syllabic, alphabetic, and others) via a unified metric. We also believe that this approach is strong on theoretical grounds, based on the intuition that transparency is related to shared structure between different modalities. Finally, we believe that it has some explanatory power. Difficult-to-compress concepts are difficult to learn, a known bias which has been demonstrated in different domains (Feldman, 2003; Chater and Vitányi, 2003) and would explain the difficulties encountered in learning to read opaque orthographies (Seymour et al., 2003; Aro and Wimmer, 2003).

Limitations

The present method has some practical limitations. Firstly, the Wiktionary data varies by transcription

narrowness. This makes comparisons between languages less accurate than they might otherwise be, and they could be theoretically improved by adopting transcriptions from a more consistent source than a Wiktionary scrape. There are also more mundane engineering questions. There is the model selection question: Can we home in on more accurate scores—and more humanlike ones—by using a better class of statistical model? There is also the question of the MI/RS method for prequential estimation. We could use other methods (like block encoding, Blier and Ollivier 2018) or optimize the hyperparameters to achieve tighter bounds on the true measure. We believe these issues worthy of future investigations.

One potential confounding source of opacity could be in the genre of writing itself. We use news corpora to provide frequency information. News sources, especially in what might be termed exoteric communities (Wray and Grace, 2007), can incorporate many loan words and foreign names, whose presence in the data could impact opacity scores. A more careful analysis of genre-related effects, and of effects from particular sources of words, is also warranted.

Ethical Considerations

We foresee no ethical issues with this work.

References

Mikko Aro and Heinz Wimmer. 2003. Learning to read: English in comparison to six more regular orthographies. *Applied psycholinguistics*, 24(4):621–635.

Léonard Blier and Yann Ollivier. 2018. The description length of deep learning models. *Advances in Neural Information Processing Systems*, 31.

Susanne R Borgwaldt, Frauke M Hellwig, and Annette MB De Groot. 2005. Onset entropy matters–letter-to-phoneme mappings in seven languages. *Reading and writing*, 18:211–229.

Elisabeth Borleffs, Ben AM Maassen, Heikki Lyytinen, and Frans Zwarts. 2017. Measuring orthographic transparency and morphological-syllabic complexity in alphabetic orthographies: a narrative review. *Reading and writing*, 30:1617–1638.

Jorg Bornschein, Yazhe Li, and Marcus Hutter. 2022. Sequential learning of neural networks for prequential MDL. *arXiv preprint arXiv:2210.07931*.

Anna MT Bosman and Guy C Van Orden. 1997. Why spelling is more difficult than reading. *Learning to spell: Research, theory, and practice across languages*, 10:173–194.

Jim Breen. 2013. The edict dictionary file. Last updated July 2017.

Nick Chater and Paul Vitányi. 2003. Simplicity: a unifying principle in cognitive science? *Trends in cognitive sciences*, 7(1):19–22.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*. John Wiley & Sons, Hoboken, NJ.

A Philip Dawid. 1984. Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2):278–290.

Joshua Dong. 2018. python-jamo: Hangul syllable decomposition and synthesis using jamo. `https://github.com/JDongian/python-jamo`. Accessed: 18 May 2025.

Eric Elmoznino, Thomas Jiralerspong, Yoshua Bengio, and Guillaume Lajoie. 2024. A complexity-based theory of compositionality. *arXiv preprint arXiv:2410.14817*.

Jacob Feldman. 2003. The simplicity principle in human concept learning. *Current directions in psychological science*, 12(6):227–232.

Ram Frost, Leonard Katz, and Shlomo Bentin. 1987. Strategies for visual word recognition and orthographical depth: a multilingual comparison. *Journal of experimental psychology: human perception and performance*, 13(1):104.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR.

Dirk Goldhahn, Thomas Eckart, Uwe Quasthoff, et al. 2012. Building large monolingual dictionaries at the Leipzig Corpora Collection: From 100 to 200 languages. In *LREC*, volume 29, pages 31–43.

