PrOnto: Language Model Evaluations for 859 Languages

Luke Gessler
Department of Linguistics
Georgetown University
{lg876@georgetown.edu

Abstract

Evaluation datasets are critical resources for measuring the quality of pretrained language models. However, due to the high cost of dataset annotation, these resources are scarce for most languages other than English, making it difficult to assess the quality of language models. In this work, we present a new method for evaluation dataset construction which enables any language with a New Testament translation to receive a suite of evaluation datasets suitable for pretrained language model evaluation. The method critically involves aligning verses with those in the New Testament portion of English OntoNotes, and then projecting annotations from English to the target language, with no manual annotation required. We apply this method to 1051 New Testament translations in 859 and make them publicly available. Additionally, we present experiments which demonstrate the soundness and the utility of our method in creating evaluation tasks which are effective in assessing language model quality.

1 Introduction

Language models such as BERT (Devlin et al., 2019) and other Transformer-based (Vaswani et al., 2017) language models (TLMs) are notoriously difficult to understand. Evaluation datasets such as SuperGLUE (Wang et al., 2019), BLiMP (Warstadt et al., 2020), and others have been essential resources for understanding and comparing different models' capabilities. By measuring two models' performance on a question-answering task, for example, we are able to make an assessment about the models' capabilities relative to each other. Unfortunately, these evaluation tasks almost always require *annotated* data produced by a human being, and these datasets are therefore very scarce except for the most well-resourced languages, especially English. This scarcity of evaluation datasets has been a significant hindrance for research on TLMs for low-resource languages, as it is much harder to assess the quality and properties of models without them.

Here, we present PrOnto, a dataset consisting of **pr**ojections of **Onto**Notes' New Testament annotations into New Testament translations in 859 different languages. OntoNotes (Hovy et al., 2006) is a corpus with many annotation types covering a wide variety of phenomena in grammar and meaning. A subset of the English portion of OntoNotes contains the Easy-to-Read Version (ERV) translation of the New Testament, complete with a segmentation of each sentence into the book, chapter, and verse of the Bible that it appeared in. Using these verse alignments, we can create new annotations for a given target language, yielding high-quality annotated data for the target language, ready to use in an evaluation, without requiring more human annotation. We focus on annotations which do not require token alignments (e.g., number of referential noun phrases that appear in a verse), as this ensures that annotation quality will remain high.

In this work, we describe our methods for creating the PrOnto dataset, and also provide experimental results demonstrating its utility as an evaluation resource. We summarize our contributions as follows:

- We publish evaluation datasets for 5 tasks across 1051 New Testament translations in 859 languages.[1]
- We perform experiments covering a wide range of languages with respect to typological variables and data-richness which demonstrate the utility of this dataset for assessing pretrained language model quality.
- We publish the annotation projection system we used to create this dataset, which is usable

[1]These datasets and all of our code are available at https://github.com/lgessler/pronto

with any additional language that has a New Testament translation or a part of one.

2 Related Work

Beginning with the publication of the first modern TLM, BERT (Devlin et al., 2019), pretrained TLMs have had their quality assessed by applying them to a wide array of downstream tasks. It is typical to apply the TLM in question to as many downstream evaluations as practically possible, since downstream tasks vary considerably in which properties of language they are sensitive to. A syntactic parsing task, for example, is presumably more discriminative of formal aspects of grammar, while a sentiment analysis task is presumably more discriminative of meaning-related aspects of grammar. All 11 of the tasks used to evaluate BERT are meaning-oriented tasks, with natural language understanding (NLU) and question answering (QA) being heavily represented.

Most post-BERT English TLMs have followed its lead in favoring meaning-related tasks (e.g. Liu et al., 2019; Zhang, 2022, *inter alia*). The English TLM evaluation dataset ecosystem has continued to grow, and some evaluation dataset suites have grown to encompass over 200 tasks (BIG-bench collaboration, 2021). Among other high-resource languages, there is more variation: MacBERT (Cui et al., 2020), a Mandarin Chinese BERT, is evaluated using tasks comparable in kind and quantity to those used with BERT, while CamemBERT (Martin et al., 2020), a French BERT, is evaluated with a large proportion of Universal Dependencies (UD) (Nivre et al., 2016) tasks.

