What Kind of Language Is Hard to Language-Model?

Sebastian J. Mielke[1] **Ryan Cotterell**[1] **Kyle Gorman**[2,3] **Brian Roark**[3] **Jason Eisner**[1]
[1] Department of Computer Science, Johns Hopkins University
[2] Program in Linguistics, Graduate Center, City University of New York [3] Google
{sjmielke@,ryan.cotterell@}jhu.edu kgorman@gc.cuny.edu
roark@google.com jason@cs.jhu.edu

Abstract

How language-agnostic are current state-of-the-art NLP tools? Are there some types of language that are easier to model with current methods? In prior work (Cotterell et al., 2018) we attempted to address this question for language modeling, and observed that recurrent neural network language models do not perform equally well over all the high-resource European languages found in the Europarl corpus. We speculated that inflectional morphology may be the primary culprit for the discrepancy. In this paper, we extend these earlier experiments to cover 69 languages from 13 language families using a multilingual Bible corpus. Methodologically, we introduce a new paired-sample multiplicative mixed-effects model to obtain language difficulty coefficients from at-least-pairwise parallel corpora. In other words, the model is aware of inter-sentence variation and can handle missing data. Exploiting this model, we show that "translationese" is not any easier to model than natively written language in a fair comparison. Trying to answer the question of what features difficult languages have in common, we try and fail to reproduce our earlier (Cotterell et al., 2018) observation about morphological complexity and instead reveal far simpler statistics of the data that seem to drive complexity in a much larger sample.

1 Introduction

Do current NLP tools serve all languages? Technically, yes, as there are rarely hard constraints that prohibit application to specific languages, as long as there is data annotated for the task. However, in practice, the answer is more nuanced: as most studies seem to (unfairly) assume English is representative of the world's languages (Bender, 2009), we do not have a clear idea how well models perform cross-linguistically in a controlled setting. In this work, we look at current methods for language modeling and attempt to determine whether there are typological properties that make certain languages harder to language-model than others.

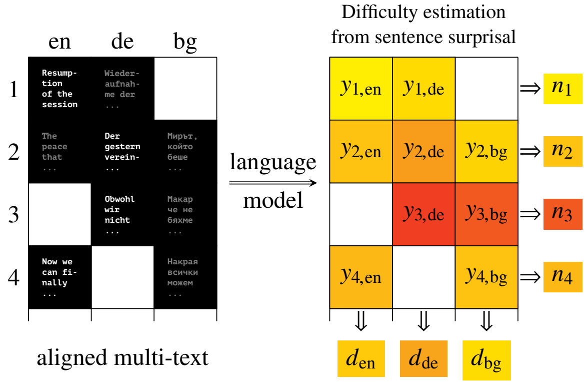


Figure 1: Jointly estimating the information n_i present in each multi-text intent i *and* the difficulty d_j of each language j. At left, gray text indicates translations of the original (white) sentence in the same row. At right, darker cells indicate higher surprisal/difficulty. Empty cells indicate missing translations. English (en) is missing a hard sentence and Bulgarian (bg) is missing an easy sentence, but this does not mislead our method into estimating English as easier than Bulgarian.

One of the oldest tasks in NLP (Shannon, 1951) is language modeling, which attempts to estimate a distribution $p(\mathbf{x})$ over strings $\mathbf{x}$ of a language. Recent years have seen impressive improvements with recurrent neural language models (e.g., Merity et al., 2018). Language modeling is an important component of tasks such as speech recognition, machine translation, and text normalization. It has also enabled the construction of contextual word embeddings that provide impressive performance gains in many other NLP tasks (Peters et al., 2018)—though those downstream evaluations, too, have focused on a small number of (mostly English) datasets.

In prior work (Cotterell et al., 2018), we compared languages in terms of the difficulty of language modeling, controlling for differences in content by using a multi-lingual, fully parallel text corpus. Few such corpora exist: in that paper, we made

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4975–4989
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

use of the Europarl corpus which, unfortunately, is not very typologically diverse. Using a corpus with relatively few (and often related) languages limits the kinds of conclusions that can be drawn from any resulting comparisons. In this paper, we present an alternative method that does not require the corpus to be *fully* parallel, so that collections consisting of many more languages can be compared. Empirically, we report language-modeling results on 62 languages from 13 language families using Bible translations, and on the 21 languages used in the European Parliament proceedings.

We suppose that a language model's surprisal on a sentence—the negated log of the probability it assigns to the sentence—reflects not only the length and complexity of the specific sentence, but also the general difficulty that the model has in predicting sentences of that language. Given language models of diverse languages, we jointly recover each language's difficulty parameter. Our regression formula explains the variance in the dataset better than previous approaches and can also deal with missing translations for some purposes.

Given these difficulty estimates, we conduct a correlational study, asking which typological features of a language are predictive of modeling difficulty. Our results suggest that simple properties of a language—the word inventory and (to a lesser extent) the raw character sequence length—are statistically significant indicators of modeling difficulty within our large set of languages. In contrast, we fail to reproduce our earlier results from Cotterell et al. (2018),[1] which suggested morphological complexity as an indicator of modeling complexity. In fact, we find no tenable correlation to a wide variety of typological features, taken from the WALS dataset and other sources. Additionally, exploiting our model's ability to handle missing data, we directly test the hypothesis that translationese leads to easier language-modeling (Baker, 1993; Lembersky et al., 2012). We ultimately cast doubt on this claim, showing that, under the strictest controls, translationese is *different*, but not any *easier* to model according to our notion of difficulty.

We conclude with a recommendation: The world being small, typology is in practice a small-data problem. there is a real danger that cross-linguistic studies will under-sample and thus over-extrapolate. We outline directions for future, more robust, investigations, and further caution that future work of this sort should focus on datasets with far more languages, something our new methods now allow.

2 The Surprisal of a Sentence

When trying to estimate the difficulty (or complexity) of a language, we face a problem: the predictiveness of a language model on a domain of text will reflect not only the language that the text is written in, but also the topic, meaning, style, and information density of the text. To measure the effect due only to the language, we would like to compare on datasets that are matched for the other variables, to the extent possible. The datasets should all contain the same content, the only difference being the language in which it is expressed.

2.1 Multitext for a Fair Comparison

To attempt a fair comparison, we make use of **multitext**—sentence-aligned[2] translations of the *same content* in multiple languages. Different surprisals on the translations of the same sentence reflect quality differences in the language models, unless the translators added or removed information.[3]

In what follows, we will distinguish between the i^{th} **sentence** in language j, which is a specific string s_{ij}, and the i^{th} **intent**, the shared abstract thought that gave rise to all the sentences $s_{i1}, s_{i2}, \ldots$. For simplicity, suppose for now that we have a fully parallel corpus. We select, say, 80% of the intents.[4] We use the English sentences that express these intents to train an English language model, and test it on the sentences that express the remaining 20% of the intents. We will later drop the assumption of a fully parallel corpus (§3), which will help us to estimate the effects of translationese (§6).

[1] We can certainly **replicate** those results in the sense that, using the surprisals from those experiments, we achieve the same correlations. However, we did not **reproduce** the results under new conditions (Drummond, 2009). Our new conditions included a larger set of languages, a more sophisticated difficulty estimation method, and—perhaps crucially—improved language modeling families that tend to achieve better surprisals (or equivalently, better perplexity).

[2] Both corpora we use align small paragraphs instead of sentences, but for simplicity we will call them "sentences."

[3] A translator might add or remove information out of helpfulness, sloppiness, showiness, consideration for their audience's background knowledge, or deference to the conventions of the target language. For example, English conventions make it almost obligatory to express number (via morphological inflection), but make it optional to express evidentiality (e.g., via an explicit modal construction); other languages are different.

[4] In practice, we use 2/3 of the raw data to train our models, 1/6 to tune them and the remaining 1/6 to test them.

2.2 Comparing Surprisal Across Languages

Given some test sentence s_{ij}, a language model p defines its **surprisal**: the negative log-likelihood $\mathrm{NLL}(s_{ij}) = -\log_2 p(s_{ij})$. This can be interpreted as the number of bits needed to represent the sentence under a compression scheme that is derived from the language model, with high-probability sentences requiring the fewest bits. Long or unusual sentences tend to have high surprisal—but high surprisal can also reflect a language's model's *failure to anticipate* predictable words. In fact, language models for the same language are often comparatively evaluated by their *average surprisal* on a corpus (the **cross-entropy**). Cotterell et al. (2018) similarly compared language models for different languages, using a multitext corpus.

