Multilingual Language Models for Named Entity Recognition in German and English

Antonia Baumann
Trinity College Dublin
Dublin 2, Ireland
abaumann@tcd.ie

Abstract

We assess the language specificity of recent language models by exploring the potential of a multilingual language model. In particular, we evaluate Google's multilingual BERT (mBERT) model on Named Entity Recognition (NER) in German and English. We expand the work on language model fine-tuning by Howard and Ruder (2018), applying it to the BERT architecture.

We successfully reproduce the NER results published by Devlin et al. (2019). Our results show that the multilingual language model generalises well for NER in the chosen languages, matching the native model in English and comparing well with recent approaches for German. However, it does not benefit from the added fine-tuning methods.

1 Introduction

Language modelling (LM) has proven to improve many natural language processing (NLP) tasks across a wide set of tasks and domains (Dai and Le, 2015; Peters et al., 2018; Radford et al., 2018; Ruder, 2016; Devlin et al., 2019). These language models encompass the requirements "for natural language understanding technology to be maximally useful" generalising to multiple tasks, genres and datasets (Wang et al., 2018).

We argue that language models could also generalise along the language axis. Cross-lingual language understanding (XLU) significantly increases the usability of language technologies for international products such as Word, Facebook, or Google (all utilising varying levels NLP, for example translation, autocompletion or grammar correction). This interest is supported by Conneau et al. (2018) from Facebook AI[1], who laid one of the first milestones by creating a multilingual natural language inference corpus (XNLI) for XLU evaluation.

Therefore, our first research aim is to investigate the cross-lingual potential of Google's multilingual BERT (mBERT). Our experiments aim to establish a baseline under good transfer learning conditions: closely related languages with enough native data for fine-tuning. We expand the baselines Google published on natural lanugage inference (NLI) to named entity recognition (NER).

The second aim is to analyse if the BERT architecture benefits from special fine-tuning methods proposed by Howard and Ruder (2018). These showed significant performance increase for an LSTM-based architecture, but have not been generalised to other architectures. Besides LSTMs, Transformers are becoming an increasingly popular choice for language models, making BERT an ideal candidate to incorporate these fine-tuning methods.

Contributions: We make the following contributions to current LM research:

- We validate the original results published by Devlin et al. (2019), by replicating their NER experiment in Pytorch. For this we compare the method outlined in their paper and other replication attempts.
- We show that for NER, Google's multilingual BERT model matches the monolingual BERT model for English, and for German compares with most of the recent native models.
- We adapt the fine-tuning methods by Howard and Ruder (2018) for Google's BERT model. Our results show that slanted triangular learning rates improve the model, but gradual

[1] in collaboration with New York University

Proceedings of the Student Research Workshop associated with RANLP-2019, pages 21–27,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/issn.2603-2821.2019_004

unfreezing and discriminative learning rates have no effect.

2 Related Work

There is a vast amount of pre-trained language model research. We briefly review the ones that this paper directly builds on.

2.1 Language Models

Bengio et al. (2003) published the first neural language model in 2003. Their basic architecture of (a) Embedding, (b) Encoding and (c) Pooling layer(s) is still used by neural language and word embedding models today. [2]

With the rise of recurrent neural networks (RNNs) in NLP, they became a better choice for (b) the Encoding layer of the LM (Mikolov et al., 2010). Especially the variation of a long-short-term memory (LSTM) RNN (Jozefowicz et al., 2016) which are still used by recent papers like Howard and Ruder (2018) and Peters et al. (2018). By combining a forward LM and a backward LM Peters et al. (2018) created a bidirectional language model (biLM).

Overall, the widely successful approach for language models is to (1) pre-train the LM on general text to predict the next sentence.[3] Then this language knowledge is transferred by (2) fine-tuning the model for the target task (Devlin et al., 2019; Howard and Ruder, 2018; Peters et al., 2018; Radford et al., 2018). The target tasks range from sentiment analysis in the movie domain (Howard and Ruder, 2018); named entity recognition for newspaper articles (Devlin et al., 2019); to question answering on Wikipedia data (Devlin et al., 2019; Peters et al., 2018; Radford et al., 2018).

