

# Probing Multilingual Sentence Representations With X-PROBE

**Vinit Ravishankar**

Language Technology Group  
Department of Informatics  
University of Oslo  
vinitr@ifi.uio.no

**Lilja Øvrelid**

Language Technology Group  
Department of Informatics  
University of Oslo  
liljao@ifi.uio.no

**Erik Velldal**

Language Technology Group  
Department of Informatics  
University of Oslo  
erikve@ifi.uio.no

## Abstract

This paper extends the task of probing sentence representations for linguistic insight in a multilingual domain. In doing so, we make two contributions: first, we provide datasets for multilingual probing, derived from Wikipedia, in five languages, viz. English, French, German, Spanish and Russian. Second, we evaluate six sentence encoders for each language, each trained by mapping sentence representations to English sentence representations, using sentences in a parallel corpus. We discover that cross-lingually mapped representations are often better at retaining certain linguistic information than representations derived from English encoders trained on natural language inference (NLI) as a downstream task.

## 1 Introduction

In recent years, there has been a considerable amount of research into attempting to represent contexts longer than single words with fixed-length vectors. These representations typically tend to focus on attempting to represent sentences, although phrase- and paragraph-centric mechanisms do exist. These have moved well beyond relatively naïve compositional methods, such as additive and multiplicative methods (Mitchell and Lapata, 2008), one of the earlier papers on the subject. There have been several proposed approaches to learning these representations since, both unsupervised and supervised. Naturally, this has also sparked interest in evaluation methods for sentence representations; the focus of this paper is on *probing*-centric evaluations, and their extension to a multilingual domain.

In Section 2, we provide a literature review of prior work in the numerous domains that our paper builds upon. Section 3 motivates the principle of cross-lingual probing and describes our

goals. In Section 4, we describe our probing tasks and relevant modifications, if any. Section 5 describes our sentence encoders, as well as the procedure we follow for training, mapping and probing. Section 6 describes our data and relevant preprocessing methods we applied. Section 7 presents a detailed evaluation from several perspectives, which we discuss in Section 8. We conclude, as well as describe avenues for future work, in Section 9. Our hyperparameters are described in Appendix A.1, and further detailed results that are not critical to the paper are tabulated in A.2.

## 2 Background

### 2.1 Sentence representation learning

Numerous methods for learning sentence representations exist. Many of these methods are unsupervised, and thus do not have much significant annotation burden. Most of these methods are, however, structured: they rely on the sentences in training data being ordered and not randomly sampled. The aptly named *SkipThoughts* (Kiros et al., 2015) is a well-known earlier work, and uses recurrent encoder-decoder models to ‘decode’ sentences surrounding the encoded sentence, using the final encoder state as the encoded sentence’s representation. Cer et al. (2018) evaluate two different encoders, a deep averaging network and a transformer, on unsupervised data drawn from a variety of web sources. Hill et al. (2016) describe a model based on denoising auto-encoders, and a simplified variant of *SkipThoughts*, that sums up source word embeddings, that they dub (*FastSent*). Another *SkipThoughts* variant (Logeswaran and Lee, 2018) uses a multiple-choice objective for contextual sentences, over the more complicated decoder-based objective.

Several supervised approaches to building representations also exist. An earlier work is Chara-

gram (Wieting et al., 2016), which uses paraphrase data and builds on character representations to arrive at sentence representations. More recent papers use a diverse variety of target tasks to ground representations, such as visual data (Kiela et al., 2017), machine translation data (McCann et al., 2017), and even multiple tasks, in a multi-task learning framework (Subramanian et al., 2018). Relevant to this paper is Conneau et al.’s (2017a) *InferSent*, that uses natural language inference (NLI) data to ground representations: they learn these representations on the well-known SNLI dataset (Bowman et al., 2015).

## 2.2 Multilingual representations

Whilst sentence representation is a thriving research domain, there has been relatively less work on multilingualism in the context of sentence representation learning: most prior work has been focussed on multilingual word representation. For sentence representations, an early work (Schwenk and Douze, 2017) proposes a seq2seq-based objective, using machine learning encoders to map source sequences to fixed-length vectors. Along similar lines, Conneau et al. (2018b) propose using machine translation data to transfer sentence representations pre-trained on NLI, using a mean squared error (MSE) loss - this is the approach we follow.

Artetxe and Schwenk (2018) present a ‘language agnostic’ sentence representation system learnt over machine translation; the agnosticism refers to the joint BPE vocabulary that they construct over all languages, giving their encoders no language information, whilst their decoders are told what language to generate. Similarly, Lample and Conneau (2019) present pre-trained cross-lingual models (XLM), based on modern pretraining mechanisms; specifically, a variant of the masked LM pretraining scheme used in BERT (Devlin et al., 2018).

Contemporaneous with this work, Aldarmaki and Diab (2019) present an evaluation of three cross-lingual sentence transfer methods. Their methods include joint cross-lingual modelling methods that extend monolingual objectives to cross-lingual training, representation transfer learning methods that attempt to ‘optimise’ sentence representations to be similar to parallel representations in another language, and sentence mapping methods based on orthogonal word em-

bedding transfer: the authors use a parallel corpus as a ‘seed dictionary’ to fit a transformation matrix between their source and target languages.

## 2.3 On evaluation

Work on evaluating sentence representations was encouraged by the release of the SentEval toolkit (Conneau and Kiela, 2018), which provided an easy-to-use framework that sentence representations could be ‘plugged’ into, for rapid downstream evaluation on numerous tasks: these include several classification tasks, textual entailment and similarity tasks, a paraphrase detection task, and caption/image retrieval tasks. Conneau et al. (2018a) also created a set of ‘probing tasks’, a variant on the theme of diagnostic classification (Hupkes et al., 2017; Belinkov et al., 2017), that would attempt to quantify precisely what sort of linguistic information was being retained by sentence representations. The authors, whose work focussed on evaluating representations for English, provided Spearman correlations between the performance of a particular representation mechanism on being probed for specific linguistic properties, and the downstream performance on a variety of NLP tasks. Along similar lines, and contemporaneously with this work, Liu et al. (2019) probe three pretrained contextualised word representation models – ELMo (Peters et al., 2018), BERT (Devlin et al., 2018) and the OpenAI transformer (Radford et al., 2018) – with a “suite of sixteen diverse probing tasks”.

