An Empirical Study of Pre-trained Transformers for Arabic Information Extraction

Wuwei Lan[1], Yang Chen[2], Wei Xu[2], Alan Ritter[2]
[1] Department of Computer Science and Engineering, Ohio State University
[2] School of Interactive Computing, Georgia Institute of Technology
lan.105@osu.edu {yang.chen, wei.xu, alan.ritter}@cc.gatech.edu

Abstract

Multilingual pre-trained Transformers, such as mBERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020a), have been shown to enable the effective cross-lingual zero-shot transfer. However, their performance on Arabic information extraction (IE) tasks is not very well studied. In this paper, we pre-train a customized bilingual BERT, dubbed GigaBERT, that is designed specifically for Arabic NLP and English-to-Arabic zero-shot transfer learning. We study GigaBERT's effectiveness on zero-short transfer across four IE tasks: named entity recognition, part-of-speech tagging, argument role labeling, and relation extraction. Our best model significantly outperforms mBERT, XLM-RoBERTa, and AraBERT (Antoun et al., 2020) in both the supervised and zero-shot transfer settings. We have made our pre-trained models publicly available at https://github.com/lanwuwei/GigaBERT.

1 Introduction

Fine-tuning pre-trained Transformer models (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019) has recently achieved state-of-the-art results on a wide range of NLP tasks where supervised training data is available. When trained on multilingual corpora, BERT-based models have demonstrated the ability to learn multilingual representations that support zero-shot cross-lingual transfer learning surprisingly effectively (Wu and Dredze, 2019; Pires et al., 2019; Lample and Conneau, 2019).

Without access to any parallel text or target language annotations, multilingual BERT (mBERT; Devlin et al., 2019) even supports cross-lingual transfer for language pairs that are written in different scripts, for example, English-to-Arabic. However, the transfer learning performance still lags far behind where supervised data is available in the target language. In this paper, we explore to what extent it is possible to improve performance in the zero-shot scenario by building a customized bilingual BERT for English and Arabic, a particularly challenging language pair for cross-lingual transfer learning.

We present GigaBERT, a customized BERT for English-to-Arabic cross-lingual transfer that is trained on newswire text in the Gigaword corpus (Graff et al., 2003; Parker et al., 2009) in addition to Wikipedia and web crawl data. We systematically compare our pre-trained models of different configurations against the mBERT (Devlin et al., 2019) and XLM-RoBERTa (XLM-R; Conneau et al., 2020a). By using a customized vocabulary and code-switched data specifically created for English-to-Arabic transfer learning, our GiagBERT outperforms mBERT and XLM-R_{base} (both support more than 100 languages) on a range of IE tasks, including named entity recognition, part-of-speech tagging, argument role labeling, and relation extraction. Further performance gains are demonstrated by augmenting the pre-training corpus with synthetically generated code-switched data. This demonstrates the usefulness of anchor points for zero-shot cross-lingual transfer learning. GigaBERT also performs well when annotated Arabic data is available, outperforming AraBERT (Antoun et al., 2020), the state-of-the-art Arabic-specific BERT model, on various Arabic IE tasks.

2 Related Work

The existing Arabic pre-trained models are either monolingual, such as hULMonA (ElJundi et al., 2019) and AraBERT (Antoun et al., 2020); or multilingual with several or over a hundred languages, such as mBERT (Devlin et al., 2019), XLM (Lample and Conneau, 2019), and XLM-RoBERTa (Conneau et al., 2020a). There is no bilingual pre-

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 4727–4734,
November 16–20, 2020.

Models	Training Data		Vocabulary			Configuration	
	source	#tokens (all/en/ar)	tokenization	size (all/en/ar)	cased	size	#parameters
AraBERT	newswire	2.5B/ – /2.5B	SentencePiece	64k/ – / 58k	no	base	136M
mBERT	Wiki	21.9B/2.5B/153M	WordPiece	110k/53k/5k	yes	base	172M
XLM-R_{base}	CommonCrawl	295B/55.6B/2.9B	SentencePiece	250k/80k/14k	yes	base	270M
XLM-R_{large}	CommonCrawl	295B/55.6B/2.9B	SentencePiece	250k/80k/14k	yes	large	550M
GiagBERT-v0	Gigaword	4.7B/3.6B/1.1B	SentencePiece	50k/28k/19k	yes	base	125M
GigaBERT-v1	Gigaword, Wiki	7.4B/6.1B/1.3B	WordPiece	50k/25k/23k	yes	base	125M
GigaBERT-v2/3	Gigaword, Wiki, Oscar	10.4B/6.1B/4.3B	WordPiece	50k/21k/26k	no	base	125M
GigaBERT-v4	Gigaword, Wiki, Oscar (+ code-switch)	10.4B/6.1B/4.3B	WordPiece	50k/21k/26k	no	base	125M

