Evaluation and Adaptation of Language Models for Under-Resourced Languages

Wietse de Vries
Overview

- Part 1: Do language models actually model Dutch during pre-training?
- Part 2: How well do language models perform on various Dutch tasks?
- Part 3: Can we adapt English models to Dutch and Italian with little training?
- Part 4: Can we adapt models to low-resource languages without labeled data?
- Part 5: How does cross-lingual training work with any source and target language?
Probing BERT’s layers for a Dutch NLP pipeline

Introduction

- Diagnostic probing has revealed a pipeline-like behaviour for English BERT (Tenney et al. 2019)
  - Simple models trained on hidden transformer layer representations
- E.g. low level tasks like POS tagging can be found in early layers and higher-level tasks like coreference resolution at later layers
- Is this pipeline actually this neat?
- Can this behaviour be found for other languages such as Dutch?
Methodology

- **Simple probes**: token label prediction with a linear model using hidden layer representations

- **Scalar mixing probes**: use a weighted sum of all hidden layers and evaluate the learned layer weights

- **Models**: BERTje (Dutch) and mBERT

- **POSS tagging (POS)**
  - Lassy Small corpus
  - Alpino corpus

- **Dependency edge labeling (DEP)**
  - Lassy Small corpus
  - Alpino corpus

- **Named Entity Recognition (NER)**
  - CoNLL-2002

- **Coreference resolution (Coref)**
  - SoNaR-1
Scalar mixing results

- Scalar mixing probes show higher accuracies than single-layer probes
- mBERT most informative layers are more central
- Word embeddings are more informative for BERTje
- Final layer is relatively uninformative
Label differences within one task: POS tagging (BERTje; single layers)
Conclusions

- BERTje and mBERT show a similar pipeline structure for Dutch as BERT for English but task differences are not very strong.
- The most informative mBERT layers are earlier layers than those of BERTje.
- Task information is spread out over multiple layers:
  - Rule of thumb: the word embeddings and the layers at 2/3 of the model may be most informative.
- BERTje shows consistent results across datasets.
- More general: task-specific information is learned during pre-training.
DUMB: A Dutch Model Benchmark

Introduction

- There are multiple Dutch and multilingual pre-trained language models
  - Unclear which model is most useful for which task
  - New models tend to be re-trained models of the same type: RobBERT-v1, RobBERT-v2, RobBERT-2022, RobBERT-2023...

- English (and other monolingual) benchmarks such as GLUE are not perfect:
  - Task duplication (e.g. 4/10 tasks in GLUE are just Natural Language Inference)
  - Averaging absolute scores undervalue improvement of already high scores

- Our benchmark:
  - 9 tasks of which 4 not previously available in Dutch
  - A different task scoring method: Relative Error Reduction
Tasks

- **Word tasks:**
  - Part-Of-Speech tagging (POS): New standardized train/dev/test splits with Lassy Small corpus
  - Named Entity Recognition (NER): New standardized train/dev/test splits with SoNaR-1 corpus

- **Word pair tasks:**
  - Word Sense Disambiguation (WSD): New Words in Context (WiC) task based on DutchSemCor
  - Pronoun Resolution (PR): New task data based on coreference annotations in SemEval 2010 Task 1

- **Sentence pair tasks:**
  - Causal Reasoning: Choice of Plausible Alternatives (COPA) translated from English to Dutch
  - Natural Language Inference (NLI): Existing SICK-NL dataset (translated SICK from English)

- **Document tasks:**
  - Sentiment Analysis (SA): Existing Dutch Book Reviews Dataset (DBRD)
  - Abusive Language Detection (ALD): Existing Dutch Abusive Language Corpus (DALC)
  - Question Answering (QA): Translated SQuAD (v2) from English to Dutch
Evaluation metric: Relative Error Reduction

- Problem with normal averaging:
  - Absolute score differences are weighted equally for every task
  - An accuracy improvement from 50% to 55% has the same effect on the average as 90% to 95%
  - My assumption: a small absolute improvement on a high score can be very meaningful

- Solution: Evaluate on Relative Error Reduction
  - E.g. 50% to 55% is only a 10% error reduction while 90% to 95% is a 50% error reduction

- In our benchmark, we use the BERTje model as a baseline for all other models
Models