2.2 Universal Language Model Fine-Tuning

Howard and Ruder (2018) introduced special LM fine-tuning methods, including a further step in between (1.5) where the language model is fine-tuned on the unlabelled task data using the language modelling objective.

In addition, they propose three more methods: Slanted triangular learning rates, an adaptation of the cyclic learning rates by Smith (2017, 2018). An individual learning rate for each layer (Discriminative learning); and gradual unfreezing where layers are slowly added to the training pool. Howard and Ruder (2018) found that the combination of all these additions worked best, reducing error rates by 18-24% on 6 text classification sets.

2.3 BERT

At the end of 2018, Google's BERT was the best performing model for the GLUE Benchmark[4] (Devlin et al., 2019; Wang et al., 2018). In contrast to previous language models they utilise a deeply bidirectional architecture for their transformer; meaning the model receives the whole sentence (or sentence pair) as input and each cell depends on the context of the previous and subsequent word in the sequence.

Due to this, BERT's training differs from other language models. The non-sequential input makes the next-word prediction task impossible. Instead, Devlin et al. (2019) train the model to predict masked words in the input sentence. For further cross-sentence context, they also trained it to classify if two sentences follow each other.

They argue that the added context improves the model, making more suited for sentence level tasks (Devlin et al., 2019). This is supported by their results on the tasks in the GLUE Benchmark, overall achieving an absolute improvement of 7.7%.

2.4 Multilingual Language Models

Out of the established language model architecture, BERT is the only one that also provides multilingual versions on their repository.[5] The mBERT model has been pre-trained on Wikipedia text from the top 104 languages. They evaluated their multilingual model on the cross-lingual natural language inference dataset (XNLI), showing good performance for the 6 languages they reported on (Conneau et al., 2018).

3 Multilingual BERT for NER

We use the multilingual BERT as our pre-trained LM. To evaluate its cross-lingual potential we select a task and multiple language for the experiments.

[2] Retrieved May 20th, 2019, from http://ruder.io/word-embeddings-1/index.html#classicneurallanguagemodel

[3] This is the most common language modelling objective.

[4] Retrieved May 20th, 2019, from https://gluebenchmark.com/

[5] Retrieved May 20th, 2019, from https://github.com/google-research/bert

3.1 Dataset & Languages

The CoNLL 2003 NER task (Tjong Kim Sang and De Meulder, 2003) was used by Devlin et al. (2019) to evaluate the English BERT model on NER. Since it also provides German data, it was the ideal candidate to validate our re-implementation of the model, evaluate the performance of the multilingual model on multiple languages, and compare against the monolingual model. The dataset is widely used for German NER and provides a baseline evaluation for the German model, that can be expanded to more recent datasets such as GermEval 2014 (Benikova et al., 2014).

The CoNLL dataset has been used with two different annotation types: IOB1 (described in the original paper (Tjong Kim Sang and De Meulder, 2003)) and BIO.[6] Since the BERT paper itself uses both annotation types in the examples they provide, it is unclear which one they used (Devlin et al., 2019). Our experiments compare the results of both annotation types.

3.2 Method/Architecture

The overall structure of the NER experiments is abstracted in figure 1. It shows that all experiments only differ in the data pre-processing and BERT model selected.

We follow the same structure outlined in the BERT paper: The data is pre-processed using Google's WordPiece tokenization, and then converted into a BERT input feature consisting of token ids, segment mask and attention mask. A tokenoptimal one classification layer[7] is added to convert the BERT output into label probabilities over the set of annotations. We use the softmax cross-entropy loss and the standard hyperparameter optimisation for BERT.[8]

We evaluate the model using the F1 score following the original CoNLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003).

3.3 Adaptation of Fine-Tuning Methods

Howard and Ruder (2018) described their fine-tuning methods for their 4-layer LSTM. This sections described our adaptations to apply them to BERT, a 12-layer transformer.

