Morphosyntactic Tagging with Pre-trained Language Models for Arabic and its Dialects

Anonymous ACL submission

Abstract

We present state-of-the-art results on morphosyntactic tagging across different varieties of Arabic using fine-tuned pre-trained transformer language models. Our models consistently outperform existing systems in Modern Standard Arabic and all the Arabic dialects we study, achieving 2.6% absolute improvement over the previous state-of-the-art in Modern Standard Arabic, 2.8% in Gulf, 1.6% in Egyptian, and 8.3% in Levantine. We explore different training setups for fine-tuning pre-trained transformer language models, including training data size, the use of external linguistic resources, and the use of annotated data from other dialects in a low-resource scenario. Our results show that strategic fine-tuning using datasets from other high-resource dialects is beneficial for a low-resource dialect. Additionally, we show that high-quality morphological analyzers as external linguistic resources are beneficial especially in low-resource settings.

1 Introduction

Fine-tuning pre-trained language models like BERT (Devlin et al., 2019) have achieved great success in a wide variety of natural language processing (NLP) tasks, e.g. sentiment analysis (Abu Farha et al., 2021), question answering (Antoun et al., 2020), and named entity recognition (Ghaddar et al., 2022), and dialect identification (Abdelali et al., 2021). Pre-trained LMs have also been used for enabling technologies such as part-of-speech (POS) tagging (Lan et al., 2020; Khalifa et al., 2021; Inoue et al., 2021), to produce features for downstream processes. Previous POS tagging results using pre-trained LMs focused on core POS tagsets; however, it is still not clear how these models perform on the full morphosyntactic tagging task of very morphologically rich languages, where the size of the full tagset can be in the thousands. One such language is Arabic, where lemmas inflect to a large number of forms through different combinations of morphological features and cliticization. Additionally, Arabic orthography omits the vast majority of its optional diacritical marks which increases morphosyntactic ambiguity.

A third challenge for Arabic is its numerous variants. Modern Standard Arabic (MSA) is the primarily written variety used in formal settings. Dialectal Arabic (DA), by contrast, is the primarily spoken unstandardized variant. MSA and different DAs, e.g., Gulf (GLF), Egyptian (EGY), and Levantine (LEV), vary in terms of their grammar and lexicon to the point of impeding usability cross-dialectally (Habash et al., 2012). Furthermore, these variants differ in the degree of data availability: MSA is the highest resourced variant, followed by GLF and EGY, and then LEV.

In this paper, we explore different training setups for fine-tuning Arabic pre-trained language models in the complex morphosyntactic tagging task for four Arabic variants (MSA, GLF, EGY, and LEV) under controlled experimental settings.

We aim to answer the following questions:

- How does the size of the fine-tuning data affect the performance?
- What kind of tagset scheme is suitable for modeling morphosyntactic features?
- Is there any additional value of using external linguistic resources?
- How can we make use of annotated data in other dialects to improve performance in a low-resourced dialect?

Our system[1] achieves state-of-the-art (SOTA) performance in full morphosyntactic tagging accuracy in all the variants we study, resulting in 2.6% absolute improvement over previous SOTA in MSA, 2.8% in GLF, 1.6% in EGY, and 8.3% in LEV.

[1] We will make our models and data publicly available.

	diac	lex	gloss	pos	prc3	prc2	prc1	prc0	per	gen	num	asp	vox	mod	stt	cas	enc0	Variant
(a)	حَفِيدَكَ *Hafiydaka*	حَفِيد *Hafiyd*	grandchild	noun	-	-	-	-	-	m	s	-	-	-	c	a	2ms_poss	MSA
(b)	حَفِيدَكِ *Hafiydaki*	حَفِيد *Hafiyd*	grandchild	noun	-	-	-	-	-	m	s	-	-	-	c	a	2fs_poss	MSA
(c)	حَفِيدُكَ *Hafiyduka*	حَفِيد *Hafiyd*	grandchild	noun	-	-	-	-	-	m	s	-	-	-	c	n	2ms_poss	MSA
(d)	حَفِيدُكِ *Hafiyduki*	حَفِيد *Hafiyd*	grandchild	noun	-	-	-	-	-	m	s	-	-	-	c	n	2fs_poss	MSA
(e)	حَفِيدِكَ *Hafiydika*	حَفِيد *Hafiyd*	grandchild	noun	-	-	-	-	-	m	s	-	-	-	c	g	2ms_poss	MSA
(f)	حَفِيدِكِ *Hafiydiki*	حَفِيد *Hafiyd*	grandchild	noun	-	-	-	-	-	m	s	-	-	-	c	g	2fs_poss	MSA
(g)	حَفِيدِك *Hafiydik*	حَفِيد *Hafiyd*	grandchild	noun	-	-	-	-	-	m	s	-	-	-	c	-	2ms_poss	GLF
(h)	حَفِيدَك *Hafiydak*	حَفِيد *Hafiyd*	grandchild	noun	-	-	-	-	-	m	s	-	-	-	c	-	2ms_poss	EGY,LEV
(i)	حَفِيدِك *Hafiydik*	حَفِيد *Hafiyd*	grandchild	noun	-	-	-	-	-	m	s	-	-	-	c	-	2fs_poss	EGY,LEV
(j)	حَفِيدَك *Hafiydak*	فاد *fAd*	benefit	verb	-	-	-	fut	1	-	s	i	-	-	-	-	2ms_dobj	EGY,LEV
(k)	حَفِيدِك *Hafiydik*	فاد *fAd*	benefit	verb	-	-	-	fut	1	-	s	i	-	-	-	-	2fs_dobj	EGY,LEV

Table 1: This is an example of multiple readings of the word حفيدك *Hfydk* in the different variants of Arabic. The table also shows the full range of morphological features: part-of-speech (**pos**), aspect (**asp**), mood (**mod**), voice (**vox**), person (**per**), gender (**gen**), number (**num**), case (**cas**), state (**stt**) and clitics: proclitics (**prc3, prc2, prc1, prc0**) and enclitic (**enc0**). In addition to the lemma (**lex**), fully diacritized form (**diac**), and English gloss (**gloss**).

