Improving Arabic Diacritization by Learning to Diacritize and Translate

Brian Thompson*
AWS AI Labs
brianjt@amazon.com

Ali Alshehri
Apple
a_alshehri@apple.com

Abstract

We propose a novel multitask learning method for diacritization which trains a model to both diacritize and translate. Our method addresses data sparsity by exploiting large, readily available bitext corpora. Furthermore, translation requires implicit linguistic and semantic knowledge, which is helpful for resolving ambiguities in diacritization. We apply our method to the Penn Arabic Treebank and report a new state-of-the-art word error rate of 4.79%. We also conduct manual and automatic analysis to better understand our method and highlight some of the remaining challenges in diacritization. Our method has applications in text-to-speech, speech-to-speech translation, and other NLP tasks.

1 Introduction

Arabic is typically written without short vowels and other pronunciation indication markers,[1] collectively referred to as diacritics. A longstanding task in Natural Language Processing (NLP) is to take undiacritized text and add the diacritics, referred to as diacritization (see Figure 1). Diacritics indicate both how to pronounce the word and resolve ambiguities in meaning between different words with the same (undiacritized) written form.

Diacritic prediction is the dominant source of errors in Arabic grapheme to phoneme conversion (Ali et al., 2020), a crucial component in many text-to-speech and speech-to-speech translation systems.

Diacritization also has applications in Automatic Speech Recognition (ASR) (Vergyri and Kirchhoff, 2004; Ananthakrishnan et al., 2005; Biadsy et al., 2009), Machine Translation (MT) (Diab et al., 2007) morphological analysis (Habash et al., 2016), lexical recognition tests (Hamed and Zesch, 2018; Hamed, 2019), and homograph resolution (Alqahtani et al., 2019a).

هيا لنذهب → هَيَّا لِنَذْهَبْ
[hjaː lnðhb] [hajːaː linaðhab]

Figure 1: Arabic diacritization is the task of adding diacritics (markings above and below characters, shown in red) to Arabic text. Diacritics clarify how a word is pronounced, including short vowels and elongation, and disambiguate word meaning. Here, we show the diacritization of هيا لنذهب (let's go). The IPA pronunciations below each word demonstrate that the diacritics are crucial for pronouncing each word: the undiacritized form maps to an incorrect pronunciation, while the diacritized form maps to the correct pronunciation (the contributions the diacritics make to the pronunciation are also shown in red).

We focus on Modern Standard Arabic (MSA), a standardized dialect of Arabic used in most academic, legal, and news publications, and an obvious choice for Text-to-Speech (TTS) systems. MSA is the 5th most spoken[2] language in the world with about 274M speakers (Eberhard et al., 2021).

1.1 Challenge #1: Data Sparsity

Arabic is a Morphologically Rich Language (MRL), where significant information concerning syntactic units and relations is expressed at word-level. For example, a word like فأسقيناكموه is roughly translated to: 'and we gave it to you to drink'. In this example, linguistic units that are typically expressed by individual words in English such as coordinating conjunctions and personal pronouns are expressed within the word form in Arabic. This fact results in Arabic having a large vocabulary (by way of example, the number of unique, undiacritized words in the Arabic bible from Christodouloupoulos and Steedman (2015)

* Work done while at Apple.

[1] Notable exceptions include the Quran and many children's books.

[2] "Speaker" is a bit of a misnomer: Most Arabic speakers can understand MSA but would not typically produce it.

Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022), pages 11 - 21

is about 4.38x larger than the number of unique, lower-cased words in the English equivalent.) Finally, high-quality diacritized datasets tend to be quite small: The Penn Arabic Treebank (PATB) training subset used in this work is only 15,789 lines, and data available in other dialects can be substantially smaller. These factors result in Arabic being quite data sparse, with diacritics models typically needing to handle a large number of unseen words.

1.2 Challenge #2: Ambiguity

Many of the morphological variants in Arabic are differentiated by only diacritics. This results in un-diacritized Arabic having a huge number of homographs which must be resolved when adding diacritics. Furthermore, as mentioned above, Arabic is a MRL, where information such as gender (male, female), number (singular, dual, plural), case (nominative, accusative, genitive), aspect (perfect, imperfect), voice (active, passive) and mood (indicative, imperative, subjunctive) is expressed on the word-level, sometime with as little as one diacritic. These factors result in undiacritized Arabic being highly ambiguous; Debili et al. (2002) reported an average of 11.6 possible diacritizations for every non-diacritized word in Arabic. For example, the form كتب could be diacritized as كَتَبَ 'he wrote', كُتِبَ 'it was written', كُتِّبَ 'it was written repeatedly', كُتُبٌ 'books' (nominative case), or كُتُبٍ 'books' (genitive case).