- Only transformer encoder models
- Three model types:
  - BERT (MLM + Sentence pair task)
  - RoBERTa (MLM)
  - DeBERTaV3 (ELECTRA-style generator-discriminator)
- Two model sizes:
  - Base: 12 layers (768 dimensions)
  - Large: 24 layers (1024 dimensions)
- Three pre-training language groups:
  - Dutch
  - Multilingual (including Dutch)
  - English
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg</th>
<th>POS</th>
<th>NER</th>
<th>WSD</th>
<th>PR</th>
<th>CR</th>
<th>NLI</th>
<th>SA</th>
<th>ALD</th>
<th>QA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERTje</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RobBERTv1</td>
<td>-16.3</td>
<td>12.5</td>
<td>-19.4</td>
<td>-15.3</td>
<td>-24.0</td>
<td>-14.7</td>
<td>-12.7</td>
<td>-58.2</td>
<td>4.8</td>
<td>-19.4</td>
</tr>
<tr>
<td>RobBERTv2</td>
<td>1.6</td>
<td>16.2</td>
<td>4.1</td>
<td>-5.3</td>
<td>0.1</td>
<td>-10.2</td>
<td>-3.8</td>
<td>-0.5</td>
<td>12.0</td>
<td>2.2</td>
</tr>
<tr>
<td>RobBERT2022</td>
<td>3.6</td>
<td>17.3</td>
<td>7.6</td>
<td>-6.4</td>
<td>-1.8</td>
<td>-10.1</td>
<td>3.1</td>
<td>4.0</td>
<td>18.9</td>
<td>-0.2</td>
</tr>
<tr>
<td>mBERTcased</td>
<td>-5.8</td>
<td>6.2</td>
<td>9.2</td>
<td>7.7</td>
<td>-11.0</td>
<td>-18.4</td>
<td>-6.2</td>
<td>-41.7</td>
<td>-4.5</td>
<td>6.9</td>
</tr>
<tr>
<td>XLM-R_base</td>
<td>-0.3</td>
<td>13.9</td>
<td>10.8</td>
<td>1.9</td>
<td>-16.2</td>
<td>-26.8</td>
<td>2.0</td>
<td>-3.6</td>
<td>3.4</td>
<td>12.3</td>
</tr>
<tr>
<td>mDeBERTaV3_base</td>
<td>12.8</td>
<td>18.2</td>
<td>17.2</td>
<td>10.8</td>
<td>-20.8</td>
<td>19.7</td>
<td>25.2</td>
<td>3.3</td>
<td>12.4</td>
<td>29.2</td>
</tr>
<tr>
<td>XLM-R_large</td>
<td>14.4</td>
<td>26.5</td>
<td>29.7</td>
<td>21.3</td>
<td>-15.8</td>
<td>-25.8</td>
<td>24.4</td>
<td>13.2</td>
<td>19.0</td>
<td>37.2</td>
</tr>
<tr>
<td>BERT_base</td>
<td>-42.8</td>
<td>-19.8</td>
<td>-30.8</td>
<td>-22.4</td>
<td>-18.7</td>
<td>-28.0</td>
<td>-19.2</td>
<td>-203.9</td>
<td>-16.1</td>
<td>-26.2</td>
</tr>
<tr>
<td>RoBERTa_base</td>
<td>-25.6</td>
<td>-6.5</td>
<td>-27.3</td>
<td>-14.0</td>
<td>-20.4</td>
<td>-24.1</td>
<td>-19.7</td>
<td>-99.9</td>
<td>-16.0</td>
<td>-2.1</td>
</tr>
<tr>
<td>DeBERTaV3_base</td>
<td>-1.6</td>
<td>6.5</td>
<td>1.7</td>
<td>-4.2</td>
<td>-25.3</td>
<td>-20.5</td>
<td>8.6</td>
<td>-14.6</td>
<td>3.5</td>
<td>29.7</td>
</tr>
<tr>
<td>BERT_large</td>
<td>-35.1</td>
<td>-12.0</td>
<td>-25.9</td>
<td>-25.4</td>
<td>-29.3</td>
<td>-31.2</td>
<td>-15.4</td>
<td>-158.7</td>
<td>-7.8</td>
<td>-10.4</td>
</tr>
<tr>
<td>RoBERTa_large</td>
<td>-14.1</td>
<td>6.4</td>
<td>-12.3</td>
<td>-19.8</td>
<td>-23.3</td>
<td>-26.1</td>
<td>-8.5</td>
<td>-63.8</td>
<td>1.2</td>
<td>19.7</td>
</tr>
<tr>
<td>DeBERTaV3_large</td>
<td>15.7</td>
<td>17.9</td>
<td>10.9</td>
<td>12.7</td>
<td>-14.4</td>
<td>35.4</td>
<td>24.1</td>
<td>-6.4</td>
<td>12.5</td>
<td>48.4</td>
</tr>
</tbody>
</table>
## Correlations between tasks