The situation for low-resource languages is quite different. Since annotated datasets are so rare and small for low-resource languages, most low-resource TLM evaluation has been centered on just a few datasets, all of which are fairly form-oriented in terms of what they are assessing models for. Occasionally, a family of low-resource languages might have a high-quality evaluation dataset: for example, Ogueji et al. (2021) train a low-resource TLM for 11 African languages, and evaluate on named-entity recognition (NER) using the MasakhaNER dataset (Adelani et al., 2021). However, more often, low-resource languages do not have resources like this.

Much recent work on low-resource TLMs (Chau et al., 2020; Chau and Smith, 2021; Muller et al., 2021; Gessler and Zeldes, 2022, *inter alia*) uses only two datasets. The first is UD corpora, which consist of human-annotated syntactic trees and tags which can be used for form-related tasks such as part-of-speech tagging and syntactic dependency parsing. The second is the WikiAnn (Pan et al., 2017) dataset, an NER dataset that was automatically generated for 282 languages based on the structure of Wikipedia hyperlinks. While evaluations that use both of these datasets have proven to be useful, the UD dataset and to a lesser extent the WikiAnn dataset are both more form- than meaning-based in terms of what they assess in models. This could mean that many low-resource TLM evaluations are missing important dimensions of model quality that cannot be assessed well by existing evaluation datasets.

```
Plain sentence:
---------------
    Jesus cried.

Treebanked sentence:
--------------------
    Jesus cried .

Speaker information:
--------------------
    name: John
    start time: 11_35_0
    stop time: 11_35_3
```

```
Tree:
-----
    (TOP (S (NP-SBJ (NNP Jesus))
            (VP (VBD cried))
            (. .)))

Leaves:
-------
    0   Jesus
          coref: IDENT        16    0-0    Jesus
    1   cried
          prop:  cry.02
           v           * -> 1:0,  cried
           ARG0        * -> 0:1,  Jesus
    2   .
```

Figure 1: A sample verse, John 11:35, taken from OntoNotes. Note the annotations for tokenization, part-of-speech, constituency syntax, coreference, and argument structure. This file is in "OntoNotes Normal Form" (ONF), a human-readable format which OntoNotes provides its annotations in.

3 OntoNotes

Before we describe our work, we briefly describe some important details of OntoNotes (Hovy et al., 2006). OntoNotes is a multilayer annotated corpus whose English portion contains the Easy-to-Read Version (ERV) translation of the New Testament of the Christian Bible. OntoNotes' major annotation types include coreference, Penn Treebank–style constituency syntax, NER, WordNet sense annotations, and PropBank argument structure annotations. The ERV New Testament subcorpus of OntoNotes has all of these major annotation types with the notable exception of NER and WordNet sense annotation (except within PropBank annotations), which was not done for the New Testament.

An example annotation of John 11:35 is given in Figure 1. The "Tree" annotation has a Penn Treebank–style parse which includes an analysis of the sentence's syntactic structure as well as part-

of-speech tags. The "Leaves" section contains multiple annotation types which are anchored on the annotation's head token. The `coref` type indicates a coreference annotation, which is then followed by coreference type, coreference chain ID, and token span information. The annotation in Figure 1 tells us that: token 0, *Jesus*, is the beginning of a new coreference mention; the coreference type of this mention is IDENT; the mention belongs to coreference chain 16; and this mention begins at token 0 and ends at token 0.

The `prop` type indicates the a PropBank annotation headed at the exponent of a predicate, typically a verb, and gives the WordNet sense of the predicate as well as the arguments of the predicate. In the example in Figure 1, the annotation tells us that: `cried` is the head of a PropBank predicate; the sense of the predicate is `cry.02`; the beginning of the `v` argument is headed at token 1, and its corresponding constituent is 0 levels up in the parse tree; and the beginning of the `ARG0` argument is headed at token 0 and its corresponding constituent is 1 level up in the parse tree.