1.3 Overview of Proposed Method

We propose a novel Multitask Learning (MTL) (Caruana, 1997) based approach to improve the semantic and linguistic knowledge of a diacritization model. Specifically, we propose augmenting diacritics training data with bitext to train a model to both diacritize Arabic and translate into and out of Arabic.

Our approach addresses data sparsity by substantially increasing the amount of training data seen by the model. Our approach also enables the use of large, readily available MT datasets, which are available not only in Arabic but in many other languages with diacritics as well.[3] In our experiments on the PATB, adding bitext increases training data from 502k to 138M Arabic words, and decreases the Out of Vocabulary (OOV) rate from 7.33% to 1.14%.

Our approach also addresses ambiguity, since the task of translation requires (implicit) semantic and linguistic knowledge. Training on bitext injects semantic and linguistic knowledge into the model which is helpful for resolving ambiguities in diacritization (see Table 1).

These factors contribute to our method achieving a new State-of-the-Art (SOTA) Word Error Rate (WER) of 4.79% on the PATB, vs 7.49% for an equivalent baseline without MTL.

1.4 Main Contributions of This Work

The main contributions of this work are:

- We present a novel MTL approach for diacritization, which does not require a morphological analyzer or specialized annotations (and thus is likely extensible to other languages, dialects and domains).
- We achieve a new SOTA WER of 4.79% on the PATB test set.
- We perform extensive automatic analysis of our method to see how it performs on various conditions including different parts of speech, genders, word frequencies, and sentence lengths.
- We perform detailed manual error analysis of our method, illustrating both issues in the PATB dataset as well as the remaining challenges in Arabic diacritization.

2 Related Work

2.1 Diacritization

Many works have explored using neural networks for Arabic diacritization (Zalmout and Habash, 2017, 2019; Alqahtani and Diab, 2019; Alqahtani et al., 2019b).

Alqahtani et al. (2020) and Zalmout and Habash (2020) both explore MTL regimes in which a model learns to predict Arabic diacritics simultaneously with other features in the PATB. Alqahtani et al. (2020) uses additional features of syntactic diacritization, word segmentation, and POS tagging, while Zalmout and Habash (2020) use additional features of lemmas, aspect, case, gender, person, POS, number, mood, state, voice, enclitics, and proclitics. By also report further improvements by adding an external morphological analyzer. These papers illustrate the potential of MTL, but they re-

[3] In contrast, prior MTL work in diacritization has used hand-curated features such as Part of Speech (POS), gender, and case (see §2.1), severely limiting both the size of available data and the applicability to other languages, which may not have such resources.

#	Arabic Sentence	English Sentence	Diacritized	Pronunciation	Translation
0	علم السعودية أخضر وأبيض اللون	The **flag** of Saudi Arabia is green and white	عَلَمُ	[ʕalamu]	flag
1	أحب علم الفلك	I love space **science**	عِلمَ	[ʕilma]	science
2	علم ناصر أحمد السباحة	Nasser **taught** Ahmad how to swim	عَلَّمَ	[ʕalːama]	taught

Table 1: Adding bitext to our training data improves the semantic and linguistic knowledge of our diacritization model. For example, in order to correctly translate علم out of Arabic, the model must learn to implicitly perform homographic resolution to determine if the word is being used to mean "flag," "science," "taught," or other meanings. This knowledge is helpful for diacritization since diacritized forms are intrinsically linked with word meaning. The model can also implicitly learn, for example, that علم in example #2 is being used as a causative past tense verb. This can help the model diacritize this use of علم correctly (عَلَّمَ), even if عَلَّمَ does not appear in the diacritization training data, since عَلَّمَ follows a common diacritization pattern for causative past tense verbs.

quire additional hand-curated features. This limits both the datasets they can use (neither are able to take advantage of large outside datasets) and the languages they could be applied to.

2.1.1 Contextual Embeddings

Náplava et al. (2021) show that contextual embeddings can result in substantial improvements in diacritization error rates in several languages, but unfortunately they do not report results on Arabic.