Concretely, recall that s_{ij} and $s_{ij'}$ should contain, at least in principle, the same information for two languages j and j'—they are translations of each other. But, if we find that $\mathrm{NLL}(s_{ij}) > \mathrm{NLL}(s_{ij'})$, we must assume that either s_{ij} contains more information than $s_{ij'}$, or that our language model was simply able to predict it less well.[5] If we were to assume that our language models were perfect in the sense that they captured the true probability distribution of a language, we could make the former claim; but we suspect that much of the difference can be explained by our imperfect LMs rather than inherent differences in the expressed information (see the discussion in footnote 3).

2.3 Our Language Models

Specifically, the crude tools we use are recurrent neural network language models (RNNLMs) over different types of subword units. For fairness, it is of utmost importance that these language models are **open-vocabulary**, i.e., they predict the entire string and cannot cheat by predicting only UNK ("unknown") for some words of the language.[6]

Char-RNNLM The first open-vocabulary RNNLM is the one of Sutskever et al. (2011), whose model generates a sentence, not word by word, but rather character by character. An obvious drawback of the model is that it has no explicit representation of reusable substrings (Mielke and Eisner, 2018), but the fact that it does not rely on a somewhat arbitrary word segmentation or tokenization makes it attractive for this study. We use a more current version based on LSTMs (Hochreiter and Schmidhuber, 1997), using the implementation of Merity et al. (2018) with the char-PTB parameters.

BPE-RNNLM BPE-based open-vocabulary language models make use of sub-word units instead of either words or characters and are a strong baseline on multiple languages (Mielke and Eisner, 2018). Before training the RNN, **byte pair encoding** (BPE; Sennrich et al., 2016) is applied globally to the training corpus, splitting each word (i.e., each space-separated substring) into one or more units. The RNN is then trained over the sequence of units, which looks like this: "The |ex|os|kel|eton |is |gener|ally |blue". The set of subword units is finite and determined from training data only, but it is a superset of the alphabet, making it possible to explain any novel word in held-out data via some segmentation.[7] One important thing to note is that the size of this set can be tuned by specifying the number of BPE **merges**, allowing us to smoothly vary between a word-level model (∞ merges) and a kind of character-level model (0 merges). As Figure 2 shows, the number of merges that maximizes log-likelihood of our dev set differs from language to language.[8] However, as we will see in Figure 3, tuning this parameter does not substantially influence our results. We therefore will refer to the model with $0.4|\mathcal{V}|$ merges as BPE-RNNLM.

3 Aggregating Sentence Surprisals

Cotterell et al. (2018) evaluated the model for language j simply by its total surprisal $\sum_i \mathrm{NLL}(s_{ij})$. This comparative measure required a complete multitext corpus containing every sentence s_{ij} (the expression of the intent i in language j). We relax this requirement by using a fully probabilistic regression model that can deal with missing data

[5] The former might be the result of overt marking of, say, evidentiality or gender, which adds information. We hope that these differences are taken care of by diligent translators producing faithful translations in our multitext corpus.

[6] We restrict the set of characters to those that we see at least 25 times in the training set, replacing all others with a new symbol ◇, as is common and easily defensible in open-vocabulary language modeling (Mielke and Eisner, 2018). We make an exception for Chinese, where we only require each character to appear at least twice. These thresholds result in negligible "out-of-alphabet" rates for all languages.

[7] In practice, in both training and testing, we only evaluate the probability of the canonical segmentation of the held-out string, rather than the total probability of all segmentations (Kudo, 2018; Mielke and Eisner, 2018, Appendix D.2).

[8] Figure 2 shows the 21 languages of the Europarl dataset. Optimal values: 0.2 (et); 0.3 (fi, lt); 0.4 (de, es, hu, lv, sk, sl); 0.5 (da, fr, pl, sv); 0.6 (bg, ru); 0.7 (el); 0.8 (en); 0.9 (it, pt).

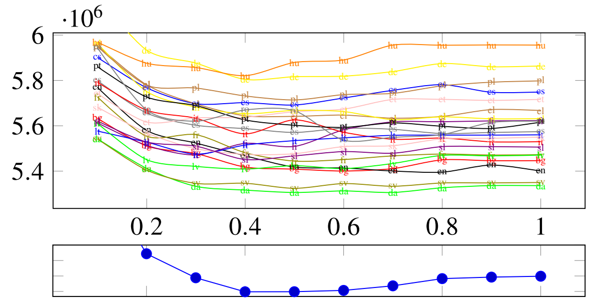


Figure 2: Top: For each language, total NLL of the dev corpus varies with the number of BPE merges, which is expressed on the x-axis as a fraction of the number of observed word types $|\mathcal{V}|$.[8] Bottom: Averaging over all 21 languages motivates a global value of 0.4.

(Figure 1).[9] Our model predicts each sentence's surprisal $y_{ij} = \text{NLL}(s_{ij})$ using an intent-specific "information content" factor n_i, which captures the inherent surprisal of the intent, combined with a language-specific difficulty factor d_j. This represents a better approach to varying sentence lengths and lets us work with missing translations in the test data (though it does not remedy our need for fully parallel language model training data).

3.1 Model 1: Multiplicative Mixed-effects

Model 1 is a multiplicative mixed-effects model:

$$y_{ij} = n_i \cdot \exp(d_j) \cdot \exp(\epsilon_{ij}) \tag{1}$$

$$\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2) \tag{2}$$

This says that each intent i has a latent **size** of n_i—measured in some abstract "informational units"—that is observed indirectly in the various sentences s_{ij} that express the intent. Larger n_i tend to yield longer sentences. Sentence s_{ij} has y_{ij} bits of surprisal; thus the multiplier y_{ij}/n_i represents the number of bits that *language* j used to express each informational unit of intent i, under our language model of language j. Our mixed-effects model assumes that this multiplier is log-normally distributed over the sentences i: that is, $\log(y_{ij}/n_i) \sim \mathcal{N}(d_j, \sigma^2)$, where mean d_j is the **difficulty** of language j. That is, $y_{ij}/n_i = \exp(d_j + \epsilon_{ij})$ where $\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$ is residual noise, yielding equations (1)–(2).[10] We jointly fit the intent sizes n_i and the language difficulties d_j.

3.2 Model 2: Heteroscedasticity

Because it is multiplicative, Model 1 appropriately predicts that in each language j, intents with large n_i will not only have larger y_{ij} values but these values will vary more widely. However, Model 1 is **homoscedastic**: the variance σ^2 of $\log(y_{ij}/n_i)$ is assumed to be independent of the independent variable n_i, which predicts that the distribution of y_{ij} should spread out *linearly* as the information content n_i increases: e.g., $p(y_{ij} \geq 13 \mid n_i = 10) = p(y_{ij} \geq 26 \mid n_i = 20)$. That assumption is questionable, since for a longer sentence, we would expect $\log y_{ij}/n_i$ to come closer to its mean d_j as the random effects of individual translational choices average out.[11] We address this issue by assuming that y_{ij} results from $n_i \in \mathbb{N}$ independent choices:

$$y_{ij} = \exp(d_j) \cdot \left(\sum_{k=1}^{n_i} \exp \epsilon_{ijk} \right) \tag{3}$$

$$\epsilon_{ijk} \sim \mathcal{N}(0, \sigma^2) \tag{4}$$

The number of bits for the k^{th} informational unit now varies by a factor of $\exp \epsilon_{ijk}$ that is log-normal and independent of the other units. It is common to approximate the sum of independent log-normals by another log-normal distribution, matching mean and variance (Fenton-Wilkinson approximation; Fenton, 1960),[12] yielding Model 2:

$$y_{ij} = n_i \cdot \exp(d_j) \cdot \exp(\epsilon_{ij}) \tag{1}$$

$$\sigma_i^2 = \ln \left(1 + \tfrac{\exp(\sigma^2)-1}{n_i} \right) \tag{5}$$

$$\epsilon_{ij} \sim \mathcal{N}\left(\tfrac{\sigma^2 - \sigma_i^2}{2}, \; \sigma_i^2 \right), \tag{6}$$

in which the noise term ϵ_{ij} now depends on n_i. Unlike (4), this formula no longer requires $n_i \in \mathbb{N}$; we allow any $n_i \in \mathbb{R}_{>0}$, which will also let us use gradient descent in estimating n_i.