Usha Goswami, Jean Emile Gombert, and Lucia Fraca de Barrera. 1998. Children's orthographic representations and linguistic transparency: Nonsense word reading in English, French, and Spanish. *Applied Psycholinguistics*, 19(1):19–52.

Shaunak Kishore. 2016. Make Me a Hanzi: Free, open-source Chinese character data. `https://www.skishore.me/makemeahanzi/`. Accessed: 18 May 2025.

Jackson L. Lee, Lucas F.E. Ashby, M. Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D. McCarthy, and Kyle Gorman. 2020. Massively multilingual pronunciation modeling with WikiPron. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4223–4228, Marseille, France. European Language Resources Association.

Ming Li, Paul Vitányi, et al. 2008. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer.

Xavier Marjou. 2019. OTEANN: Estimating the transparency of orthographies with an artificial neural network. *arXiv preprint arXiv:1912.13321*.

Athanassios Protopapas and Eleni L Vlahou. 2009. A comparative quantitative analysis of greek orthographic transparency. *Behavior research methods*, 41:991–1008.

Jorma Rissanen. 1989. *Stochastic complexity in statistical inquiry*, volume 15. World scientific.

Xenia Schmalz, Eva Marinus, Max Coltheart, and Anne Castles. 2015. Getting to the bottom of orthographic depth. *Psychonomic bulletin & review*, 22:1614–1629.

Philip HK Seymour, Mikko Aro, Jane M Erskine, and Collaboration with COST Action A8 Network. 2003. Foundation literacy acquisition in European orthographies. *British Journal of psychology*, 94(2):143–174.

Alexander Shen, Vladimir A Uspensky, and Nikolay Vereshchagin. 2017. *Kolmogorov complexity and algorithmic randomness*, volume 220. American Mathematical Soc.

Rupert Snell and Simon Weightman. 2016. *Complete Hindi: Beginner to Intermediate Course: Learn to read, write, speak and understand a new language*. Teach Yourself.

Richard William Sproat. 2000. *A computational theory of writing systems*. Cambridge University Press.

The Unicode Consortium. 2023. The Unicode standard, version 15.1. Technical Report 15.1.0, The Unicode Consortium, Mountain View, CA.

Antal Van den Bosch, Alain Content, Walter Daelemans, and Beatrice De Gelder. 1994. Measuring the complexity of writing systems. *Journal of Quantitative Linguistics*, 1(3):178–188.

Alison Wray and George W Grace. 2007. The consequences of talking to strangers: Evolutionary corollaries of socio-cultural influences on linguistic form. *Lingua*, 117(3):543–578.

Johannes C Ziegler, Arthur M Jacobs, and Gregory O Stone. 1996. Statistical analysis of the bidirectional inconsistency of spelling and sound in french. *Behavior Research Methods, Instruments & Computers*, 28:504–515.

Johannes C Ziegler, Conrad Perry, and Max Coltheart. 2000. The DRC model of visual word recognition and reading aloud: An extension to German. *European Journal of Cognitive Psychology*, 12(3):413–430.

A Hyperparameter Tuning

In order to select the best hyperparameters for the task a grid search was performed across a small set of possible parameters using English, Chinese, and Finnish (selected for representation across the suspected transparency spectrum) and using both the phonology-to-orthography and orthography-to-phonology directions. Because variance between encodings could be high, we measured compressibility for these languages 20 times each and selected the hyperparameters with the greatest 95% CI lower bound.

Kernel size, learning rate, hidden dimension, and number of layers were all varied with candidate values shown in Table 1. The total number of candidate combinations came out to 24 evaluations, but given the need to check each language multiple times in both directions the combined runs precluded a more exhaustive search. The best hyperparameter combination value is shown below in bold.

Variable	Candidate values
Hidden size	**100**, 200
Num. layers	**1**, 2, 3
Learning rate	0.01, **0.005**
Kernel size	3, **5**

Table 1: Candidate values evaluated during hyperparameter tuning.

B Between-Language Significance Tests

We report in Table 2 the results of TukeyHSD tests on both the orthography-to-phonology and phonology-to-orthography mappings for those interested. Colored cells are significant with $p < .05$.