On a different note, Saphra and Lopez (2018) present a CCA-based method to compare representation learning dynamics across time and models, without explicitly requiring annotated probing corpora. They motivate the use of SVCCA (Raghu et al., 2017) to quantify precisely what an encoder learns by comparing the representations it generates with representations generated by an architecture trained specifically for a certain task, with the intuition that a higher similarity between the representations generated by the generic encoder and the specialised representations would indicate that the encoder is capable of encapsulating more task-relevant information. Their method has numerous advantages over traditional diagnostic classification, such as the elimination of the classifier, which reduces the risk of an additional component obfuscating results.

A visible limitation of the datasets provided by

these probing tasks is that most of them were created with the idea of evaluating representations built for English language data. In this spirit, what we propose is analogous to [Abdou et al.’s \(2018\)](#) work on generating multilingual evaluation corpora for word representations. Within the realm of evaluating *multilingual* sentence representations, [Conneau et al. \(2018b\)](#) describe the XNLI dataset, a set of translations of the development and test portions of the multi-genre MultiNLI inference dataset ([Williams et al., 2018](#)). This, in a sense, is an extension of a predominantly monolingual task to the multilingual domain; the authors evaluate sentence representations derived by *mapping* non-English representations to an English representation space.

The original XNLI paper provides a baseline representation mapping technique, based on minimising the mean-squared error (MSE) loss between sentence representations across a parallel corpus. Their English language sentence representations are derived from an encoder trained on NLI data ([Conneau et al., 2017a](#)), and their target language representations are randomly initialised for a parallel sentence. While this system does perform reasonably well, a more naive machine-translation based approach performs better.

### 3 Multilingual evaluation

The focus of this paper is twofold. First, we provide five datasets for probing mapped sentence representations, in five languages (including an additional dataset for English), drawn from a different domain to [Conneau et al.’s](#) probing dataset: specifically, from Wikipedia. Second, we probe a selection of mapped sentence representations, in an attempt to answer precisely what linguistic features are retained, and to what extent, post mapping. The emphasis of this evaluation is therefore, crucially, not a probing-oriented analysis of representations *trained* on different languages, but an analysis of the effects of MSE-based mapping procedures on the ability of sentence representations to retain linguistic features. In this sense, our focus is less on the correlation between probing performance and downstream performance, and more on the relative performance of our representations on probing tasks.

Despite having described (in Section 2) numerous methods, both for learning monolingual sentence representations, and for mapping them

cross-linguistically, we restrict our work to a smaller subset of these. Specifically, we evaluate six encoders, each trained in a supervised fashion on NLI data.

Whilst our choice of languages could have been more typologically diverse, we were restricted by three factors:

1. the availability of a parallel corpus with English for our mapping procedure
2. the availability of a large enough Wikipedia to allow us to extract sufficient data (for instance, the Arabic Wikipedia was not large enough to fully extract data for all our tasks)
3. the inclusion of the language in XNLI. Despite not being necessary, we believed it would be interesting to have a ‘real’ downstream task to compare to.

### 4 Probing

We use most of the probing tasks described in [Conneau et al. \(2018a\)](#). Due to the differences in corpus domain, we alter some of their word-frequency parameters. We also exclude the top constituent (**TopConst**) task; we noticed that Wikipedia tended to have far less diversity in sentence structure than the original Toronto Books corpus, due to the more encyclopaedic style of writing. A brief description of the tasks follows, although we urge the reader to refer to the original paper for more detailed descriptions.

1. Sentence length: In **SentLen**, sentences are divided into multiple bins based on their length; the job of the classifier is to predict the appropriate bin, creating a 6-way classification task.
2. Word count: In **WC**, we sample sentences that feature exactly one amongst a thousand mid-frequency words, and train the classifier to predict the word: this is the most ‘difficult’ task, in that it has the most possible classes.
3. Tree depth: The **TreeDepth** task simply asks the representation to predict the depth of the sentence’s syntax tree. Unlike the original paper, we use the depth of the dependency tree instead of the constituency tree: this has the added benefit of being faster to extract,

due to the relative speed of dependency parsing, as well as having better multilingual support. We also replace the authors’ sentence-length-decorrelation procedure with a naïver one, where we sample an equal number of  $d$ -depth trees for each sentence length bin.

4. Bigram shift: In **BiShift**, for half the sentences in the dataset, the order of words in a randomly sampled bigram is reversed. The classifier learns to predict whether or not the sentence contains a reversal.
5. Subject number: The **SubjNum** task asks the classifier to predict the number of the subject of the head verb of the sentence. Only sentences with exactly one subject (annotated with the `nsubj` relation) attached to the root verb were considered.
6. Object number: **ObjNum**, similar to the subject number task, was annotated with the number of the direct object of the head verb (annotated with the `obj` relation).
7. Coordination inversion: In **CoordInv**, two main clauses joined by a coordinating conjunction (annotated with the `cc` and `conj` relations) have their orders reversed, with a probability of one in two. Only sentences with exactly two top-level conjuncts are considered.
8. (Semantic) odd man out: **SOMO**, one of the more difficult tasks in the collection, replaces a randomly sampled word with another word with comparable corpus bigram frequencies, for both bigrams formed with the preceding and the succeeding words. We defined ‘comparable’ as having a log-frequency difference not greater than 1.
9. Tense prediction: The **Tense** prediction asks the classifier to predict the tense of the main verb: the task uses a rather simple division of tenses; two tenses, `Past` and `Pres`. Tense information was extracted from UD morphological annotation.

## 5 Encoders

The NLI-oriented training approach for all our encoders is based on *InferSent* (Conneau et al.,

2017a). Our first encoder is an RNN-based encoder (specifically, an LSTM); we use two variants of this encoder, one that uses max-pooling over bidirectional RNN states, and another that uses the last recurrent state. Our second encoder is a self-attention based encoder Lin et al. (2017), with the same max-pool/last-state variants. Finally, we include a convolutional sentence representation model inspired by Gan et al. (2016); this model has an order of magnitude fewer parameters than the RNN- and attention-based variants. A variant of this CNN-based encoder has the maximum pooling replaced with average pooling.