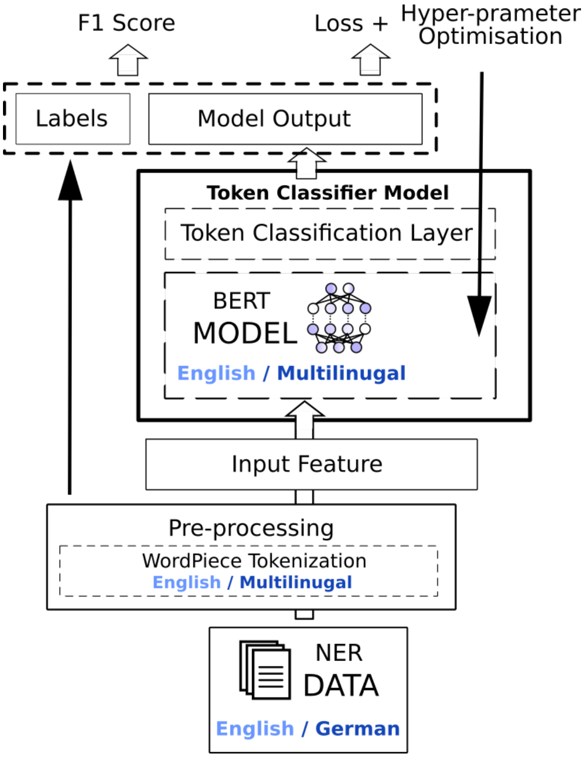

Figure 1: NER model architecture

Slanted Triangular Learning Rates This fine-tuning method is already used by BERT, however, Devlin et al. (2019) call it linear warmup. Therefore, we do not need to adapt this method, instead we compare BERT's performance with and without.

Discriminative Fine-Tuning Howard and Ruder (2018) used the following formula to calculate the learning rate for each layer:

$$\eta^n = \frac{\eta^0}{\delta^n} \tag{1}$$

where η^0, the learning rate of the top layer is manually selected. They empirically found a $\delta = 2.6$ to work well for their model. In the most recent ULMFiT implementation taught by Howard in his new course[9] η^0, on the other hand, decreases after every epoch.

Since a δ of 2.6 would lead to minuscule learning rates for the lower levels for BERT, we compare δ values: 2.6, 1.6 and 1[10]. Further, we measure several η^0's for each epoch to find the most optimal one.

[6] Also called IOB2.

[7] Linear classification layer

[8] Linear learning rate warmup.

[9] Retrieved May 20th, 2019, from https://nbviewer.jupyter.org/github/fastai/course-v3/blob/master/nbs/dl1/lesson3-imdb.ipynb

[10] Meaning a constant learning rate for all layers.

Gradual Unfreezing The difference in layer count also affects the unfreezing procedure. Going from top to bottom, Howard and Ruder (2018) added a single layer to the set of trained layers after each epoch, resulting in 4 epochs of fine-tuning.

Applying the same procedure to BERT would lead to 12 epochs, which is 3 to 4 times as much as the standard BERT task fine-tuning of 3-4 epochs. Instead we unfreeze the layers in groups of 3, thus, fine-tuning the model for 5 epochs.

4 Experiments

This section discusses the results of the NER experiments.

hyperparameter	options
batch size	16, 32
learning rate	2e-5, 3e-5, 5e-5
epochs	3,4

Table 1: BERT fine-tuning hyperparameters

4.1 Replication of English Results

For the replication, we performed a grid search over the hyperparameter options listed by Devlin et al. (2019) (see Table 1) and evaluate them on the development set. The best parameters[11] were then used to evaluate on the test set.

Table 2 shows that our implementation of BERT$_{BASE}$ for NER matches the original for the IOB1 annotation style. Therefore, validating the original results and our re-implementation. We use the same structure for the multilingual experiments.