Table 1: Configuration comparisons for AraBERT (Antoun et al., 2020), mBERT (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020a), and GigaBERT (this work).

trained language model designed specifically for English-Arabic. K et al. (2020) pre-trained small-scale (e.g., 1GB data and 2M training steps) bilingual BERT for English-Hindi, English-Spanish, and English-Russian to study the impact of linguistic properties of the languages, the architecture of the model, and the learning objectives on cross-lingual transfer. Kim et al. (2019) presented a bilingual BERT using multi-task learning for translation quality estimation with regards to English-Russian and English-German. Conneau et al. (2020b) focused on the bilingual XLM for English-French, English-Russian, and English-Chinese to analyze the cross-lingual transfer ability with domain similarity, anchor points, parameter sharing, and language similarity.

3 GigaBERT

We present five versions of GigaBERT pre-trained using the Transformer encoder (Vaswani et al., 2017) with $BERT_{base}$ configurations: 12 attention layers, each has 12 attention heads and 768 hidden dimensions, which attributes 110M parameters. Table 1 shows a detailed summary of the training data and model parameters.

3.1 Training Data

We pre-train our GigaBERT models using the fifth edition of English and Arabic Gigaword corpora.[1] The Gigaword data consists of 13 million news articles[2] and matches the domain of many NLP tasks. We split English and Arabic sentences without tokenization by a modified version of the Stanford CoreNLP tool (Manning et al., 2014).[3] We also add Wikipedia data processed by WikiExtractor[4] for better coverage. As the English Wikipedia (total 2.5B tokens) is much larger than the Arabic Wikipedia (total 0.15B tokens), we balance the pre-training data by (1) up-sampling the Arabic data by repeating the Wikipedia portion five times and the Gigaword portion three times; (2) adding the Arabic section of the Oscar corpus (Ortiz Suárez et al., 2019), a large-scale multilingual dataset filtered from the Common Crawl.

Code-Switched Data Augmentation. To further improve cross-lingual transfer capability, we leverage English-Arabic dictionaries to create synthetic code-switched training data (Conneau et al., 2020a). We experimented with three dictionaries: PanLex (Kamholz et al., 2014), MUSE (Conneau et al., 2018), and Wikipedia parallel titles. We extract parallel article titles in Wikipedia based on the inter-language links and the entities based on the Wikidata (Jiang et al., 2020).[5] The dictionaries of PanLex, MUSE, Wikipedia contain 24K, 44K, 2M entries, respectively, and on overage 4.6, 1.4 and 1 translations per entry (English or Arabic). For training GigaBERT-v4, we code-switch up to 50% random sentences for both English and Arabic and up to 30% of tokens for each sentence. During the replacement process, we prioritize substitutions based on the Wikipedia titles, then PanLex and MUSE if the proportion of tokens being replaced has not reached 30% for a given sentence.

[1] https://catalog.ldc.upenn.edu/LDC2011T07 and https://catalog.ldc.upenn.edu/LDC2011T11

[2] We flattened the Gigaword data with https://github.com/nelson-liu/flatten_gigaword.

[3] In the early versions of GigaBERT (v0/1/2/3), we split Arabic sentences at period, exclamation, and question mark.