For full details, we refer readers to the official documentation at https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf.

4 Methods

We would like to have more evaluation datasets for low-resource TLM evaluation, though constructing these for each individual language is expensive, as the creation of new datasets generally requires human annotation of some kind. However, in this work, we propose a method for creating evaluation datasets without requiring additional human annotation. New Testament translations are also highly common for low-resource languages because of missionary work, and OntoNotes' New Testament subcorpus is richly annotated. Because the New Testament is partitioned into verses that are highly consistent across translations, it is possible to view verse boundaries as sentence-like alignments across translations, which would allow the projection of sentence-level annotations from OntoNotes to another New Testament translation.

This is the approach we take up: we propose five annotation projection methods, apply them to Bible translations, and perform evaluations to assess their utility. More specifically, our goal is to take a New Testament translation in a *target language*, align its verses with the verses present in OntoNotes, and then use OntoNotes' annotations to annotate the target language's translation, verse by verse. Here, we describe the steps we take to process the data.

[5-6] The Son of Man has power on earth to forgive sins. But how can I prove this to you? Maybe you are thinking it was easy for me to say, 'Your sins are forgiven.' There's no proof that it really happened. But what if I say to the man, 'Stand up and walk'? Then you will be able to see that I really have this power." So Jesus said to the paralyzed man, "Stand up. Take your mat and go home."

[5] For which is easier: to say, 'Your sins are forgiven,' or to say, 'Stand up and walk'? [6] But so that you may know that the Son of Man has authority on earth to forgive sins"—he then said to the paralytic—"Stand up, take your bed, and go to your home."

Figure 2: Matthew 9:5-6, as translated by the ERV (above) and the NRSVUE (below). In the ERV translation, verses 5 and 6 are fused, which means that no boundary between the two is indicated, and that their contents have been altered in linear ordering.

4.1 Bible Translations

We use all permissively-licensed New Testament translations available at ebible.org, a repository of Bible data, processing the proprietary XML format of these translations into our simple TSV format. Some translations are very small or do not contain any of the New Testament, and we discard any with fewer than 500 verses overlapping with OntoNotes, which we do not count in our totals. The final 1051 translations cover a total of 859 languages.

4.2 Alignment

We parse OntoNotes' ONF files, and we assume that the target translation is given in a simple TSV format where each row contains the textual content of the verse as well as the verse's book, chapter, and verse number. In an ideal situation, an OntoNotes sentence would correspond to exactly one verse in both the ERV and the target translation, but this is not always the case. These are the possible complications:

1. A verse contains more than one OntoNotes sentence. Some verses simply contain more than one sentence.

2. An OntoNotes sentence spans more than one ERV verse. Verse boundaries are not guaranteed to coincide with sentence boundaries, so sometimes a sentence will begin in one verse and end in another. In OntoNotes, a sentence never spans more than two verses.

3. The verse in either the ERV or the target translation has been combined with one or

more other verses. Bible translators sometimes choose to combine verses and in such cases do not provide internal boundaries for the verses that have been merged.

For determining a mapping, (1) presents no problem—we simply associate multiple OntoNotes sentences with a single verse. For (2), we associate the sentence with both verses, retaining the information that a sentence spanned a verse boundary. (For all of the tasks described in this paper, we discard verses that have sentences that cross verse boundaries, but the alignments are still constructed and ready to use.) For (3), if verses have been combined in either the ERV or the target translation, we simply remove the combined verses from consideration. In the ERV, combined verses are very rare, accounting for well under 1% of all verses. In other translations, this figure is also quite small.

4.3 Tasks

Once alignment is complete, we are prepared to generate task data. We propose five tasks, all of which are sequence classification tasks either on single sequences or on paired (à la BERT's next sentence prediction) sequences. While we do not pursue this in our present work, we expect that it may also be possible to produce annotations for token-level tasks using high-quality automatically generated word alignments.

