COMP0087 Statistical Natural Language Processing

Christian Martín Ríos
chris.rios.17@ucl.ac.uk

Ali Alktebi
ali.alktebi.20@ucl.ac.uk

Jakob Wessel
jakob.wessel.20@ucl.ac.uk

Udi Ibgui
udi.ibgui.17@ucl.ac.uk

May 2021

Abstract

The standard research in multilingual models usually explores the changes in performance when fine-tuning for specific tasks by probing entire language models. This project digs deep into language model architecture and compares the learning that occurs within layers of two different multilingual models (`mBERT` and `XLM`) when tasked to perform Question Answering (QA). The authors probe with a QA-task across the twelve different layers and examine where improvements in performance occur when evaluated on different languages. The authors conclude that QA-abilites seem to congregate in some of the middle-layers and that learning for QA occurs very similarly across multiple languages.

1 Introduction

Language Models (LM), such as BERT, have seen increased popularity in recent times. This is largely due to their high success in Natural language Processing (NLP) tasks, ranging from sentiment analysis, question answering tasks to machine translation.

Researchers have now turned their focus to building universal LM. Classical LM generally perform best on English tasks, where training data is most abundant. Having LM limited to only one language is extremely undesirable, shutting out billions of people from having effective utility from the technology. Several multilingual LM have been developed in recent times but their performance in most tasks still does not come on par with the English alternatives.

In this project, the authors tried to understand what sort of learning occurs within these models on the specific tasks of Question Answering (QA). Specific to both multilingual-QA, where the context-language is different from the question-language, as well as monolingual QA, where both are identical. In particular, the authors wanted to see if certain layers are more adept at picking up and specialising in multilingualism. In order to do so,

[1]Whilst SQuAD v2 is more recent, it only adds 50.000 unanswerable questions. Such a type of questions is

they explored the standard SQuAD v1.1 task [Rajpurkar et al., 2016] [1] as well as the MLQA task [Lewis et al., 2019].

For this purpose, the authors use the so-called probing technique. This consists of attaching model heads to a LM, after a certain number of frozen layers, and training the heads on certain tasks. The performance of the sub-LM together with the model-head then indicates how *rich* the encodings generated by the LM-layers are. Here, one uses a *direct* probing approach attaching a QA-head directly to the frozen layers [Tenney et al., 2019].

The authors argue that, if some layers are more specialised for understanding multilingualism, it would show by having a different performance across the layers when comparing tests on `MLQA` sets and the `SQuAD` v1.1 dataset. If this were to be true, a greater focus should be placed on improving those layers. In the course of this paper, the authors show that the QA-abilities develop really similarly across languages and that these are centered on only a few layers. This means that, firstly, one can generalise results from other studies on how LMs perform English QA tasks in relation to other languages, and secondly, one can focus on training only a few layers when wanting to do *multilingual*-QA tasks.

This report particularly relates the exploration of the multilingual-`BERT` [Devlin et al., 2018] and `XLM-RoBERTa` [Conneau et al., 2019] language models.

The main contributions of this project comprise the addition of a new direct probing technique via QA-tasks not yet seen in current literature. Secondly, it is shown that QA-abilities are similarly distributed across layers for different languages. Finally, it is shown that QA-abilities are centered in only a few layers, meaning one can also directly perform fine-tuning on those.

2 Literature Review

2.1 Cross-Lingual Models

BERT was originally trained on English corpora and therefore specialised in English-language tasks. This was then extended by [Devlin et al., 2018], to create a single language model, `mBERT`, product of the pre-training from monolingual corpora in 104 different languages. These top 104 languages were selected in terms of Wikipedia size, and were processed to yield zero-shot transfer learning across languages.

[Pires et al., 2019] empirically explore representations across languages and conclude that `mBERT` is fairly capable of executing zero-shot cross-lingual generalisation. Nevertheless, they conclude that successful transfer to transiliterated and typologically divergent targets would need the model to set out a multilingual training objective. They hypothesise that the reason why mBERT successfully generalises between languages is because of common word pieces, such as numbers, names and URLs. Since all of these are mapped to a collective space, the model gets to a point where it starts to stick to this behaviour with other words, yielding generalisation across languages.