<table>
<thead>
<tr>
<th></th>
<th>POS</th>
<th>NER</th>
<th>WSD</th>
<th>PR</th>
<th>CR</th>
<th>NLI</th>
<th>SA</th>
<th>ALD</th>
<th>QA</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS</td>
<td>-</td>
<td>0.85</td>
<td>0.75</td>
<td>0.31</td>
<td>0.43</td>
<td>0.77</td>
<td><strong>0.89</strong></td>
<td><strong>0.93</strong></td>
<td>0.66</td>
</tr>
<tr>
<td>NER</td>
<td>0.85</td>
<td>-</td>
<td><strong>0.92</strong></td>
<td>0.41</td>
<td>0.42</td>
<td><strong>0.88</strong></td>
<td>0.87</td>
<td>0.81</td>
<td>0.75</td>
</tr>
<tr>
<td>WSD</td>
<td>0.75</td>
<td><strong>0.92</strong></td>
<td>-</td>
<td>0.35</td>
<td>0.52</td>
<td>0.86</td>
<td>0.77</td>
<td>0.64</td>
<td>0.75</td>
</tr>
<tr>
<td>PR</td>
<td>0.31</td>
<td>0.41</td>
<td>0.35</td>
<td>-</td>
<td>0.29</td>
<td>0.15</td>
<td>0.50</td>
<td>0.38</td>
<td>-0.03</td>
</tr>
<tr>
<td>CR</td>
<td>0.43</td>
<td>0.42</td>
<td>0.52</td>
<td>0.29</td>
<td>-</td>
<td><strong>0.64</strong></td>
<td>0.48</td>
<td>0.47</td>
<td>0.51</td>
</tr>
<tr>
<td>NLI</td>
<td>0.77</td>
<td>0.88</td>
<td>0.86</td>
<td>0.15</td>
<td><strong>0.64</strong></td>
<td>-</td>
<td>0.74</td>
<td>0.79</td>
<td><strong>0.87</strong></td>
</tr>
<tr>
<td>SA</td>
<td>0.89</td>
<td>0.87</td>
<td>0.77</td>
<td>0.50</td>
<td>0.48</td>
<td>0.74</td>
<td>-</td>
<td>0.82</td>
<td>0.66</td>
</tr>
<tr>
<td>ALD</td>
<td><strong>0.93</strong></td>
<td>0.81</td>
<td>0.64</td>
<td>0.38</td>
<td>0.47</td>
<td>0.79</td>
<td>0.82</td>
<td>-</td>
<td>0.59</td>
</tr>
<tr>
<td>QA</td>
<td>0.66</td>
<td>0.75</td>
<td>0.75</td>
<td>-0.03</td>
<td>0.51</td>
<td>0.87</td>
<td>0.66</td>
<td>0.59</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td><strong>0.70</strong></td>
<td>0.74</td>
<td><strong>0.70</strong></td>
<td><strong>0.30</strong></td>
<td>0.47</td>
<td>0.71</td>
<td>0.72</td>
<td>0.68</td>
<td><strong>0.59</strong></td>
</tr>
</tbody>
</table>
Missing models: A lot of room for improvement

- Dutch pre-training is better than multilingual, which is better than English
- Large models perform better than smaller
- DeBERTaV3 models are better than RoBERTa and BERT
- More information and a leaderboard can be found on dumbench.nl
Recycling GPT-2 for Dutch and Italian

Introduction

- English models can be effective for Dutch
- At the time of this research, there was no generative Dutch model
- Can GPT-2 generate Dutch and Italian without training the transformer layers?
- Word embedding / Lexical layer retraining for Dutch and Italian
  - The lexical layer is the layer that maps hidden representations to the byte pair encoding vocabulary
Method

- Unlabeled data from Wikipedia, web scraped data, newspapers and books
- Train GPT-2 (small) with randomly initialized word embeddings and frozen transformer layers
- Result: separate new word embeddings for Dutch and Italian that should be compatible with the English transformer model
Sanity check: word embedding alignment

- Dutch/Italian word embeddings should have similar embeddings as literal translations in English
- This is actually true!