A fundamental assumption for our approach is that some linguistic properties a sentence might have ought to be *similar enough* in all languages to yield projected annotations which are useful for model evaluation. Of course, short of examining every last verse, we cannot know with certainty that just because, for example, an English sentence has declarative sentence mood, its Farsi translation would also have declarative sentence mood. But we do have reason to believe that sentence mood ought to be fairly well preserved across translations, given that sentence mood is so highly associated with semantic-pragmatic rather than formal aspects of language (Portner, 2018), and so we can have some justification in assuming that sentence mood ought to be the same between translation pairs. At any rate, regardless of the justifiability of this assumption, we contend that if this assumption does hold for a certain annotation type, then we should see differential performance across pretrained TLMs, which we will examine in §5.2.

Task 1: Non-pronominal Mention Counting (NMC) Predict the number of non-pronominal *mentions* in a verse. The intuition for this task is that it ought to require a model to understand which spans in a sentence could co-refer, which requires knowledge of both form and meaning. A mention is a span of tokens, often but not always a noun phrase, that has been annotated for coreference, according to the OntoNotes-specific coreference annotation guidelines.[2]

It is important to point out that some entity must be mentioned at least *twice* in a document in order to be annotated: if an entity is only mentioned once, then the mention is not annotated. This makes this task a little pathological, because models will only be getting verse-level (not document-level) context, and this ought to make it impossible to tell in many cases whether a given markable (some tokens that *could* be a mention) genuinely is a mention. This is unfortunate, but this is not necessarily fatal for the utility of this task.

Task 2: Proper Noun in Subject (PNS) Predict whether the subject of the first sentence in the verse contains a proper noun. To determine whether the subject contains a proper noun, we attempt to find a constituent labeled NP-SBJ in the main clause, and if we succeed in finding exactly one, we consider it a positive instance if any of the tokens within it are tagged with "NNP" or "NNPS". Note that this does not necessarily mean that the *head* of the subject is a proper noun: *scholars*/NNS *from*/IN *Burundi*/NNP would count as a positive instance by our criterion, despite the fact that a common noun heads it.

Task 3: Sentence Mood (SM) Predict whether the mood of the main clause of the first sentence is declarative, interrogative, or imperative. In Penn Treebank parse trees, sentence mood is encoded in the label of the highest constituent: for example, `S` and `S-CLF` are defined as having declarative sentence mood, `S-IMP` is imperative, and `SQ`, `SBARQ`, and `SQ-CLF` are interrogative. If the top constituent does not have a label that falls into any of these categories, which likely means it is a sentence fragment or some other unusual sentence type, we discard it.

[2] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-coreference-guidelines.pdf

Task 4: Same Sense (SS) Given two verses v_1 and v_2, and given further that v_1 contains at least one usage of the predicate identified by sense label s, predict whether v_2 also has a usage of sense label s. Note that in our formulation of this task, the sense label s is explicitly given as an input rather than left unexpressed because otherwise the model would need to look for whether *any* sense-usages overlap across the two verses, which is likely too hard. Pairs are sampled so that negative and positive instances are balanced.

This task is perhaps the most suspect of all of our five proposed tasks given the great diversity of distinctions that may or may not be made at the word sense level. For example, for the English word *go*, Bukiyip has at least three different lexical items, distinguished by vertical motion relative to the mover's position at the beginning of the going event: *nato* 'go up, ascend'; *nabəh* 'go down, descend'; and *narih* 'go around, go at a level grade'. As such, we should expect that performance will likely be nowhere close to 90% even on non-English high-resource languages, as the English sense labels will likely often reflect distinctions which are either unexpressed or not specific enough for the target language's sense-inventory. Still, we expect that for any given language, *some* sense labels will still be appropriate when projected, and if this is the case, then we expect that higher-quality models will be able to perform better than lower-quality ones.