[4] https://github.com/attardi/wikiextractor

[5] https://github.com/clab/wikipedia-parallel-titles and https://dumps.wikimedia.org/wikidatawiki/entities/

3.2 Vocabulary

The vocabulary size is critical to the performance of pre-training models, as it directly impacts the subword granularity and the number of parameters. The original English BERT (Devlin et al., 2019) uses a 30k vocabulary size for ∼3B tokens of training data, while the multilingual BERT and XLM-R have ∼5k and ∼14k Arabic subwords in their vocabularies respectively (Table 1).[6] We choose a vocabulary size of 50k for our GigaBERT models based on preliminary experiments. For GigaBERT-v0, we use the unigram language model in the SentencePiece (Kudo and Richardson, 2018) to create 30k cased English subwords and 20k Arabic subwords separately.[7] For GigaBERT-v1/2/3/4, we did not distinguish Arabic and English subword units, instead, we train a unified 50k vocabulary using WordPiece (Wu et al., 2016).[8] The vocabulary is cased for GigaBERT-v1 and uncased for GigaBERT-v2/3/4, which use the same vocabulary.

3.3 Optimization

We use the official implementation of BERT (Devlin et al., 2019) in TensorFlow for pre-training. We use Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1e-4, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L2 weight decay of 0.01. The learning rate is warmed up over the first 100,000 steps to a peak value of 1e-4, then linearly decayed. The dropout is set to 0.1 for all layers. We use the whole word mask for GigaBERT-v0 and the regular subword mask for v1/2/3/4. The batch size is set to 512. GigaBERT-v0/1/2 are trained for 1.2 million steps on Google Cloud TPUs with a max sequence length of 128. GigaBERT-v3 is additionally trained for 140k steps with a max sequence length of 512. The maximum number of masked LM predictions per sequence is set to 20 when max sequence length is 128 and set to 80 when max sequence length is 512. GigaBERT-v4 is trained from the GigaBERT-v3 checkpoint for another 140K steps on the code-switched data. We also experiment with different thresholds for the code-switched data augmentation, as well as training models from scratch on the code-switched data (Appendix A).

[6]We check the Unicode range of characters to classify word pieces as English or Arabic.

[7]There are 633 word pieces shared by both languages. We add 633 unused symbols (e.g., unused-1, unused-2, etc.) to make up the 50k combined vocabulary.

[8]We use Hugging Face's implementation: https://github.com/huggingface/tokenizers

Task	#Train (en/ar)	#Dev (en/ar)	#Test (en/ar)	Metric
NER	7634/2683	1005/322	1095/238	F_1
POS	12543/6174	2002/786	2077/704	Acc
ARL	21875/11587	3345/1221	2603/1568	F_1
RE	63177/32984	10218/4482	6861/4638	F_1

Table 2: Statistics of the datasets for IE tasks.

4 Experiments

4.1 Downstream IE Tasks

We demonstrate the effectiveness of GigaBERT on named entity recognition (NER), part-of-speech tagging (POS), argument role labeling (ARL), and relation extraction (RE) tasks. We use the ACE 2005 corpus (Walker et al., 2006) in the NER, ARL, and RE evaluations, and use the Universal Dependencies Treebank v1.4 (Nivre et al., 2016) in the POS experiments. All of these datasets are from the news domain, as summarized in Table 2. For NER, we use the same English document splits as Lu and Roth (2015) and randomly shuffle Arabic documents into train/dev/test (80%/10%/10%). For ARL and RE, we randomly shuffle both English and Arabic documents into train/dev/test (80%/10%/10%). For POS, we follow the train/dev/test split by Wu and Dredze (2019). In the ARL fine-tuning experiment, we pair each trigger with its argument mentions as positive instances and with other entities in the sentence as negative instances. As for RE, we use gold relation mentions as positive examples and create negative examples by randomly pairing two entities in a sentence. We perform these tasks following the same fine-tuning pipeline as BERT (Devlin et al., 2019). We feed input sentences into a pre-trained model, then extract the necessary hidden representations, i.e., all token representations for NER/POS and argument/entity spans for ARL/RE, before applying one linear layer for classification. We evaluate for each language in the standard supervised learning setting, as well as the zero-shot transfer learning setting from English to Arabic, where the model is trained on the annotated English training data and evaluated on the Arabic test set.

4.2 Implementations

We implement the fine-tuning experiments with the PyTorch framework (Paszke et al., 2019) and choose hyperparameters by grid search.[9] We set the

[9]The search range includes learning rate (1e-5. 2e-5. 5e-5. 1e-4), batch size (4, 8, 16, 32) and epoch number (3, 7, 10).