2 Arabic Language and Resources

2.1 Arabic and its Dialects

MSA is the primarily written form of Arabic used in official media communications, official documents, news, and education. In contrast, the primarily spoken varieties of Arabic are its dialects. Arabic dialects vary among themselves and can be categorized at different levels of regional classifications (Salameh et al., 2018). They are also different from MSA in most linguistic aspects (namely phonology, morphology, and syntax). Moreover, dialects have no official status despite being widely used in different means of daily communication – spoken as well as increasingly written on social media. In this work we focus on MSA, Gulf Arabic (GLF), Egyptian Arabic (EGY), and Levantine Arabic (LEV).

2.2 Orthography

In this paper, we focus on Arabic written in Arabic script for MSA and DA. An important feature of Arabic orthography is the omission of diacritical marks which are mostly used to indicate short vowels and consonantal doubling. This omission introduces ambiguity to the text, e.g., the word كتب *ktb*[2] could mean 'to write' (كَتَبَ *katab*) or 'books' (كُتُب *kutub*) among other readings.

Unlike MSA, Arabic dialects have no official standard orthography. Depending on the writer, words are sometimes spelled phonetically or closer to an MSA spelling through cognates or a mix of both. It has been found that in extreme cases a word can have more than 20 different spellings (Habash et al., 2018). This results in highly inconsistent and sparse datasets and models. The Conventional Orthography for Dialectal Arabic (CODA) (Habash et al., 2018) has been proposed and used in manual annotations of many datasets including some of those used in this paper. Ideally, the process of morphological disambiguation should take raw text as input, as this is more authentic than conventionalized spelling. We follow this principle for EGY and LEV where analyses are paired with the raw text. However, the GLF dataset analyses are linked to the CODA version only, since orthographic conventionalization was applied as an independent step during manual data annotations and there are no simple direct mappings between the raw text and the analyses (Khalifa et al., 2018).

2.3 Morphology

Arabic is a morphologically rich language where a single lemma inflects to a large number of forms through different combinations of morphological features (gender, number, person, case, state, mood, voice, aspect) and cliticization (prepositions, conjunctions, determiners, pronominal objects, and possessives). As some of the morphological features are primarily expressed with optional diacritical marks, orthographic ambiguity results in different morphological analyses, e.g., MSA can have up to 12 analyses per word (out-of-context) on average (Pasha et al., 2014). MSA and DA differ in the degree of morphological complexity, for example, MSA retains nominal case and verbal mood features; but these are absent in DA. On the other hand, many dialects take more clitics than MSA, e.g., the

[2] Arabic transliteration is presented in the HSB scheme (Habash et al., 2007).

Variant	Resource	Size	Orthography	Analyzer
MSA	PATB	629k	Standard	Manual
GLF	Gumar	202k	CODA	Automatic
EGY	ARZTB	175k	Spontaneous	Manual
LEV	Curras	57k	Spontaneous	Automatic

Table 2: An overview of the current status of the data and morphological analyzers used in this work.

ما+ +ش *mA+ +š* negation circumclitic structure found in EGY and not MSA (Habash et al., 2012).

Table 1 shows different possible readings for the word حفيدك *Hfydk* among MSA, EGY, GLF, and LEV. Rows (a) to (i) are different inflections for case or possessive pronouns or both of the lemma حَفِيد *Hafiyd* 'grandchild' for all variants. Rows (j) and (k) show different readings that are inflections of the verb lemma فَاد *fAd* 'to benefit', the inflections are for different object pronouns. Note that even between the different POS inflections words can sound and look exactly the same, this shows the degree of morphological complexity and ambiguity in Arabic and its dialects.

2.4 Resources

In this work, we use datasets that have been fully annotated for morphological features and cliticization among other lexical features such as lemmas. We use the Penn Arabic Treebank for MSA (Maamouri et al., 2004), ARZTB (Maamouri et al., 2012) for EGY, the Gumar corpus (Khalifa et al., 2018) for GLF, and the Curras corpus (Jarrar et al., 2014) for LEV. We also use morphological analyzers that provides out-of-context analyses for a given word, those analyzers provide the same set of features that are seen in the annotated data. For MSA we use the SAMA database (Graff et al., 2009), and for EGY we use CALIMA (Habash et al., 2012). Both GLF and LEV do not have morphological analyzers, instead we use automatically generated analyzers from their training data using paradigm completion as described in Eskander et al. (2013, 2016) and Khalifa et al. (2020). The quality and coverage of analyzers in general can differ depending on how they were created. Manually created analyzers (MSA and EGY in this work) tend to have a better quality and lexical coverage over automatically created ones (GLF and LEV in this work). The quality of automatically generated analyzers are also highly dependent on the quality and size of the training data used to create them.