### 5.1 Representation learning

We train all our encoders to represent sentences using the same NLI-based objective followed by Conneau et al. (2017a). More precisely, we first convert the word indices for both our premise and our hypothesis into dense word representations using pretrained fastText word embeddings (Bojanowski et al., 2016). These representations are then passed to our encoder of choice, which returns two fixed-length vectors:  $u$  for the premise, and  $v$  for the hypothesis. These vectors are combined and concatenated, as  $[u, v, u * v, |u - v|]$ , and then passed through a classifier with a softmax layer that outputs a probability distribution over the three NLI labels.

### 5.2 Mapping

Our procedure for mapping our encoders cross-linguistically broadly follows the principled mapping approach described in Conneau et al. (2018b). The procedure begins by mapping our *word* representations to the same vector space. Unlike the original paper, we use the supervised variant of VecMap (Artetxe et al., 2016) for representation mapping; however, we use seed dictionaries described in Conneau et al. (2017b). Having mapped our word representations, we proceed to map our sentence representations. To do so, we first build an English-language encoder, using (frozen) word representations and (frozen) encoder weights obtained in Section 5.1. We then build a target language encoder, using word embeddings mapped to the same space as the English embeddings. The sentence encoder itself is initialised with random parameters.

We then encode the source and target sentences in an en-trg machine translation corpus, where trg is our target language. Our English encoder re-

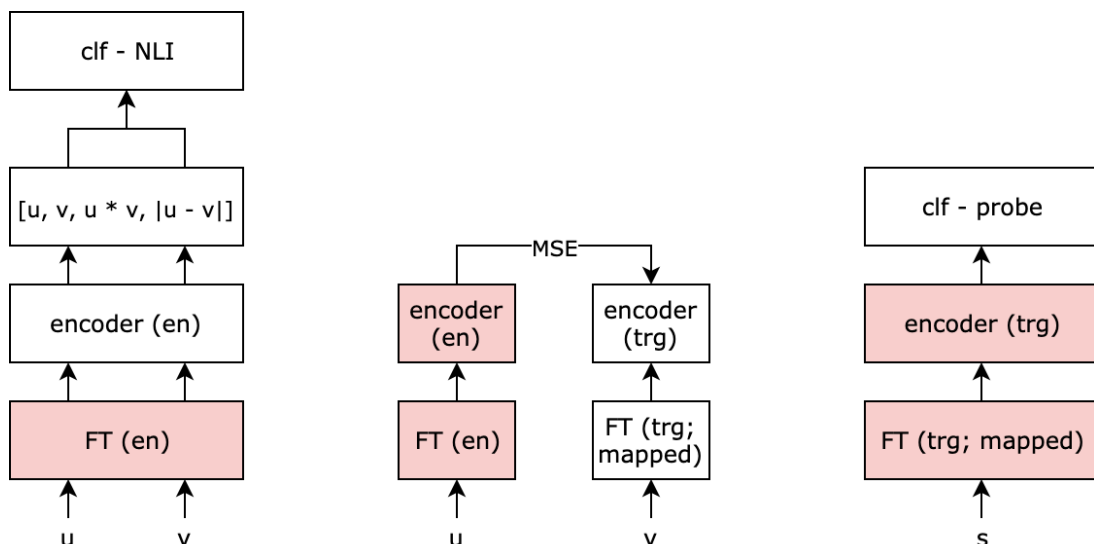


Figure 1: (a) an English-language encoder is trained on NLI data; (b) parallel sentences are encoded in English and the target language, and the MSE loss between them is minimised; (c) the mapped target encoders are used downstream in probing. Greyed-out blocks represent ‘frozen’ components that do not further adjust their parameters.

turns a ‘meaningful’ representation: recall that the encoder has weights trained on NLI data. We then use a mean-square error loss function to reduce the distance between our target-language representation and the English representation; the system then backpropagates through the target language encoder to obtain better parameters.

Our MSE loss function, similar to [Conneau et al.](#)’s function, attempts to minimise the distance between representations of parallel sentences, whilst simultaneously maximising the distance between random sentences sampled from either language in the pair. Mathematically, the alignment loss is given by:

$$\mathcal{L}_{align} = \|\mathbf{x} - \mathbf{y}\|_2 - \lambda(\|\mathbf{x}_c - \mathbf{y}\|_2 + \|\mathbf{x} - \mathbf{y}_c\|_2)$$

where  $\lambda$  is a hyperparameter.

We evaluate our mapped encoder on the relevant validation data section from the XNLI corpus per epoch, and terminate the mapping procedure when our validation accuracy does not improve for two consecutive epochs.

### 5.3 Multilingual probing

Having obtained our mapped sentence representation encoder, we proceed to plug the encoder into our probing architecture downstream, and evaluate classifier performance.

First, we load our mapped word representations for the language that we intend to analyse. We use these word representations to build sentence

representations, using the encoder architecture of choice. We then add a simple multi-layer perceptron (MLP) that learns to predict the appropriate label for each task: the MLP consists of a dense layer, a non-linearity (we use the sigmoid function), and another dense layer that we softmax over to arrive at per-class probabilities. During training, we keep the encoder’s parameters fixed. Mathematically, therefore, given an encoder  $f$  with parameters  $\theta$ , and word representations  $w_k$  for each word  $k$ :

$$\begin{aligned} s &= f(w_0, w_1, \dots, w_n; \theta) \\ z &= \text{MLP}(s) \\ \mathbf{y} &= \text{softmax}(z) \end{aligned}$$

where ‘MLP’ refers to a multi-layer perceptron with one sigmoid hidden layer.

Finally, we evaluate our representations on the relevant test portion. Whilst [Conneau et al.](#) used grid search to determine the best hyperparameters for each probing task, we did not do so, due to both time constraints, and in an attempt to ensure classifier uniformity across languages. We describe our probing results in Section 7.