Models	NER (F_1)			POS (Accuracy)			ARL (F_1)			RE (F_1)		
	en	ar	en→ ar	en	ar	en→ ar	en	ar	en→ ar	en	ar	en→ ar
AraBERT	-	78.6	-	-	97.6	-	-	73.3	-	-	83.1	-
mBERT	80.3	72.9	30.8/31.1	97.0	97.3	50.8/50.8	70.4	64.5	44.4/45.9	77.9	75.3	30.1/30.1
XLM-R_{base}	81.0	81.5	43.5/43.5	**97.8**	97.6	59.6/61.1	69.4	56.4	**54.4**/53.7	78.2	79.2	40.4/36.0
GigaXLM-R_{base}	82.0	80.8	45.4/45.0	97.3	97.7	**60.7/61.4**	70.1	71.4	52.6/52.6	79.6	79.5	43.3/44.0
GigaBERT-v0	79.1	76.6	43.9/45.9	96.8	97.5	49.7/54.1	69.1	66.1	42.3/42.2	76.6	72.5	21.5/20.9
GigaBERT-v1	82.8	72.9	49.1/49.1	97.2	96.6	51.9/52.2	**72.8**	67.7	44.6/45.5	**80.4**	73.2	36.0/31.1
GigaBERT-v2	82.5	75.2	48.3/48.2	97.2	**97.8**	53.1/53.4	72.0	66.7	42.5/44.1	79.4	74.2	31.9/36.8
GigaBERT-v3	83.4	83.1	48.9/48.3	97.1	**97.8**	53.3/54.7	72.3	**76.5**	51.0/51.0	79.9	**84.3**	**48.2**/46.8
GigaBERT-v4	**83.8**	**84.1**	**51.5/51.5**	97.1	97.7	54.6/55.5	71.9	73.9	52.7/**56.1**	79.1	83.6	43.3/**48.2**
XLM-R_{large}	85.8	84.8	49.3/50.4	98.0	97.8	61.7/61.2	72.3	73.4	58.0/57.4	83.2	82.1	52.5/57.5
GigaXLM-R_{large}	85.8	84.5	51.0/51.0	97.9	97.8	62.0/63.6	73.1	71.1	56.5/51.9	82.5	82.3	54.0/58.2

Table 3: Evaluation on four Arabic IE tasks that compares AraBERT (Antoun et al., 2020), multilingual BERT (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020a), GigaBERT/GigaXLM-R (this work). All models use $BERT_{base}$ architecture except XLM-R_{large}. GigaBERT-v4 is continued pre-training of GigaBERT-v3 on code-switched data. GigaXLM-R is domain adapted pre-training of XLM-R on Gigaword data.

learning rate to 2e-5, batch size to 8, max sequence length to 128, and the number of fine-tuning epochs to 7. Some exceptions include a learning rate of 1e-4 in NER experiments, max sequence length of 512, and batch size of 4 in RE experiments. For RE, we also use gradient accumulation to simulate the larger batch size of 32 when using models based on $BERT_{large}$ architecture.

4.3 Results and Analysis

Table 3 shows experimental results for the pre-trained models on both English and Arabic IE tasks. For the zero-shot transfer (en → ar), we report two scores on the Arabic test set, where the best checkpoint is selected based on the English dev set and the Arabic dev set, respectively. In summary, we find the key factors of improved pre-training performance are a large amount of training data in the target language, customized vocabulary, longer max length of sentence, and more anchor points from code-switched data. We also add experiments with XLM-R_{large} models as a reference, but the comparison focuses on the pre-trained models with $BERT_{base}$ configuration for fairness.

Single-language Performance. All versions of GigaBERT perform very competitively, especially the GigaBERT-v3/4. After adding Wikipedia and Oscar data, GigaBERT-v2 starts to outperform mBERT and XLM-R_{base} on most tasks. We find it crucial to continue training GigaBERT-v2 with a longer max sentence length of 512 word pieces, as the resulting GigaBERT-v3 model shows improvements in all four IE tasks. GigaBERT-v3 also outperforms AraBERT (Antoun et al., 2020), the state-of-the-art Arabic-specific BERT model by a large margin, showing that our bilingual GigaBERT does not sacrifice per-language performance. It is worth noting that GigaBERT-v4 also has competitive single-language performance after training on the synthetically created code-switched data.