Qin et al. (2021) start with a strong baseline built on ZEN 2.0 (Song et al., 2021), an n-gram aware BERT variant. Their BERT-based baseline outperforms prior work on PATB. They then claim even stronger results on PATB with two methods that incorporate multitask training with a second, auxiliary decoder trained to predict the diacritics produced by the Farasa morphological analyzer (Abdelali et al., 2016). We argue that their experimental setup is fundamentally flawed, since Farasa was trained on the PATB test set[4] and can leak information about the test set to the model.[5] They also report results on the Tashkeela training/test data (Zerrouki and Balla, 2017; Fadel et al., 2019), which does not have a potential testset contamination problem, and find that their method underperforms a straightforward bidirectional LSTM,[6] which supports the hypothesis that their strong PATB results are due to training on a derivative of the test set.

2.2 Character-Level and Multilingual MT

Multilingual MT (Dong et al., 2015) has been shown to dramatically improve low-resource translation, including enabling transfer from higher resource language pairs to lower-resource language pairs (Zoph et al., 2016; Nguyen and Chiang, 2017; Neubig and Hu, 2018). In our case, we set up learning to encourage transfer from undiacritized Arabic to much lower-resourced diacritized Arabic.

Most MT systems operate at the subword (Sennrich et al., 2016; Kudo and Richardson, 2018); however, such approaches would result in diacritized and undiacritized versions of the same word having little to no overlap in subwords. We instead train a character-level encoder-decoder model (Lee et al., 2017; Cherry et al., 2018), to maximize the number of shared representations between diacritized and undiacritized words. Character-level diacritics models have also been shown to outperform subword-level models (Alqahtani and Diab, 2019).

3 Method

We train a single Transformer-based (Vaswani et al., 2017) encoder-decoder model to both translate and diacritize, with the hypothesis that the translation task is complementary to diacritization. To maximize the number of shared representations between diacritized and undiacritized words, we train our model at the character-level. Following

[4]Farasa was trained on PATB parts 1, 2 and 3 *in their entirety*, and then tested on a separate collection of hand curated news articles (Abdelali et al., 2016).

[5]To understand how leakage from the test set can occur, consider the word النجمة (the star; female). النجمة appears three times in the training data, once without diacritics (likely an error) and twice as النَّجْمَة . However, it appears 9 times in the test set, each time diacritized as النَّجْمَة . Farasa is trained on both the training and test data, so from it's perspective, النَّجْمَة is by far the most likely diacritization of النجمة . Thus when the model sees النجمة in training, Farasa can artificially bias the model toward producing the diacritized form in the test set, despite that form never appearing in the training data.

[6]Qin et al. (2021) claim to achieve state-of-the-art performance on both datasets, but this is not supported by their results (see their Table 2, noting that bold does *not* denote the best performing system).

work in multilingual MT, we prepend a tag to each output sentence to tell the model whether the output is undiacritized Arabic, diacritized Arabic, English, French, or Spanish during training. At inference time we force decode the tag to request that the model produce diacritized Arabic.

3.1 Decoding

In Arabic, simple rules dictate where diacritics can be placed. During decoding, we enforce these rules by keeping track of which input characters the decoder has produced (i.e. copied from input to output) and constrain the decoder as follows: If the previous output is a non-diacritic Arabic character, we restrict the decoder to produce any diacritic or the next input character. If the previous output is a shadda, we restrict the decoder to produce a non-shadda diacritic or the next input character. Otherwise, the model is forced to produce the next input character. Without these restrictions, we found that the model would occasionally produce minor paraphrastic variations of the input.[7]

3.2 Long Sentence Handling

The computational complexity of Transformer layers is proportional to sequence length squared (Vaswani et al., 2017), so we do not want to train or evaluate on an arbitrarily long sequences of characters. Instead, we limit the maximum input and output sequence to 600. To diacritize a sentence with more than 300 input characters, we take overlapping windows of 300 characters with a step size of 100 characters. We predict diacritics independently for each window, and reconstruct the original sentence using the first 200 characters from the first window, the input characters of the last window excluding the first 100 characters, and the middle 100 characters from any windows in between. This ensures that we only use output with at least 100 characters of context. For the bitext data, we simply discard sentence pairs with greater than 600 input or output characters.

4 Experiments

We train a character-level transformer encoder-decoder model on both diacritics data and the bitext. Our primary model performs diacritization, translation from Arabic (Ar) to English (En), French (Fr), and Spanish (Es), and translation from English, French, and Spanish to Arabic. However, we also perform ablations for analysis purposes, leaving out (1) the Ar→{En,Fr,Es} data, (2) the {En,Fr,Es}→Ar data, and (3) all of the bitext data. Each model uses a single encoder and decoder for all tasks.

