The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor,[†] Nizar Habash
Computational Approaches to Modeling Language (CAMeL) Lab
New York University Abu Dhabi
[†]Carnegie Mellon University in Qatar
{go.inoue,alhafni,nurpeiis,nizar.habash}@nyu.edu
hbouamor@qatar.cmu.edu

Abstract

In this paper, we explore the effects of language variants, data sizes, and fine-tuning task types in Arabic pre-trained language models. To do so, we build three pre-trained language models across three variants of Arabic: Modern Standard Arabic (MSA), dialectal Arabic, and classical Arabic, in addition to a fourth language model which is pre-trained on a mix of the three. We also examine the importance of pre-training data size by building additional models that are pre-trained on a scaled-down set of the MSA variant. We compare our different models to each other, as well as to eight publicly available models by fine-tuning them on five NLP tasks spanning 12 datasets. Our results suggest that the variant proximity of pre-training data to fine-tuning data is more important than the pre-training data size. We exploit this insight in defining an optimized system selection model for the studied tasks.

1 Introduction

Pre-trained language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) have shown significant success in a wide range of natural language processing (NLP) tasks in various languages. Arabic has benefited from extensive efforts in building dedicated pre-trained language models, achieving state-of-the-art results in a number of NLP tasks, across both Modern Standard Arabic (MSA) and Dialectal Arabic (DA) (Antoun et al., 2020; Abdul-Mageed et al., 2020a).

However, it is hard to compare these models to understand what contributes to their performances because of their different design decisions and hyperparameters, such as data size, language variant, tokenization, vocabulary size, number of training steps, and so forth. Practically, one may empirically choose the best performing pre-trained model by fine-tuning it on a particular task; however, it is still unclear why a particular model is performing better than another and what design choices are contributing to its performance.

To answer this question, we pre-trained various language models as part of a controlled experiment where we vary pre-training data sizes and language variants while keeping other hyperparameters constant throughout pre-training. We started by scaling down MSA pre-training data size to measure its impact on performance in fine-tuning tasks. We then pre-trained three different variants of Arabic: MSA, DA, and classical Arabic (CA), as well as a mix of these three variants.

We evaluate our models along with eight other recent Arabic pre-trained models across five different tasks covering all the language variants we study, namely, named entity recognition (NER), part-of-speech (POS) tagging, sentiment analysis, dialect identification, and poetry classification, spanning 12 datasets.

Our contributions can be summarized as follows:

- We create and release eight Arabic pre-trained models, which we name CAMeLBERT, with different design decisions, including one (CAMeLBERT-Mix) that is trained on the largest dataset to date.[1]
- We investigate the interplay of data size, language variant, and fine-tuning task type through controlled experimentation. Our results show that variant proximity of pre-training data and task data is more important than pre-training data size.
- We exploit this insight in defining an optimized system selection model.

[1] Our pre-trained models are available at https://huggingface.co/CAMeL-Lab, and the fine-tuning code and models are available at https://github.com/CAMeL-Lab/CAMeLBERT.

Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 92–104
Kyiv, Ukraine (Virtual), April 19, 2021.

Ref	Model	Variants	Size	#Word	Tokens	Vocab	#Steps
	BERT (Devlin et al., 2019)	-	-	3.3B	WP	30k	1M
X_1	mBERT (Devlin et al., 2019)	MSA	-	-	WP	120k	-
X_2	AraBERTv0.1 (Antoun et al., 2020)	MSA	24GB	-	SP	60k	1.25M
X_3	AraBERTv0.2 (Antoun et al., 2020)	MSA	77GB	8.6B	WP	60k	3M
X_4	ArabicBERT (Safaya et al., 2020)	MSA	95GB	8.2B	WP	32k	4M
X_5	Multi-dialect-Arabic-BERT (Talafha et al., 2020)	MSA/DA	-	-	WP	32k	-
X_6	GigaBERTv4 (Lan et al., 2020)	MSA	-	10.4B	WP	50k	1.48M
X_7	MARBERT (Abdul-Mageed et al., 2020a)	MSA/DA	128GB	15.6B	WP	100K	17M
X_8	ARBERT (Abdul-Mageed et al., 2020a)	MSA	61GB	6.5B	WP	100K	8M
	CAMeLBERT-MSA	MSA	107GB	12.6B	WP	30k	1M
	CAMeLBERT-DA	DA	54GB	5.8B	WP	30k	1M
	CAMeLBERT-CA	CA	6GB	847M	WP	30k	1M
	CAMeLBERT-Mix	MSA/DA/CA	167GB	17.3B	WP	30k	1M

Table 1: Configurations of existing models and CAMeLBERT models. Ref is a model identifier used in Table 5. WP is WordPiece and SP is SentencePiece.

2 Related Work

There have been several research efforts on Arabic pre-trained models achieving state-of-the-art results in a number of Arabic NLP tasks. One of the earliest efforts includes AraBERT (Antoun et al., 2020), where they pre-trained a monolingual BERT model using 24GB of Arabic text in the news domain. Safaya et al. (2020) pre-trained ArabicBERT using 95GB of text mainly from the Arabic portion of the OSCAR corpus. Based on ArabicBERT, Talafha et al. (2020) further pre-trained their model using 10 million tweets, which included dialectal data. Lan et al. (2020) released several English-Arabic bilingual models dubbed GigaBERTs, where they studied the effectiveness of cross-lingual transfer learning and code-switched pre-training using Wikipedia, Gigaword, and the OSCAR corpus. Most recently, Abdul-Mageed et al. (2020a) developed two models, ARBERT and MARBERT, pre-trained on a large collection of datasets in MSA and DA. They reported new state-of-the-art results on the majority of the datasets in their fine-tuning benchmark.