In effect, fitting the model chooses each n_i so that the resulting intent-specific but language-independent distribution of $n_i \cdot \exp(\epsilon_{ij})$ values,[13]

[9] Specifically, we deal with data missing completely at random (MCAR), a strong assumption on the data generation process. More discussion on this can be found in Appendix A.

[10] It is tempting to give each language its own σ_j^2 parameter, but then the MAP estimate is pathological, since infinite likelihood can be attained by setting one language's σ_j^2 to 0.

[11] Similarly, flipping a fair coin 10 times results in 5 ± 1.58 heads where 1.58 represents the standard deviation, but flipping it 20 times does not result in $10 \pm 1.58 \cdot 2$ heads but rather $10 \pm 1.58 \cdot \sqrt{2}$ heads. Thus, with more flips, the ratio heads/flips tends to fall closer to its mean 0.5.

[12] There are better approximations, but even the only slightly more complicated Schwartz-Yeh approximation (Schwartz and Yeh, 1982) already requires costly and complicated approximations in addition to lacking the generalizability to non-integral n_i values that we will obtain for the Fenton-Wilkinson approximation.

[13] The distribution of ϵ_{ij} is the same for every j. It no longer has mean 0, but it depends only on n_i.

after it is scaled by $\exp(d_j)$ for each language j, will assign high probability to the observed y_{ij}. Notice that in Model 2, the scale of n_i becomes meaningful: fitting the model will choose the size of the abstract informational units so as to predict how rapidly σ_i falls off with n_i. This contrasts with Model 1, where doubling all the n_i values could be compensated for by halving all the $\exp(d_j)$ values.

3.3 Model 2L: An Outlier-Resistant Variant

One way to make Model 2 more outlier-resistant is to use a Laplace distribution[14] instead of a Gaussian in (6) as an approximation to the distribution of ϵ_{ij}. The Laplace distribution is heavy-tailed, so it is more tolerant of large residuals. We choose its mean and variance just as in (6). This heavy-tailed ϵ_{ij} distribution can be viewed as approximating a version of Model 2 in which the ϵ_{ijk} themselves follow some heavy-tailed distribution.

3.4 Estimating model parameters

We fit each regression model's parameters by L-BFGS. We then evaluate the model's fitness by measuring its held-out data likelihood—that is, the probability it assigns to the y_{ij} values for held-out intents i. Here we use the previously fitted d_j and σ parameters, but we must newly fit n_i values for the new i using MAP estimates or posterior means. A full comparison of our models under various conditions can be found in Appendix C. The primary findings are as follows. On Europarl data (which has fewer languages), Model 2 performs best. On the Bible corpora, all models are relatively close to one another, though the robust Model 2L gets more consistent results than Model 2 across data subsets. We use MAP estimates under Model 2 for all remaining experiments for speed and simplicity.[15]

3.5 A Note on Bayesian Inference

As our model of y_{ij} values is fully generative, one could place priors on our parameters and do full inference of the posterior rather than performing MAP inference. We did experiment with priors but found them so quickly overruled by the data that it did not make much sense to spend time on them.

Specifically, for full inference, we implemented all models in STAN (Carpenter et al., 2017), a toolkit for fast, state-of-the-art inference using Hamiltonian Monte Carlo (HMC) estimation. Running HMC unfortunately scales sublinearly with the number of sentences (and thus results in very long sampling times), and the posteriors we obtained were unimodal with relatively small variances (see also Appendix C). We therefore work with the MAP estimates in the rest of this paper.

4 The Difficulties of 69 languages

Having outlined our method for estimating language difficulty scores d_j, we now seek data to do so for all our languages. If we wanted to cover the most languages possible with parallel text, we should surely look at the Universal Declaration of Human Rights, which has been translated into over 500 languages. Yet this short document is far too small to train state-of-the-art language models. In this paper, we will therefore follow previous work in using the Europarl corpus (Koehn, 2005), but also for the first time make use of 106 Bibles from Mayer and Cysouw (2014)'s corpus.

Although our regression models of the surprisals y_{ij} can be estimated from incomplete multitext, the surprisals themselves are derived from the language models we are comparing. To ensure that the language models are comparable, we want to train them on completely parallel data in the various languages. For this, we seek complete multitext.

4.1 Europarl: 21 Languages

The Europarl corpus (Koehn, 2005) contains decades worth of discussions of the European Parliament, where each intent appears in up to 21 languages. It was previously used by Cotterell et al. (2018) for its size and stability. In §6, we will also exploit the fact that each intent's original language is known. To simplify our access to this information, we will use the "Corrected & Structured Europarl Corpus" (CoStEP) corpus (Graën et al., 2014). From it, we extract the intents that appear in all 21 languages, as enumerated in footnote 8. The full extraction process and corpus statistics are detailed in Appendix D.

4.2 The Bible: 62 Languages

The Bible is a religious text that has been used for decades as a dataset for massively multilingual NLP (Resnik et al., 1999; Yarowsky et al., 2001; Agić et al., 2016). Concretely, we use the

[14] One could also use a Cauchy distribution instead of the Laplace distribution to get even heavier tails, but we saw little difference between the two in practice.

[15] Further enhancements are possible: we discuss our "Model 3" in Appendix B, but it did not seem to fit better.

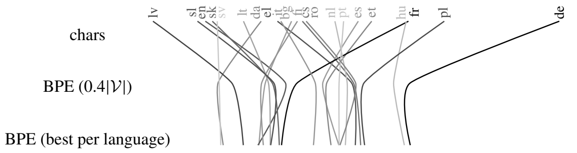


Figure 3: The Europarl language difficulties appear more similar, and are ordered differently, when the RNN models use BPE units instead of character units. Tuning BPE per-language has a small additional effect.

tokenized[16] and aligned collection assembled by Mayer and Cysouw (2014). We use the smallest annotated subdivision (a single *verse*) as a sentence in our difficulty estimation model; see footnote 2.

Some of the Bibles in the dataset are incomplete. As the Bibles include different sets of verses (intents), we have to select a set of Bibles that overlap strongly, so we can use the verses shared by all these Bibles to comparably train all our language models (and fairly test them: see Appendix A). We cast this selection problem as an integer linear program (ILP), which we solve exactly in a few hours using the Gurobi solver (more details on this selection in Appendix E). This optimal solution keeps 25996 verses, each of which appears across 106 Bibles in 62 languages,[17] spanning 13 language families.[18] We allow j to range over the 106 Bibles, so when a language has multiple Bibles, we estimate a separate difficulty d_j for each one.

4.3 Results

The estimated difficulties are visualized in Figure 4. We can see that general trends are preserved between datasets: German and Hungarian are hardest, English and Lithuanian easiest. As we can see in Figure 3 for Europarl, the difficulty estimates are hardly affected when tuning the number of BPE merges per-language instead of globally, validating our approach of using the BPE model for our experiments. A bigger difference seems to be the choice of char-RNNLM vs. BPE-RNNLM, which changes the ranking of languages both on Europarl data and on Bibles. We still see German as the hardest language, but almost all other languages switch places. Specifically, we can see that the variance of the char-RNNLM is much higher.

4.4 Are All Translations the Same?

Texts like the Bible are justly infamous for their sometimes archaic or unrepresentative use of language. The fact that we sometimes have multiple Bible translations in the same language lets us observe variation by translation style.

The sample standard deviation of d_j among the 106 Bibles j is 0.076/0.063 for BPE/char-RNNLM. Within the 11 German, 11 French, and 4 English Bibles, the sample standard deviations were roughly 0.05/0.04, 0.05/0.04, and 0.02/0.04 respectively: so style accounts for less than half the variance. We also consider another parallel corpus, created from the NIST OpenMT competitions on machine translation, in which each sentence has 4 English translations (NIST Multimodal Information Group, 2010a,b,c,d,e,f,g, 2013b,a). We get a sample standard deviation of 0.01/0.03 among the 4 resulting English corpora, suggesting that language difficulty estimates (particularly the BPE estimate) depend less on the translator, to the extent that these corpora represent individual translators.