[Lample and Conneau, 2019] build on past work to create a more general representation for multiple languages. Their cross-lingual model, `XLM`, pre-processes the data with a Byte Pair Encoding (BPE) [Sennrich et al., 2015] to create created a shared vocabulary. Further, it uses dual-language training using parallel sentences to

not yet present in the MLQA-task. Therefore the more comparable and smaller SQuAD v1.1 was used.

learn relations between words in different languages. Overall, the model outperforms others in cross-lingual tasks. Still, cross-lingual performance is considerably lower than single-language LM.

2.2 Understanding BERT

Due to BERT's overwhelming success and increasing popularity, much research has been devoted to better understand the inner working of these models, what is it they are actually learning? Research in this spirit has been named *BERTology*. Surveys such as that published by [Rogers et al., 2020] review more than 150 papers that study the inner working of BERT. Understanding these models is an important step in order to be able to improve and specialise them for tasks such as cross-lingual modelling.

In their paper, [Devlin et al., 2018] perform and investigate several ablation studies with BERT. These include the effect of model size, the effect of pre-training tasks and feature-based approaches with BERT. [Lee et al., 2019] investigate freezing layers and fine-tuning others. They find that, in order to preserve 90% of the original quality, only a fourth of the layers need to be fine-tuned. [Lin et al., 2019] find that the bottom four layers contain the most information about linear word order, whilst later layers have an increased knowledge of hierarchical sentence structure. The general consensus seems to be that syntactic information is most prominent within the middle layers, as seen in Section 4 of [Jawahar et al., 2019]. On the other hand, semantic information is suggested to be spread across the whole model [Tenney et al., 2019]. Various probing techniques are used for these investigations, yet research in probing done within multilingual language models appears to be scarce. The authors of this project want to investigate where the cross-lingual skills of these models start appearing and whether QA-abilities are shared.

2.3 Probing

One of the main tools for inspecting the abilities of a LM is probing. Probing generally refers the training of auxiliary classifiers that are designed to tell something about the information present in the encodings generated by an LM.

There seem to be two different ways of probing that are present in the literature [Choenni and Shutova, 2020]. One approach – layer-probing – is to take the first n layers of a model and attach a head to them (e.g. a classifier). This is then trained to perform a certain task and, upon termination, examine how much information there is in these first n layers. This can then be repeated in batches of n layers. The other approach, full-model probing, works generally similarly, where only the encodings generated by the full LM are taken.

There are a variety of different probing techniques that are present in the literature to test the QA-abilities of LMs. One of the most popular ones is the edge-probing framework. [Tenney et al., 2019] evaluate a model on relatively low-level tasks like speech-tagging or constituent-labeling. These probing tasks are extended upon by [van Aken et al., 2019], who also perform a layer-wise analysis, similarly to what the authors do in the course of this paper. [Choenni and Shutova, 2020] probe for typological information and [Peters et al., 2019] do layer-wise probing using among others sentiment-classification on SST-2. Probing has had great success in interpreting the abilities of models like BERT. Naturally, the authors use this for their own experiments.

3 Method

The authors' aim is to understand the multilingual QA abilities across all the 12 layers of `BERT` or `XLM-RoBERTa` (in the following referred to as XLM). They attempt layer probing across all layers, comparing four different models:

1. A finetuned `mBERT` for QA. This model consists of the full `mBERT` together with a model head for QA and is trained for 3 epochs on `SQuAD` v1.1. The authors use an already published model from `BERT`-Base-Multilingual-Cased-Finetuned-`SQuAD`, HuggingFace.

2. A non-finetuned `mBERT`, just consisting of the pretrained `mBERT` combined with a model head from `BERT`-Base-Multilingual-Cased, HuggingFace.

3. A finetuned `XLM-RoBERTa` for QA. This model consists of the full `XLM` together with a model head for QA and is trained for 3 epochs on `SQuAD` v1.1, comparable to the model on 1. The authors trained this model themselves and uploaded it to HuggingFace to share it with the NLP-community. This can be found here.

4. A non-finetuned XLM-roberta, just consisting of the pretrained XLM-roberta combined with a model head from `XLM`-roberta-base, HuggingFace.