Moreover, there have been various studies explaining why pre-trained language models perform well on downstream tasks either in monolingual (Hewitt and Manning, 2019; Jawahar et al., 2019; Liu et al., 2019a; Tenney et al., 2019a,b) or multilingual settings (Wu and Dredze, 2019; Chi et al., 2020; Kulmizev et al., 2020; Vulić et al., 2020). Most of these efforts leveraged probing techniques to explore the linguistic knowledge that is captured by pre-trained language models such as morphosyntactic and semantic knowledge. More recently, there have been additional efforts investigating the effects of pre-training data size and tokenization on the performance of pre-trained language models. Zhang et al. (2020) showed that pre-training RoBERTa requires 10M to 100M words to learn representations that reliably encode most syntactic and semantic features. However, a much larger quantity of data is needed for the model to perform well on typical downstream NLU tasks. Rust et al. (2020) empirically compared multilingual pre-trained language models to their monolingual counterparts on a set of nine typologically diverse languages. They showed that while the pre-training data size is an important factor, the designated tokenizer of each monolingual model plays an equally important role in the downstream performance.

In this work, we primarily focus on understanding the behavior of pre-trained models against variables such as data sizes and language variants. We compare against eight existing models. We find that AraBERTv02 (X_3) is the best on average and it wins or ties for a win in six out of 12 subtasks. Our CAMeLBERT-Star model is second overall on average, and it wins or ties for a win in five out of 12 subtasks. Interestingly, these systems are complementary in their performance and between the two, they win or tie for a win in ten out of 12 subtasks.

3 Pre-training CAMeLBERT

We describe the datasets and the procedure we use to pre-train our models. We use the original implementation released by Google for pre-training.[2]

3.1 Data

MSA Training Data For MSA, we use the Arabic Gigaword Fifth Edition (Parker et al., 2011), Abu El-Khair Corpus (El-Khair, 2016), OSIAN corpus (Zeroual et al., 2019), Arabic Wikipedia,[3] and the unshuffled version of the Arabic OSCAR corpus (Ortiz Suárez et al., 2020).

DA Training Data For DA, we collect a range of dialectal corpora: LDC97T19-CALLHOME Transcripts (Gadalla et al., 1997); LDC2002T38-CALLHOME Supplement Transcripts (Linguistic Data Consortium, 2002); LDC2005S08-Babylon Levantine Arabic Transcripts (BBN Technologies, 2005); LDC2005S14-CTS Levantine Arabic Transcripts (Maamouri et al., 2005); LDC2006T07-Levantine Arabic Transcripts (Maamouri et al., 2006); LDC2006T15-Gulf Arabic Transcripts (Appen Pty Ltd, 2006a); LDC2006T16-Iraqi Arabic Transcripts (Appen Pty Ltd, 2006b); LDC2007T01-Levantine Arabic Transcripts (Appen Pty Ltd, 2007); LDC2007T04-Fisher Levantine Arabic Transcripts (Maamouri et al., 2007); Arabic Online Commentary Dataset (AOC) (Zaidan and Callison-Burch, 2011); LDC2012T09-English/Arabic Parallel text (Raytheon BBN Technologies et al., 2012); Arabic Multi Dialect Text Corpora (Almeman and Lee, 2013); A Multidialectal Parallel Corpus of Arabic (Bouamor et al., 2014); Multi-Dialect, Multi-Genre Corpus of Informal Written Arabic (Cotterell and Callison-Burch, 2014); YouDACC (Salama et al., 2014); PADIC (Meftouh et al., 2015); Curras (Jarrar et al., 2016); WERd (Ali et al., 2017); LDC2017T07-BOLT Egyptian SMS (Chen et al., 2017); Shami (Abu Kwaik et al., 2018); SUAR (Al-Twairesh et al., 2018); Arap-Tweet (Zaghouani and Charfi, 2018); Gumar (Khalifa et al., 2018); MADAR (Bouamor et al., 2018); Habibi (El-Haj, 2020); NADI (Abdul-Mageed et al., 2020b); and QADI (Abdelali et al., 2020).

CA Training Data For CA, we use the OpenITI corpus (v1.2) (Nigst et al., 2020).

3.2 Pre-processing

After extracting the raw text from each corpus, we apply the following pre-processing. We first remove invalid characters and normalize white spaces using the utilities provided by the original BERT implementation. We also remove lines without any Arabic characters. We then remove diacritics and kashida using CAMeL Tools (Obeid et al., 2020). Finally, we split each line into sentences with a heuristic-based sentence segmenter.

3.3 Preparing Data for BERT Pre-training

We follow the original English BERT model's hyperparameters for pre-training. We train a WordPiece (Schuster and Nakajima, 2012) tokenizer on the entire dataset (167 GB text) with a vocabulary size of 30,000 using Hugging Face's tokenizers.[4] We do not lowercase letters nor strip accents. We use whole word masking and a duplicate factor of 10. We set maximum predictions per sequence to 20 for the datasets with a maximum sequence length of 128 tokens and 80 for the datasets with a maximum sequence length of 512 tokens.

3.4 Pre-training Procedure

We use a Google Cloud TPU (v3-8) for model pre-training. We use a learning rate of 1e-4 with a warmup over the first 10,000 steps. We pre-trained our models with a batch size of 1,024 sequences with a maximum sequence length of 128 tokens for the first 900,000 steps. We then continued pre-training with a batch size of 256 sequences with a maximum sequence length of 512 tokens for another 100,000 steps. In total, we pre-trained our models for one million steps. Pre-training one model took approximately 4.5 days.

4 Fine-tuning Tasks

We evaluate our pre-trained language models on five NLP tasks: NER, POS tagging, sentiment analysis, dialect identification, and poetry classification. Specifically, we fine-tune and evaluate the models using 12 datasets (corresponding to 12 subtasks). We used Hugging Face's transformers (Wolf et al., 2020) to fine-tune our CAMeLBERT models.[5] The fine-tuning was done by adding a fully connected linear layer to the last hidden state.