5 What Correlates with Difficulty?

Making use of our results on these languages, we can now answer the question: what features of a language correlate with the difference in language complexity? Sadly, we cannot conduct all analyses on all data: the Europarl languages are well-served by existing tools like UDPipe (Straka et al., 2016), but the languages of our Bibles are often not. We therefore conduct analyses that rely on automatically extracted features only on the Europarl corpora. Note that to ensure a false discovery rate of at most $\alpha = .05$, all reported p-values have to be corrected using Benjamini and Hochberg (1995)'s procedure: only $p \leq .05 \cdot 5/28 \approx 0.009$ is significant.

Morphological Counting Complexity Cotterell et al. (2018) suspected that inflectional morphology (i.e., the grammatical requirement to

[16] The fact that the resource is tokenized is (yet) another possible confound for this study: we are not comparing performance on languages, but on languages/Bibles *with some specific translator and tokenization*. It is possible that our y_{ij} values for each language j depend to a small degree on the tokenizer that was chosen for that language.

[17] afr, aln, arb, arz, ayr, bba, ben, bqc, bul, cac, cak, ceb, ces, cmn, cnh, cym, dan, deu, ell, eng, epo, fin, fra, guj, gur, hat, hrv, hun, ind, ita, kek, kjb, lat, lit, mah, mam, mri, mya, nld, nor, plt, poh, por, qub, quh, quy, quz, ron, rus, som, tbz, tcw, tgl, tlh, tpi, tpm, ukr, vie, wal, wbm, xho, zom

[18] 22 Indo-European, 6 Niger-Congo, 6 Mayan, 6 Austronesian, 4 Sino-Tibetan, 4 Quechuan, 4 Afro-Asiatic, 2 Uralic, 2 Creoles, 2 Constructed languages, 2 Austro-Asiatic, 1 Totonacan, 1 Aymaran. For each language, we are reporting here the first family listed by Ethnologue (Paul et al., 2009), manually fixing tlh ↦ Constructed language. It is unfortunate not to have more families or more languages per family. A broader sample could be obtained by taking only the New Testament—but unfortunately that has < 8000 verses, a meager third of our dataset that is already smaller that the usually considered tiny PTB dataset (see details in Appendix E).

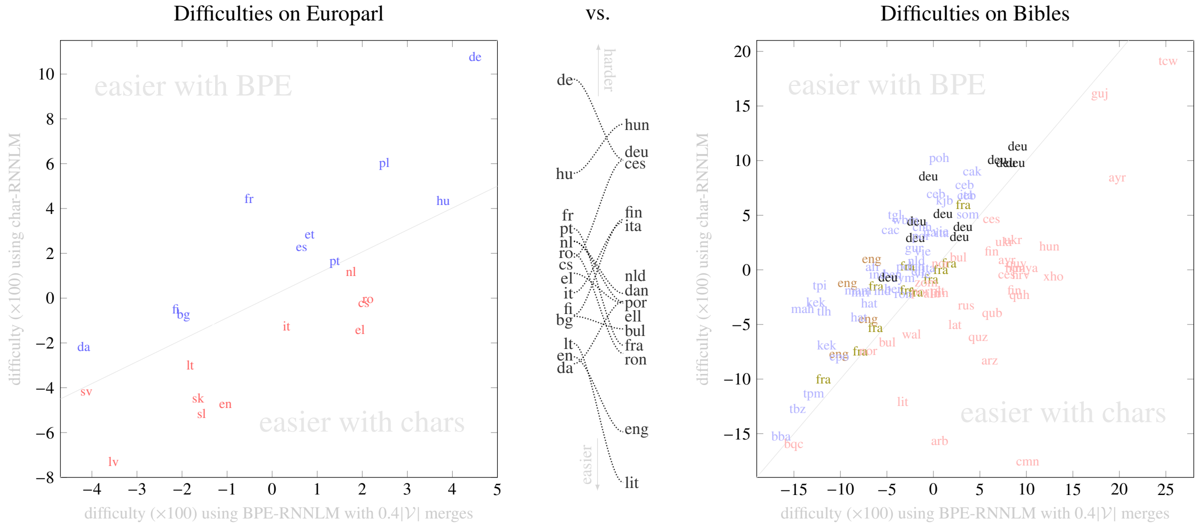


Figure 4: Difficulties of 21 Europarl languages (left) and 106 Bibles (right), comparing difficulties when estimated from BPE-RNNLMs vs. char-RNNLMs. Highlighted on the right are deu and fra, for which we have many Bibles, and eng, which has often been prioritized even over these two in research. In the middle we see the difficulties of the 14 languages that are shared between the Bibles and Europarl aligned to each other (averaging all estimates), indicating that the general trends we see are not tied to either corpus.

choose among forms like “talk,” “talks,” “talking”) was mainly responsible for difficulty in modeling. They found a language’s Morphological Counting Complexity (Sagot, 2013) to correlate positively with its difficulty. We use the reported MCC values from that paper for our 21 Europarl languages, but to our surprise, find no statistically significant correlation with the newly estimated difficulties of our new language models. Comparing the scatterplot for both languages in Figure 5 with Cotterell et al. (2018)’s Figure 1, we see that the high-MCC outlier Finnish has become much easier in our (presumably) better-tuned models. We suspect that the reported correlation in that paper was mainly driven by such outliers and conclude that MCC is not a good predictor of modeling difficulty. Perhaps finer measures of morphological complexity would be more predictive.

Head-POS Entropy Dehouck and Denis (2018) propose an alternative measure of morphosyntactic complexity. Given a corpus of dependency graphs, they estimate the conditional entropy of the POS tag of a random token’s parent, conditioned on the token’s type. In a language where this **HPE-mean** metric is low, most tokens can predict the POS of their parent even without context. We compute HPE-mean from dependency parses of the Europarl data, generated using UDPipe 1.2.0 (Straka et al., 2016) and freely-available tokenization, tagging, parsing models trained on the Universal Dependencies 2.0 treebanks (Straka and Strakov, 2017).

HPE-mean may be regarded as the mean over all corpus tokens of *Head POS Entropy* (Dehouck and Denis, 2018), which is the entropy of the POS tag of a token’s parent given that *particular* token’s type. We also compute **HPE-skew**, the (positive) skewness of the empirical distribution of HPE on the corpus tokens. We remark that in each language, HPE is 0 for most tokens.

As predictors of language difficulty, HPE-mean has a Spearman’s $\rho = .004/{-.045}$ ($p > .9/.8$) and HPE-skew has a Spearman’s $\rho = .032/.158$ ($p > .8/.4$), so this is not a positive result.

Average dependency length It has been observed that languages tend to minimize the distance between heads and dependents (Liu, 2008). Speakers prefer shorter dependencies in both production and processing, and average dependency lengths tend to be much shorter than would be expected from randomly-generated parses (Futrell et al., 2015; Liu et al., 2017). On the other hand, there is substantial variability between languages, and it has been proposed, for example, that head-final languages and case-marking languages tend to have longer dependencies on average.