(a) Orthography differences

	ara	bul	ces	deu	ell	eng	fin	fra	hin	hun	ita	jpn	kor	nld	pol	por	rus	spa	tur	ukr	vie	zho
ara	—	**-0.127**	**-0.163**	0.001	**-0.092**	0.013	**-0.129**	**-0.082**	**-0.108**	**-0.142**	**-0.115**	**0.404**	**-0.055**	**-0.080**	**-0.147**	**-0.105**	**-0.046**	**-0.168**	**-0.057**	**-0.103**	**-0.065**	**0.495**
bul	**0.127**	—	**-0.036**	**0.128**	**0.035**	**0.140**	-0.002	**0.045**	**0.019**	**-0.014**	0.013	**0.531**	**0.072**	**0.048**	**-0.020**	**0.022**	**0.082**	**-0.041**	**0.070**	**0.024**	**0.063**	**0.623**
ces	**0.163**	**0.036**	—	**0.164**	**0.071**	**0.176**	**0.034**	**0.081**	**0.055**	**0.022**	**0.048**	**0.567**	**0.108**	**0.083**	**0.016**	**0.058**	**0.118**	-0.005	**0.106**	**0.060**	**0.099**	**0.659**
deu	-0.001	**-0.128**	**-0.164**	—	**-0.093**	0.012	**-0.130**	**-0.083**	**-0.109**	**-0.143**	**-0.116**	**0.403**	**-0.056**	**-0.081**	**-0.148**	**-0.106**	**-0.047**	**-0.169**	**-0.058**	**-0.104**	**-0.066**	**0.494**
ell	**0.092**	**-0.035**	**-0.071**	**0.093**	—	**0.105**	**-0.037**	0.010	**-0.016**	**-0.050**	**-0.023**	**0.496**	**0.037**	0.012	**-0.055**	-0.013	**0.046**	**-0.076**	**0.035**	-0.011	**0.027**	**0.587**
eng	-0.013	**-0.140**	**-0.176**	-0.012	**-0.105**	—	**-0.142**	**-0.095**	**-0.121**	**-0.155**	**-0.128**	**0.391**	**-0.068**	**-0.093**	**-0.160**	**-0.118**	**-0.059**	**-0.181**	**-0.070**	**-0.116**	**-0.078**	**0.482**
fin	**0.129**	0.002	**-0.034**	**0.130**	**0.037**	**0.142**	—	**0.047**	**0.021**	-0.013	**0.014**	**0.533**	**0.074**	**0.049**	**-0.018**	**0.024**	**0.083**	**-0.039**	**0.072**	**0.026**	**0.064**	**0.624**
fra	**0.082**	**-0.045**	**-0.081**	**0.083**	-0.010	**0.095**	**-0.047**	—	**-0.026**	**-0.059**	**-0.033**	**0.486**	**0.027**	0.002	**-0.065**	**-0.023**	**0.037**	**-0.086**	**0.025**	**-0.021**	**0.018**	**0.578**
hin	**0.108**	**-0.019**	**-0.055**	**0.109**	**0.016**	**0.121**	**-0.021**	**0.026**	—	**-0.034**	-0.007	**0.512**	**0.053**	**0.028**	**-0.039**	0.003	**0.062**	**-0.060**	**0.051**	0.005	**0.043**	**0.603**
hun	**0.142**	**0.014**	**-0.022**	**0.143**	**0.050**	**0.155**	0.013	**0.059**	**0.034**	—	**0.027**	**0.545**	**0.086**	**0.062**	-0.005	**0.037**	**0.096**	**-0.026**	**0.085**	**0.038**	**0.077**	**0.637**
ita	**0.115**	-0.013	**-0.048**	**0.116**	**0.023**	**0.128**	**-0.014**	**0.033**	0.007	**-0.027**	—	**0.518**	**0.059**	**0.035**	**-0.032**	0.010	**0.069**	**-0.053**	**0.058**	0.012	**0.050**	**0.610**