[2] https://github.com/google-research/bert

[3] https://archive.org/details/arwiki-20190201

[4] https://github.com/huggingface/tokenizers

[5] We used transformers v3.1.0 along with PyTorch v1.5.1

Task	Dataset/Subtask	#Label	#Train	#Test	Unit	Variant	%MSA
NER	ANERcorp (Benajiba et al., 2007)	9	125,102	25,008	Token	MSA	83.6
POS	PATB (MSA) (Maamouri et al., 2004)	32	503,015	63,172	Token	MSA	85.1
	ARZTB (EGY) (Maamouri et al., 2012)	33	133,751	20,464	Token	DA	21.5
	Gumar (GLF) (Khalifa et al., 2018)	35	162,031	20,100	Token	DA	30.0
SA	ASTD (Nabil et al., 2015)	3	23,327	663	Sent	MSA	56.9
	ArSAS (Elmadany et al., 2018)	3	23,327	3,705	Sent	MSA	60.5
	SemEval (Rosenthal et al., 2017)	3	23,327	6,100	Sent	MSA	77.7
DID	MADAR-26 (Salameh et al., 2018)	26	41,600	5,200	Sent	DA	14.1
	MADAR-6 (Salameh et al., 2018)	6	54,000	12,000	Sent	DA	17.2
	MADAR-Twitter-5 (Bouamor et al., 2019)	21	39,836	9,116	Grp	MSA	92.3
	NADI (Abdul-Mageed et al., 2020b)	21	21,000	5,000	Sent	DA	38.3
Poetry	APCD (Yousef et al., 2019)	23	1,391,541	173,963	Sent	CA	71.3

Table 2: Statistics of our fine-tuning datasets. Unit refers to a unit we use to calculate the number of examples in Train and Test. For MADAR-Twitter-5, we use a group of five tweets as a unit (Grp). A language variant is determined based on dataset design and the estimated proportion of MSA sentences in the dataset.

Tasks and Variants We selected the fine-tuning datasets and subtasks to represent multiple variants of Arabic by design. For some of the datasets, the variant is readily known. However, other datasets contain a lot of social media text where the dominant variant of Arabic is unknown. Therefore, we estimate the proportion of MSA sentences in each dataset by identifying whether the text is MSA or DA using the Corpus 6 dialect identification model in Salameh et al. (2018) as implemented in CAMeL Tools (Obeid et al., 2020). This technique does not model CA. Of course, none of the datasets was purely MSA or DA; however, based on known dataset variants, we observe that having about 40% or fewer MSA labels strongly suggests that the dataset is dialectal (or a strong dialectal mix).

Table 2 presents the number of labels, size, unit, variant, and MSA percentage for the datasets used in the subtasks.

4.1 Named Entity Recognition

Dataset We fine-tuned our models on the publicly available Arabic NER Dataset ANERcorp (∼150K words) (Benajiba et al., 2007) which is in MSA and we followed the splits defined by Obeid et al. (2020). We also kept the same IOB (inside, outside, beginning) tagging format defined in the dataset covering four classes: Location (LOC), Miscellaneous (MISC), Organization (ORG), and Person (PERS).

Experimental Setup During fine-tuning, we used the representation of the first sub-token as an input to the linear layer. All models were fine-tuned on a single GPU for 3 epochs with a learning rate of 5e-5, batch size of 32, and a maximum sequence length of 512. Since ANERcorp does not have a dev set, we used the last checkpoint after the fine-tuning is done to report results on the test set using the F_1 score.

4.2 Part-of-Speech Tagging

Dataset We fine-tuned our models on three different POS tagging datasets: (1) the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) which is in MSA and includes 32 POS tags; (2) the Egyptian Arabic Treebank (ARZATB) (Maamouri et al., 2012) which is in Egyptian (EGY) and includes 33 POS tags; and (3) the GUMAR corpus (Khalifa et al., 2018) which is in Gulf (GLF) and includes 35 POS tags.

Experimental Setup Similar to NER, we used the representation of the first sub-token as an input to the linear layer. Our models were fine-tuned on a single GPU for 10 epochs with a learning rate of 5e-5, batch size of 32, and a maximum sequence length of 512. We used the same hyperparameters for the fine-tuning across the three POS tagging datasets. After the fine-tuning, we used the best checkpoints based on the dev sets to report results on the test sets using the F_1 score.

4.3 Sentiment Analysis

Dataset We used a combination of sentiment analysis datasets to fine-tune our models. The datasets are: (1) the Arabic Speech-Act and Sentiment Corpus of Tweets (ArSAS) (Elmadany et al., 2018); (2) the Arabic Sentiment Tweets Dataset (ASTD) (Nabil et al., 2015); (3) SemEval-2017 task 4-A benchmark dataset (Rosenthal et al., 2017); and (4) the Multi-Topic Corpus for Target-based Sentiment Analysis in Arabic Levantine Tweets (ArSenTD-Lev) (Baly et al., 2019). We combined and preprocessed the datasets in a similar way to what was done by Abu Farha and Magdy (2019) and Obeid et al. (2020). That is, we removed diacritics, URLs, and Twitter usernames from all the tweets.

Experimental Setup Our models were fine-tuned on ArSenTD-Lev and the train splits from SemEval, ASTD, and ArSAS (23,327 tweets) on a single GPU for 3 epochs with a learning rate of 3e-5, batch size of 32, and a maximum sequence length of 128. After the fine-tuning, we used the best checkpoint based on a single dev set from SemEval, ASTD, and ArSAS to report results on the test sets. We used the F_1^{PN} score which was defined in the SemEval-2017 task 4-A; F_1^{PN} is the macro F_1 score over the positive and negative classes only while neglecting the neutral class.