<table>
<thead>
<tr>
<th>English</th>
<th>Italian</th>
<th>Dutch</th>
</tr>
</thead>
<tbody>
<tr>
<td>while</td>
<td>mentre</td>
<td>terwijk</td>
</tr>
<tr>
<td>genes</td>
<td>geni</td>
<td>genen</td>
</tr>
<tr>
<td>clothes</td>
<td>vestiti</td>
<td>kleren</td>
</tr>
<tr>
<td>musicians</td>
<td>composi[...]</td>
<td>artiesten</td>
</tr>
<tr>
<td>permitted</td>
<td>ammessa</td>
<td>toegelaten</td>
</tr>
<tr>
<td>Finally</td>
<td>infine</td>
<td>Eindelijk</td>
</tr>
<tr>
<td>satisfied</td>
<td>soddisfatto</td>
<td>tevreden</td>
</tr>
</tbody>
</table>

**Accuracy:** 85% 89%

*Table 4.1* | Alignment of closest tokens in the lexical embeddings of *smlrl* for Italian and Dutch. Accuracy scores are based on a manual evaluation by the authors of 200 randomly selected aligned tokens.
Scaling to larger models by using alignments

- We have aligned GPT-2 word embeddings for English/Dutch/Italian
- A transformation that converts GPT-2 small to GPT-2 medium embeddings can be applied to the Dutch/Italian embeddings
- Transformation strategies:
  - Linear regression (lstsq; least-squares regression)
  - Orthogonal Procrustes (proc)
  - Weighted K-Nearest Neighbors (knn)

<table>
<thead>
<tr>
<th>Model</th>
<th>Italian</th>
<th>Dutch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Int@1k</td>
<td>PPL</td>
</tr>
<tr>
<td>med_cle (1 epoch)</td>
<td>0.38</td>
<td>-</td>
</tr>
<tr>
<td>sml_cle $\xrightarrow{proc}$ med</td>
<td>0.61</td>
<td>8.12 $\times 10^{12}$</td>
</tr>
<tr>
<td>sml_cle $\xrightarrow{lstsq}$ med</td>
<td>0.56</td>
<td>364.06</td>
</tr>
<tr>
<td>sml_cle $\xrightarrow{1-NN}$ med</td>
<td>0.37</td>
<td>2,764.19</td>
</tr>
<tr>
<td>sml_cle $\xrightarrow{10-NN}$ med</td>
<td>0.37</td>
<td>20,715.80</td>
</tr>
</tbody>
</table>

Table 4.3 | Scores for different transformation methods. Int@1K are the average 1k nearest English neighbors intersection (int) fractions between sml and transformed med embeddings. PPL is the perplexity on the test sets for Italian and Dutch. PPL (1 epoch) indicates the perplexity after one epoch of training, which is low if the transformed embeddings were close to a good local optimum.
Quality: Quite good but with anglicisms

(b) Human judgment scores for Dutch texts.
Conclusion

- GPT-2 can be adapted to Dutch and Italian with only word embedding retraining
- However, extra full model fine-tuning is needed for better performance
- This cheaper adaptation generates the same quality of Italian as an Italian model of the same size trained from scratch (with more data and much longer training)
- We did not find a meaningful difference between Dutch and Italian as target languages
Adapting monolingual models to low-resource languages


*equal contribution
Introduction

- Word embedding retraining can be effective, but can we use that for real low-resource languages?

- Target languages: **Gronings** (Low Saxon) and **Frisian**

- Source languages: **Dutch**, **German** and **English**
  - All languages are Germanic languages, Frisian and Gronings are most similar to Dutch

- Independent word embedding retraining and Transformer layer fine-tuning

- Tested with monolingual BERT models and mBERT
Separate fine-tuning and word embedding retraining

Frozen original transformer layers
Results: original word embeddings

(a) Monolingual POS accuracies for BERT, gBERT and BERTje.

(b) Multilingual POS accuracies for mBERT.
Results

(a) Monolingual POS accuracies for BERT, gBERT and BERTje.