Rather than performing probing with auxiliary tasks the authors decided to directly use a QA-task to evaluate the information an encoded text contains after being run through the first n layers of each model. This means that, for the first $n \in \{1, \ldots, 12\}$ layers in each model, they created a sub-neural network consisting of these first n layers, together with a model head for QA. In this sub-network, all the layers except for the model head were frozen, keeping the weights constants during training. The sub-network was trained for 3 epochs with a learning rate of *1e-5*, batch-size of *8*, `AdamW`-optimizer on the `SQuAD` v1.1 dataset. Training only the model head allows to infer how much QA-abilities are present in the first n layers. To infer those, the sub-network was evaluated on the `SQuAD`-evaluation dataset and different versions of `MLQA`. The authors decided to evaluate on *mlqa.en.en*, *mlqa.ar.en*, *mlqa.en.ar*, *mlqa.ar.ar*, *mlqa.es.es*, *mlqa.en.de* and *mlqa.de.de*. These combinations refer to, firstly, the language of the context (e.g. Arabic in *mlqa.ar.en*) followed by the language of the question. This selection of `MLQA`-languages allowed to see if there are differences in performance between English questions or context and similar Germanic based languages, like German. The authors also explored Arabic, which is morphologically rich and a Romanic language like Spanish. This allowed for some generalisation, whilst still limiting evaluation-time. In evaluation, the authors were mainly interested in the F1-scores on the evaluation-sets.

The advantage of this more direct approach in comparison to what is present in the literature is that one is more easily able to compare languages and to infer in which layers QA-abilities are actually present. By comparing the performance of QA-probes of the non-finetuned and the finetuned models, one can see to which layers the QA-abilities propagate to during finetuning (or if they just stay in the model head) and also whether the performance of the QA-probes rises in a similar fashion with the layers across different languages or if at some point, the model suddenly picks up multiligual QA-abilities.

For training and evaluation, the authors used the *HuggingFace* framework. Especially, the datasets and transformers library. Other publicly-available NLP toolkits and methods like `jiant` or self-coded model heads were examined, but not deemed suitable. Probing for all layers of the models and evaluation took around 40 hours of computation-time, done with *Google Colab*.

4 Results

All the extracted data of the experiments carried out, together with the code is publicly available in here. In Figure 1, we have visualised the F1-scores on different evaluation datasets obtained when probing the first n layers (shown on the x-axis). The plots and the probing framework allows to infer what information is stored in the first n layers. During probing, the authors only train the model heads, which work with the given encodings recovered after the n layers. The *goodness* of this model, evaluated as F1-score, then tells how *rich* these encodings are. In addition, the comparison of finetuned and non-finetuned models allows to infer what and where the models learn during finetuning, and the different `MLQA`-languages allow to see if there are differences in the QA-abilities across those.

Generally, it seems like the non-finetuned model (where only training the head) does not perform well. In this model, it appears that there is not much QA-knowledge, meaning that, when finetuning, it seems like the QA-abilities spread well across all layers and are not only fixed in the head. Nevertheless, non-finetuned `mBERT` performs slightly better than non-finetuned `XLM`.

For the finetuned model, it is no surprise that the results on the `SQuAD`-evaluation dataset are, layer-independently, the best. This is soon followed by the *mlqa.en.en* results. Peculiarly enough, the QA-abilities seem to be distributed very similarly across all languages, except for *en.ar*. The curves show the same behavior across all plots, indicating that there is no substantial difference in where the QA-abilities in one language or language-pair get picked up. They seem to be learned in a very parallel fashion and there is not necessarily a multilingual-ability picked up in one of the layers.

For the finetuned `XLM`, the F1-curves over the layers are all sigmoid-shaped (except for *mlqa.en.ar*), whilst for finetuned `mBERT`, all the curves plateau at around layer 6 and make a big step in the F1-scores after layer 8. That indicates that the QA-abilities are mostly learned by the model between layers 4 and 9 for both models. Specifically for `mBERT`, layer 9 seems to be crucial as, when this one is added, the encodings seem to be much richer and the model head can pick up a lot on the QA-abilities. Furthermore, it seems that in each plot there is a point where the finetuned `XLM`-curve is over the finetuned `mBERT`-curve, implying that `XLM` seems to pick up the QA abilities earlier on in the layers.

It is also captivating to compare the final results after 12 layers for the finetuned and non-finetuned models, visualised in Figures 3 and 4. It seems that for the finetuned models, `mBERT` performs surprisingly better in the cross-lingual case, whilst `XLM` is better in the monolingual model (in red), although not by a lot. For the non-finetuned models where learning occurs only in the head, the F1-scores are really low, but as seen in the line-plots, `mBERT` performs a lot better there.