Table 2 shows the overall state of the resources for each dialect studied in this work. In terms of the size of fully annotated corpora in tokens, MSA is approximately three times larger than GLF and EGY and 11 times larger than LEV. Both MSA and GLF have consistent orthography whereas EGY and LEV are more noisy. When it comes to external morphological analyzers, only MSA and EGY have manually created and checked morphological analyzers, while both GLF and LEV have analyzers created automatically. This contrast of resource availability allows us to study how challenging the morphosyntactic tagging task can be in different real world situations.

3 Related Work

Arabic morphological modeling proved to be useful in a number of downstream NLP tasks such as machine translation (Sadat and Habash, 2006; El Kholy and Habash, 2012) speech synthesis (Halabi, 2016), dependency parsing (Marton et al., 2013), sentiment analysis (Baly et al., 2017), and gender reinflection (Alhafni et al., 2020). We expect all of these applications and others to benefit from improvements in morphosyntactic tagging.

There have been multiple approaches to morphological modeling for Arabic. Those approaches differ depending on the target tagset (POS vs full morphology) and the availability of linguistic resources. When it comes to MSA and DA full morphological tagging, MADAMIRA (Pasha et al., 2014), trained separate SVM taggers for each morphological feature (including cliticization) and selected the most probable answer provided by an external morphological analyzer all in one step for both MSA and EGY. AMIRA (Diab et al., 2004) on the other hand used a cascading approach where it performed POS tagging after automatically segmenting the text.

A more recent similar approach to MADAMIRA was introduced by Zalmout and Habash (2017) but using a neural architecture instead. Inoue et al. (2017) presented a multitask neural architecture that jointly models individual morphological features for MSA. Zalmout and Habash (2019) extended Zalmout and Habash (2017)'s work using multitask learning and adversarial training for full morphological tagging in MSA and EGY. Similarly, Zalmout and Habash (2020) proposed an approach where they jointly model lemmas, diacritized forms, and morphosyntactic features, providing the current state-of-the-art in MSA. The same approach was used in Khalifa et al. (2020), where they focused on the effect of the size of the data and the available linguistic resources and the impact on the overall performance on morphosyntactic tagging for GLF. Zalmout (2020) provides the current state-of-the-art performance in LEV by extending Khalifa et al. (2020)'s work to LEV.

Another line of research that works with DA includes Darwish et al. (2018), where they presented a multi-dialectal CRF POS tagger, using a small set of 350 manually annotated tweets for each of EGY, GLF, LEV, and Maghrebi Arabic (Samih et al., 2017). We do not evaluate on their data because their task is defined as shallow morpheme segmentation and tagging; this is quite different from, and not easily mappable to, our task, where we disambiguate morphosyntactic features of the whole word without identifying its morpheme segments. Additionally, their tagset includes social media specific tags, such as HASH, EMOT, and MENTION, which are not in any of the large standard dataset and analyzers we study in this paper.

Pre-trained LM-based efforts in Arabic morphosyntactic tagging are relatively limited and either assume gold segmentation or only produce core POS tags. Kondratyuk (2019) leveraged the multilingual BERT model with additional word-level and character-level LSTM layers for lemmatization and morphological tagging, assuming gold segmentation. They reported the results for the SIGMORPHON 2019 Shared Task (McCarthy et al., 2019), which includes MSA. Inoue et al. (2021) reported POS tagging results in MSA, GLF, and EGY using BERT models pre-trained on Arabic text with various pre-training configurations. They do not assume pre-segmentation of the text, however, they only consider the core POS tag, rather than the fully specified morphosyntactic tag. Khalifa et al. (2021) proposed a self-training approach for core POS tagging where they iteratively improve the model by incorporating the predicted examples into the training set used for fine-tuning.

In this paper, we work with full morphosyntactic modeling on unsegmented text in four different variants of Arabic: MSA, GLF, EGY, and LEV. Furthermore, we explore the behavior of the pre-trained LM with respect to fine-tuning data size under different training setups. Given the available resources, we recognize our results' limitations in terms of applicability to different genres and styles, as well as noisy social media text and Roman script Arabic text (Darwish, 2014).

4 Methodology

4.1 Morphosyntactic Tagging with Pre-trained LMs

To obtain a fully specified morphosyntactic tag sequence, we build a classifier for each morphosyntactic feature independently, inspired by MADAMIRA. Unlike MADAMIRA where they use an SVM classifier, we use two pre-trained LM based classifiers: CAMeLBERT-Mix for DA and CAMeLBERT-MSA for MSA (Inoue et al., 2021). In selecting these pre-trained language models, we considered the results from Inoue et al. (2021) who showed that CAMeLBERT-Mix, their largest Arabic BERT model by training data size, gives the best results on DA tasks. CAMeLBERT-MSA, which outperforms CAMeLBERT-Mix on MSA tasks, is only second to AraBERT (Antoun et al., 2020), but since it was created under the same setting as CAMeLBERT-Mix, it minimizes experimental variations in our study.[3] Following the work of Devlin et al. (2019), fine-tuning the CAMeLBERT models is done by appending a linear layer on top of its architecture. We use the representation of the first sub-token as an input to the linear layer.

4.2 Factored and Unfactored Tagset

One of the challenges of morphosyntactic tagging is the large size of the full tagset due to morphological complexity of the language, where a complete single tag is a concatenation of all the morphosyntactic features. For example, MSA and EGY data have approximately 2,000 unique complete tags in the training data, whereas GLF and LEV have around 1,400 and 1,000 tags, respectively. These are not the full tagsets as there are many feature combinations that are not seen in the data.