[7] The tendency of a multilingual MT model to paraphrase the input has been noted (and exploited) in Tiedemann and Scherrer (2019) and Thompson and Post (2020b).

Name	Form	Sound [IPA]
Fatha	◌َ	/a/
Fathatan	◌ً	/an/
Kasra	◌ِ	/i/
Kasratan	◌ٍ	/in/
Damma	◌ُ	/u/
Dammatan	◌ٌ	/un/
Dagger Alif	◌ٰ	/aː/
Maddah	◌ّ	/ʕaː/
Shadda	◌ّ	Elongation (ː)
Sukun	◌ْ	None

Table 2: Diacritics considered in this work.

	Ar-En	Ar-Es	Ar-Fr	Diacs
Global Voices	0.9	0.9	0.5	-
CCAligned	-	21.9	21.7	-
News Commentary	5.0	5.0	4.3	-
United Nations	20.7	19.9	19.5	-
WikiMatrix	15.0	1.7	1.6	-
PATB	-	-	-	0.5
Total	40.8	48.4	47.1	0.5

Table 3: Size (millions of Arabic words) of training datasets used in this work. Note that total bitext is about 275x larger than diacritics data.

4.1 Diacritics Data

We chose to use PATB part 1 v4.1 (LDC2010T13), part 2 v3.1 (LDC2011T09) and part 3 v3.2 (LDC2010T08), following the train/dev/test splits proposed by Diab et al. (2013). The PATB was chosen because in addition to diacritics, it contains many carefully annotated features which we use to analyze the performance of our models (see §6). We perform unicode NFKD normalization on the text in order to (1) split Unicode characters which contain both a non-diacritic and diacritic (e.g. the Unicode character for alif with maddah above (U+0622) is split into alif (U+0627) and maddah (U+0653)) and (2) normalize the order of characters (e.g. alif + high hamza + fatha and alif + fatha + high hamza both render as أَ and are normalized to alif + high hamza + fatha). The diacritics considered in this work are shown in Table 2.

Training Data	OOV Rate (Undiacritized)
PATB	7.33%
PATB + Bitext	1.14%

Table 4: OOV rates (rate of seeing a word at inference time that was not seen in training), for the encoder, which sees words without diacritics.

4.2 MT Data

We use Ar↔{En,Fr,Es} data from Wikimatrix (Schwenk et al., 2019), Global Voices,[8] United Nations (Ziemski et al., 2016), and NewsCommentary,[9] and Ar↔{Fr,Es} data from CCAligned (El-Kishky et al., 2020), after joining on English urls. We filter out noisy sentence pairs (Khayrallah and Koehn, 2018) using the scripts[10] provided by Thompson and Post (2020a), using more aggressive thresholds of min_laser_score=1.06, max_3gram_overlap=0.1 for the CCAligned data and using values from Thompson and Post (2020a) otherwise. We limit each dataset to 1M lines per language pair, so that no one data type dominates training. Data size are shown in Table 3. We upsample PATB by 20x when combining it with the bitext, since it is much smaller than the bitext.

We filter out the (very infrequent) diacritics from the MT data to ensure that any benefits observed are due to MTL and not simply the result of including more diacritized data in training.[11]

The impact that adding bitext has on the OOV rate is shown in Table 4.

4.3 Models & Training

We train character-level Transformer models in fairseq (Ott et al., 2019). Metaparameters are tuned on the development set. The (non-MTL) baseline has 6 encoder and decoder layers, encoder and decoder embedding dimensions of 1024, encoder and decoder feed-forward network embedding dimensions of 8192, and 16 heads. All embeddings are shared. The model is trained with learning rate of 0.0004, label smoothing of 0.1, dropout of 0.4 with no attention or activation dropout, 40k characters per batch, for 50 epochs. All MTL models have 6 encoder and decoder layers, encoder and decoder embedding dimensions of 1280, encoder and decoder feed-forward network embedding dimensions of 12288, and 20 heads. All embeddings are shared. The model is trained with learning rate of 0.0004, label smoothing of 0.1, dropout of 0.2 with no attention and activation dropout each set to 0.1, 40k characters per batch, for 20 epochs. We select the best performing model for each run using WER on the development set.