## 6 Data

### 6.1 Probing data

We build our probing datasets using the relevant language’s Wikipedia dump as a corpus. Specifically, we use Wikipedia dumps (dated 2019-02-01), which we process using the WikiExtractor

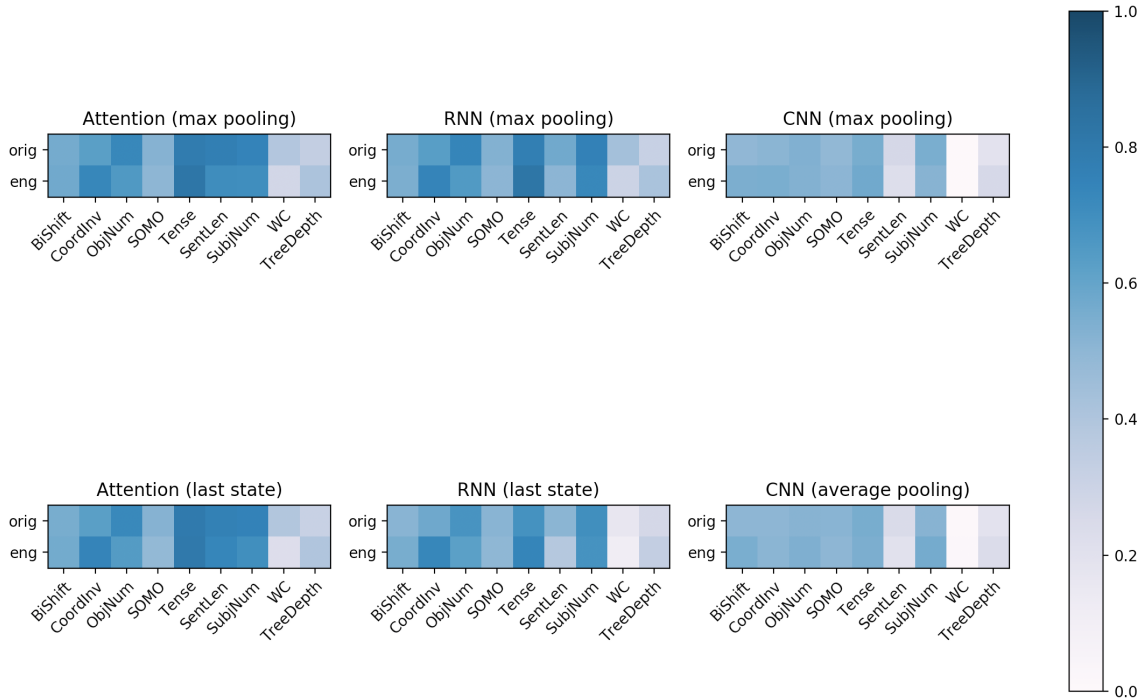


Figure 2: Probing accuracies for our six encoders on [Conneau et al.](#)’s dataset (`orig`), compared to our Wikipedia-derived dataset (`eng`)

utility<sup>1</sup>. We use the Punkt tokeniser ([Kiss and Strunk, 2006](#)) to segment our Wikipedia dumps into discrete sentences. For Russian, which lacked a Punkt tokenisation model, we used the UD-Pipe ([Straka and Strakova, 2017](#)) toolkit to perform segmentation.

Having segmented our data, we used the Moses ([Koehn et al., 2007](#)) tokeniser for the appropriate language, falling back to English tokenisation when unavailable: this was similar to XNLI’s tokenisation schema, and therefore necessary for appropriate evaluation on XNLI.

Next, we obtained dependency parses for our sentences, again using the UDPipe toolkit’s pre-trained models, trained on Universal Dependencies treebanks ([Nivre et al., 2015](#)). We then processed these dependency parsed corpora to extract the appropriate sentences; each task had 120k extracted sentences, divided into training/validation/test splits with 100k, 10k and 10k sentences respectively.

<sup>1</sup><https://github.com/attardi/wikiextractor/>

## 6.2 Mapping data

For mapping our sentence representations, we were restricted by the availability of large parallel corpora we could use for our mapping procedure. We used two such corpora: the Europarl corpus ([Koehn, 2005](#)), a multilingual collection of European Parliament proceedings, and the MultiUN corpus ([Tiedemann, 2012](#)), a collection of translated documents from the United Nations. We used Europarl for the official EU languages we analysed: German and Spanish. For Russian, we used MultiUN. We used both corpora for French, to attempt to analyse what, if any, effect the mapping corpus would have. We also truncated our MultiUN corpora to 2 million sentences, to keep the corpus size roughly equivalent to Europarl, and also due to time and resource constraints: mapping representations on the complete 10 million sentence corpus would have required significant amounts of time.

Both our corpora were pre-segmented: we followed the same Moses-based tokenisation scheme that we did for our probing corpora, falling back to English for languages that lacked appropriate tokeniser models.

## 7 Evaluation

As a preface to this section, we reiterate that the goal of this work was not to attempt to reach state-of-the-art on the tasks we describe; our goal was primarily to study the effect of transfer on sentence representations.

Our first step during evaluation, therefore, was to probe all our encoders using [Conneau et al.](#)'s original probing corpus, and compare these results to our English-language results on our Wikipedia-generated corpus. We present these results in the form of a heatmap in Figure 2.

Similarities between results on our corpora are instantly visible; these also appear to hold across encoders. Tasks with minor visible differences include **WC**, the most 'difficult' classification task (1k classes), and **TreeDepth**, where we use dependency tree depth instead of constituency tree depth, as well as a different sampling mechanism.

Next, we present Spearman correlations between the performance of our encoders on probing tasks and on the only 'true' cross-lingual downstream task we evaluated our systems on: cross-lingual natural language inference, via the XNLI ([Conneau et al., 2018b](#)) corpus. A caveat here is that we make no claims about the statistical significance of these results; given just six data points per language per task, our  $p$ -values tend to be well below acceptable for statistical significance. We refer the reader to [Conneau et al.](#)'s original probing work, where despite having results for 30 encoders, correlations between many downstream and probing tasks were not statisti-

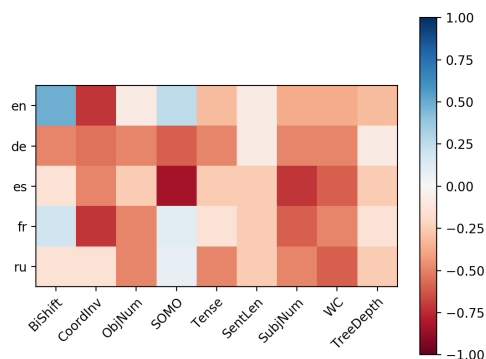


Figure 3: Spearman correlation between probing performance and XNLI; results are not statistically significant.

cally significant. Our correlations are presented, again in the form of a heatmap, in Figure 3. Our absolute results on XNLI are presented in the appendix. These are not a focus for this work: we did not attempt to obtain state-of-the-art, nor, indeed, perform any sort of hyperparameter optimisation to get the 'best' possible results. Given these caveats, we draw the reader's attention to the fact that the overwhelming majority of correlations are negative.