Do language *models* find short dependencies easier? We find that average dependency lengths estimated from automated parses are very closely correlated with those estimated from (held-out) manual parse trees. We again use the automatically-parsed Europarl data and compute dependency lengths using the Futrell et al. (2015) procedure,

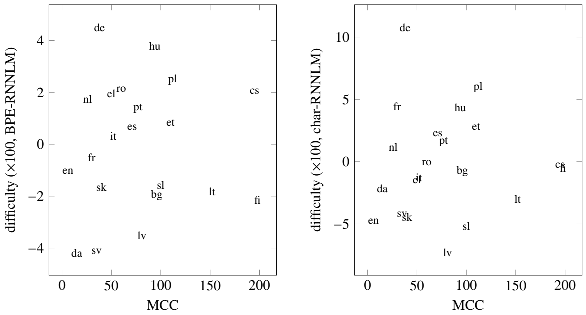


Figure 5: MCC does not predict difficulty on Europarl. Spearman's ρ is .091 / .110 with $p > .6$ for BPE-RNNLM (left) / char-RNNLM (right).

which excludes punctuation and standardizes several other grammatical relationships (e.g., objects of prepositions are made to depend on their prepositions, and verbs to depend on their complementizers). Our hypothesis that scrambling makes language harder to model seems confirmed at first: while the non-parametric (and thus more weakly powered) Spearman's $\rho = .196/.092$ ($p = .394/.691$), Pearson's $r = .486/.522$ ($p = .032/.015$). However, after correcting for multiple comparisons, this is also non-significant.[19]

WALS features The World Atlas of Language Structures (WALS; Dryer and Haspelmath, 2013) contains nearly 200 binary and integer features for over 2000 languages. Similarly to the Bible situation, not all features are present for all languages—and for some of our Bibles, no information can be found at all. We therefore restrict our attention to two well-annotated WALS features that are present in enough of our Bible languages (foregoing Europarl to keep the analysis simple): 26A "Prefixing vs. Suffixing in Inflectional Morphology" and 81A "Order of Subject, Object and Verb." The results are again not quite as striking as we would hope. In particular, in Mood's median null hypothesis significance test neither 26A ($p > .3$ / .7 for BPE/char-RNNLM) nor 81A ($p > .6$ / .2 for BPE/char-RNNLM) show any significant differences between categories (detailed results in Appendix F.1). We therefore turn our attention to much simpler, yet strikingly effective heuristics.

Raw character sequence length An interesting correlation emerges between language difficulty for the char-RNNLM and the raw length in characters of the test corpus (detailed results in Appendix F.2). On both Europarl and the more reliable Bible corpus, we have positive correlation for the char-RNNLM at a significance level of $p < .001$, passing the multiple-test correction. The BPE-RNNLM correlation on the Bible corpus is very weak, suggesting that allowing larger units of prediction effectively eliminates this source of difficulty (van Merriënboer et al., 2017).

Raw word inventory Our most predictive feature, however, is the *size of the word inventory*. To obtain this number, we count the number of distinct types $|\mathcal{V}|$ in the (tokenized) training set of a language (detailed results in Appendix F.3).[20] While again there is little power in the small set of Europarl languages, on the bigger set of Bibles we do see the biggest positive correlation of any of our features—but only on the BPE model ($p < 1e{-}11$). Recall that the char-RNNLM has no notion of words, whereas the number of BPE units increases with $|\mathcal{V}|$ (indeed, many whole words are BPE units, because we do many merges but BPE stops at word boundaries). Thus, one interpretation is that the Bible corpora are too small to fit the parameters for all the units needed in large-vocabulary languages. A similarly predictive feature on Bibles—whose numerator is this word inventory size—is the type/token ratio, where values closer to 1 are a traditional omen of undertraining.

An interesting observation is that on Europarl, the size of the word inventory and the morphological counting complexity of a language correlate quite well with each other (Pearson's $\rho = .693$ at $p = .0005$, Spearman's $\rho = .666$ at $p = .0009$), so the original claim in Cotterell et al. (2018) about MCC may very well hold true after all. Unfortunately, we cannot estimate the MCC for all the Bible languages, or this would be easy to check.[21]

Given more nuanced linguistic measures (or more languages), our methods may permit discovery of specific linguistic correlates of modeling difficulty, beyond these simply suggestive results.

[19] We also caution that the significance test for Pearson's assumes that the two variables are bivariate normal. If not, then even a significant r does not allow us to reject the null hypothesis of zero covariance (Kowalski, 1972, Figs. 1–2, §5).

[20] A more sophisticated version of this feature might consider not just the existence of certain forms but also their rates of appearance. We did calculate the entropy of the unigram distribution over words in a language, but we found that is strongly correlated with the size of the word inventory and not any more predictive.

[21] Perhaps in a future where more data has been annotated by the UniMorph project (Kirov et al., 2018), a yet more comprehensive study can be performed, and the null hypothesis for the MCC can be ruled out after all.

6 Evaluating Translationese

Our previous experiments treat translated sentences just like natively generated sentences. But since Europarl contains information about which language an intent was originally expressed in,[22] here we have the opportunity to ask another question: is translationese harder, easier, indistinguishable, or impossible to tell? We tackle this question by splitting each language j into two sub-languages, "native" j and "translated" j, resulting in 42 sub-languages with 42 difficulties.[23] Each intent is expressed in at most 21 sub-languages, so this approach *requires* a regresssion method that can handle missing data, such as the probabilistic approach we proposed in §3. Our mixed-effects modeling ensures that our estimation focuses on the differences between languages, controlling for content by automatically fitting the n_i factors. Thus, we are not in danger of calling native German more complicated than translated German just because German speakers in Parliament may like to talk about complicated things in complicated ways.

In a first attempt, we simply use our already-trained BPE-best models (as they perform the best and are thus most likely to support claims about the language itself rather than the shortcomings of any singular model), limit ourselves to only splitting the eight languages that have at least 500 native sentences[24] (to ensure stable results). Indeed we *seem* to find that native sentences are slightly more difficult: their d_j is 0.027 larger (± 0.023, averaged over our selected 8 languages).

But are they? This result is confounded by the fact that our RNN language models *were trained* mostly on translationese text (even the English data is mostly translationese). Thus, translationese might merely be *different* (Rabinovich and Wintner, 2015)—not necessarily easier to model, but over-represented when training the model, making the translationese test sentences more predictable. To remove this confound, we must train our language models on equal parts translationese and native text. We cannot do this for multiple languages at once, given our requirement of training all language models on the same intents. We thus choose to balance only *one* language—we train all models for all languages, making sure that the training set for one language is balanced—and then perform our regression, reporting the translationese and native difficulties only for the balanced language. We repeat this process for every language that has enough intents. We sample equal numbers of native and non-native sentences, such that there are ~1M words in the corresponding English column (to be comparable to the PTB size). To raise the number of languages we can split in this way, we restrict ourselves here to fully-parallel Europarl in only 10 languages[25] instead of 21, thus ensuring that each of these 10 languages has enough native sentences.

On this level playing field, the previously observed effect practically disappears (-0.0044 ± 0.022), leading us to question the widespread hypothesis that translationese is "easier" to model (Baker, 1993).[26]

7 Conclusion

There is a real danger in cross-linguistic studies of over-extrapolating from limited data. We re-evaluated the conclusions of Cotterell et al. (2018) on a larger set of languages, requiring new methods to select fully parallel data (§4.2) or handle missing data. We showed how to fit a paired-sample multiplicative mixed-effects model to probabilistically obtain language difficulties from at-least-pairwise parallel corpora. Our language difficulty estimates were largely stable across datasets and language model architectures, but they were not significantly predicted by linguistic factors. However, a language's vocabulary size and the length in characters of its sentences were well-correlated with difficulty on our large set of languages. Our mixed-effects approach could be used to assess other NLP systems via parallel texts, separating out the influences on performance of language, sentence, model architecture, and training procedure.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. 1718846.

[22] It should be said that using Europarl for translationese studies is not without caveats (Rabinovich et al., 2016), one of them being the fact that not all language pairs are translated equally: a natively Finnish sentence is translated first into English, French, or German (**pivoting**) and only from there into any other language like Bulgarian.

[23] This method would also allow us to study the effect of source language, yielding $d_{j\leftarrow j'}$ for sentences translated from j' into j. Similarly, we could have included surprisals from *both* models, *jointly* estimating $d_{j,\text{char-RNN}}$ and $d_{j,\text{BPE}}$ values.

[24] en (3256), fr (1650), de (1275), pt (1077), it (892), es (685), ro (661), pl (594)

[25] da, de, en, es, fi, fr, it, nl, pt, sv

[26] Of course we *cannot* claim that it is just as hard to *read* or *translate* as native text—those are different claims altogether—but only that it is as easy to monolingually language-model.

References

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Mona Baker. 1993. Corpus linguistics and translation studies: Implications and applications. *Text and Technology: In Honour of John Sinclair*, pages 233–250.

Emily M. Bender. 2009. Linguistically naïve != language independent: Why NLP needs linguistic typology. In *EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics*, pages 26–32.

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300.

Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. Stan: A probabilistic programming language. *Journal of Statistical Software, Articles*, 76(1):1–32.