Cross-lingual Zero-shot Transfer Learning. All pre-trained models show varied performance when we select checkpoints based on the English dev set and Arabic dev set, indicating that the best single-language performance does not necessarily imply the best cross-lingual performance. Compared to GigaBERT-v0, additional data used to train GigaBERT-v1/2 helps improve zero-shot transfer capability, even though the added data is not from the news domain. Different from previous works (Wu and Dredze, 2019; Pires et al., 2019) that attribute cross-lingual ability to shared subwords, GigaBERT-v3 has nearly no shared word pieces or scripts between English and Arabic, but still shows strong cross-lingual performance. We hypothesize the Transformer encoder projects similar contextual representations and enables cross-lingual transfer (Conneau et al., 2020b).

Code-Switched Pre-training. We show that we can further improve GigaBERT's cross-lingual transfer capability with a carefully designed code-switching procedure. Our GigaBERT-v4 pre-trained with code-switched data shows significant improvement over GigaBERT-v3, achieving

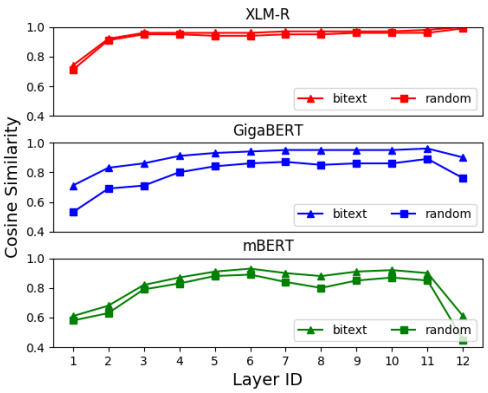

Figure 1: Cosine similarity between sentence representations of parallel sentences (bitext) and randomly paired sentences (random).

new state-of-the-art for zero-shot transfer from English to Arabic on NER, ARL, and RE. Our code-switched pre-training differs from Conneau et al. (2020b) in two aspects: 1) we explored multiple bilingual dictionaries, including PanLex (Kamholz et al., 2014), MUSE (Conneau et al., 2018) and Wikipedia titles, while MUSE appears to be the most effective; 2) we keep at least half of the sentences unchanged to balance between real data and artificial data. In practice, the generated data for GigaBERT-v4 has 47.4% of the sentences code-switched. We present more comparison experiments using varied code-switching mixes and different bilingual lexicons in Appendix A.

Domain-adapted Pre-training. We also explore whether XLM-RoBERTa can be improved by additional pre-training on Gigaword data, as Gururangan et al. (2020) have shown that the continued pre-training with in-domain data is helpful. We create GigaXLM-R models by continuing pre-training from XLM-R_{base} and XLM-R_{large} checkpoints in the Fairseq toolkit (Ott et al., 2019) for 500k steps on shuffled Arabic and English Gigaword corpus (max sequence length 512 and batch size 4). Although only $\sim$1% of the Gigaword corpus is used in this continued training step due to computing resource limit, GigaXLM-R still improves zero-shot transfer performance for NER, POS, and RE over the original XLM-R models as shown in Table 3. We could expect more performance improvement with a larger batch size and longer training time.

Embedding Space Analysis. We further analyze the semantic similarity of parallel English-Arabic sentence representations and find that GigaBERT is able to distinguish parallel sentences from randomly paired sentences more effectively compared to its counterparts. Our hypothesis is that cross-lingual representations for parallel English-Arabic sentences should be similar, but randomly paired sentences should be dissimilar. To evaluate cross-lingual similarity, we extract sentence representation of 5340 English-Arabic parallel sentences from the GALE corpus[10] and the same number of randomly paired sentences with pre-trained models across all 12 layers. We use the average of hidden representations, excluding [CLS] and [SEP], as a sentence representation. Cosine similarity is calculated for each sentence pairs and averaged across the whole corpus. In Figure 1, GigaBERT shows high similarity between parallel sentences and low similarity between randomly paired sentences. A clear separation for two types of paired sentences is shown across all the layers. In contrast, XLM-R is not able to distinguish between them but shows high similarity scores. mBERT shows low similarity in both cases. This suggests that our GigaBERT preserves language independent semantic information in the sentence representations, which might contribute to the competitive performance in downstream IE tasks.