4.4 Dialect Identification

Dataset We fine-tuned our models on four different dialect identification datasets: (1) MADAR Corpus 26 which includes 26 labels; (2) MADAR Corpus 6 which includes six labels; (3) MADAR Twitter Corpus (Bouamor et al., 2018; Salameh et al., 2018; Bouamor et al., 2019) which includes 21 labels; and (4) NADI Country-level (Abdul-Mageed et al., 2020b) which includes 21 labels. The datasets were preprocessed by removing diacritics, URLs, and Twitter usernames while maintaining the same train, dev, and test splits for each dataset. Moreover, we collated the tweets belonging to a particular user in the MADAR Twitter Corpus in groups of 5 before feeding them to the model. We refer to this preprocessed version as MADAR-Twitter-5 to avoid confusion with the publicly available original MADAR Twitter Corpus.

Experimental Setup Our models were fine-tuned for 10 epochs with a learning rate of 3e-5, batch size of 32, and a maximum sequence length of 128. After the fine-tuning, we used the best checkpoints based on the dev sets to report results on the test sets using the F_1 score. Moreover, for the MADAR-Twitter-5 evaluation, we took a voting approach. That is, each user in the dev and test sets is assigned to the most frequent predicted country label. In case of a tie, we always pick the most frequent predicted country label based on the training set.

4.5 Poetry Meter Classification

Dataset We used the Arabic Poem Comprehensive Dataset (APCD) (Yousef et al., 2019), which is mostly in CA, to fine-tune our models to identify the meters of Arabic poems. The dataset contains around 1.8M poems and covers 23 meters. We preprocessed the dataset by removing diacritics from the poems and separated the halves of each verse by using the `[SEP]` token. We applied an 80/10/10 random split to create train, dev, and test sets respectively.

Experimental Setup We fine-tuned our models on a single GPU for 3 epochs with a learning rate of 3e-5, batch size of 32, and a maximum sequence length of 128. After the fine-tuning, we used the best checkpoint based on the dev set to report results on the test set using the F_1 score.

5 Evaluation Results and Discussion

We first present an experiment where we investigate the effect of pre-training data size. We then report on CAMeLBERT models pre-trained on MSA, DA, and CA data, in addition to a model that is pre-trained on a mixture of these variants. We then provide a comparison against publicly available models.

5.1 Models with Different Data Sizes

To investigate the effect of pre-training data size on fine-tuning tasks, we pre-train MSA models in a controlled setting where we scale down the MSA pre-training size by a factor of two while keeping all other hyperparameters constant. We pre-train four CAMeLBERT models on MSA data as follows: MSA-1/2 (54GB, 6.3B words), MSA-1/4 (27GB, 3.1B words), MSA-1/8 (14GB, 1.5B words), and MSA-1/16 (6GB, 636M words). In Table 3, we show the results on our fine-tuning subtasks.

We observe that the full MSA model is on average the highest performing system by a slight margin, and it wins or ties for a win in seven out of 12 subtasks. It is also the best model on average in the MSA and DA subtasks. However, we note that

			%Performance					
Task	Subtask	Variant	MSA (107GB)	MSA-1/2 (53GB)	MSA-1/4 (27GB)	MSA-1/8 (14GB)	MSA-1/16 (6GB)	Max-Min
NER	ANERcorp	MSA	**82.4**	82.3	82.0	82.3	80.5	1.9
POS	PATB (MSA)	MSA	**97.4**	**97.4**	**97.4**	**97.4**	**97.4**	0.1
	ARZTB (EGY)	DA	**90.8**	90.3	90.5	90.5	90.4	0.5
	Gumar (GLF)	DA	**97.1**	97.0	97.0	**97.1**	97.0	0.1
SA	ASTD	MSA	**76.9**	76.0	76.8	76.7	75.3	1.6
	ArSAS	MSA	**93.0**	92.6	92.5	92.5	92.3	0.8
	SemEval	MSA	72.1	70.7	**72.8**	71.6	71.2	2.0
DID	MADAR-26	DA	62.6	62.0	**62.8**	62.0	62.2	0.8
	MADAR-6	DA	91.9	91.8	**92.2**	92.1	92.0	0.4
	MADAR-Twitter-5	MSA	77.6	**78.5**	77.3	77.7	76.2	2.3
	NADI	DA	**24.9**	24.6	24.6	**24.9**	23.8	1.1
Poetry	APCD	CA	79.7	79.9	**80.0**	79.7	79.8	0.3
Variant-wise-average		MSA	**83.2**	82.9	83.1	83.0	82.1	1.1
		DA	**73.5**	73.1	73.4	73.3	73.1	0.4
		CA	79.7	79.9	**80.0**	79.7	79.8	0.3
Macro-average			**78.9**	78.6	78.8	78.7	78.2	0.7

Table 3: Performance of CAMeLBERT models trained on MSA datasets with different sizes. We use F_1 score as a metric for all tasks. Max-Min refers to the difference in performance among the models for each dataset. Variant-wise-average refers to average over a group of tasks in the same language variant. The best results among the models are in bold.

the MSA-1/4 wins or ties for a win in five subtasks and performs best in the CA subtask, even though it was pre-trained on a quarter of the full MSA data.

We also observe that different subtasks have different patterns. For some subtasks, plateauing in performance happens rather early. For instance, the performance on PATB (MSA) does not change even if we increase the size. Similarly, the difference in performance on Gumar (GLF) is very small (0.1%). For other subtasks, the improvement is not consistent with the size, particularly in SemEval. When we calculate the correlation between the performance and the pre-training data size, we note that ArSAS has a strong positive correlation of 0.96, however, MADAR-6 has a negative correlation of -0.62. In fact, the average of the correlation of each of the 12 experiments is 0.37, which is not a strong pattern correlating size with performance. These observations suggest that the size of pre-training data has limited and inconsistent effect on the fine-tuning performance. This is consistent with Micheli et al. (2020), where they concluded that pre-training data size does not show a strong monotonic relationship with fine-tuning performance in their controlled experiments on French corpora.