Ryan Cotterell, Sebastian J. Mielke, Jason Eisner, and Brian Roark. 2018. Are all languages equally hard to language-model? In *Proceedings of NAACL*, pages 536–541.

Mathieu Dehouck and Pascal Denis. 2018. A framework for understanding the role of morphology in universal dependency parsing. In *Proceedings of EMNLP*, pages 2864–2870.

Chris Drummond. 2009. Replicability is not reproducibility: Nor is it good science. In *Proceedings of the Evaluation Methods for Machine Learning Workshop at the 26th ICML*.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Lawrence Fenton. 1960. The sum of log-normal probability distributions in scatter transmission systems. *IRE Transactions on Communications Systems*, 8(1):57–67.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Johannes Graën, Dolores Batinic, and Martin Volk. 2014. Cleaning the Europarl corpus for linguistic applications. In *Konvens*, pages 222–227.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Arya McCarthy, Sebastian J. Mielke, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. Unimorph 2.0: Universal morphology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA).

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, pages 79–86.

Charles J. Kowalski. 1972. On the effects of non-normality on the distribution of the sample product-moment correlation coefficient. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 21(1):1–12.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia.

Gennadi Lembersky, Noam Ordan, and Shuly Wintner. 2012. Adapting translation models to translationese improves SMT. In *Proceedings of EACL*, pages 255–265.

Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191.

Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of Life Reviews*, 21:171–193.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of LREC*, pages 3158–3163.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. An analysis of neural language modeling at multiple scales. *arXiv preprint arXiv:1803.08240*.

Bart van Merriënboer, Amartya Sanyal, Hugo Larochelle, and Yoshua Bengio. 2017. Multiscale sequence modeling with a learned dictionary. *arXiv preprint arXiv:1707.00762*.

Sebastian J. Mielke and Jason Eisner. 2018. Spell once, summon anywhere: A two-level open-vocabulary language model. *arXiv preprint arXiv:1804.08205*.

NIST Multimodal Information Group. 2010a. NIST 2002 Open Machine Translation (OpenMT) evaluation LDC2010T10.

NIST Multimodal Information Group. 2010b. NIST 2003 Open Machine Translation (OpenMT) evaluation LDC2010T11.

NIST Multimodal Information Group. 2010c. NIST 2004 Open Machine Translation (OpenMT) evaluation LDC2010T12.

NIST Multimodal Information Group. 2010d. NIST 2005 Open Machine Translation (OpenMT) evaluation LDC2010T14.

NIST Multimodal Information Group. 2010e. NIST 2006 Open Machine Translation (OpenMT) evaluation LDC2010T17.

NIST Multimodal Information Group. 2010f. NIST 2008 Open Machine Translation (OpenMT) evaluation LDC2010T21.

NIST Multimodal Information Group. 2010g. NIST 2009 Open Machine Translation (OpenMT) evaluation LDC2010T23.

NIST Multimodal Information Group. 2013a. NIST 2008-2012 Open Machine Translation (OpenMT) progress test sets LDC2013T07.

NIST Multimodal Information Group. 2013b. NIST 2012 Open Machine Translation (OpenMT) evaluation LDC2013T03.

Lewis M. Paul, Gary F. Simons, Charles D. Fennig, et al. 2009. *Ethnologue: Languages of the world*, 19 edition. SIL International, Dallas.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL*, pages 2227–2237.

Ella Rabinovich and Shuly Wintner. 2015. Unsupervised identification of translationese. *Transactions of the Association for Computational Linguistics*, 3:419–432.

Ella Rabinovich, Shuly Wintner, and Ofek Luis Lewinsohn. 2016. A parallel corpus of translationese. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 140–155. Springer.

Philip Resnik, Mari Broman Olsen, and Mona Diab. 1999. The bible as a parallel corpus: Annotating the 'book of 2000 tongues'. *Computers and the Humanities*, 33(1):129–153.

Benoît Sagot. 2013. Comparing complexity measures. In *Computational Approaches to Morphological Complexity*.

S. C. Schwartz and Y. S. Yeh. 1982. On the distribution function and moments of power sums with log-normal components. *The Bell System Technical Journal*, 61(7):1441–1462.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725.

Claude E. Shannon. 1951. Prediction and entropy of printed English. *Bell Labs Technical Journal*, 30(1):50–64.

Milan Straka, Jan Haji, and Jana Strakov. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of LREC*, pages 4290–4297.

Milan Straka and Jana Strakov. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *CoNLL 2017 Shared Task: Multilingual parsing from raw text to Universal Dependencies*, pages 88–99. Documented models at `http://hdl.handle.net/11234/1-2364`.

Ilya Sutskever, James Martens, and Geoffrey Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of ICML*, pages 1017–1024.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–8.

A A Note on Missing Data

We stated that our model can deal with missing data, but this is true only for the case of data **missing completely at random** (MCAR), the strongest assumption we can make about missing data: the missingness of data is neither influenced by what the value would have been (had it not been missing), nor by any covariates. Sadly, this assumption is rarely met in real translations, where difficult, useless, or otherwise *distinctive* sentences may be skipped. This leads to data **missing at random** (MAR), where the missingness of a translation is correlated with the original sentence it should have been translated from—or even data **missing not at random** (MNAR), where the missingness of a translation is correlated with that translation, i.e., the original sentence was translated, but the translation was then deleted for a reason that depends on the translation itself). For this reason we use fully parallel data where possible; in fact, we only make use of the ability to deal with missing data in §6.[27]

B Regression, Model 3: Handling outliers cleverly

Consider the problem of outliers. In some cases, sloppy translation will yield a y_{ij} that is unusually high or low given the y'_{ij} values of other languages j'. Such a y_{ij} is not good evidence of the quality of the language model for language j since it has been corrupted by the sloppy translation. However, under Model 1 or 2, we could not simply explain this corrupted y_{ij} with the random residual ϵ_{ij} since large $|\epsilon_{ij}|$ is highly unlikely under the Gaussian assumption of those models. Rather, y_{ij} would have significant influence on our estimate of the per-language effect d_j. This is the usual motivation for switching to L1 regression, which replaces the Gaussian prior on the residuals with a Laplace prior.[28]

How can we include this idea into our models? First let us identify two failure modes:

(a) part of a sentence was omitted (or added) during translation, changing the n_i additively; thus we should use a noisy $n_i + \nu_{ij}$ in place of n_i in equations (1) and (5)

(b) the style of the translation was unusual throughout the sentence; thus we should use a noisy $n_i \cdot \exp \nu_{ij}$ instead of n_i in equations (1) and (5)

In both cases $\nu_{ij} \sim \text{Laplace}(0, b)$, i.e., ν_{ij} specifies sparse additive or multiplicative noise in ν_{ij} (on language j only).[29]

Let us write out version (b), which is a modification of Model 2 (equations (1), (5) and (6)):

$$\begin{aligned} y_{ij} &= (n_i \cdot \exp \nu_{ij}) \cdot \exp(d_j) \cdot \exp(\epsilon_{ij}) \\ &= n_i \cdot \exp(d_j) \cdot \exp(\epsilon_{ij} + \nu_{ij}) && (7) \\ \nu_{ij} &\sim \text{Laplace}(0, b) && (8) \\ \sigma_i^2 &= \ln\left(1 + \tfrac{\exp(\sigma^2)-1}{n_i \cdot \exp \nu_{ij}}\right) && (9) \\ \epsilon_{ij} &\sim \mathcal{N}\left(\tfrac{\sigma^2 - \sigma_i^2}{2},\ \sigma_i^2\right), && (10) \end{aligned}$$

Comparing equation (7) to equation (1), we see that we are now modeling the residual error in $\log y_{ij}$ as a *sum of two noise terms* $a_{ij} = \nu_{ij} + \epsilon_{ij}$ and penalizing it by (some multiple of) the weighted sum of $|\nu_{ij}|$ and ϵ_{ij}^2, where large errors can be more cheaply explained using the former summand, and small errors using the latter summand.[30] The weighting of the two terms is a tunable hyperparameter.

We did implement this model and test it on data, but not only was fitting it much harder and slower, it also did not yield particularly encouraging results, leading us to omit it from the main text.