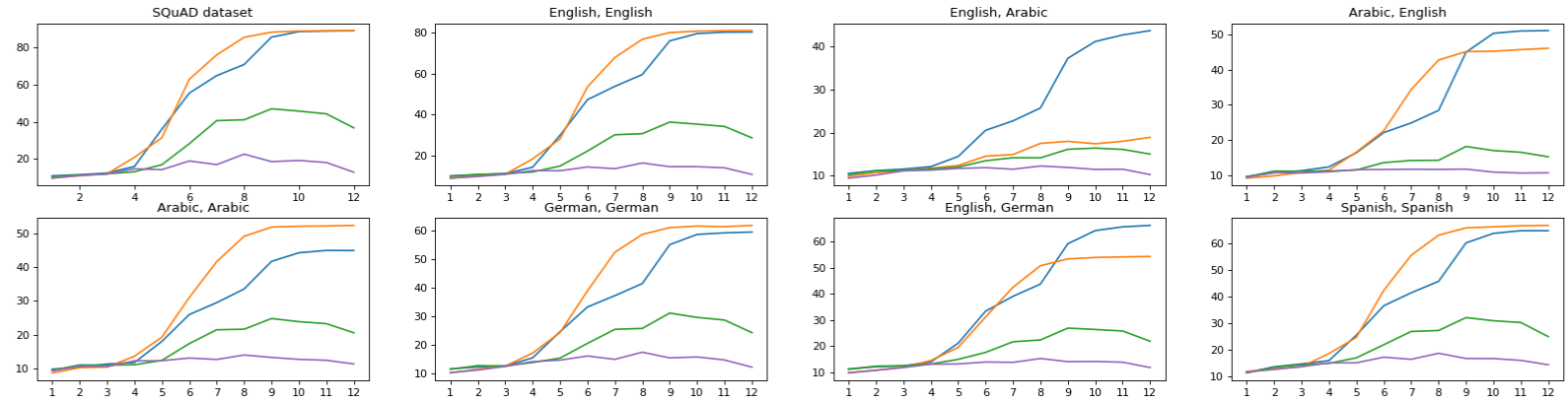

Figure 1: F1 scores (y) for probes attached after the first n layers (x).
Refer to Figure 2 for the legend.

This behavior, in particular the big difference between finetuned models and non-finetuned ones is really interesting. One can compare it these results with the work of [Peters et al., 2019] or [van Aken et al., 2019], where the authors did not seem to get much of a difference between pretraining and finetuning. This may be due to the fact that they mainly ran *diagnostic classifiers*, namely, relatively low-level tasks, whilst QA tasks are generally quite complex. The finetuning could therefore really make a difference. The non-finetuned model does not yet really have a lot of QA abilities anywhere, but when finetuned, they spread in the model, especially in the middle layers. The model needs to learn them first since it does not inherently have them. In addition, in both publications the authors find a way more linear relationship across the layers: it seems to be a QA-specificity that the learning is concentrated in only a few layers and a sigmoid shape appears. The authors of this report found nothing similar in the literature.

One can use those results to now train faster, better QA-models. Firstly, the similarity of curves indicates that the research on how to optimise English QA-systems can be generalised to other languages and even multilingual QA-systems. Secondly, one can use the concentration of the QA-abilities in certain layers to solely finetune those. As an experiment, the authors trained a QA-model with `XLM-roberta-base`, finetuning only the layers 4-8 and the model head, whilst keeping the other layers frozen. In Figure 5, one can see the F1-score of this model as evaluated on `SQuAD` and the usual `MLQA`-languages, compared with an *XLM-roberta-base* QA-model trained under the same parameters (number of epochs, batch size, etc.) where the latter is a finetuned full model without any layer freezing. This shows that the predictive accuracy across the languages is very similar, but only training layers 4-8 cuts down training time by over 60%. Therefore, only finetuning certain identified layers proves viable for QA-tasks and also extends to the multilingual case. This can be read together with the findings of [Lee et al., 2019], who find that only a fourth of the final layers need to be finetuned for certain downstream tasks contained in the `GLUE`-benchmark[2]. Simi-

[2]The authors of [Lee et al., 2019] look at CoLA, SST-2, MPRC and STS-B.

larly, the authors here show that also for QA, only a small part of the layers should be finetuned and that finetuning some yields better results than others. It is established that the middle ones (4-8), rather than the later ones like in [Lee et al., 2019], yield better results, and that this is also language-independent for QA.