MADAMIRA's basic approach is to use a factored feature tagset that comprises multiple tags, each representing a corresponding morphosyntactic category.[4] This approach remedies the issue of the large tagset size by dividing it into multiple sub-tagsets of small sizes, however, it may produce inconsistent tag combinations.

Alternatively, one can combine the individual tags into a single tag. This approach has the advantage of guaranteeing consistency of morphosyntac-

[3] We leave engineering optimization using other pre-trained language models to future work.

[4] For example, the tagset for MSA comprises POS (34 tags), per (4), gen (3), num (5), asp (4), vox (4), mod (5), stt (5), cas (5), prc3 (3), prc2 (9), prc1 (17), prc0 (7), enc0 (48).

tic feature combination. However, it may not be optimal in terms of tag coverage due to the large number of unseen tags in the test data in addition to the large space of classes.

To determine which approach is most suitable for modeling, we build morphosyntactic taggers with both the factored tagset and the unfactored tagset for each variant. Additionally, we explore the effect of the training data size for both settings.

4.3 Retagging via Morphological Analyzers

In previous efforts (Zalmout and Habash, 2017; Khalifa et al., 2020), it has been shown that lexical resources such as morphological analyzers can boost the performance of morphosyntactic tagging through in-context ranking of out-of-context answers provided by the analyzer.

In this work, we follow their approach, where we use the morphological analyzers as a later step after tagging with the fine-tuned pre-trained model. We use the analyzers described in Section 2.4 to provide out-of-context analyses. For each word, the analyzer may provide more than one answer.[5] The analyses are then ranked based on the unweighted sum of successful matches between the values of the predictions from the individual taggers and those provided by the analyzer. To break ties during the ranking, we take the sum of the probability of the *unfactored* feature tag and the probability of all the individual tags happening together as follows:

$$\frac{1}{2}P(t_{unfactored}) + \frac{1}{2}\prod_{m \in M} P(t_m) \qquad (1)$$

where t is the tag for the feature m and M is the set of morphosyntactic features. The probabilities are obtained through unigram models based on the respective training data split.

4.4 Merged and Continued Training

Morphosyntactic modeling for DA is especially challenging because of data scarcity. Among the datasets that we use, LEV is the least resourced variant, having 11 times less training data than MSA. Therefore, we want to investigate an optimal approach to utilize data from other variants to improve upon the performance of morphosyntactic tagging for LEV.

[5] Both the MSA and EGY analyzers provide backoff modes. We use the recommended setting by Zalmout and Habash (2017). For GLF and LEV analyzers we keep the original predictions if no answer is returned.

Split	MSA	GLF	EGY	LEV
TRAIN	478k	154k	127k	43k
TUNE	26k	8k	7k	2k
DEV	63k	20k	21k	6k
TEST	63k	20k	20k	6k
ALL	629k	202k	175k	57k

Table 3: Statistics on TRAIN, TUNE, DEV, and TEST for each variant in terms of number of words.

In this work, we experiment with the following two settings: (a) We merge all the datasets together and fine-tune a pre-trained LM on the merged datasets in a single step; and (b) Similar to Zalmout (2020), we start fine-tuning a pre-trained LM on a mix of high-resource datasets (MSA, GLF, and EGY), and then continue fine-tuning on a low-resource dataset (LEV).

5 Experiments

5.1 Experimental Settings

Data To be able to compare with previous SOTA (Zalmout and Habash, 2020, 2019; Khalifa et al., 2020; Zalmout, 2020), we follow the same conventions they used for data splits: MSA and EGY (Diab et al., 2013), GLF (Khalifa et al., 2018), and LEV (Eskander et al., 2016). In Table 3, we show the statistics of our datasets.

Fine-tuning We fine-tuned the CAMeLBERT models (Inoue et al., 2021) on each morphosyntactic tagging task. Following their recommendation, we used CAMeLBERT-MSA for MSA and CAMeLBERT-Mix for the dialects. We used Hugging Face's transformers (Wolf et al., 2020) for implementation. We trained our models for 10 epochs with a learning rate of 5e-5, a batch size of 32, and a maximum sequence length of 512. We pick the best checkpoint based on TUNE and report results on DEV and TEST from a single run.

Learning Curve To investigate the effect of fine-tuning data sizes, we randomly sample training examples on a scale of 5k, 10k, 20k, 40k, 80k, 120k, and 150k tokens. We use 150k, 120k, and 40k since they are comparable to the number of tokens in GLF, EGY, and LEV datasets, respectively. This allows us to measure the performance difference across different dialects in a controlled manner. This also gives us insight into the amount of annotated data required to achieve a certain performance, which is useful when creating annotated