Finally, and most importantly, we measure downstream performance on probing tasks for all our cross-lingually mapped encoders. For visualisation relevant to our goals, and for brevity, we present these results, in Figure 4, as a heatmap of probing results *relative* to (our) English probing results; a full table with numeric results is presented in Appendix A.2.

## 8 Discussion

Our cross-lingual results display some very interesting characteristics, that we enumerate and attempt to explain in this section. These results can be analysed along three dimensions: that of language, encoder mechanism, and the probing task itself.

### 8.1 Language

Whilst our results are broadly similar across languages, Russian appears to be an exception to this: our probing performance for most tasks is considerably worse when transferred to Russian than other languages. Transfer corpus does not appear to be a factor in this case: most of our encoders perform very similarly on both the Europarl and the UN variants of our transferred French representations. These are interesting preliminary results, that would require further analysis: as we mentioned in an earlier section, we were rather limited in our choice of languages, however, we foresee a possible extension to this work including more typologically diverse languages. One possible explanation for the relatively poor results on Russian is the nature of the word embeddings themselves: whilst we did not use the same methods, we did map our embeddings to the same space using the same dictionaries as [Conneau et al. \(2017b\)](#). The results they describe for word translation retrieval are considerably poorer for English and Russian than they are for English and Spanish, French or German.

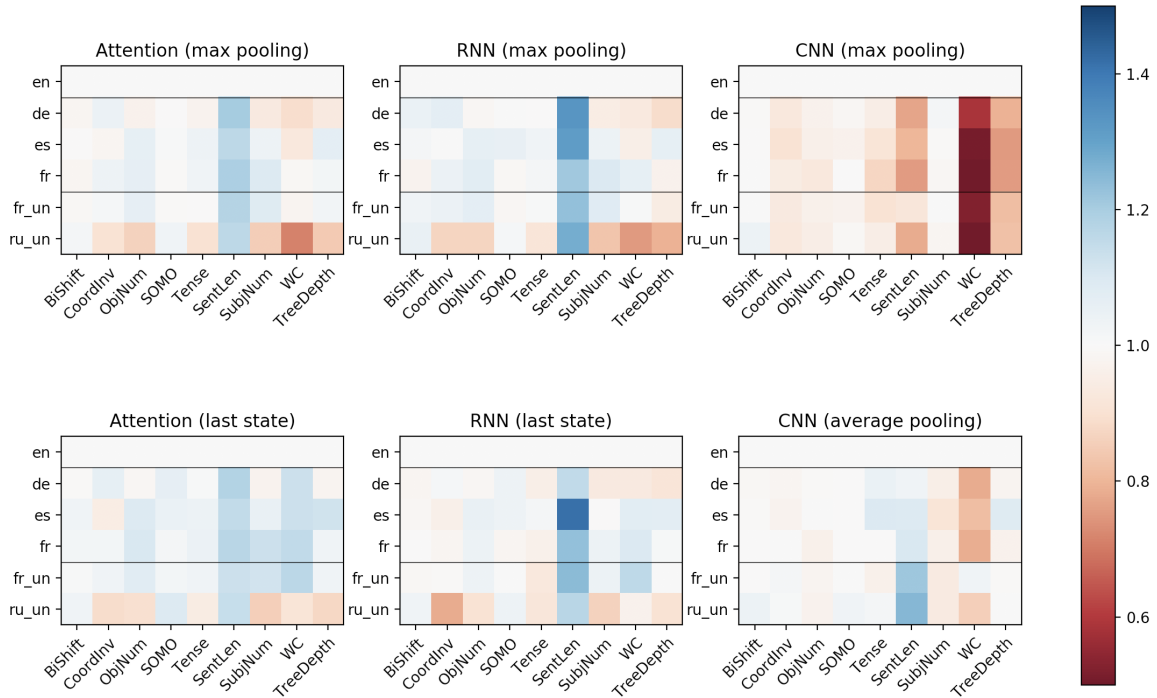


Figure 4: Probing results for each encoder relative to results on English. The second horizontal line indicates a switch in corpora. A white square indicates a value of 1, i.e. a parity in performance

## 8.2 Probing task

An immediate surprising takeaway from our results is the (perhaps counter-intuitive) fact that transferred representations are not necessarily worse at probing tasks than trained representations are. To help with the analysis of Figure 4, we present Table 1, where several trends are easily visible. In particular, a task that appears to stand out is **SentLen**, with transferred encoders displaying considerably improved performance in five out of six cases.

Apart from sentence length, both number prediction tasks – **SubjNum** and **ObjNum** – show noticeable improvements when transferred to a non-English language. The fact that this improvement is consistent across both number tasks likely also rules out mere coincidence. We hypothesise that the explanation for these three tasks in particular showing improvements on transfer lies in the specific nature of the mapping task. While it is plausible that this is due to these specific phenomena being less critical to NLI (on which our English encoders were trained) than to the attempt made by our target encoders to *emulate* these English representations, it is not immediately clear how these encoders are capable of exceeding the

predictive capabilities of the encoders they are attempting to emulate.