System	Annot.	Dev	Test
BERT$_{LARGE}$	-	96.6	92.8
BERT$_{BASE}$	-	96.4	92.4
our BERT$_{BASE}$	IOB1	96.4	**92.6**
	BIO	95.9	92.2
Multilingual BERT	IOB1	96.4	91.9
	BIO	**96.5**	92.1

Table 2: [English Data] BERT model F1 results, compared to original paper. All results recorded are averaged out of 5 randomly initialised runs.

System	Annot.	Dev	Test
Ahmed & Mehler	IOB1	-	83.64
Riedl & Pado	-	-	84.73
Akbik et al. (2018)	-	-	**88.33**
Multilingual BERT	IOB1	**88.44**	85.81
	BIO	87.49	84.98

Table 3: [German Data] F1 Score evaluation on German CoNLL-2003 data, development and test set. Comparing our results with the state-of-the-art native models.

4.1.1 Multilingual BERT

We evaluate the multilingual BERT model on both the German and English dataset.

German We compare our results for the German data against the most recent state-of-the-art: for example Ahmed and Mehler (2018) used a Long-Short-Term Memory (LSTM) model with a Conditional Random Field (CRF) on top. Riedl and Padó (2018) lead with their bidirectional LSTM, which has been pre-trained on GermEval NER data.

As seen in Table 3 the multilingual model outperforms these first two models; notably for IOB1 annotation, and slightly exceeding Riedl and Padó (2018) with BIO. Riedl and Padó (2018) pre-trained on German data and fine-tuned for 15 epochs, in contrast to our multilingual pre-training and 3 epochs of fine-tuning.

The most recent and leading approach by Akbik et al. (2018), uses an LSTM + CRF with their novel contextual string embeddings [12] concatenated with Glove embeddings (Pennington et al., 2014), and task-trained character features. The contextual string embeddings were trained on half a million German words.

Using only these proposed contextual string embeddings, their models achieves 85.78 for F1 on the CoNLL dataset, similar to our multilingual model. Their research shows that the embeddings chosen strongly influences the models performance. We find that further comparison and analysis is needed to see how the multilingual model might benefit from concatenating multiple embeddings.

Overall, the multilingual Bert model compares well against the current state-of-the-art given that it is the only model using non-native embeddings.

[11] Batch size: 16; learning rate: 3e-5; epochs: 4

[12] Forward + Backward character embeddings

Model	Optim	Epoch 3		Epoch 4	
English BERT	with BertAdam	96.31	± 0.13	96.42	± 0.09
	without BertAdam	95.76	± 0.47	94.80	± 2.39
Multilingual BERT (English)	with BertAdam	96.51	± 0.18	96.55	± 0.21
	without BertAdam	95.88	± 0.74	96.25	± 0.06
Multilingual BERT (German)	with BertAdam	88.44	± 0.35	88.23	± 0.46
	without BertAdam	86.69	± 1.12	85.24	± 2.17

Table 4: Comparing the multilingual/English model with and without the BertAdam optimiser using the learning rate warmup. The scores reported are on the development set (IBO1). The optimal hyperparameters from the previous section were used for each model. Scores are averaged out of 4 random initialised runs.

The results by Akbik et al. (2018) show that our LM could be improved through richer embeddings

English Table 2 shows that the multilingual model matches the native for the development scores, yet it does not generalise as well to the test set.

4.2 Additional Fine-Tuning

First, we analyse the effect of the linear learning rate warmup: the results in Table 4 show that the warmup improves the scores and their stability.

Second, the other fine-tuning methods are added to the task fine-tuning step. For each layer the best discriminative learning rate is selected, from a set of manually selected η^0 values and the varying δ.

We measure the effectiveness of the additional LM fine-tuning on the target data by comparing (1) the "plain" BERT for task fine-tuning, (2) adding the additional fine-tuning methods and (3) adding the LM fine-tuning for 10/20 epochs.