[8] casmacat.eu/corpus/global-voices.html

[9] data.statmt.org/news-commentary/

[10] github.com/thompsonb/prism_bitext_filter

[11] In practice, there may be some benefit to retaining diacritics in the MT data, but this was not explored in this work.

5 Results

The word error rates for our method (main model, both ablation models, and baseline) are shown in Table 5, along with error rates reported by prior work. Our main model achieves 4.71% WER on the development set, a relative improvement of 22.8% over the previous best development set result from Zalmout and Habash (2020), who trained a multitask model on PATB features and incorporated a morphological analyzer. On the test set, it achieves 4.79% WER, a relative improvement of 18.8% over the best previously reported test set result from Qin et al. (2021), who trained a BERT-based model.

Our ablation models also outperform all prior work, with the model trained on Ar→{En,Es,Fr} (denoted Ar→∗) bitext outperforming the model trained on {En,Es,Fr}→Ar (denoted ∗→Ar) bitext, but neither perform as well as the main model trained on both Ar→∗ and ∗→Ar. (See §6 for more detailed comparisons between the models trained in this work.)

Finally, our baseline model, consisting of a character-based Transformer with no augmentation or word embeddings, slightly outperforms prior models from Alqahtani et al. (2019b) and Alqahtani and Diab (2019), that also do not use MTL, morphological analyzers, or contextual embeddings.

6 Automatic Analysis

6.1 Case Endings

We compute the Diacritic Error Rate (DER) for all models trained in this work for several different settings: all characters (including whitespace, punctuation, and non-Arabic characters), Arabic characters, Arabic case endings, and Arabic characters excluding case endings: see Table 6. We use POS tags to determine which words have case end-

	Multitask	Morphological Analyzer	Word Embeddings	Dev WER ↓	Test WER ↓
Alqahtani et al. (2019b)	No	No	No		8.20%
Alqahtani and Diab (2019)	No	No	No		7.60%
Alqahtani et al. (2020)	PATB Features	No	fastText		7.51%
Zalmout and Habash (2019)	PATB Features	Train & Test	fastText	7.30%	7.50%
Zalmout and Habash (2020)	PATB Features	Train & Test	fastText	6.10%	
Qin et al. (2021)†	No	No	Zen 2.0	6.49%	5.90%‡
This word (baseline)	No	No	No	7.46%	7.49%
This work (ablation)	Translate ∗→Ar	No	No	5.60%	5.83%
This work (ablation)	Translate Ar→∗	No	No	5.24%	5.32%
This work	Translate ∗→Ar & Ar→∗	No	No	**4.71%**	**4.79%**

Table 5: Development and Test WER (lower is better) for our main system, ablation systems, and baseline, compared to recent work. Our main system outperforms all prior work, as do both ablation systems. †:We exclude the experiments of Qin et al. (2021) which use Farasa in training, as Farasa was trained on the test set (see §2.1.1). ‡:Mean of 5 runs with different random seeds.

	Baseline	Multitask Learning ∗→Ar	Ar→∗	Both
All	2.34%	1.85%	1.73%	**1.52%**
Arabic	2.97%	2.35%	2.21%	**1.94%**
Arabic CE	6.90%	4.71%	4.18%	**3.61%**
Arabic non-CE	2.48%	2.06%	1.96%	**1.73%**

Table 6: Diacritic error rate for all characters (including whitespace and non-Arabic characters), Arabic characters only, Arabic case endings (CE), and Arabic characters excluding case endings (non-CE). We use POS tags to determine which words contain case endings.

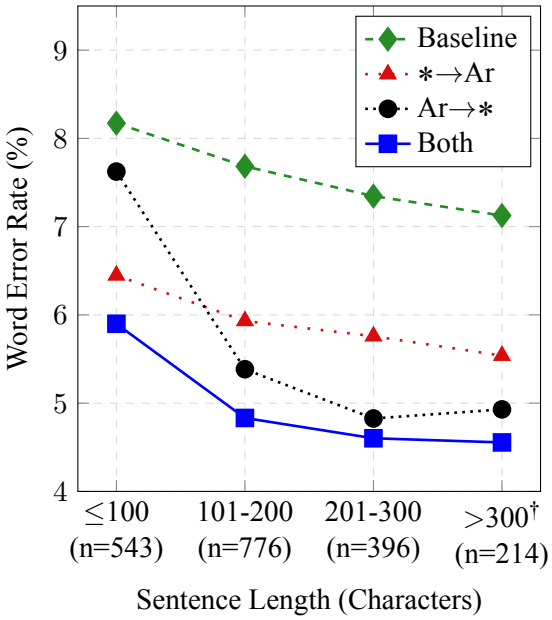

Figure 2: Word error rate vs (undiacritized) character length. †:Sentences over 300 characters are processed in overlapping windows of 300 characters (see §3.2).