Another interesting observation is the variance in performance for the word content (**WC**) task, which also happens to be the ‘hardest’ task with the most output classes. We further note that, regrettably, none of our encoders were able to learn anything on **SOMO**.

| Task      | $\mu$ | $\sigma$ |
|-----------|-------|----------|
| BiShift   | 0.558 | 0.013    |
| CoordInv  | 0.656 | 0.111    |
| ObjNum    | 0.605 | 0.073    |
| SOMO      | 0.505 | 0.011    |
| Tense     | 0.708 | 0.124    |
| SentLen   | 0.523 | 0.259    |
| SubjNum   | 0.643 | 0.099    |
| WC        | 0.152 | 0.115    |
| TreeDepth | 0.330 | 0.082    |

Table 1: Mean and standard deviations for the absolute performance for each probing task, across languages and encoders



### 8.3 Encoder

All our encoders do appear to display very distinctive probing patterns, with variants of each encoder being more similar to each other than to different encoders. We enumerate some of the key observations:

1. Both our CNNs appear to perform worse than attentive or recurrent mechanisms; this is, however, perfectly understandable, as our CNN-based models had an order of magnitude fewer parameters than the recurrent ones. The choice of pooling mechanism, however, appears to have a more significant effect on convolutional encoders than on others.
2. Attentive encoders appear to be better at probing in general, whilst recurrent encoders show extremely strong performance on certain tasks, such as sentence length.
3. The max-pooled CNN is the only encoder that shows considerably worse performance on sentence length. This is also true for English, as is visible from Figure 2. We hypothesise that the fixed-length filters used in convolutional encoders do not store much information about sentence length, as they only observe chunks of the sentence. A max-pooling mechanism further compounds this inability to store length by eliminating possible compositional length information that mean-pooling does ignore.

## 9 Conclusions

Our analysis reveals several interesting patterns that appear to hold during cross-lingual transfer. Several of our probing tasks give us clearer insight into the sentence representations that we have generated by cross-lingual mapping, which is much needed: the principle of learning a sentence representation in parallel, combined with the fact that these representations actually appear to ‘work’ downstream, raises a lot of questions both about what information sentence representations hold, but more interestingly, in a cross-lingual context, about what *mutual* information a sentence and its translation contain.

We open-source both our training code and the probing datasets (that we dub X-PROBE)<sup>2</sup> that we

<sup>2</sup><https://github.com/lsgoslo/xprobe>

generated in the hope that the domain of cross-lingual analysis sees further work. There are several avenues for expansion, the most obvious being a probing-oriented analysis of more complex contextualisers, such as BERT, as well as of massively multilingual or language agnostic model.

We also hypothesise that more can be said about probing with a different selection of probing tasks; indeed, Liu et al. (2019) do provide a set of tasks that do not overlap with the tasks we have used. Selecting probing tasks that might tell allow us to better interpret cross-lingual modelling is another logical path one might follow. On a similar theme, an interesting research direction also involve adaptations of simple probing tasks describing linguistic phenomena to specialised architectures, for better comparison using SVCCA-style analyses (Saphra and Lopez, 2018).

Finally, we would also like to expand these datasets to more typologically diverse languages. A challenge in doing so is the availability of corpora that are large enough; none of our probing tasks have any sentences in common, which, given the size of each task’s corpus, requires a fairly large corpus for extraction. However, this process could possibly be simplified massively by removing this mutual exclusivity requirement, which would vastly simplify the process.

## References

- Mostafa Abdou, Artur Kulmizev, and Vinit Ravishanker. 2018. *MGAD: Multilingual Generation of Analogy Datasets*. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.
- Hanan Aldarmaki and Mona Diab. 2019. *Scalable Cross-Lingual Transfer of Neural Sentence Embeddings*. *arXiv:1904.05542 [cs]*. ArXiv: 1904.05542.
- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294.
- Mikel Artetxe and Holger Schwenk. 2018. *Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond*. *arXiv:1812.10464 [cs]*. ArXiv: 1812.10464.
- Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. *What do Neural*

- [Machine Translation Models Learn about Morphology?](#) In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. [Enriching Word Vectors with Subword Information](#). *arXiv:1607.04606 [cs]*. ArXiv: 1607.04606.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv:1508.05326 [cs]*. ArXiv: 1508.05326.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, and Chris Tar. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.
- Alexis Conneau and Douwe Kiela. 2018. [SentEval: An Evaluation Toolkit for Universal Sentence Representations](#). *arXiv:1803.05449 [cs]*. ArXiv: 1803.05449.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017a. [Supervised Learning of Universal Sentence Representations from Natural Language Inference Data](#). *arXiv:1705.02364 [cs]*. ArXiv: 1705.02364.
- Alexis Conneau, German Kruszewski, Guillaume Lample, Loc Barrault, and Marco Baroni. 2018a. [What you can cram into a single vector: Probing sentence embeddings for linguistic properties](#). *arXiv:1805.01070 [cs]*. ArXiv: 1805.01070.
- Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Herv Jgou. 2017b. [Word Translation Without Parallel Data](#). *arXiv:1710.04087 [cs]*. ArXiv: 1710.04087.
- Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018b. [XNLI: Evaluating Cross-lingual Sentence Representations](#). *arXiv:1809.05053 [cs]*. ArXiv: 1809.05053.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. 2016. Learning generic sentence representations using convolutional neural networks. *arXiv preprint arXiv:1611.07897*.
- Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. [Learning Distributed Representations of Sentences from Unlabelled Data](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics.
- the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1367–1377, San Diego, California. Association for Computational Linguistics.
- Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2017. [Visualisation and ‘diagnostic classifiers’ reveal how recurrent and recursive neural networks process hierarchical structure](#). *arXiv:1711.10203 [cs]*. ArXiv: 1711.10203.
- Douwe Kiela, Alexis Conneau, Allan Jabri, and Maximilian Nickel. 2017. [Learning Visually Grounded Sentence Representations](#). *arXiv:1707.06320 [cs]*. ArXiv: 1707.06320.
- Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. [Skip-Thought Vectors](#). *arXiv:1506.06726 [cs]*. ArXiv: 1506.06726.
- Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.
- Guillaume Lample and Alexis Conneau. 2019. [Cross-lingual Language Model Pretraining](#). *arXiv:1901.07291 [cs]*. ArXiv: 1901.07291.
- Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. [A Structured Self-attentive Sentence Embedding](#). *arXiv:1703.03130 [cs]*. ArXiv: 1703.03130.
- Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. 2019. Linguistic Knowledge and Transferability of Contextual Representations. page 22.
- Lajanugen Logeswaran and Honglak Lee. 2018. [An efficient framework for learning sentence representations](#). *arXiv:1803.02893 [cs]*. ArXiv: 1803.02893.
- Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. [Learned in Translation: Contextualized Word Vectors](#).
- Jeff Mitchell and Mirella Lapata. 2008. [Vector-based Models of Semantic Composition](#). In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio. Association for Computational Linguistics.