C Goodness of fit of our difficulty estimation models

Figure 6 shows the log-probability of held-out data under the regression model, by fixing the estimated difficulties d_j (and sometimes also the estimated variance σ^2) to their values obtained from training data, and then finding either MAP estimates or posterior means (by running HMC using STAN) of

[27] Note that this application counts as data MAR and not MCAR, thus technically violating our requirements, but only in a minor enough way that we are confident it can still be applied.

[28] An alternative would be to use a method like RANSAC to discard y_{ij} values that do not appear to fit.

[29] However, version (a) is then deficient since it then incorrectly allocates some probability mass to $n_i + \nu_{ij} < 0$ and thus $y_{ij} < 0$ is possible. This could be fixed by using a different sparsity-inducing distribution.

[30] The cheapest penalty or explanation of the weighted sum $\delta|\nu_{ij}| + \frac{1}{2}\epsilon_{ij}^2$ for some weighting or threshold δ (which adjusts the relative variances of the two priors) is $\nu = 0$ if $|a| \leq \delta$, $\nu = a - \delta$ if $a \geq \delta$, and $\nu = -(a - \delta)$ if $a < -\delta$ (found by minimizing $\delta|\nu| + \frac{1}{2}(a - \nu)^2$, a convex function of ν). This implies that we incur a quadratic penalty $\frac{1}{2}a^2$ if $|a| \leq \delta$, and a linear penalty $\delta(|a| - \frac{1}{2}\delta)$ for the other cases; this penalty function is exactly the Huber loss of a, and essentially imposes an L2 penalty on small residuals and an L1 penalty on large residuals (outliers), so our estimate of d_j will be something between a mean and a median.

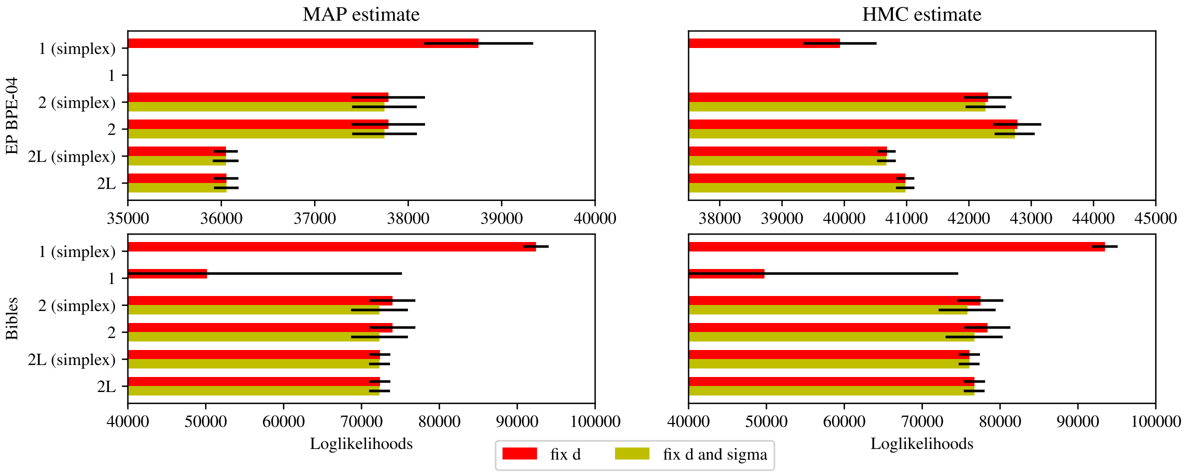


Figure 6: Achieved log-likelihoods on held-out data. Top: Europarl (BPE), Bottom: Bibles, Left: MAP inference, Right: HMC inference (posterior mean).

the other parameters, in particular n_i for the new sentences i. The error bars are the standard deviations when running the model over different subsets of data. The "simplex" versions of regression in Figure 6 force all d_j to add up to the number of languages (i.e., encouraging each one to stay close to 1). This is *necessary* for Model 1, which otherwise is unidentifiable (hence the enormous standard deviation). For other models, it turns out to only have much of an effect on the posterior means, not on the log-probability of held out data under the MAP estimate. For stability, we in all cases take the best result when initializing the new parameters randomly or "sensibly," i.e., the n_i of an intent i is initialized as the average of the corresponding sentences' y_{ij}.

D Data selection: Europarl

In the "Corrected & Structured Europarl Corpus" (CoStEP) corpus (Graën et al., 2014), sessions are grouped into *turns*, each turn has one speaker (that is marked with clean attributes like native language) and a number of aligned *paragraphs* for each language, i.e., the actual multitext.

We ignore all paragraphs that are in *ill-fitting* turns (i.e., turns with an unequal number of paragraphs across languages, a clear sign of an incorrect alignment), losing roughly 27% of intents. After this cleaning step, only 14% of *intents* are represented in all 21 languages, see the distribution in Figure 7 (the peak at 11 languages is explained by looking at the raw number of sentences present in each language, shown in Figure 8).

Since we want a fair comparison, we use the

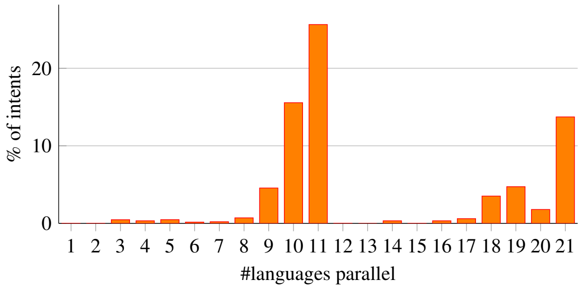


Figure 7: In how many languages are the intents in Europarl translated? (intents from ill-fitting turns included in 100%, but not plotted)

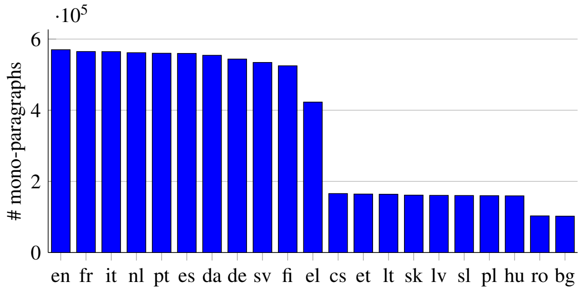


Figure 8: How many sentences are there per Europarl language?

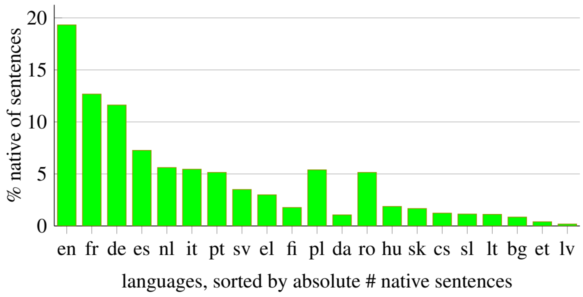


Figure 9: How many of the Europarl sentences in one language are "native"?

aforementioned 14% of Europarl, giving us 78169 intents that are represented in all 21 languages.

Finally, it should be said that the text in CoStEP itself contains some markup, marking reports, ellipses, etc., but we strip this additional markup to obtain the raw text. We tokenize it using the reversible language-agnostic tokenizer of Mielke and Eisner (2018)[31] and split the obtained 78169 paragraphs into training set, development set for tuning our language models, and test set for our regression, again by dividing the data into blocks of 30 paragraphs and then taking 5 sentences for the development and test set each, leaving the remainder for the training set. This way we ensure uniform division over sessions of the parliament and sizes of 2/3, 1/6, and 1/6, respectively.

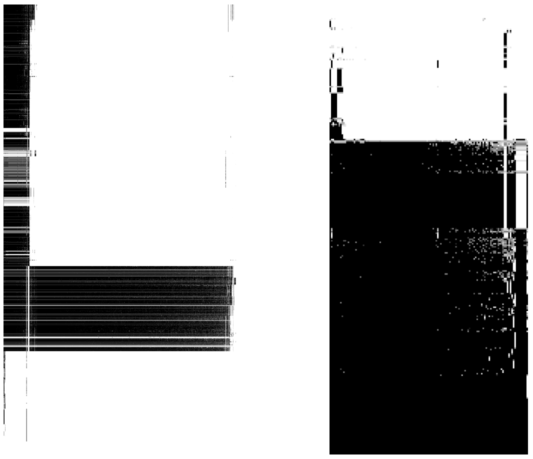
D.1 How are the source languages distributed?