Variant	Model	ALL TAGS 5k	10k	20k	40k	80k	120k	150k	480k	POS 5k	10k	20k	40k	80k	120k	150k	480k	Ortho	Morph
MSA	Unfactored	43.2	65.5	79.2	88.1	91.6	93.3	93.9	95.5	80.1	90.5	94.1	96.9	97.7	98.0	98.1	98.5	Consistent	Manual
	+Morph	63.4	77.6	85.4	91.3	93.3	94.4	94.8	95.9	81.6	91.6	95.1	97.4	98.1	98.3	98.5	98.7		
	Factored	75.3	86.1	90.8	93.0	94.1	94.7	94.9	95.5	93.0	96.4	97.6	98.1	98.3	98.3	98.4	98.6		
	+Morph	**86.5**	**91.3**	**93.6**	**94.7**	**95.2**	**95.5**	**95.7**	**96.1**	**95.1**	**97.1**	**98.0**	**98.5**	**98.6**	**98.6**	**98.7**	**98.8**		
GLF	Unfactored	75.1	81.0	89.6	93.3	**94.8**	**95.3**	**95.8**		90.3	92.6	95.6	96.8	97.2	97.7	97.8		Consistent	Auto
	+Morph	86.4	87.1	90.7	92.3	93.1	93.4	93.8		93.9	94.1	95.5	96.1	96.4	96.7	96.6			
	Factored	87.1	89.8	**92.4**	**94.0**	94.7	95.1	95.5		94.6	95.5	**96.6**	**97.1**	**97.5**	**97.9**	**98.0**			
	+Morph	**90.8**	**90.6**	92.1	92.9	93.4	93.8	93.9		**95.4**	95.5	96.0	96.3	96.6	96.8	96.8			
EGY	Unfactored	64.6	77.3	83.0	86.1	87.7	88.8			84.0	87.8	90.5	92.0	92.7	93.0			Spontaneous	Manual
	+Morph	76.4	83.8	87.4	89.2	89.9	90.5			81.9	87.9	91.5	93.1	93.7	94.0				
	Factored	77.1	82.0	84.1	85.7	86.8	87.4			89.9	91.0	92.0	92.6	92.9	93.2				
	+Morph	**86.3**	**88.3**	**89.2**	**89.8**	**90.3**	**90.6**			**90.9**	**92.6**	**93.4**	**93.7**	**94.0**	**94.1**				
LEV	Unfactored	73.6	80.8	85.0	88.1					86.7	91.0	93.1	94.5					Spontaneous	Auto
	+Morph	77.0	80.6	83.2	85.4					87.8	90.2	92.0	93.1						
	Factored	80.6	**84.6**	**86.6**	**88.9**					**91.4**	**93.2**	**94.1**	**94.7**						
	+Morph	**81.2**	83.4	84.7	86.2					90.5	91.7	92.7	93.4						

Table 4: DEV results on a learning curve of the training data size. Morph refers to the model with an additional step of retagging using a morphological analyzer. We bold the best score for each variant. Underlined scores denote that the differences between those scores and the best scores are statistically insignificant with McNemar's test ($p < 0.05$).

resources for new dialects. We use this setup in all the experimental setups.

Pre-processing for Merged and Continued Training Although the different datasets provide the same set of the morphosyntactic features, there exist some inconsistencies between them. The datasets were annotated by different groups using slightly different annotation guidelines, therefore, we need to bring all the feature values into a common space with LEV. We performed the following steps to address those inconsistencies: (a) We drop the state, case, mood, and voice features; (b) We remove the diactization from the lexical parts of the proclitic features, e.g. the conjunction w realized as *wa_conj* in MSA and *wi_conj* in EGY both maps to *w_conj* in LEV; and (c) For certain POS classes some features have default values in case they are not present, those default values were different for different datasets. Thus, we mapped those default values to match whatever was specified as default in LEV. We only performed these modifications for the experiments on merged and continued training.

Evaluation Metrics We compute the accuracy in terms of the core POS and the combined morphosyntactic features (ALL TAGS).

5.2 Results

Factored vs Unfactored Models Table 4 shows the DEV results for the models trained with the factored and unfactored tagset (henceforth, factored and unfactored models, respectively) on a learning curve of the training data size. In the extremely low-resource setting of 5k tokens in the ALL TAGS metric, we observe that factored models consistently outperform unfactored models across all the variants (15.9% absolute increase on average). In particular, MSA benefited most with 32.1% absolute increase, followed by EGY (12.5%), GLF (12.0%), and LEV (7.1%).

However, this gap shrinks as the data size increases. For instance in MSA, the differences between the scores of the factored model and the unfactored model become statistically insignificant by McNemar's test (McNemar, 1947) with $p < 0.05$ when trained on the full data. This is presumably due to the decrease in the number of unseen unfactored tags in DEV. In fact, 3.9% of the unfactored tags in DEV are not seen in TRAIN in the 5k setting, whereas only 0.1% of tags are unseen in DEV when we use the full data.

The factored model performs better than the unfactored model across all the data sizes in MSA and LEV. The EGY and GLF models follow a similar pattern in the low resourced settings, however, the unfactored models begin to perform better than the factored ones from 20k for EGY and 40k for GLF. Our results suggest that the factored tagset is optimal compared to the unfactored tagset, especially in low-resource settings.

	ALL TAGS				POS			
	5k	10k	20k	40k	5k	10k	20k	40k
SINGLE	81.5	85.4	87.4	89.2	91.4	93.2	94.1	94.7
MERGED	77.9	80.6	82.7	85.0	87.3	89.4	90.9	92.3
CONTINUED	**85.1**	**86.9**	**88.2**	**89.5**	**92.0**	**93.3**	**94.2**	**94.8**

Table 5: DEV results on LEV for the merged training setup (MERGED) and the continued training setup (CONTINUED). SINGLE refers to the model trained only on LEV.

Retagging with Morphological Analyzer We observe that the use of a morphological analyzer consistently improves performance of both unfactored and factored models across all the different training data sizes in MSA and EGY in ALL TAGS. The value of a morphological analyzer is especially apparent in the very low resourced setting (5k), with an increase of 20.2% (MSA) and 11.8% (EGY) in the unfactored model and 11.2% (MSA) and 9.2% (EGY) in the factored model. However, the effect of retagging with a morphological analyzer diminishes as the data size increases, yet providing a performance gain of and 0.4% in the unfactored model with the analyzer and 0.5% in its factored counterpart in the high resourced setting in MSA.