5.2 Models with Different Language Variants

Next, we explore the relationship between language variants in pre-training and fine-tuning datasets.

5.2.1 MSA, DA, and CA

Task Type Difference We compare the behavior of three models pre-trained on MSA, DA, and CA data. From Table 4, we observe that the difference in performance (Max-Min) among CAMeLBERT's MSA, DA, and CA models is 4.9% on average, ranging from 0.7% to 16.2%. To study trends by task type, we compute the average performance difference across the subtasks for each task. NER is the most sensitive to the pre-trained model variant (16.2%), followed by sentiment analysis (8.2%), dialect identification (3.8%), poetry classification (1.3%), and POS tagging (1.3%). This indicates the importance of optimal pairing of pre-trained models and fine-tuning tasks.

On average the CAMeLBERT-MSA model performs best, and is the winner in 10 out of 12 subtasks. The following are the two exceptions: (a) the CAMeLBERT-DA model performs best in the highly dialectal MADAR-6 subtask; and (b) the CAMeLBERT-CA model outperforms other models in the poetry classification task, which is

Task	Dataset	Variant	%Performance						%OOV			
			Star	Mix	MSA	DA	CA	Max-Min	Mix	MSA	DA	CA
NER	ANERcorp	MSA	**82.4**	80.2	**82.4**	74.2	66.2	16.2	0.2	0.2	1.4	4.2
POS	PATB (MSA)	MSA	**97.4**	97.3	**97.4**	96.5	96.6	1.0	0.2	0.2	0.9	3.0
	ARZTB (EGY)	DA	90.1	90.1	**90.8**	89.4	88.6	2.2	0.6	0.8	1.0	7.3
	Gumar (GLF)	DA	**97.3**	**97.3**	97.1	97.0	96.5	0.7	0.2	0.8	0.3	5.4
SA	ASTD	MSA	**76.9**	76.3	**76.9**	74.6	69.4	7.5	0.9	1.1	1.2	5.3
	ArSAS	MSA	**93.0**	92.7	**93.0**	91.8	89.4	3.6	1.3	1.5	1.8	7.4
	SemEval	MSA	**72.1**	69.0	**72.1**	68.4	58.5	13.6	1.9	2.1	2.4	6.6
DID	MADAR-26	DA	**62.9**	**62.9**	62.6	61.8	61.9	0.8	0.4	0.8	0.8	7.5
	MADAR-6	DA	**92.5**	**92.5**	91.9	92.2	91.5	0.7	0.1	1.1	0.2	8.1
	MADAR-Twitter-5	MSA	**77.6**	75.7	**77.6**	74.2	71.4	6.2	2.4	2.6	3.0	6.7
	NADI	DA	24.7	24.7	**24.9**	20.1	17.3	7.6	1.6	2.0	2.4	8.1
Poetry	APCD	CA	**80.9**	79.8	79.7	79.6	**80.9**	1.3	0.4	1.1	2.7	0.9
Variant-wise-average		MSA	**83.2**	81.9	**83.2**	79.9	75.3	8.0	1.2	1.3	1.8	5.5
		DA	**73.5**	**73.5**	**73.5**	72.1	71.1	2.3	0.6	1.1	0.9	7.3
		CA	**80.9**	79.8	79.7	79.6	**80.9**	1.3	0.4	1.1	2.7	0.9
Macro-average			**79.0**	78.2	78.9	76.6	74.0	4.9	0.9	1.2	1.5	5.9

Table 4: Performance of CAMeLBERT models trained on MSA, DA, CA, and their Mix data. Star refers to a way of choosing CAMeLBERT models based on the language variant of the fine-tuning dataset. We use F_1 score as a metric for all tasks. Max-Min refers to the difference in performance among CAMeLBERT's MSA, DA, and CA models only. Variant-wise-average refers to average over a group of tasks in the same language variant. The best results among CAMeLBERT's MSA, DA, and CA models are underlined. The best results among CAMeLBERT's MSA, DA, CA, Mix, and Star are in bold. The OOV rate for each dataset is calculated based on the data used for pre-training each model. We underline the lowest OOV value per dataset.

in classical Arabic. These two exceptions suggest that performance in fine-tuning tasks may be associated with the variant proximity of the pre-training data to fine-tuning data; although we also acknowledge that CAMeLBERT-MSA's data is two times the size of CAMeLBERT-DA's, and 18 times the size of CAMeLBERT-CA's, which may give CAMeLBERT-MSA an advantage.

OOV Effect To further investigate the effect of variant proximity on performance, we compute the word out-of-vocabulary (OOV) rate of all fine-tuning test sets against the pre-training data, as a way to estimate their similarity.[6] Note that CAMeLBERT-Mix, where we concatenate MSA, DA, and CA pre-training data, has the lowest OOV rate by design. In Table 4, we show the OOV rates for each dataset.

In all the cases except Gumar (11 out of 12), we obtain the best performance where the model has the lowest OOV rate. To better understand the relationship between fine-tuning performance and OOV rates, we assessed the correlation between model performance and OOV rates for each dataset. We found a strong negative correlation of -0.82 on average. Interestingly, the CAMeLBERT-CA model which was pre-trained only on 6 GB of data outperforms other models that are pre-trained on significantly larger data in the poetry classification task. It is also worth mentioning that the CAMeLBERT-CA model has the lowest OOV rate on the poetry dataset (0.9%), while having access to approximately 18 times less data compared to the CAMeLBERT-MSA model (6GB vs 107GB). This again suggests that the variant proximity of pre-training data to fine-tuning data is more important than the size of pre-training data.