5 Conclusions

In this paper, we show that the performance of zero-shot cross-lingual transfer can be improved by training customized bilingual BERT for a given language pair and text domain. We pre-trained several masked language models (GigaBERTs) for Arabic-English and conducted a focused study on information extraction tasks in the newswire domain. The experiments show that our GigaBERT model outperforms multilingual BERT, XLM-RoBERTa, and the monolingual AraBERT on NER, POS, ARL and RE tasks. We also achieve the new state-of-the-art performance fro zero-shot transfer learning from English to Arabic. We additionally studied code-switched pre-training for GigaBERT and domain-adapted pre-training for XLM-RoBERTa.

Acknowledgement

We thank Nizar Habash and anonymous reviewers for their valuable suggestions. We also thank the Google TFRC program for providing free TPU access. This material is based in part on research sponsored by the NSF (IIS-1845670), ODNI,

[10] https://catalog.ldc.upenn.edu/LDC2014T10

and IARPA via the BETTER program (2019-19051600004), DARPA via the ARO (W911NF-17-C-0095) in addition to an Amazon Research Award. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, ARO, DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Wissam Antoun, Fady Baly, and Hazem M. Hajj. 2020. AraBERT: Transformer-based model for arabic language understanding. In *Proceedings of the International Conference on Language Resources and Evaluation*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the Association for Computational Linguistics*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of International Conference on Learning Representation*.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Obeida ElJundi, Wissam Antoun, Nour El Droubi, Hazem Hajj, Wassim El-Hajj, and Khaled Shaban. 2019. hULMonA: The universal language model in Arabic. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English Gigaword. *Linguistic Data Consortium*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the Association for Computational Linguistics*.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural crf model for sentence alignment in text simplification. In *Proceedings of the Association for Computational Linguistics*.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: An empirical study. In *Proceedings of the International Conference on Learning Representations*.

David Kamholz, Jonathan Pool, and Susan Colowick. 2014. PanLex: Building a resource for panlingual lexical translation. In *Proceedings of the International Conference on Language Resources and Evaluation*.

Hyun Kim, Joon-Ho Lim, Hyun-Ki Kim, and Seung-Hoon Na. 2019. QE BERT: Bilingual BERT using multi-task learning for neural quality estimation. In *Proceedings of the Fourth Conference on Machine Translation*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representation*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of Emperical Methods in Natural Language Processing*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Proceedings of Advances in Neural Information Processing Systems*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of Emperical Methods in Natural Language Processing*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford CoreNLP natural language processing toolkit. In *Proceedings of the Association for Computational Linguistics*.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Yevgeni Berzak, Riyaz Ahmad Bhat, Eckhard Bick, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A.

Celano, Fabricio Chalub, Çağrı Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Claudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Jan Hajič, Linh Hà Mỹ, Dag Haug, Barbora Hladká, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Jessica Kenney, Natalia Kotsyba, Simon Krek, Veronika Laippala, Lucia Lam, Phuong Lê H`ông, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Keiko Sophie Mori, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Lu'o'ng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Vitaly Nikolaev, Hanna Nurmi, Petya Osenova, Robert Östling, Lilja Øvrelid, Valeria Paiva, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Baiba Saulīte, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Carolyn Spadine, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Mats Wirén, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2016. Universal Dependencies 1.4.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *Proceedings of the 7th Workshop on the Challenges in the Management of Large Corpora*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Robert Parker, David Graff, Ke Chen, Junbo Kong, and Kazuaki Maeda. 2009. Arabic Gigaword.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of Advances in Neural Information Processing Systems*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the Association for Computational Linguistics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.

Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of Advances in Neural Information Processing Systems*.

A Comparison Experiments for Code-Switched Pre-training

Given the English and Arabic monolingual corpus and the bilingual lexicons, we have different thresholds to control the code-switched data generation: 1) the percentage of sentences being code-switched within the whole corpus, we set sentence replacement threshold to limit the changed sentences; 2) the percentage of tokens being replaced within the sentence, we set token replacement threshold to limit the changed tokens; 3) the choice of bilingual lexicons, where we explore different combinations of PanLex, MUSE and Wiki titles. With the generated code-switched data, we can pre-train GigaBERT from scratch or load the existing checkpoint (GigaBERT-v3) for continued pre-training, which are s1 and s2 in Table 4, respectively.