- Joakim Nivre, Željko Agić, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Cristina Bosco, Sam Bowman, Giuseppe G. A. Celano, Miriam Connor, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Daniel Galbraith, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Berta Gonzales, Bruno Guillaume, Jan Hajič, Dag Haug, Radu Ion, Elena Irimia, Anders Johannsen, Hiroshi Kanayama, Jenna Kanerva, Simon Krek, Veronika Laippala, Alessandro Lenci, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Shunsuke Mori, Hanna Nurmi, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenek Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Prokopis Prokopidis, Sampo Pyysalo, Loganathan Ramasamy, Rudolf Rosa, Shadi Saleh, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Kiril Simov, Aaron Smith, Jan Štěpánek, Alane Suhr, Zolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Sumire Uematsu, Larraitz Uria, Viktor Varga, Veronika Vincze, Zdeněk Žabokrtský, Daniel Zeman, and Hanzhi Zhu. 2015. *Universal dependencies 1.2*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, pages 6076–6085.
- Naomi Saphra and Adam Lopez. 2018. *Understanding Learning Dynamics Of Language Models with SVCCA*. *arXiv:1811.00225 [cs]*. ArXiv: 1811.00225.
- Holger Schwenk and Matthijs Douze. 2017. *Learning Joint Multilingual Sentence Representations with Neural Machine Translation*. *arXiv:1704.04154 [cs]*. ArXiv: 1704.04154.
- Milan Straka and Jana Straková. 2017. *Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipes*. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.
- Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J. Pal. 2018. *Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning*. *arXiv:1804.00079 [cs]*. ArXiv: 1804.00079.
- Jrg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. *Charagram: Embedding Words and Sentences via Character n-grams*. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515, Austin, Texas. Association for Computational Linguistics.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. *A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference*. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

## A Appendices

### A.1 Hyperparameters

| Component        | Layer          | Value          |
|------------------|----------------|----------------|
| Global           | Embeddings     | 300 (FastText) |
|                  | Batch size     | 10             |
|                  | Optimiser      | Adam           |
|                  | Learning rate  | $10^{-3}$      |
| RNN              | biLSTM dim     | 512            |
|                  | biLSTM layers  | 2              |
|                  | Dropout        | 10%            |
| CNN              | Filter sizes   | (3, 4, 5)      |
|                  | Padding        | (1, 2, 2)      |
|                  | Channels       | 800            |
|                  | Projection dim | 1024           |
| Attention        | biLSTM dim     | 512            |
|                  | biLSTM layers  | 2              |
|                  | Dropout        | 10%            |
|                  | MLP dim        | 150            |
|                  | Activation     | tanh           |
|                  | Attn. heads    | 60             |
| Mapper           | $\lambda$      | 0.25           |
| Probe classifier | Hidden dim     | 150            |
|                  | Activation     | $\sigma$       |

Table 2: Hyperparameters, divided by the ‘component’ that each layer belongs to. Note that biRNN dims are per direction.

### A.2 Additional results

| Encoder         | Language    |             |             |             |             |             |
|-----------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                 | English     | German      | Spanish     | French      | French (UN) | Russian     |
| RNN (maxpool)   | <b>0.71</b> | <b>0.66</b> | 0.68        | 0.68        | 0.65        | 0.61        |
| RNN (last)      | 0.66        | 0.63        | 0.65        | 0.65        | 0.63        | 0.59        |
| CNN (maxpool)   | 0.51        | 0.39        | 0.41        | 0.36        | 0.44        | 0.43        |
| CNN (avg. pool) | 0.51        | 0.50        | 0.51        | 0.50        | 0.50        | 0.48        |
| Attn. (maxpool) | <b>0.71</b> | 0.64        | 0.67        | 0.67        | 0.67        | 0.60        |
| Attn. (last)    | 0.70        | 0.65        | <b>0.69</b> | <b>0.69</b> | <b>0.66</b> | <b>0.62</b> |