5.2.2 Mix of MSA, DA, and CA

To further study the interplay of language variants and pre-training data size, we pre-trained a model (CAMeLBERT-Mix) on the concatenation of the MSA, DA, and CA datasets. This is the largest dataset used to pre-train an Arabic language model to date. As shown in Table 4, the CAMeLBERT-Mix model improves over other models in three cases, all of which are dialectal, suggesting that the

[6] We use a simple token as a unit, where we segment text with white space and punctuation.

Task	Dataset	Variant	%Performance												
			Star	**Mix**	**MSA**	**DA**	**CA**	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
NER	ANERcorp	MSA	82.4	80.2	82.4	74.2	66.2	76.7	82.4	82.0	80.3	77.3	82.0	79.3	**83.6**
POS	PATB (MSA)	MSA	**97.4**	97.3	**97.4**	96.5	96.6	96.5	97.2	97.3	97.2	96.8	**97.4**	96.8	**97.4**
	ARZTB (EGY)	DA	90.1	90.1	90.8	89.4	88.6	88.1	89.9	**91.2**	89.8	90.1	90.6	90.3	90.5
	Gumar (GLF)	DA	**97.3**	**97.3**	97.1	97.0	96.5	96.3	97.0	**97.3**	96.9	96.7	97.1	97.0	97.0
SA	ASTD	MSA	76.9	76.3	76.9	74.6	69.4	64.5	74.2	**78.1**	69.5	71.7	72.0	77.6	72.0
	ArSAS	MSA	93.0	92.7	93.0	91.8	89.4	88.4	91.5	**93.3**	89.4	90.8	90.8	92.3	90.8
	SemEval	MSA	72.1	69.0	72.1	68.4	58.5	57.5	69.5	**72.7**	66.8	67.2	66.7	69.7	69.8
DID	MADAR-26	DA	**62.9**	**62.9**	62.6	61.8	61.9	60.4	61.9	62.2	59.6	59.6	59.5	61.9	59.3
	MADAR-6	DA	**92.5**	**92.5**	91.9	92.2	91.5	90.8	91.9	92.3	90.6	90.8	90.4	91.1	90.4
	MADAR-Twitter-5	MSA	77.6	75.7	77.6	74.2	71.4	71.8	**79.0**	**79.0**	77.0	76.5	74.4	77.2	74.9
	NADI	DA	24.7	24.7	24.9	20.1	17.3	16.7	21.1	24.5	23.5	25.4	20.5	**26.1**	24.2
Poetry	APCD	CA	**80.9**	79.8	79.7	79.6	**80.9**	78.8	79.6	79.9	75.1	73.6	74.5	74.2	74.9
Variant-wise-average		MSA	83.2	81.9	83.2	79.9	75.3	75.9	82.3	**83.7**	80.0	80.1	80.6	82.2	81.4
		DA	**73.5**	**73.5**	**73.5**	72.1	71.1	70.5	72.4	**73.5**	72.1	72.5	71.6	73.3	72.3
		CA	**80.9**	79.8	79.7	79.6	**80.9**	78.8	79.6	79.9	75.1	73.6	74.5	74.2	74.9
Macro-average			79.0	78.2	78.9	76.6	74.0	73.9	77.9	**79.1**	76.3	76.4	76.3	77.8	77.1

Table 5: Performance of CAMeLBERT models and other existing models. We use F_1 score as a metric for all the tasks. Star refers to a way of choosing CAMeLBERT models based on the language variant of the fine-tuning dataset. $X_1, \cdots, X_8$ corresponds to the models in Table 1. Variant-wise-average refers to average over a group of tasks in the same language variant. The best results among the models are in bold.

CAMeLBERT-Mix model does better in some dialectal context. However, we do not see an increase in performance in other cases when compared with the best performing model, although the size of the pre-training data and the variety of the data are increased. This suggests that having a wide language variety in pre-training data can be beneficial for DA subtasks, whereas variant proximity of pre-training data to fine-tuning data is important MSA and CA subtasks.

5.2.3 Selecting an Optimal Model

Taking these insights into consideration, one cannot help but consider the exciting possibility of a system-selection ensembling approach that can help users make decisions with reasonable expectations using what they know of their specific tasks. We outline here such a setup: the user has access to three versions of the models: CAMeLBERT's CA, MSA, and Mix. If the task data is known a priori to be CA, then we select the CAMeLBERT-CA model; if the task data is known to be MSA, we select the CAMeLBERT-MSA model; otherwise, we use the CAMeLBERT-Mix model (for dialects, i.e.). We report on this model in Table 4 and 5 as CAMeLBERT-Star.

It is noteworthy that this model is not the same as oracularly selecting the best performer among our four models (MSA, DA, CA, and Mix). In fact, it is lower in performance than such oracular system as the CAMeLBERT-MSA model performs better than CAMeLBERT-Mix model in ARZTB and NADI. We do not claim here that this is a foolproof method; however, it is an interesting candidate for *common wisdom* of the kind we are hoping to develop through this effort.

5.3 Comparison with Existing Models

Table 5 compares our work with other existing models. We do not use models that require morphological pre-tokenization to allow direct comparison, and also because existing tokenization systems are mostly focused on MSA or EGY (Pasha et al., 2014; Abdelali et al., 2016; Obeid et al., 2020).

We are aware that design decisions such as vocabulary size and number of training steps are not the same across these eight existing pre-trained models, which might be a contributing factor to their varying performances. We plan to investigate the effects of such decisions in future work.

Task Performance Complementarity The best model on average is AraBERTv02 (X_3); it wins or ties for a win in six out of 12 subtasks (four MSA and two DA). Our CAMeLBERT-Star is second overall on average, and it wins or ties for a win in five out of 12 subtasks (three DA, one MSA, one CA). Interestingly, the two systems are complementary in their performance and between the two they win or tie for a win in 10 out of 12 subtasks. The two remaining subtasks are uniquely won by MARBERT (X_7) (NADI, DA), and ARBERT (X_8) (ANERcorp, MSA). In practice, such complementarity can be exploited by system developers to achieve higher overall performance.