(b) Multilingual POS accuracies for mBERT.
## Results

<table>
<thead>
<tr>
<th>Test language:</th>
<th>Source</th>
<th>Gronings</th>
<th>West Frisian</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>orig.</td>
<td>gro.</td>
<td>fri.</td>
</tr>
<tr>
<td><strong>Train language:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EN</td>
<td>GUM BERT</td>
<td>93.5</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>mBERT</td>
<td>93.5</td>
<td>22.0</td>
</tr>
<tr>
<td></td>
<td>ParTUT BERT</td>
<td>94.0</td>
<td>16.6</td>
</tr>
<tr>
<td></td>
<td>mBERT</td>
<td>94.0</td>
<td>41.3</td>
</tr>
<tr>
<td>DE</td>
<td>GSD gBERT</td>
<td>92.6</td>
<td>23.3</td>
</tr>
<tr>
<td></td>
<td>mBERT</td>
<td>92.2</td>
<td>25.1</td>
</tr>
<tr>
<td></td>
<td>HDT gBERT</td>
<td>94.0</td>
<td>28.5</td>
</tr>
<tr>
<td></td>
<td>mBERT</td>
<td>93.7</td>
<td>26.1</td>
</tr>
<tr>
<td>NL</td>
<td>Alpino BERTje</td>
<td>96.0</td>
<td>90.8</td>
</tr>
<tr>
<td></td>
<td>mBERT</td>
<td>96.2</td>
<td>87.8</td>
</tr>
<tr>
<td></td>
<td>LassySmall BERTje</td>
<td>96.8</td>
<td>89.6</td>
</tr>
<tr>
<td></td>
<td>mBERT</td>
<td>96.8</td>
<td>80.4</td>
</tr>
</tbody>
</table>

**Table 5.2** Accuracy per target language variety (columns) per lexical layer (subcolumns). This table shows that not all datasets are equally effective for transfer to Gronings and West Frisian.
<table>
<thead>
<tr>
<th></th>
<th>Gronings</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>West Frisian</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1MB</td>
<td>5MB</td>
<td>10MB</td>
<td>20MB</td>
<td>40MB</td>
<td>43MB</td>
<td>1MB</td>
<td>5MB</td>
<td>10MB</td>
</tr>
<tr>
<td>BERT EN</td>
<td>GUM ParTUT</td>
<td>29.2 47.8 66.1 67.1 58.9 55.4</td>
<td>48.0 69.5 76.6 79.8 79.4 78.5</td>
<td>37.8 55.1 70.4 72.0 67.8 85.0</td>
<td>53.1 70.4 75.9 78.1 77.8 88.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBERT DE</td>
<td>GUM ParTUT</td>
<td>19.6 73.5 84.8 84.9 84.8 67.7</td>
<td>69.7 87.1 88.0 88.4 88.5 77.0</td>
<td>30.0 76.7 84.0 84.2 84.1 84.3</td>
<td>74.3 88.1 88.4 89.7 89.4 89.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gBERT DE</td>
<td>GSD HDT</td>
<td>48.8 82.3 83.9 84.0 83.8 84.2</td>
<td>77.7 87.3 88.8 88.5 88.7 89.1</td>
<td>30.9 84.5 86.5 87.0 86.3 83.9</td>
<td>73.8 86.3 86.6 87.6 87.1 88.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBERT DE</td>
<td>GSD HDT</td>
<td>24.0 74.0 82.4 82.4 82.7 86.7</td>
<td>71.1 87.1 87.3 88.1 88.1 89.3</td>
<td>03.7 44.2 75.1 72.2 79.5 81.1</td>
<td>34.4 72.0 79.1 81.2 81.2 83.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERTje NL</td>
<td>Alpino LassySmall</td>
<td>73.2 90.3 92.0 91.9 92.0 92.4</td>
<td>43.5 94.2 94.8 95.1 94.9 95.4</td>
<td>67.0 88.3 90.0 90.2 89.9 90.5</td>
<td>44.3 93.6 94.9 94.4 94.6 95.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBERT NL</td>
<td>Alpino LassySmall</td>
<td>31.0 79.6 89.1 88.5 89.3 90.9</td>
<td>74.9 93.7 93.8 94.5 94.7 94.9</td>
<td>15.9 57.4 85.0 85.7 86.7 88.1</td>
<td>67.8 91.6 93.0 93.7 94.1 94.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 5.3** | POS-tagging accuracy for Gronings and West Frisian with subsets of the unlabeled lexical layer retraining data.
Conclusion