jpn	**-0.404**	**-0.531**	**-0.567**	**-0.403**	**-0.496**	**-0.391**	**-0.533**	**-0.486**	**-0.512**	**-0.545**	**-0.518**	—	**-0.459**	**-0.483**	**-0.551**	**-0.509**	**-0.449**	**-0.572**	**-0.461**	**-0.507**	**-0.468**	**0.092**
kor	**0.055**	**-0.072**	**-0.108**	**0.056**	**-0.037**	**0.068**	**-0.074**	**-0.027**	**-0.053**	**-0.086**	**-0.059**	**0.459**	—	**-0.025**	**-0.092**	**-0.050**	0.010	**-0.113**	-0.002	**-0.048**	-0.009	**0.551**
nld	**0.080**	**-0.048**	**-0.083**	**0.081**	-0.012	**0.093**	**-0.049**	-0.002	**-0.028**	**-0.062**	**-0.035**	**0.483**	**0.025**	—	**-0.067**	**-0.025**	**0.034**	**-0.088**	**0.023**	**-0.023**	**0.015**	**0.575**
pol	**0.147**	**0.020**	**-0.016**	**0.148**	**0.055**	**0.160**	**0.018**	**0.065**	**0.039**	0.005	**0.032**	**0.551**	**0.092**	**0.067**	—	**0.042**	**0.101**	**-0.021**	**0.090**	**0.044**	**0.082**	**0.643**
por	**0.105**	**-0.022**	**-0.058**	**0.106**	0.013	**0.118**	**-0.024**	**0.023**	-0.003	**-0.037**	-0.010	**0.509**	**0.050**	**0.025**	**-0.042**	—	**0.059**	**-0.063**	**0.048**	0.002	**0.040**	**0.600**
rus	**0.046**	**-0.082**	**-0.118**	**0.047**	**-0.046**	**0.059**	**-0.083**	**-0.037**	**-0.062**	**-0.096**	**-0.069**	**0.449**	-0.010	**-0.034**	**-0.101**	**-0.059**	—	**-0.122**	-0.011	**-0.058**	**-0.019**	**0.541**
spa	**0.168**	**0.041**	0.005	**0.169**	**0.076**	**0.181**	**0.039**	**0.086**	**0.060**	**0.026**	**0.053**	**0.572**	**0.113**	**0.088**	**0.021**	**0.063**	**0.122**	—	**0.111**	**0.065**	**0.103**	**0.663**
tur	**0.057**	**-0.070**	**-0.106**	**0.058**	**-0.035**	**0.070**	**-0.072**	**-0.025**	**-0.051**	**-0.085**	**-0.058**	**0.461**	0.002	**-0.023**	**-0.090**	**-0.048**	0.011	**-0.111**	—	**-0.046**	-0.008	**0.552**
ukr	**0.103**	**-0.024**	**-0.060**	**0.104**	0.011	**0.116**	**-0.026**	**0.021**	-0.005	**-0.038**	-0.012	**0.507**	**0.048**	**0.023**	**-0.044**	-0.002	**0.058**	**-0.065**	**0.046**	—	**0.038**	**0.599**
vie	**0.065**	**-0.063**	**-0.099**	**0.066**	**-0.027**	**0.078**	**-0.064**	**-0.018**	**-0.043**	**-0.077**	**-0.050**	**0.468**	0.009	**-0.015**	**-0.082**	**-0.040**	**0.019**	**-0.103**	0.008	**-0.038**	—	**0.560**
zho	**-0.495**	**-0.623**	**-0.659**	**-0.494**	**-0.587**	**-0.482**	**-0.624**	**-0.578**	**-0.603**	**-0.637**	**-0.610**	**-0.092**	**-0.551**	**-0.575**	**-0.643**	**-0.600**	**-0.541**	**-0.663**	**-0.552**	**-0.599**	**-0.560**	—