Task 5: Same Argument Count (SAC) Given two verses v_1 and v_2 which both feature a usage of the predicate identified by sense label s, predict whether both usages of s have the same number of arguments. Pairs are sampled so that negative and positive instances are balanced. We do not require that the verses have *exactly* one usage of s, which we do in the interest of using as many distinct verses as possible, though this may be interesting to consider in future work.

5 Evaluation

In order to evaluate our dataset, we implement a simple sequence classification model and apply it to our tasks using a wide range of pretrained TLMs. We evaluate a wide range of languages and models in order to get as much information as possible about the utility of our methods. These include several low-resource languages, but we also include some high- and medium-resource languages in order to get additional perspective.

5.1 Languages

The only work we were able to locate in the literature on low-resource TLMs that both worked on a wide range of languages and made all of their pretrained TLMs publicly available is Gessler and Zeldes (2022), and we therefore include all of the languages they studied in their work. These include the the low-resource languages Wolof, Sahidic Coptic, Uyghur, and Ancient Greek. (Gessler and Zeldes also published models for Maltese, but we were unable to locate a permissively-licensed Maltese Bible.) These also include Tamil and Indonesian, two medium-resource languages.

We additionally consider the high-resource languages French and Japanese, which may be interesting to look at given that they are both high resource and typologically similar to and divergent from English, respectively. Any differences that emerge between French and Japanese could be indicative of typological distance degrading the quality of our projected annotations. Additionally, both of these languages have high-quality monolingual TLMs, and it would be interesting to examine if different patterns emerge in high-resource settings.

Finally, we include two different English translations. First, we include the original translation used in OntoNotes, the ERV, because it ought to give us an upper bound on projected annotation quality: ERV annotations projected to the same ERV verses ought to have the highest possible quality. Second, we include the Noah Webster's revision of the King James Version. The Webster Bible differs from the KJV only in that mechanical edits were made to replace archaic words and constructions, and we include it in order to see if relatively small differences across translations (same language, slightly different register) are enough to cause major differences in task performance, which would then indicate differences in projected annotation quality.

Bible Manifest

5.2 Model Implementation

We use HuggingFace's (Wolf et al., 2020) off-the-shelf AutoModelForSequenceClassification model. This model takes a pretrained TLM and adds a sequence classification head (with pretrained weights, if available). The architectural details of this head vary depending on which exact model a pre-

trained TLM is for (e.g. BertModel or RobertaModel), but most major models, including BERT and RoBERTa, simply use one (BERT) or two (RoBERTa) linear transformations that are applied to the [CLS] (or equivalent) token. The model is trained with a low learning rate for a small number of epochs before it is evaluated on a held-out test set for each task.

Hyperparamters Specifically, we use the default parameters for the `transformers` package, version 4.28.1, for the Trainer class, with the following exceptions. Learning rate is set to 2e-5, batch size is set to 16, training epochs is set to 10 except for SM in which case it is 20, and weight decay for AdamW is set to 0.01.

NMC Capping For NMC, while we always provide the genuine number of non-pronominal mentions in our dataset, in our experiments, we cap the maximum number of mentions at 3, labeling any sentence with more than 3 mentions as if it only had 3. This was done to make the task easier, as the number of sentences with more than 3 mentions is very low, and the model subsequently suffers while trying to learn how to count higher than three.

Sequence Packing for SS and SAC Recall that for the SS and SAC tasks, the inputs include not only two verses but also a sense label. First, we pack the two verses into a single input sequence, obeying any model-specific rules about where to put special tokens. In a BERT style model, for example, the sequence would look like [CLS] v_1 [SEP] v_2 [SEP]. There are many ways the sense label s could be provided as an input, but we choose to provide the label as an extra token after the final token of the base sequence. To do this, we extend the vocabulary $\mathcal{V}$ with $|\mathcal{S}|$ more entries, where $\mathcal{S}$ is the inventory of sense labels, so that the new vocabulary has size $|\mathcal{V}| + |\mathcal{S}|$. Senses are individually assigned to the new entries, and each sense is put after the final token, e.g. [CLS] v_1 [SEP] v_2 [SEP] s.