Figure 2: Legend for Figure 1.

5 Conclusion & Future Work

In this paper, the authors introduced QA-probing with evaluation in multiple languages. They showed that LM need to be finetuned for the generated encodings to contain QA-information and that the QA-abilities then concentrate in a few of the middle layers. Furthermore, the authors demonstrated that QA-abilities are distributed very similarly across languages and that there is not necessarily a multilingual ability picked up in any of the layers. Both these findings can be used to train faster, better multilingual QA-systems in the future.

This research presents only the beginning in an investigation into the QA-learning-abilities of multilingual models. As future research, one could investigate how the learning changes if QA-training sets in other languages are used. Additionally, one could examine whether there are positive effects of language similarities. For example, German and English are quite similar and there is a chance that models perform well on one of those, when trained on the other language. It would also be worth looking if, as one would expect from this study, similar sigmoid-shapes occur in mixed-language training sets. In addition, one could try to extend the framework of edge-probing introduced by [Tenney et al., 2019] into the multilingual case. At the moment, only edge-probes exist for the English language. However, the OntoNotes-dataset, already used for some of the probes, exists for 4 languages. Therefore, extending existing probing frameworks onto those 4 should be straightforward.

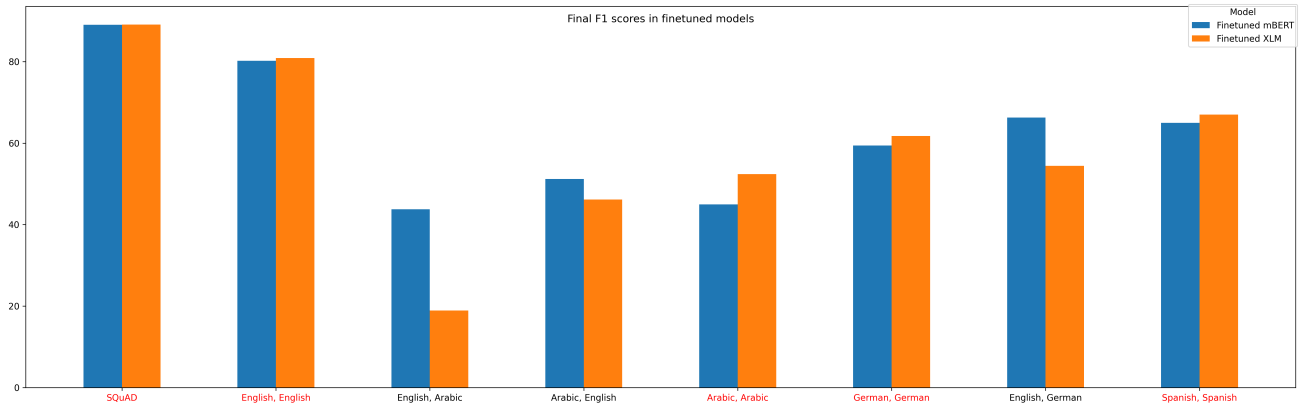

Figure 3: F1 scores for probes attached after the final layers for the finetuned models

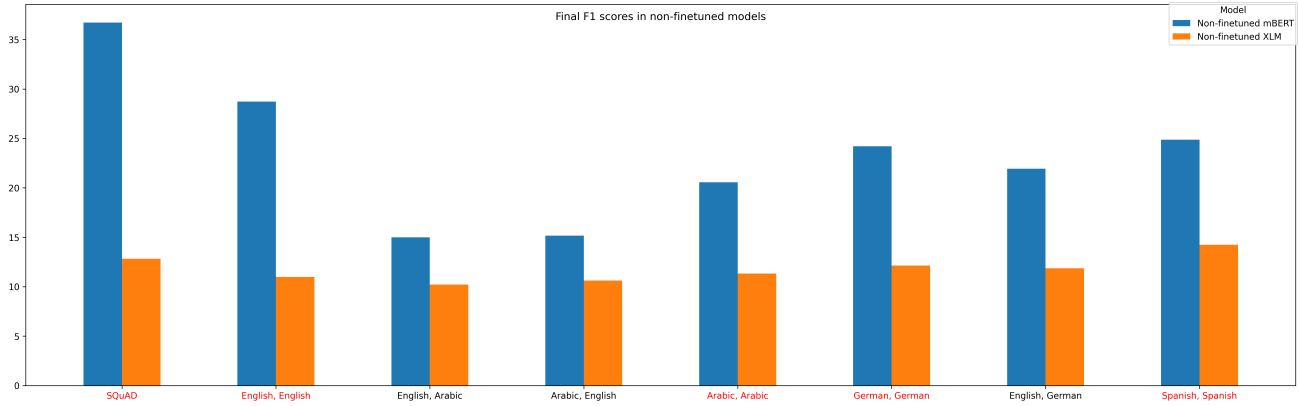

Figure 4: F1 scores for probes attached after the final layers for the non-finetuned models