As shown in Table 4, it's better to keep some sentences unchanged for code-switched pre-training. The continued pre-training (s2) shows slightly better performance than that training from scratch (s1). During the data augmentation, we need to keep a relatively low ratio for token replacement. The results also reveal that the MUSE dictionary is very promising, which outperforms the combinations of all dictionaries in some cases.

Models	NER (F_1)			POS (Accuracy)			ARL (F_1)			RE (F_1)		
	en	ar	en→ ar	en	ar	en→ ar	en	ar	en→ ar	en	ar	en→ ar
s1-0.5-0.3-all	82.1	83.3	49.7	97.0	97.7	58.3	72.4	74.4	48.6	80.0	84.1	47.0
s1-1.0-0.5-all	83.1	82.9	48.3	97.0	97.7	55.0	71.1	74.6	46.9	79.0	82.7	40.2
s1-0.5-0.3-pm	83.5	83.9	51.3	97.2	97.8	56.9	71.4	73.4	38.3	74.9	82.8	47.6
s1-0.5-0.3-m	82.4	84.7	52.9	97.1	97.8	58.6	70.7	72.4	52.1	77.3	83.7	46.0
s1-0.5-0.1-mw	83.1	83.9	52.2	97.2	97.6	55.0	71.7	72.7	49.0	78.2	84.1	54.0
s1-0.5-0.3-mw	83.3	83.3	53.5	97.1	97.7	56.0	71.9	72.8	46.8	79.2	84.2	44.3
s1-1.0-0.3-mw	82.7	84.4	48.2	97.1	97.7	56.1	70.6	72.7	51.4	77.9	84.6	47.3
s1-1.0-0.001-mw	83.4	83.8	54.1	97.2	97.7	55.1	72.3	73.3	48.0	78.7	83.5	41.2
s1-0.5-0.3-w	82.8	83.8	49.9	97.1	97.8	53.8	71.4	73.8	50.8	77.1	82.7	54.3
s2-0.5-0.3-all	83.8	84.1	51.5	97.1	97.7	55.5	71.9	73.9	56.1	79.1	83.6	48.2
s2-1.0-0.5-all	82.2	83.7	51.7	97.0	97.8	56.1	71.3	74.5	51.1	79.2	82.0	45.8
s2-0.5-0.3-pm	83.2	83.8	50.9	97.1	97.7	55.7	72.0	73.7	48.4	79.3	82.9	45.3
s2-0.5-0.3-m	83.4	83.4	52.9	97.2	97.7	52.9	71.0	73.9	55.0	78.8	83.5	52.5
s2-0.5-0.1-mw	83.0	85.1	52.7	97.2	97.8	53.6	71.9	75.0	50.0	79.0	83.7	52.2
s2-0.5-0.3-mw	83.4	85.0	51.0	97.1	97.7	52.4	72.2	74.9	49.3	81.0	83.7	49.8
s2-1.0-0.3-mw	83.2	83.7	50.2	97.0	97.7	53.5	71.0	71.8	54.7	67.2	81.3	42.9
s2-1.0-0.001-mw	83.6	84.2	49.6	97.4	97.7	52.3	71.8	73.2	51.2	79.0	84.0	42.6
s2-0.5-0.3-w	83.7	83.9	50.4	97.2	97.7	53.1	72.6	74.4	48.2	76.2	83.6	47.4

Table 4: Comparison experiments of different code-switching configurations. The model name is composed of four parts: **s1** (pre-train from scratch)/ **s2** (continue pre-training), sentence replacement threshold, token replacement threshold and bilingual lexicons, where **all** uses PanLex, MUSE and Wiki titles, **pm** uses PanLex and MUSE, **mw** uses MUSE and Wiki, **m** uses MUSE only and **w** uses Wiki only. The model s2-0.5-0.3-all is GigaBERT-v4 in the paper. The best checkpoint for en→ ar is selected with Arabic dev set.