Metrics We report accuracy on all tasks. Other more specialized metrics might be more informative for some tasks where e.g. the task is a binary classification problem or the label distribution is highly imbalanced, but we find that accuracy alone is sufficient to support our findings here, and choose to work with it exclusively to simplify the discussion.

5.3 List of Bibles

Our complete list of Bibles for the evaluation is as follows. We format them so that our own abbreviation for them comes first, the full title follows, and the code for ebible.org's page follows in parentheses (append this code to `ebible.org/details.php?id=`).

1. ERV: Easy-to-Read version (engerv)
2. WBT: Webster Bible (engwebster)
3. IND: Indonesian New Testament (ind)
4. TAM: Tamil Indian Revised Bible (tam2017)
5. FRA: French Free Holy Bible for the World (frasbl)
6. JPN: New Japanese New Testament (jpn1965)
7. GRC: Greek Majority Text New Testament (grcmt)
8. COP: Coptic Sahidic New Testament (copshc)
9. UIG: Uyghur Bible (uigara)
10. WOL: Wolof Bible 2020 Revision (wolKYG)

5.4 List of Pretrained Models

Our complete list of pretrained models from HuggingFace Hub for the evaluation is as follows. Note that some abbreviations are repeated because language will disambiguate which one is meant. The models beginning with `lgessler/microbert` are taken from Gessler and Zeldes (2022), and the suffixes indicate whether pretraining took place with just MLM (`-m`) or the combination of MLM and part-of-speech tagging (`-mx`). (We refer readers to their paper for further details.)

1. `bert-base-multilingual-cased`: mBERT
2. `xlm-roberta-base`: XLM-R
3. `bert-base-cased`: BERT
4. `distilbert-base-cased`: DistilBERT
5. `roberta-base`: RoBERTa
6. `camembert-base`: BERT
7. `cl-tohoku/bert-base-japanese`: BERT
8. `l3cube-pune/tamil-bert`: BERT
9. `cahya/bert-base-indonesian-522M`: BERT
10. `lgessler/microbert-...-m`: μBERT-M (where `...` is one of `wolof`, `ancient-greek`, `indonesian`, `coptic`, `uyghur`, `tamil`
11. `lgessler/microbert-...-mx`: μBERT-MX

Model	NMC	PNS	SM	SS	SAC
ERV	49.59	72.60	91.56	50.43	50.69
DistilBERT	71.93	99.07	99.72	94.48	61.34
BERT	71.12	99.23	100.00	97.75	63.25
RoBERTa	70.03	98.76	99.86	89.53	50.69
mBERT	67.30	99.23	99.86	96.11	61.04
XLM-R	69.35	99.07	100.00	49.57	50.69
WBT	49.73	70.43	90.87	50.53	50.78
DistilBERT	55.99	84.67	92.81	72.15	61.54
BERT	52.86	83.13	94.88	76.06	64.51
RoBERTa	55.31	82.04	91.15	57.11	50.78
mBERT	53.54	85.29	93.22	79.08	60.55
XLM-R	53.68	84.67	93.22	49.47	50.78

Table 1: Task accuracy for English by model and translation. ERV is the Easy-to-Read Version, WBT is the Webster Bible.

5.5 Results

English Results for our two English datasets are given in Table 1. A majority-label baseline is given in the row labeled with the translation (ERV or WBT), and results with several common pretrained English models as well as two multilingual models are given.

Looking first at our "control" dataset, the projection from the ERV translation onto itself, we can see that overall our models perform well above the majority class baseline, indicating that all of our tasks are not intractable, at least in the most easy setting. It's worth noting that the Sentence Mood task is very easy in this condition, with two models getting a perfect score. The hardest task is Same Argument Count, with the best model performing only 13% higher than the baseline. A striking pattern with the sequence-pair tasks is that the RoBERTa-family models perform at chance in three out of four cases. The only obvious reason why this might be is that the other, BERT-family models are pretrained with a sequence-pair task (next sentence prediction), while RoBERTa does not. We set this matter aside for now and note that even very popular and generally high-quality models can have anomalous performance on some tasks.