(b) Phonology differences

	ara	bul	ces	deu	ell	eng	fin	fra	hin	hun	ita	jpn	kor	nld	pol	por	rus	spa	tur	ukr	vie	zho
ara	—	-0.011	-0.003	**0.147**	**0.155**	**0.184**	**0.014**	**0.159**	**-0.025**	**0.015**	0.006	**0.638**	0.008	**0.085**	0.006	**0.061**	**0.043**	**0.016**	**0.019**	-0.003	**0.065**	**0.651**
bul	0.011	—	0.008	**0.158**	**0.167**	**0.195**	**0.025**	**0.170**	**-0.014**	**0.026**	**0.017**	**0.650**	**0.019**	**0.096**	**0.017**	**0.072**	**0.054**	**0.027**	**0.031**	0.008	**0.077**	**0.662**
ces	0.003	-0.008	—	**0.150**	**0.159**	**0.187**	**0.017**	**0.163**	**-0.021**	**0.018**	0.009	**0.642**	0.011	**0.088**	0.009	**0.065**	**0.046**	**0.019**	**0.023**	0.001	**0.069**	**0.655**
deu	**-0.147**	**-0.158**	**-0.150**	—	0.009	**0.037**	**-0.133**	0.013	**-0.172**	**-0.132**	**-0.141**	**0.492**	**-0.139**	**-0.062**	**-0.141**	**-0.085**	**-0.104**	**-0.131**	**-0.127**	**-0.150**	**-0.081**	**0.505**
ell	**-0.155**	**-0.167**	**-0.159**	-0.009	—	**0.028**	**-0.142**	0.004	**-0.180**	**-0.140**	**-0.149**	**0.483**	**-0.147**	**-0.071**	**-0.150**	**-0.094**	**-0.112**	**-0.139**	**-0.136**	**-0.158**	**-0.090**	**0.496**
eng	**-0.184**	**-0.195**	**-0.187**	**-0.037**	**-0.028**	—	**-0.170**	**-0.025**	**-0.209**	**-0.169**	**-0.178**	**0.455**	**-0.176**	**-0.099**	**-0.178**	**-0.122**	**-0.141**	**-0.168**	**-0.164**	**-0.187**	**-0.118**	**0.467**
fin	**-0.014**	**-0.025**	**-0.017**	**0.133**	**0.142**	**0.170**	—	**0.146**	**-0.039**	0.001	-0.008	**0.625**	-0.006	**0.071**	-0.008	**0.048**	**0.029**	0.002	0.006	**-0.017**	**0.052**	**0.638**
fra	**-0.159**	**-0.170**	**-0.163**	-0.013	-0.004	**0.025**	**-0.146**	—	**-0.184**	**-0.144**	**-0.153**	**0.479**	**-0.151**	**-0.074**	**-0.154**	**-0.098**	**-0.116**	**-0.143**	**-0.140**	**-0.162**	**-0.094**	**0.492**
hin	**0.025**	**0.014**	**0.021**	**0.172**	**0.180**	**0.209**	**0.039**	**0.184**	—	**0.040**	**0.031**	**0.663**	**0.033**	**0.110**	**0.031**	**0.086**	**0.068**	**0.041**	**0.044**	**0.022**	**0.090**	**0.676**
hun	**-0.015**	**-0.026**	**-0.018**	**0.132**	**0.140**	**0.169**	-0.001	**0.144**	**-0.040**	—	-0.009	**0.623**	-0.007	**0.070**	-0.009	**0.046**	**0.028**	0.001	0.004	**-0.018**	**0.051**	**0.636**
ita	-0.006	**-0.017**	-0.009	**0.141**	**0.149**	**0.178**	0.008	**0.153**	**-0.031**	0.009	—	**0.632**	0.002	**0.079**	-0.000	**0.055**	**0.037**	0.010	0.013	-0.009	**0.059**	**0.645**