An obvious question we should ask is: how many "native" sentences can we actually find in Europarl? One could assume that there are as many native *sentences* as there are *intents* in total, but there are three issues with this: the first is that the *president* in any Europarl session is never annotated with name or native language (leaving us guessing what the native version of any president-uttered intent is; 12% of all intents in Europarl that can be extracted have this problem), the second is that a number of speakers are labeled with "unknown" as native language (10% of sentences), and finally some speakers have their native language annotated, but it is nowhere to be found in the corresponding sentences (7% of sentences).

Looking only at the native sentences that we could identify, we can see that there are native sentences in every language, but unsurprisingly, some languages are overrepresented. Dividing the number of *native* sentences in a language by the number of *total* sentences, we get an idea of how "natively spoken" the language is in Europarl, shown in Figure 9.

E Data selection: Bibles

The Bible is composed of the *Old Testament* and the *New Testament* (the latter of which has been much more widely translated), both consisting of individual *books*, which, in turn, can be separated into *chapters*, but we will only work with the smallest subdivision unit: the *verse*, corresponding roughly to a sentence. Turning to the collection assembled by Mayer and Cysouw (2014), we see that it has over 1000 New Testaments, but far fewer complete Bibles.



(a) All 1174 Bibles, in packets of 20 verses, Bibles sorted by number of verses present, verses in chronological order. The New Testament (third quarter of verses) is present in almost every Bible.

(b) The 131 Bibles with at least 20000 verses, in packets of 150 verses (this time, both sorted). The optimization task is to remove rows and columns in this picture until only black remains.

Figure 10: Presence (black) of verses (y-axis) in Bibles (x-axis). Both pictures are downsampled, resulting in grayscale values for all packets of N values.

Despite being a fairly standardized book, not all Bibles are fully parallel. Some verses and sometimes entire books are missing in some Bibles—some of these discrepancies may be reduced to the question of the legitimacy of certain biblical books, others are simply artifacts of verse numbering and labeling of individual translations.

For us, this means that we can neither simply take all translations that have "the entire thing" (in fact, no single Bible in the set covers the union of all others' verses), nor can we take all Bibles and work with the verses that they all share (because, again, no single verse is shared over all given Bibles). The whole situation is visualized in Figure 10.

We have to find a tradeoff: take as many Bibles as possible that share as many verses as possible. Specifically, we cast this selection process as an optimization problem: select Bibles such that the number of verses overall (i.e., the number of verses shared times the number of Bibles) is maximal, breaking ties in favor of including more Bibles and ensuring that we have at least 20000 verses overall to ensure applicability of neural language models. This problem can be cast as an integer linear program and solved using a standard optimization tool

[31] http://sjmielke.com/papers/tokenize/

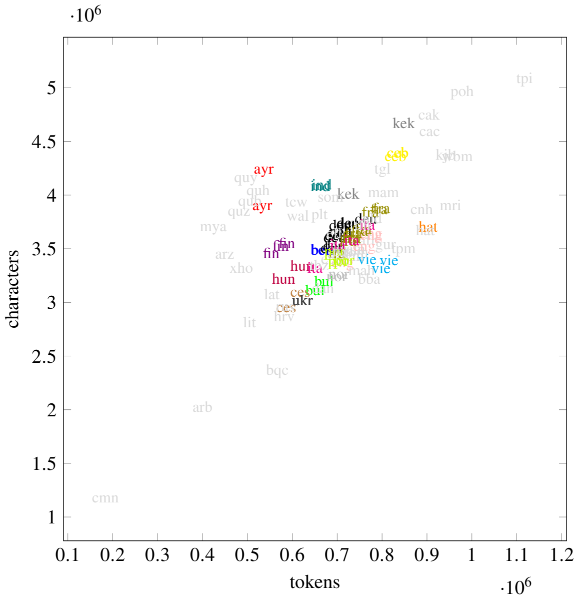


Figure 11: Tokens and characters (as reported by `wc -w/-m`) of the 106 Bibles. Equal languages share a color, all others are shown in faint gray. Most Bibles have around 700k tokens and 3.6M characters; outliers like Mandarin Chinese (cmn) are not surprising.

(Gurobi) within a few hours.

The optimal solution that we find contains 25996 verses for 106 Bibles in 62 languages,[32] spanning 13 language families.[33] The sizes of the selected Bible subsets are visualized for each Bible in Figure 11 and in relation to other datasets in Table 1.

We split them into train/dev/test by dividing the data into blocks of 30 paragraphs and then taking 5 sentences for the development and test set each, leaving the remainder for the training set. This way we ensure uniform division over books of the Bible and sizes of ²⁄₃, ¹⁄₆, and ¹⁄₆, respectively.

F Detailed regression results

F.1 WALS

We report the mean and sample standard deviation of language difficulties for languages that lie in the corresponding categories in Table 2:

English corpus	lines	words	chars
WikiText-103	1809468	101880752	543005627
Wikipedia ($\underset{\in[\text{a-z }]*}{\texttt{text8}}$)	1	17005207	100000000
Europarl	78169	6411731	37388604
WikiText-2	44836	2507005	13378183
PTB	49199	1036580	5951345
62/106-parallel Bible	25996	~700000	~3600000

Table 1: Sizes of various language modeling datasets, numbers estimated using `wc`.

26A (Inflectional Morphology)	BPE	chars
1 Little affixation (5)	-0.0263 (± .034)	0.0131 (± .033)
2 Strongly suffixing (22)	0.0037 (± .049)	-0.0145 (± .049)
3 Weakly suffixing (2)	0.0657 (± .007)	-0.0317 (± .074)
6 Strong prefixing (1)	0.1292	-0.0057
81A (Order of S, O and V)	BPE	chars
1 SOV (7)	0.0125 (± .106)	0.0029 (± .099)
2 SVO (18)	0.0139 (± .058)	-0.0252 (± .053)
3 VSO (5)	-0.0241 (± .041)	-0.0129 (± .089)
4 VOS (2)	0.0233 (± .026)	0.0353 (± .078)
7 No dominant order (4)	0.0252 (± .059)	0.0206 (± .029)

Table 2: Average difficulty for languages with certain WALS features (with number of languages).

F.2 Raw character sequence length

We report correlation measures and significance values when regressing on raw character sequence length in Table 3:

		BPE		char	
dataset	statistic	ρ	p	ρ	p
Europarl	Pearson	.509	.0185	**.621**	**.00264**
	Spearman	.423	.0558	**.560**	**.00832**
Bibles	Pearson	.015	.917	**.527**	**.000013**
	Spearman	.014	.915	**.434**	**.000481**

Table 3: Correlations and significances when regressing on raw character sequence length. Significant correlations are boldfaced.

F.3 Raw word inventory

We report correlation measures and significance values when regressing on the size of the raw word inventory in Table 4:

		BPE		char	
dataset	statistic	ρ	p	ρ	p
Europarl	Pearson	.040	.862	.107	.643
	Spearman	.005	.982	.008	.973
Bibles	Pearson	**.742**	**8e-12**	.034	.792
	Spearman	**.751**	**3e-12**	-.025	.851

Table 4: Correlations and significances when regressing on the size of the raw word inventory.

[32] afr, aln, arb, arz, ayr, bba, ben, bqc, bul, cac, cak, ceb, ces, cmn, cnh, cym, dan, deu, ell, eng, epo, fin, fra, guj, gur, hat, hrv, hun, ind, ita, kek, kjb, lat, lit, mah, mam, mri, mya, nld, nor, plt, poh, por, qub, quh, quy, quz, ron, rus, som, tbz, tcw, tgl, tlh, tpi, tpm, ukr, vie, wal, wbm, xho, zom

[33] 22 Indo-European, 6 Niger-Congo, 6 Mayan, 6 Austronesian, 4 Sino-Tibetan, 4 Quechuan, 4 Afro-Asiatic, 2 Uralic, 2 Creoles, 2 Constructed languages, 2 Austro-Asiatic, 1 Totonacan, 1 Aymaran; we are reporting the first category on Ethnologue (Paul et al., 2009) for all languages, manually fixing tlh ↦ Constructed language.