Turning now to the other English translation, WBT, we see that performance is lower on the whole but remains discernably higher than the baseline in all cases. It is worth noting that the variety of English used in WBT, a slightly modernized form of Early Modern English, is likely quite out of domain for all of our models, and in this sense, the WBT could be thought of as a few-shot setting. A pattern similar to the one for the ERV emerges where the RoBERTa-family models fail to do anything meaningful for the Same Argument Count task.

Model	NMC	PNS	SM	SS	SAC
FRA	49.86	76.78	89.76	50.40	51.14
BERT	57.63	82.35	92.81	67.43	64.04
mBERT	56.27	84.83	92.67	77.43	64.88
XLM-R	57.49	84.21	92.95	49.60	51.14
JPN	51.30	76.47	91.41	50.15	50.64
BERT	58.25	89.63	94.04	73.52	62.46
mBERT	59.21	88.24	93.21	79.74	51.36
XLM-R	54.98	88.85	95.15	49.85	50.64
IND	49.15	72.95	92.36	50.37	50.87
BERT	54.40	87.92	92.80	69.25	62.79
μBERT-M	54.12	88.24	94.09	61.46	62.28
μBERT-MX	53.98	87.12	93.80	59.87	62.10
mBERT	51.28	87.44	94.52	72.08	64.35
XLM-R	55.40	86.63	92.36	50.37	49.13
TAM	49.59	74.77	91.56	50.51	50.65
BERT	54.90	86.84	92.53	49.49	50.65
μBERT-M	53.13	81.27	91.70	62.33	62.34
μBERT-MX	52.32	82.51	91.29	62.92	63.11
mBERT	55.45	85.29	92.39	70.32	64.24
XLM-R	55.86	85.14	91.56	50.51	50.65

Table 2: Task accuracy for "medium-resource" languages by language and translation.

Overall, the results are in line with what we would expect given other published results which have evaluated the quality of these five pretrained models. The monolingual models almost always do best for ERV and in three out of five tasks for WBT (SS and PNS, where mBERT does best). Among the monolingual models, excepting the anomalous RoBERTa cases described above, BERT most often performs best, with DistilBERT doing best in only two cases, which accords with findings that DistilBERT's quality is usually slightly lower than BERT's (Sanh et al., 2020). In sum, these results on English corroborate our claim that our five tasks are well-posed, not pathologically difficult, and indicative of model quality, at least in English settings.

Medium-resource Languages We turn now to our "medium-resource" languages in Table 2: French and Japanese at the higher end, and Indonesian and Tamil at the lower end. For all four languages, XLM-RoBERTa continues to struggle with sequence-pair classification tasks, performing essentially at chance for all languages.

For French and Japanese, the monolingual BERT model's performance is typically a bit better than either of the multilingual models' performance, with one exception: for the same-sense (SS) task, mBERT performs significantly better than the monolingual model. Thus the broad picture of

Model	NMC	PNS	SM	SS	SAC
GRC	50.41	76.32	90.73	50.40	50.87
µBERT-M	52.59	81.11	90.18	60.58	61.80
µBERT-MX	56.81	81.42	91.56	60.95	61.71
mBERT	57.36	83.13	91.70	65.34	50.87
XLM-R	55.99	76.32	91.42	49.60	50.87
COP	48.98	75.50	89.75	50.35	51.24
µBERT-M	50.75	78.76	89.75	61.32	62.70
µBERT-MX	53.34	80.78	91.55	61.30	61.58
mBERT	49.52	75.50	89.75	52.79	51.24
XLM-R	48.84	75.50	89.75	50.35	51.24
UIG	49.37	73.53	89.96	50.23	50.78
µBERT-M	49.37	81.30	89.96	60.65	61.78
µBERT-MX	51.19	78.45	90.10	61.51	62.12
mBERT	51.46	80.35	91.23	62.73	50.78
XLM-R	54.53	84.94	92.93	49.77	50.78
WOL	51.47	77.72	90.36	50.44	50.45
µBERT-M	51.47	77.72	90.36	59.78	61.05
µBERT-MX	59.24	79.90	90.36	63.08	63.46
mBERT	57.35	84.75	91.65	66.49	54.46
XLM-R	56.51	82.32	91.01	50.44	49.55

Table 3: Task accuracy for low-resource languages by language and translation.

performance is what we'd expect, though this one surprising result shows that our tasks are broad in what they assess models for.