Similarly, we observe an increase in performance when we include a morphological analyzer in the very low resourced settings in GLF and LEV. However, as we increase the training data size, the use of a morphological analyzer starts to hurt the performance at 40k in GLF and 10k in LEV in the unfactored model and 20k in GLF and 10k in LEV in the factored model. We observe here that the quality of the analyzer has direct implications on the performance. The analyzers used for MSA and EGY are of high quality since they were manually created and checked, whereas GLF and LEV analyzers are impacted by the quality and size of the annotated data used to create them. This is also consistent with the findings of Khalifa et al. (2020).

Comparison with Previous SOTA Systems Table 6 shows DEV and TEST results for our models and a number of previously published state-of-the-art morphosyntactic tagging systems. For our models, we use the best systems in terms of ALL TAGS metric, namely, the factored model with a morphological analyzer for MSA and EGY, the unfactored model for GLF, and the factored model for LEV. For existing models, we report the best results from Zalmout and Habash (2020) (ZH'20) for MSA, Khalifa et al. (2020) (K'20) for GLF, Zalmout and Habash (2019) (ZH'19) for EGY, and Zalmout (2020) (Z'20) for LEV.

Since some of these systems do not report on all of the features that we report on, but rather on different subsets of them, we include in the table our results when matched with their features (ALL TAGS∗ in Table 6). There is no difference for MSA; however the ALL TAGS∗ setting for EGY and LEV excludes *enc1* and *enc2*. As for GLF, ALL TAGS∗ consists of only 10 features: *pos, asp, per, gen, num, prc0, prc1, prc2, prc3, enc0*.

We observe that our models consistently outperform the existing systems in all variants. Our model achieves 2.6% absolute improvement over the state-of-the-art system in MSA, 2.8% in GLF, 1.6% in EGY, and 8.3% in LEV.

Merged and Continued Training Table 5 shows the results on LEV for the merged and the continued training setups. The results for merged training are consistently below those for the baseline across different data sizes, even though they have access to more data. This is most likely a result of the disproportionately small size of the LEV dataset when compared to the other variants.

In contrast, the results for continued training show consistent improvements over the LEV-only baseline model. Continued training provides a substantial increase in performance, especially in the very low resourced setting with only 5k tokens, giving 3.6% absolute improvement over the baseline. Our results show that continued training from the model trained on high resourced dialects is very beneficial with lower amounts of training data.

5.3 Error Analysis

OOV To better understand the effect of different training setups, we look at the performance of our models in terms of out-of-vocabulary (OOV) tokens alone. We observe a stronger and a more consistent pattern when evaluated on OOV tokens. In fact, the average difference between the best model and the weakest model across variants is larger in OOV tokens (6.7% in ALL TAGS) than in all tokens (2.3%). On OOV tokens, the factored model with a morphological analyzer consistently performs best in all the data sizes for all the variants except for LEV. In LEV, however, the same model without the morphological analyzer outperforms the one with the analyzer. This is presumably due to the orthographic inconsistency in the data along

	DEV								TEST					
	MSA		GLF		EGY		LEV		MSA	GLF		EGY		LEV
	Ours	ZH'20	Ours	K'20	Ours	ZH'19	Ours	Z'20	Ours	Ours	K'20	Ours	ZH'19	Ours
POS	**98.8**	98.1	**97.8**	96.8	**94.2**	93.3	**94.7**	89.4	**98.9**	**97.9**	96.9	**94.6**	93.8	**94.0**
ALL TAGS	**96.1**	93.5	**95.8**	-	**90.6**	-	**88.9**	-	**96.3**	**95.7**	-	**91.0**	-	**87.6**
ALL TAGS*	**96.1**	93.5	**95.8**	93.3	**90.7**	89.3	**89.1**	80.8	**96.3**	**95.7**	92.9	**91.0**	89.4	**87.8**

Table 6: DEV and TEST results of our systems and previously published systems on the same datasets.

	ALL TAGS Error Rate	# Error Features	*Feature Contribution to ALL TAGS Error Rate*															
			pos	per	gen	num	asp	mod	vox	stt	cas	prc0	prc1	prc2	prc3	enc0	enc1	enc2
MSA	3.9	1.5	31.1	4.2	5.1	3.5	3.2	4.9	5.1	21.9	64.1	4.0	2.3	2.2	0.7	2.2	-	-
GLF	4.2	2.0	51.7	33.9	38.0	14.3	19.7	0.8	0.8	0.8	0.8	1.3	5.9	10.7	0.8	19.5	0.8	0.8
EGY	9.4	2.4	62.2	14.6	15.9	14.0	11.0	17.4	11.3	20.0	21.5	9.2	11.3	8.9	2.1	12.9	2.3	2.3
LEV	11.1	1.9	47.6	19.8	22.9	15.3	12.7	0.5	9.6	1.4	1.9	8.2	8.5	6.8	2.2	18.7	5.7	3.7

Table 7: The number and percentage of specific feature errors among the ALL TAGS errors in the best systems on the DEV set.

with the quality of the morphological analyzer as discussed in Section 2.4.

Error Statistics Table 7 presents the number and percentage of specific feature errors among the ALL TAGS errors in the best systems on the DEV set. On average, there are two feature prediction failures within an unfactored tag across the different variants. We observe that MSA and DA exhibit different error patterns: In MSA, case is the largest contributor among other features, which is consistent with the previous findings along the line (Zalmout and Habash, 2020), whereas in dialects, POS is the largest contributor.