The results in tables 5 and 6 show that the added fine-tuning methods do not exhibit any improvement over the "plain" BERT model. Further, there is no significant difference when adding the extra LM fine-tuning.

Our adaptation of the fine-tuning methods, however, are not fine-grained enough to allow for more detailed analysis. Compared to the multilingual model, the quick conversion does not yield results, instead a more in-depth approach is required to identify how a transformer is affected by these methods.

English BERT	Dev	Test
Plain	**96.4**	**92.6**
+ Task fine-tuning	95.60	92.38
+ 10e LM & Task fine-tuning	95.58	92.42
+ 20e LM & Task fine-tuning	95.91	92.36

Table 5: English BERT fine-tuning F1 results. Averaged over 2 runs.

Multilingual BERT	Dev	Test
Plain	**88.44**	**85.81**
+ Task fine-tuning	87.50	85.78
+ 10e LM & Task fine-tuning	87.11	84.98
+ 20e LM & Task fine-tuning	87.93	85.16

Table 6: Multilingual BERT fine-tuning F1 results for German. Averaged over 2 runs.

5 Conclusion

Pre-trained language models have led to significant empirical improvements for English natural language understanding. We validate parts of those findings by replicating the BERT result for NER.

Further, our work demonstrates that the expansion to cross-lingual language models holds a lot of potential. For German we outperform most recent models, leaving some room for improvement. The English the multilingual model closely matched the native one, in contrast to the BERT results reported for the XNLI task, where the English model noticeably outperformed the multilingual one.[13]

The investigation into LM fine-tuning methods proposed by Howard and Ruder (2018) showed that they do not improve the BERT model, with exception of slanted triangular learning rates that

[13] Retrieved May 20th, 2019, from `https://github.com/google-research/bert`

are already used by BERT.

5.1 Future Work

Our experiments support the hypothesis of cross-lingual language models for general NLP. The improvements Akbik et al. (2018) achieved with their embedding work, should be used on language models; to evaluate if they provide a similar benefit, not only for NER but general NLP tasks.

In the future, this should be expanded to more tasks and languages. Such as Wu and Dredze (2019), who concurrent to our work showed mBERT's zero-shot transfer learning potential.

Possible areas of focus are morphologically complex languages such as Finish, Korean and Tamil [14] since typological properties of languages can impact "language-agnostic" models (Gerz et al., 2018).

Further, Lample and Conneau (2019) show that cross-lingual language models can be improved on by cross-lingual language model (XLM) pre-training.

Acknowledgments

I would like to thank my supervisor Alfredo Maldonado for his consistent support and guidance throughout the project. His expertise and focus really helped to shape this work. Further, I would like to thank him for the encouragement and support in submitting this student research paper.

The research in this paper was supported by the ADAPT Centre for Digital Content Technology (https://www.adaptcentre.ie), funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Sajawel Ahmed and Alexander Mehler. 2018. Resource-size matters: Improving neural named entity recognition with optimized large corpora. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pages 919–924.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pages 1638–1649. https://www.aclweb.org/anthology/C18-1139.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Pado. 2014. Germeval 2014 named entity recognition shared task: companion paper .

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. pages 2475–2485.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*. pages 3079–3087.

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

Daniela Gerz, Ivan Vulić, Edoardo Maria Ponti, Roi Reichart, and Anna Korhonen. 2018. On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. pages 316–327.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 328–339.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410* .

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291* .

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke

[14] All included in mBERT's pre-training data

Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, pages 2227–2237. https://doi.org/10.18653/v1/N18-1202.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL*`https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/languageunderstandingpaper.pdf` .

Martin Riedl and Sebastian Padó. 2018. A named entity recognition shootout for german. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. pages 120–125.

Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* .

Leslie N Smith. 2017. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pages 464–472.

Leslie N Smith. 2018. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820* .

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*. Association for Computational Linguistics, Stroudsburg, PA, USA, CONLL '03, pages 142–147. https://doi.org/10.3115/1119176.1119195.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Brussels, Belgium, pages 353–355.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077* .