Size and Performance Considering the data size and performance of the other pre-trained models (X_1 to X_8), we observe a similar trend to our CAMeLBERT models. AraBERTv02 (X_3) is the best on average, with only 77GB of pre-training data. AraBERTv01 (X_2) is the smallest (24GB); however, on average it outperforms other models pre-trained on much larger datasets, such as MARBERT (X_7, 128GB), and ArabicBERT (X_4, 95GB). This confirms that pre-training data size may not be an important factor to fine-tuning performance, as we showed in Section 5.1.

Variant Proximity and Performance When we examine the proximity in terms of language variants of the pre-training data and the fine-tuning data across the eight existing pre-trained models, we observe the following. First, the monolingual MSA models (X_2, X_3, X_4, X_8) are better performers than the mixed models (X_5, X_7) on average (77.6% and 77.1%, respectively).[7] Second, the monolingual MSA models perform better than the mixed models in MSA subtasks on average (81.9% and 81.1%, respectively), while the mixed models perform better than the MSA models in DA subtasks on average (72.9% and 72.6%, respectively).[8] This result is consistent with our analysis of the CAMeLBERT-Mix and the CAMeLBERT-MSA models in Section 5.2, where we found that the CAMeLBERT-Mix model is the best choice for DA subtasks, whereas the CAMeLBERT-MSA model is the best in MSA subtasks.

On MARBERT and ARBERT In another study that compared models pre-trained on MSA alone or a mix of MSA and DA data, Abdul-Mageed et al. (2020a) reported that MARBERT (X_7, pre-trained on MSA-DA mix) is more powerful than ARBERT (X_8, pre-trained on MSA). In our study, we do replicate their specific relative performance in terms of macro-average in our experiments (77.8% for MARBERT and 77.1% for ARBERT). It is not clear why MARBERT and ARBERT do not exhibit similar trends as observed in the analysis of our own CAMeLBERT models and other existing models. This may be attributed to numerous factors such as the degree of MSA-DA mixture, genre, and the pre-training procedure details. It is also worth noting that the data used to pre-train our CAMeLBERT-MSA model is a subset of the data used to pre-train our CAMeLBERT-Mix model, whereas the pre-training data for MARBERT and ARBERT are derived from different data sources.

6 Conclusion and Future Work

In this paper, we investigated the interplay of size, language variant, and fine-tuning task type in Arabic pre-trained language models using carefully controlled experiments on a number of Arabic NLP tasks. Our results show that pre-training data and subtask data variant proximity is more important than pre-training data size. We confirm these results on existing models. We exploit this insight in defining an optimized system selection model for the studied tasks. We make all of our created models and fine-tuning code publicly available.

In future work, we plan to explore other design decisions that may contribute to the fine-tuning performance, including vocabulary size, tokenization techniques, and additional data mixtures. We also plan to utilize CAMeLBERT models in a number of other Arabic NLP tasks, and integrate them in the open-source toolkit, CAMeL Tools (Obeid et al., 2020).

Acknowledgment

This research was supported with Cloud TPUs from Google's TensorFlow Research Cloud (TFRC). This work was also carried out on the High Performance Computing resources at New York University Abu Dhabi. The first and second authors were supported by the New York University Abu Dhabi Global PhD Student Fellowship program. We thank Salam Khalifa, and Ossama Obeid for helpful discussions. We also thank the anonymous reviewers for their valuable comments.

[7] The average over macro-average performances.

[8] The average over variant-wise-average performances.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 11–16, San Diego, California.

Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2020. Arabic dialect identification in the wild.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020a. ARBERT & MARBERT: Deep bidirectional transformers for Arabic.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020b. NADI 2020: The First Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop (WANLP 2020)*, Barcelona, Spain.

Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy. Association for Computational Linguistics.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. Shami: A corpus of Levantine Arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Nora Al-Twairesh, Rawan Al-Matham, Nora Madi, Nada Almugren, Al-Hanouf Al-Aljmi, Shahad Alshalan, Raghad Alshalan, Nafla Alrumayyan, Shams Al-Manea, Sumayah Bawazeer, Nourah Al-Mutlaq, Nada Almanea, Waad Bin Huwaymil, Dalal Alqusair, Reem Alotaibi, Suha Al-Senaydi, and Abeer Alfutamani. 2018. SUAR: Towards building a corpus for the Saudi dialect. In *Proceedings of the International Conference on Arabic Computational Linguistics (ACLing)*.

Ahmed Ali, Preslav Nakov, Peter Bell, and Steve Renals. 2017. WERD: using social text spelling variants for evaluating dialectal speech recognition. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*, pages 141–148. IEEE.

Khalid Almeman and Mark Lee. 2013. Automatic building of Arabic multi dialect text corpora by bootstrapping dialect words. In *Proceedings of the International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–6.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Appen Pty Ltd. 2006a. Gulf Arabic Conversational Telephone Speech, Transcripts LDC2006T15.

Appen Pty Ltd. 2006b. Iraqi Arabic Conversational Telephone Speech, Transcripts LDC2006T16.

Appen Pty Ltd. 2007. Levantine Arabic Conversational Telephone Speech, Transcripts LDC2007T01.

Ramy Baly, Alaa Khaddaj, Hazem Hajj, Wassim El-Hajj, and Khaled Bashir Shaban. 2019. ArSentD-LEV: A multi-topic corpus for target-based sentiment analysis in Arabic Levantine tweets.