For Indonesian and Tamil, the µBERT models perform slightly worse on average than mBERT, in line with the results reported by Gessler and Zeldes (2022). Compared to the full-size monolingual models, the µBERT models also are slightly worse on average, save for SS and SAC for Tamil, where performance is at-chance for the monolingual BERT.

Low-resource Languages Results for low-resource languages are given in Table 3. Something that distinguishes the low-resource languages from the medium-resource languages and English is that many models now perform no better than the majority baseline. Many of the Wolof and Coptic models perform no better than the baseline, and fewer but still some of the Uyghur and Ancient Greek models do not outperform the baseline. For the µBERT models, we note that the frequency with which this happens seems connected to dataset size: the tokens used by the µBERT developers for each language were approximately 500K for Wolof, 1M for Coptic, 2M for Uyghur, and 9M for Ancient Greek. This demonstrates that some of our tasks are too hard to be solved at all by a model if it falls below a quality threshold, which can be seen as a desirable trait.

Differences between the best-performing model and the baseline can be very small in some cases, such as for Sentence Mood in most languages. This may indicate that sentence mood annotation projection is inappropriate for some target languages, though the fact that models still do differentiate themselves in how able they are to do it demonstrates that some properties of the target language can at least be correlated with the sentence mood of a translation-equivalent English sentence. The performance gain relative to the baseline remains quite high for the two sense-related

6 Conclusion

We have presented PrOnto, a publicly available dataset of evaluation tasks for pretrained language models for 1051 New Testament translations in 859. Overall, our results show that our tasks remain meaningful even when projected to languages which are typologically very different from English, and also even when they are performed by models that were trained on very little data. The fact that the way pretrained models distribute relative to our tasks mostly in the same way they do for established evaluation tasks constitutes evidence that these tasks are indeed indicative of model quality. Moreover, while our intent was primarily to develop this resource for low-resource languages, we have shown that it is able to serve medium- and high-resource languages as well.

In future work, we intend to continue developing additional tasks. There is still much data that has not been fully used in the OntoNotes annotations, and some tasks (such as SAC) would likely benefit from refinement or reformulation. We further invite interested readers to consider contributing a task, as our annotation projection pipeline has been structured to make tasks very easy to author.

References

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiu Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa,

Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021. MasakhaNER: Named entity recognition for African languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

BIG-bench collaboration. 2021. Beyond the imitation game: Measuring and extrapolating the capabilities of language models. *In preparation*.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, Online. Association for Computational Linguistics.

Ethan C. Chau and Noah A. Smith. 2021. Specializing multilingual language models: An empirical study. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 51–61, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Luke Gessler and Amir Zeldes. 2022. MicroBERT: Effective training of low-resource monolingual BERTs through parameter reduction and multitask learning. In *Proceedings of the The 2nd Workshop on Multilingual Representation Learning (MRL)*, pages 86–99, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57–60, New York, New York. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Paul Portner. 2018. *Mood*. Oxford Surveys in Semantics and Pragmatics. Oxford University Press, Oxford, New York.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]*. ArXiv: 1910.01108.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *Advances in Neural Information Processing Systems*, 30.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*. ArXiv: 1910.03771.

Bryan Zhang. 2022. Improve MT for search with selected translation memory using search signals. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 123–131, Orlando, USA. Association for Machine Translation in the Americas.

A Acknowledgments

We thank Amir Zeldes for very helpful comments on this work.