- Word embedding retraining is an extremely effective way to adapt task-specific models!
- Only 10mb of data (~1.9 million tokens) is enough to adapt from a very similar language
- Monolingual models outperform mBERT cross-lingually
- How important is language similarity in general?
Cross-lingual training with over 100 languages

Introduction

- Previous 2 papers: adaptation from English or from highly similar source languages
- How does this generalize to other languages and language families?
- Simple setup: Fine-tune XLM-RoBERTa for POS tagging with all languages in Universal Dependencies v2.8
  - 65 languages with (enough) training data
  - 114 languages with test data
- $65 \times 114 = 7410$ test scores (!)
Introduction

- Previous 2 papers: adaptation from English or from highly similar source languages
- How does this generalize to other languages and language families?
- Simple setup: Fine-tune XLM-RoBERTa for POS tagging with all languages in Universal Dependencies v2.8
  - 65 languages with (enough) training data
  - 114 languages with test data
- $65 \times 114 = 7410$ test scores (!)
Analysis

- What are the effects of:
  - Inclusion in pre-training
  - Language similarity
    - automatic LDND measure for lexical similarity
  - Language families
  - Writing systems
  - Word order

- Mixed effects regression analysis
  - Random effects for source and target languages (no interactions)

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Coef.</th>
<th>Std. Err.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>42.2</td>
<td>3.3</td>
</tr>
<tr>
<td>Target pre-trained</td>
<td>19.2</td>
<td>2.5</td>
</tr>
<tr>
<td>LDND distance</td>
<td>-12.7</td>
<td>1.0</td>
</tr>
<tr>
<td>Both pre-trained</td>
<td>7.4</td>
<td>7.4</td>
</tr>
<tr>
<td>Same family</td>
<td>6.8</td>
<td>6.8</td>
</tr>
<tr>
<td>Source pre-trained</td>
<td>5.6</td>
<td>2.0</td>
</tr>
<tr>
<td>Same writing system type</td>
<td>3.6</td>
<td>0.4</td>
</tr>
<tr>
<td>Same writing system</td>
<td>1.4</td>
<td>0.3</td>
</tr>
<tr>
<td>Same SOV word order</td>
<td>1.3</td>
<td>0.2</td>
</tr>
</tbody>
</table>

*Table 6.1* | Coefficients and standard errors of predictors in the final mixed-effects regression model with Accuracy as the dependent variable. All predictors were significant at the $p < 0.01$ level. LDND distances were scaled between 0 (minimum) and 1 (maximum). The predictors are sorted in order of decreasing importance.
Effects of writing systems and language families
Source/Target symmetry

- Estonian and Finnish
- Icelandic and Faroese
- French and Italian
- Chinese and Japanese
- Irish and Scottish Gaelic
- Croatian and Serbian
- Catalan and Spanish
- Belarusian and Ukrainian
- Hindi and Urdu
- Armenian and Western Armenian
- English and Swedish

- From same or neighbouring countries
  - Exceptions: English-Swedish
- Genetically closest siblings (or actually two variants of the same language)
  - Exceptions: English-Swedish, Chinese-Japanese, Catalan-Spanish
What is the best source language?

- Real answer: pick the highest resource language that is closely related to the target language
- Our experiments contain multiple language families and writing systems, but Indo-European languages are still overrepresented. Therefore, aggregates are biased
What is the best source language?

- Anyway: **Romanian** and **Swedish** are the best for most target languages (10 and 7 respectively).
- They also achieve the highest global average accuracy: 67.2% and 65.9%.
- **English** is only the 19th best source language (out of 65).
- English is even just the 5th best Germanic Indo-European language...
Conclusion

- Languages need to be included in pre-training (can be overcome with the strategy of the previous paper)

- Cross-writing system performance is good for alphabetic writing systems but not for logo-syllabic systems

- Any cross-lingual experiment that you will see does not show how good a multilingual model works for a target language, but how good it will transfer from English to that target language
Conclusions
Every language except English is under-resourced

- Dutch is not considered a low-resource language, but we show that other model types and larger sizes would yield much better results than current models.

- Smarter transfer strategies such as word embedding retraining or using adapters work better than just fine-tuning a multilingual model. Especially with monolingual models.

- Cross-lingual performance of multilingual models is highly dependent on the relationship between source and target languages.

- The models that I used are small by today’s standards. How this affects huge generative models is an open question.
Thanks for your attention!

- Please get in touch if you have any questions
  - Only via email: wietse.de.vries@rug.nl