BBN Technologies. 2005. BBN/AUB DARPA Babylon Levantine Arabic Speech and Transcripts LDC2005S08.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedí Ruiz. 2007. ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153, Berlin, Heidelberg. Springer Berlin Heidelberg.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multidialectal parallel corpus of Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Reykjavik, Iceland.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR Arabic Dialect Corpus and Lexicon. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR shared task on Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207, Florence, Italy. Association for Computational Linguistics.

Song Chen, Dana Fore, Stephanie Strassel, Haejoong Lee, and Jonathan Wright. 2017. BOLT Egyptian Arabic SMS/Chat and Transliteration LDC2017T07.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Ryan Cotterell and Chris Callison-Burch. 2014. A Multi-Dialect, Multi-Genre Corpus of Informal Written Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 241–245, Reykjavik, Iceland.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Mahmoud El-Haj. 2020. Habibi - a multi dialect multi national Arabic song lyrics corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1318–1326, Marseille, France. European Language Resources Association.

Ibrahim Abu El-Khair. 2016. 1.5 billion words Arabic corpus. *CoRR*, abs/1611.04033.

AbdelRahim Elmadany, Hamdy Mubarak, and Walid Magdy. 2018. ArSAS: An Arabic speech-act and sentiment corpus of tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Hassan Gadalla, Hanaa Kilany, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, Amr Shalaby, Krisjanis Karins, Everett Rowson, Robert MacIntyre, Paul Kingsbury, David Graff, and Cynthia McLemore. 1997. CALLHOME Egyptian Arabic transcripts LDC97T19. Web Download. Philadelphia: Linguistic Data Consortium.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2016. Curras: an annotated corpus for the Palestinian Arabic dialect. *Language Resources and Evaluation*, pages 1–31.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 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.

Salam Khalifa, Nizar Habash, Fadhl Eryani, Ossama Obeid, Dana Abdulrahim, and Meera Al Kaabi. 2018. A morphologically annotated corpus of Emirati Arabic. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, and Joakim Nivre. 2020. Do neural language models show preferences for syntactic formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4077–4091, Online. Association for Computational Linguistics.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. An empirical study of pre-trained transformers for Arabic information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4727–4734, Online. Association for Computational Linguistics.

Linguistic Data Consortium. 2002. CALLHOME Egyptian Arabic Transcripts Supplement LDC2002T38.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Mohamed Maamouri, Ann Bies, Seth Kulick, Dalila Tabessi, and Sondos Krouna. 2012. Egyptian Arabic Treebank DF Parts 1-8 V2.0 - LDC catalog numbers LDC2012E93, LDC2012E98, LDC2012E89, LDC2012E99, LDC2012E107, LDC2012E125, LDC2013E12, LDC2013E21.

Mohamed Maamouri, Tim Buckwalter, David Graff, and Hubert Jin. 2006. Levantine Arabic QT Training Data Set 5, Transcripts LDC2006T07.

Mohamed Maamouri, Tim Buckwalter, David Graff, and Hubert Jin. 2007. Fisher Levantine Arabic Conversational Telephone Speech, Transcripts LDC2007T04.

Mohamed Maamouri, Tim Buckwalter, and Hubert Jin. 2005. Levantine Arabic QT Training Data Set 4 (Speech + Transcripts) LDC2005S14.

Karima Meftouh, Salima Harrat, Salma Jamoussi, Mourad Abbas, and Kamel Smaili. 2015. Machine translation experiments on PADIC: A parallel Arabic dialect corpus. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation*.

Vincent Micheli, Martin d'Hoffschmidt, and François Fleuret. 2020. On the importance of pre-training data volume for compact language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7853–7858, Online. Association for Computational Linguistics.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. ASTD: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519, Lisbon, Portugal. Association for Computational Linguistics.

Lorenz Nigst, Maxim Romanov, Sarah Bowen Savant, Masoumeh Seydi, and Peter Verkinderen. 2020. OpenITI: a Machine-Readable Corpus of Islamicate Texts.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Robert Parker, David Graff, Ke Chen, Junbo Kong, and Kazuaki Maeda. 2011. Arabic Gigaword Fifth Edition. LDC catalog number No. LDC2011T11, ISBN 1-58563-595-2.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 1094–1101, Reykjavik, Iceland.

Raytheon BBN Technologies, Linguistic Data Consortium, and Sakhr Software. 2012. Arabic-Dialect/English Parallel Text LDC2012T09.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 501–516, Vancouver, Canada.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2020. How good is your tokenizer? on the monolingual performance of multilingual language models.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Ahmed Salama, Houda Bouamor, Behrang Mohit, and Kemal Oflazer. 2014. YouDACC: the Youtube Dialectal Arabic Comment Corpus. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 1246–1251, Reykjavik, Iceland.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained Arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *International Conference on Acoustics, Speech and Signal Processing*, pages 5149–5152.

Bashar Talafha, Mohammad Ali, Muhy Eddin Za'ter, Haitham Seelawi, Ibraheem Tuffaha, Mostafa Samir, Wael Farhan, and Hussein Al-Natsheh. 2020. Multi-dialect Arabic BERT for country-level dialect identification. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 111–118, Barcelona, Spain (Online). Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Waleed A. Yousef, Omar M. Ibrahime, Taha M. Madbouly, and Moustafa A. Mahmoud. 2019. Learning meters of Arabic and English poems with recurrent neural networks: a step forward for language understanding and synthesis.

Wajdi Zaghouani and Anis Charfi. 2018. ArapTweet: A Large Multi-Dialect Twitter Corpus for Gender, Age and Language Variety Identification. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Omar F Zaidan and Chris Callison-Burch. 2011. The Arabic Online Commentary Dataset: an Annotated Dataset of Informal Arabic With High Dialectal Content. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 37–41.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182, Florence, Italy. Association for Computational Linguistics.

Yian Zhang, Alex Warstadt, Haau-Sing Li, and Samuel R. Bowman. 2020. When do you need billions of words of pretraining data?