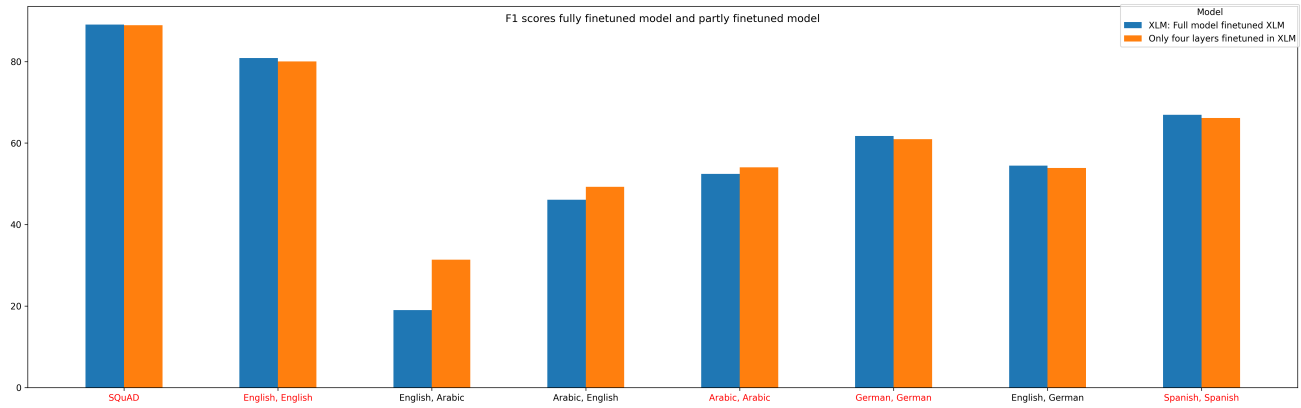

Figure 5: F1 scores for the fully vs. partly (layers 4-8) finetuned `XLM`-RoBERTa

References

[Choenni and Shutova, 2020] Choenni, R. and Shutova, E. (2020). What does it mean to be language-agnostic? probing multilingual sentence encoders for typological properties. *CoRR*, abs/2009.12862.

[Conneau et al., 2019] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

[Devlin et al., 2018] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[Jawahar et al., 2019] Jawahar, G., Sagot, B., and Seddah, D. (2019). What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

[Lample and Conneau, 2019] Lample, G. and Conneau, A. (2019). Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

[Lee et al., 2019] Lee, J., Tang, R., and Lin, J. (2019). What would elsa do? freezing layers during transformer fine-tuning. *CoRR*, abs/1911.03090.

[Lewis et al., 2019] Lewis, P., Oğuz, B., Rinott, R., Riedel, S., and Schwenk, H. (2019). Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

[Lin et al., 2019] Lin, Y., Tan, Y. C., and Frank, R. (2019). Open sesame: Getting inside bert's linguistic knowledge. *CoRR*, abs/1906.01698.

[Peters et al., 2019] Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on*

Representation Learning for NLP (RepL4NLP-2019), pages 7–14, Florence, Italy. Association for Computational Linguistics.

[Pires et al., 2019] Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual bert? *CoRR*, abs/1906.01502.

[Rajpurkar et al., 2016] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

[Rogers et al., 2020] Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in bertology: What we know about how bert works.

[Sennrich et al., 2015] Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

[Tenney et al., 2019] Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S. R., Das, D., and Pavlick, E. (2019). What do you learn from context? probing for sentence structure in contextualized word representations. *CoRR*, abs/1905.06316.

[Tenney et al., 2019] Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S. R., Das, D., and Pavlick, E. (2019). What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019*.

[van Aken et al., 2019] van Aken, B., Winter, B., Löser, A., and Gers, F. A. (2019). How does BERT answer questions? A layer-wise analysis of transformer representations. *CoRR*, abs/1909.04925.