Among the POS errors, the most common error type is mislabeling a nominal tag with a different nominal tag, at 44.2% of the errors in GLF, 67.3% in EGY, and 57.8% in LEV, while this type of error is more dominant in MSA (80.8%). Mislabeling nominals with verbs is more common in DA at 23.1% in GLF, 13.0% in EGY, and 20.1% in LEV, compared to MSA (7.7%).

The core morphological features such as per, gen, num, and asp have a higher percentage of errors in DA. Another noticeable difference is enc0 feature (MSA ∼2% vs DA on average ∼17%). This is likely due to label distribution difference: MSA has a highly skewed distribution with 90%, 1%, and 9% ration for 3rd, 2nd and 1st persons as expected in MSA news genre. In comparison, DA has less skew with 50%, 17%, and 32% respectively, which increase the likelihood of error.

Among the three dialects, we observe similar patterns in terms of feature error contribution, especially for GLF and LEV with a correlation coefficient of 0.93. However, in EGY specifically, we observe a high percentage of errors in mod, vox, stt, and cas, partly due to the difference and inconsistency in annotation schemes.

We also found some gold errors which affect all of the systems we compared (previous SOTA and ours). As the results on Arabic morphosyntactic disambiguation are reaching new heights, it may be useful for the community using these resources to revisit their annotations.

6 Conclusion and Future Work

In this paper, we presented the state-of-the-art results in the morphosyntactic tagging task for Modern Standard Arabic and three Arabic dialects that differ in terms of linguistic properties and resource availability. We conducted different experiments to examine the performance of pre-trained LMs under different fine-tuning setups. We showed that the factored model outperforms the unfactored model in low-resource settings. Additionally, high quality morphological analyzers proved to be helpful. Our results also show that fine-tuning using datasets from other dialects followed by fine-tuning using the target dialect is beneficial for low-resource settings. Our systems outperform previously published SOTA on this task.

In the future, we plan to investigate continued training further and find other ways where we can utilize resources and datasets for low-resourced dialects. We also intend to explore other architectures for morphosyntactic tagging using multi-task learning in the context of pre-trained LMs, as well as work on the task of automatic lemmatization.

7 Ethical Considerations

The experiments reported in this work rely on previously published datasets described in Section 2.4. We used the CAMeLBERT models along with morphosyntactically annotated datasets to build our morphosyntactic taggers, which is inline with their intended use. Our work is on core and generic NLP technologies that can be potentially used with malicious intention, for example, as part of the pipeline. To ensure reproducibility, we make our code publicly available. The details on the datasets and training are described in Appendix A. Given the focus of this paper and the available resources, we recognize the limitations of our findings in terms of applicability to different genres, styles, and other languages.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations.

Ibrahim Abu Farha, Wajdi Zaghouani, and Walid Magdy. 2021. Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 296–305, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Bashar Alhafni, Nizar Habash, and Houda Bouamor. 2020. Gender-aware reinflection using linguistically enhanced neural models. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 139–150, Barcelona, Spain (Online). Association for Computational Linguistics.

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.

Ramy Baly, Hazem Hajj, Nizar Habash, Khaled Bashir Shaban, and Wassim El-Hajj. 2017. A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in Arabic. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):23.

Kareem Darwish. 2014. Arabizi Detection and Conversion to Arabic. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, pages 217–224, Doha, Qatar.

Kareem Darwish, Hamdy Mubarak, Ahmed Abdelali, Mohamed Eldesouki, Younes Samih, Randah Alharbi, Mohammed Attia, Walid Magdy, and Laura Kallmeyer. 2018. Multi-dialect Arabic pos tagging: A CRF approach. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

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.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. LDC Arabic treebanks and associated corpora: Data divisions manual. *arXiv preprint arXiv:1309.5652*.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 149–152, Boston, MA.

Ahmed El Kholy and Nizar Habash. 2012. Orthographic and morphological processing for English–Arabic statistical machine translation. *Machine Translation*, 26(1-2):25–45.

Ramy Eskander, Nizar Habash, and Owen Rambow. 2013. Automatic extraction of morphological lexicons from morphologically annotated corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1032–1043, Seattle, Washington, USA.

Ramy Eskander, Nizar Habash, Owen Rambow, and Arfath Pasha. 2016. Creating resources for Dialectal Arabic from a single annotation: A case study on Egyptian and Levantine. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 3455–3465, Osaka, Japan.

Abbas Ghaddar, Yimeng Wu, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Duan Xinyu, Zhefeng Wang, Baoxing Huai, Xin Jiang, Qun Liu, and Philippe Langlais. 2022. Jaber and saber: Junior and senior arabic bert.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

Nizar Habash, Fadhl Eryani, Salam Khalifa, Owen Rambow, Dana Abdulrahim, Alexander Erdmann, Reem Faraj, Wajdi Zaghouani, Houda Bouamor, Nasser Zalmout, Sara Hassan, Faisal Al shargi, Sakhar Alkhereyf, Basma Abdulkareem, Ramy Eskander, Mohammad Salameh, and Hind Saddiki. 2018. Unified guidelines and resources for Arabic dialect orthography. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. 2012. A Morphological Analyzer for Egyptian Arabic. In *Proceedings of the Workshop of the Special*

Interest Group on Computational Morphology and Phonology (SIGMORPHON), pages 1–9, Montréal, Canada.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, pages 15–22. Springer, Netherlands.