Table 3: Language-specific results on relevant XNLI splits for each encoder

| <b>English</b>      | <b>BiShift</b>  | <b>CoordInv</b> | <b>ObjNum</b> | <b>SOMO</b>  | <b>Tense</b>   | <b>SentLen</b> | <b>SubjNum</b> | <b>WC</b>        | <b>TreeDepth</b> |
|---------------------|-----------------|-----------------|---------------|--------------|----------------|----------------|----------------|------------------|------------------|
| Attention (maxpool) | 0.57            | 0.73            | 0.65          | 0.5          | 0.82           | 0.7            | 0.7            | 0.27             | 0.41             |
| Attention (last)    | 0.56            | 0.74            | 0.64          | 0.49         | 0.8            | 0.74           | 0.7            | 0.22             | 0.4              |
| RNN (maxpool)       | 0.54            | 0.74            | 0.65          | 0.5          | 0.82           | 0.51           | 0.73           | 0.3              | 0.42             |
| RNN (last)          | 0.55            | 0.73            | 0.62          | 0.5          | 0.74           | 0.38           | 0.68           | 0.11             | 0.34             |
| CNN (maxpool)       | 0.55            | 0.55            | 0.53          | 0.51         | 0.57           | 0.22           | 0.52           | 0.01             | 0.26             |
| CNN (avg. pool)     | 0.55            | 0.51            | 0.54          | 0.5          | 0.54           | 0.21           | 0.56           | 0.02             | 0.24             |
| <b>German</b>       |                 |                 |               |              |                |                |                |                  |                  |
| <b>BiShift</b>      | <b>CoordInv</b> | <b>ObjNum</b>   | <b>SOMO</b>   | <b>Tense</b> | <b>SentLen</b> | <b>SubjNum</b> | <b>WC</b>      | <b>TreeDepth</b> |                  |
| Attention (maxpool) | 0.56            | 0.76            | 0.63          | 0.5          | 0.8            | 0.85           | 0.66           | 0.24             | 0.39             |
| Attention (last)    | 0.56            | 0.79            | 0.63          | 0.52         | 0.81           | 0.87           | 0.68           | 0.25             | 0.39             |
| RNN (maxpool)       | 0.57            | 0.8             | 0.64          | 0.51         | 0.82           | 0.68           | 0.69           | 0.28             | 0.37             |
| RNN (last)          | 0.54            | 0.74            | 0.61          | 0.52         | 0.71           | 0.44           | 0.63           | 0.11             | 0.31             |
| CNN (maxpool)       | 0.54            | 0.51            | 0.51          | 0.5          | 0.55           | 0.17           | 0.53           | 0.0              | 0.21             |
| CNN (avg. pool)     | 0.54            | 0.5             | 0.53          | 0.5          | 0.57           | 0.21           | 0.54           | 0.01             | 0.23             |
| <b>Spanish</b>      |                 |                 |               |              |                |                |                |                  |                  |
| <b>BiShift</b>      | <b>CoordInv</b> | <b>ObjNum</b>   | <b>SOMO</b>   | <b>Tense</b> | <b>SentLen</b> | <b>SubjNum</b> | <b>WC</b>      | <b>TreeDepth</b> |                  |
| Attention (maxpool) | 0.57            | 0.72            | 0.69          | 0.51         | 0.85           | 0.82           | 0.73           | 0.25             | 0.44             |
| Attention (last)    | 0.58            | 0.71            | 0.7           | 0.51         | 0.84           | 0.85           | 0.74           | 0.25             | 0.45             |
| RNN (maxpool)       | 0.55            | 0.75            | 0.69          | 0.53         | 0.85           | 0.67           | 0.76           | 0.28             | 0.44             |
| RNN (last)          | 0.55            | 0.7             | 0.65          | 0.52         | 0.75           | 0.54           | 0.68           | 0.12             | 0.36             |
| CNN (maxpool)       | 0.55            | 0.5             | 0.51          | 0.49         | 0.52           | 0.18           | 0.51           | 0.0              | 0.19             |
| CNN (avg. pool)     | 0.55            | 0.5             | 0.54          | 0.5          | 0.6            | 0.23           | 0.51           | 0.01             | 0.26             |
| <b>French</b>       |                 |                 |               |              |                |                |                |                  |                  |
| <b>BiShift</b>      | <b>CoordInv</b> | <b>ObjNum</b>   | <b>SOMO</b>   | <b>Tense</b> | <b>SentLen</b> | <b>SubjNum</b> | <b>WC</b>      | <b>TreeDepth</b> |                  |
| Attention (maxpool) | 0.56            | 0.76            | 0.7           | 0.5          | 0.85           | 0.84           | 0.76           | 0.27             | 0.42             |
| Attention (last)    | 0.58            | 0.76            | 0.71          | 0.5          | 0.84           | 0.86           | 0.79           | 0.26             | 0.41             |
| RNN (maxpool)       | 0.53            | 0.78            | 0.7           | 0.5          | 0.84           | 0.61           | 0.8            | 0.31             | 0.4              |
| RNN (last)          | 0.55            | 0.72            | 0.65          | 0.49         | 0.71           | 0.47           | 0.71           | 0.12             | 0.34             |
| CNN (maxpool)       | 0.55            | 0.52            | 0.49          | 0.51         | 0.5            | 0.17           | 0.51           | 0.0              | 0.2              |
| CNN (avg. pool)     | 0.55            | 0.51            | 0.52          | 0.5          | 0.54           | 0.23           | 0.54           | 0.01             | 0.23             |
| <b>French (UN)</b>  |                 |                 |               |              |                |                |                |                  |                  |
| <b>BiShift</b>      | <b>CoordInv</b> | <b>ObjNum</b>   | <b>SOMO</b>   | <b>Tense</b> | <b>SentLen</b> | <b>SubjNum</b> | <b>WC</b>      | <b>TreeDepth</b> |                  |
| Attention (maxpool) | 0.57            | 0.74            | 0.7           | 0.5          | 0.82           | 0.83           | 0.76           | 0.27             | 0.42             |
| Attention (last)    | 0.57            | 0.76            | 0.69          | 0.5          | 0.83           | 0.83           | 0.78           | 0.26             | 0.41             |
| RNN (maxpool)       | 0.56            | 0.78            | 0.7           | 0.5          | 0.83           | 0.62           | 0.79           | 0.3              | 0.39             |
| RNN (last)          | 0.55            | 0.73            | 0.65          | 0.5          | 0.68           | 0.47           | 0.71           | 0.13             | 0.34             |
| CNN (maxpool)       | 0.55            | 0.51            | 0.51          | 0.49         | 0.52           | 0.2            | 0.52           | 0.0              | 0.21             |
| CNN (avg. pool)     | 0.55            | 0.52            | 0.52          | 0.5          | 0.52           | 0.25           | 0.53           | 0.02             | 0.24             |
| <b>Russian</b>      |                 |                 |               |              |                |                |                |                  |                  |
| <b>BiShift</b>      | <b>CoordInv</b> | <b>ObjNum</b>   | <b>SOMO</b>   | <b>Tense</b> | <b>SentLen</b> | <b>SubjNum</b> | <b>WC</b>      | <b>TreeDepth</b> |                  |
| Attention (maxpool) | 0.58            | 0.66            | 0.56          | 0.52         | 0.74           | 0.82           | 0.6            | 0.2              | 0.35             |
| Attention (last)    | 0.58            | 0.66            | 0.57          | 0.53         | 0.76           | 0.84           | 0.6            | 0.2              | 0.35             |
| RNN (maxpool)       | 0.57            | 0.65            | 0.57          | 0.51         | 0.76           | 0.65           | 0.61           | 0.22             | 0.33             |
| RNN (last)          | 0.57            | 0.57            | 0.56          | 0.52         | 0.68           | 0.45           | 0.59           | 0.11             | 0.3              |
| CNN (maxpool)       | 0.57            | 0.51            | 0.5           | 0.5          | 0.55           | 0.17           | 0.51           | 0.0              | 0.21             |
| CNN (avg. pool)     | 0.57            | 0.51            | 0.52          | 0.52         | 0.56           | 0.26           | 0.53           | 0.01             | 0.24             |

Table 4: Complete set of absolute results per probing task, per encoder, per language. For English, these numbers are for unmapped, NLI-based encoders; for all other languages, these are post-mapping numbers