jpn	**-0.638**	**-0.650**	**-0.642**	**-0.492**	**-0.483**	**-0.455**	**-0.625**	**-0.479**	**-0.663**	**-0.623**	**-0.632**	—	**-0.630**	**-0.554**	**-0.633**	**-0.577**	**-0.595**	**-0.622**	**-0.619**	**-0.641**	**-0.573**	0.013
kor	-0.008	**-0.019**	-0.011	**0.139**	**0.147**	**0.176**	0.006	**0.151**	**-0.033**	0.007	-0.002	**0.630**	—	**0.077**	-0.002	**0.053**	**0.035**	0.008	0.011	-0.011	**0.058**	**0.643**
nld	**-0.085**	**-0.096**	**-0.088**	**0.062**	**0.071**	**0.099**	**-0.071**	**0.074**	**-0.110**	**-0.070**	**-0.079**	**0.554**	**-0.077**	—	**-0.079**	**-0.024**	**-0.042**	**-0.069**	**-0.065**	**-0.088**	**-0.019**	**0.566**
pol	-0.006	**-0.017**	-0.009	**0.141**	**0.150**	**0.178**	0.008	**0.154**	**-0.031**	0.009	0.000	**0.633**	0.002	**0.079**	—	**0.056**	**0.037**	0.010	**0.014**	-0.009	**0.060**	**0.646**
por	**-0.061**	**-0.072**	**-0.065**	**0.085**	**0.094**	**0.122**	**-0.048**	**0.098**	**-0.086**	**-0.046**	**-0.055**	**0.577**	**-0.053**	**0.024**	**-0.056**	—	**-0.018**	**-0.045**	**-0.042**	**-0.064**	0.004	**0.590**
rus	**-0.043**	**-0.054**	**-0.046**	**0.104**	**0.112**	**0.141**	**-0.029**	**0.116**	**-0.068**	**-0.028**	**-0.037**	**0.595**	**-0.035**	**0.042**	**-0.037**	**0.018**	—	**-0.027**	**-0.024**	**-0.046**	**0.023**	**0.608**
spa	**-0.016**	**-0.027**	**-0.019**	**0.131**	**0.139**	**0.168**	-0.002	**0.143**	**-0.041**	-0.001	-0.010	**0.622**	-0.008	**0.069**	-0.010	**0.045**	**0.027**	—	0.003	**-0.019**	**0.049**	**0.635**
tur	**-0.019**	**-0.031**	**-0.023**	**0.127**	**0.136**	**0.164**	-0.006	**0.140**	**-0.044**	-0.004	-0.013	**0.619**	-0.011	**0.065**	**-0.014**	**0.042**	**0.024**	-0.003	—	**-0.022**	**0.046**	**0.632**
ukr	0.003	-0.008	-0.001	**0.150**	**0.158**	**0.187**	**0.017**	**0.162**	**-0.022**	**0.018**	0.009	**0.641**	0.011	**0.088**	0.009	**0.064**	**0.046**	**0.019**	**0.022**	—	**0.068**	**0.654**
vie	**-0.065**	**-0.077**	**-0.069**	**0.081**	**0.090**	**0.118**	**-0.052**	**0.094**	**-0.090**	**-0.051**	**-0.059**	**0.573**	**-0.058**	**0.019**	**-0.060**	-0.004	**-0.023**	**-0.049**	**-0.046**	**-0.068**	—	**0.586**
zho	**-0.651**	**-0.662**	**-0.655**	**-0.505**	**-0.496**	**-0.467**	**-0.638**	**-0.492**	**-0.676**	**-0.636**	**-0.645**	-0.013	**-0.643**	**-0.566**	**-0.646**	**-0.590**	**-0.608**	**-0.635**	**-0.632**	**-0.654**	**-0.586**	—

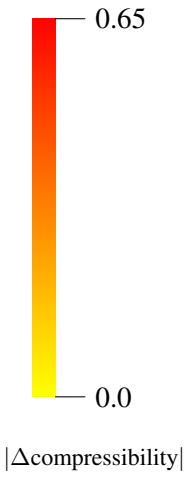

Table 2: Results of TukeyHSD tests between all languages. Each cell shows the signed difference (bold if $p < 0.05$); shading from yellow→red indicates $|\Delta|$ on a 0–0.65 scale.