Nawar Halabi. 2016. *Modern standard Arabic phonetics for speech synthesis*. Ph.D. thesis, University of Southampton.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Go Inoue, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Joint Prediction of Morphosyntactic Categories for Fine-Grained Arabic Part-of-Speech Tagging Exploiting Tag Dictionary Information. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, pages 421–431, Vancouver, Canada.

Mustafa Jarrar, Nizar Habash, Diyam Akra, and Nasser Zalmout. 2014. Building a Corpus for Palestinian Arabic: A Preliminary Study. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, pages 18–27, Doha, Qatar.

Muhammad Khalifa, Muhammad Abdul-Mageed, and Khaled Shaalan. 2021. Self-training pre-trained language models for zero- and few-shot multi-dialectal Arabic sequence labeling. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 769–782, Online. 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 Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Salam Khalifa, Nasser Zalmout, and Nizar Habash. 2020. Morphological analysis and disambiguation for Gulf Arabic: The interplay between resources and methods. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3895–3904, Marseille, France. European Language Resources Association.

Dan Kondratyuk. 2019. Cross-lingual lemmatization and morphology tagging with two-stage multilingual BERT fine-tuning. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 12–18, Florence, Italy. 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.

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.

Yuval Marton, Nizar Habash, and Owen Rambow. 2013. Dependency parsing of modern standard Arabic with lexical and inflectional features. *Computational Linguistics*, 39(1):161–194.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

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.

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.

Fatiha Sadat and Nizar Habash. 2006. Combination of Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the International Conference on Computational Linguistics and the Conference of the Association for Computational Linguistics (COLING-ACL)*, pages 1–8, Sydney, Australia.

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

Younes Samih, Mohamed Eldesouki, Mohammed Attia, Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak, and Laura Kallmeyer. 2017. Learning from relatives: Unified dialectal Arabic segmentation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 432–441, Vancouver, Canada. 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.

Nasser Zalmout. 2020. *Morphological Tagging and Disambiguation in Dialectal Arabic Using Deep Learning Architectures*. Ph.D. thesis, New York University.

Nasser Zalmout and Nizar Habash. 2017. Don't throw those morphological analyzers away just yet: Neural morphological disambiguation for Arabic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 704–713, Copenhagen, Denmark.

Nasser Zalmout and Nizar Habash. 2019. Adversarial multitask learning for joint multi-feature and multi-dialect morphological modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1775–1786, Florence, Italy. Association for Computational Linguistics.

Nasser Zalmout and Nizar Habash. 2020. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8297–8307, Online. Association for Computational Linguistics.

A Replicability

A.1 Resources

Pretrained transfromer models We fine-tuned CAMeLBERT-MSA for the morphosyntactic tagging task in MSA and CAMeLBERT-Mix (Inoue et al., 2021) for EGY, GLF, and LEV.

Fine-tuning Data We used the Penn Arabic Treebank for MSA (Maamouri et al., 2004), ARZTB (Maamouri et al., 2012) for EGY, the Gumar corpus (Khalifa et al., 2018) for GLF, and the Curras corpus (Jarrar et al., 2014) for LEV. The preprocessing of the data includes fixing inconsistent annotations and removing diacritics through CAMeL Tools (Obeid et al., 2020). This preprocessing was followed in all the previous work we compared with Zalmout and Habash (2019, 2020); Khalifa et al. (2020); Zalmout (2020).

Data Sampling For the learning curve experiment in Section 5.1, we sampled the training data up to 5k, 20k, 40k, 80k, 120k, 150k tokens after shuffling the entire dataset. Each sample after 5k is inclusive of the smaller samples.

Morphological Analyzers The morphological analyzers used in our experiments are the following: For MSA we use the SAMA database (Graff et al., 2009), and for EGY we use CALIMA (Habash et al., 2012). For GLF and LEV, we use automatically generated analyzers from their training data using paradigm completion as described in Eskander et al. (2013, 2016) and Khalifa et al. (2020).

Data Accessibility MSA and EGY related resources need a license from the Linguistic Data Consortium (LDC). GLF data is available at `https://camel.abudhabi.nyu.edu/annotated-gumar-corpus/` and the LEV data is available at `https://portal.sina.birzeit.edu/curras/`. We are happy to provide all of our preprocessed datasets, to those who provide evidence of legal access.

A.2 Implementation

We used Hugging Face's transformers (Wolf et al., 2020) for implementation. Fine-tuning is done by adding a fully connected linear layer to the last hidden state. We release our code including the hyperparameters used in the experiments at `(anonymous URL)`.

For the experiments in Section 5, we use the following hyperparameters: a random seed of 12345, training for 10 epochs, saving the model for every 500 steps, a learning rate of 5e-5, a batch size of 32, and a maximum sequence length of 512. We pick the best checkpoint based on TUNE and report results on DEV and TEST from a single run.

The number of parameters of the factored model for MSA is about 1.5 billion, while the factored model for GLF, EGY, and LEV has 1.8 billion parameters in total. The unfactored model has about 110 million parameters for MSA, GLF, EGY and LEV.

The factored model is the most computationally expensive model to train, which took about 21 hours for MSA, 16 hours for GLF, 13 hours for EGY, and five hours for LEV on a single NVIDIA-V100 card. The unfactored model took about 90 minutes to train for MSA, 60 minutes for GLF, 50 minutes for EGY, and 20 minutes for LEV on the same machine.