ings when computing DER.[12] Comparing our main model to the baseline, we see that MTL training improves case endings more than non-case endings: case ending DER is improved by a 47.7% (3.61% vs 6.90%) vs 30.2% (1.72% vs 2.48%) for non case ending characters. Furthermore, comparing the ablation models, the performance difference between them is more pronounced on case endings, where the ∗→Ar model is 12.7% worse than the Ar→∗ model, while the difference is only 5.1% for non case endings.

6.2 WER vs Sentence Length

We show WER as a function of sentence length (in undiacritized characters) in Figure 2. We note that while both the ∗→Ar and the Ar→∗ models tend to improve with sentence length, the improvement is much more pronounced for the Ar→∗ model. In other words, the Ar→∗ model is benefiting much more from increased context than the ∗→Ar model.

In conjunction with the DER results in §6.1, this indicates that training the model to translate out of Arabic is more helpful at injecting semantic and linguistic knowledge into the model to address ambiguity. The fact that the two translation directions are complementary suggests that training the model to translate into Arabic is addressing data sparsity issues in the model's decoder, despite the mismatch between the bitext being undiacritized and the model needing to produce diacritized output.

[12] Several prior works have reported DER of just the last character as a stand-in for case-ending DER. However, this analysis is muddied by the fact that not all words in Arabic have case endings; in the PATB test set, for example, the POS tags indicate that only about 46.8% of words have them.

	Male		Female		Bias
	#	WER	#	WER	
Pronoun	835	6.23%	641	8.11%	30.3%
Verb	3579	5.34%	2083	6.39%	19.6%
Suffix	901†	5.22%	10222	5.71%	9.5%

Table 7: WER for male and female pronouns, verbs, and nouns/adjectives with gendered suffixes, along with their counts in the test set. †:We include only suffixes which are explicitly marked in the PATB for gender, which tend to be female (see §6.3).

6.3 Gender Bias

Gender bias has been noted in many aspects of NLP (Sun et al., 2019) but we are not aware of any prior work looking at gender bias in diacritization. We use the PATB POS tags to isolate three types of gendered words: pronouns, verbs, and suffixes. "Suffixes" refer to nouns and adjectives that have a gendered suffix. Unsurprisingly, we find that the model is better at diacritizing male words than female words in all three cases (see Table 7), with words in the male categories being diacritized correctly 9.5% to 30.3% more often than their female equivalents. We suspect that this bias is due at least in part to representation within the data: Male pronouns and verbs are 30% and 72% more common than their female counterparts. Counts of suffixes are complicated by the fact that that PATB only marks certain nouns and adjectives for gender (including those with taa marbuta, which tend to be female). By manual inspection, the remainder appear to be male, but we were unable to confirm this in the PATB annotation guidelines so we included only those explicitly marked for gender.

6.4 WER vs POS

The PATB includes detailed POS tagging. We exploit this feature to examine how our model performs on different parts of speech: see Table 8. Note that the PATB has one *or more* POS tags per word, with about 2.19 tags per word on average in the test set. We do not attempt to split words into their respective parts, as we find cases where this is not straightforward. Instead, such words are counted multiple times. As an example, الأَوَّلُون (the first) is both a determiner and cardinal adjective, and contributes to the WER of both.

For parts of speech with at least 500 occurrences in the test set, the worst performing POS for the MTL model by far is proper nouns (count=5969) at 14.09% WER. This is followed by imperfect verbs (count=2598) at 7.89% WER, possessive pronouns (count=1609) at 6.60%, and adjectives (excluding cardinal and comparative) (count=6106) at 6.49%.

Comparative adjectives, which are relatively infrequent (count=264) also have a high WER of 9.95%, but the worst POS considered by far is the extremely infrequent (count=18) imperative verbs, with a WER of 72.22%. Imperative verbs illustrate the importance of domain; news data contains very few imperatives, and imperative verbs are often distinguished from from imperfect or perfect verbs by diacritics alone. For example, استمر على الطريق can be diacritized اِستَمِر عَلَى الطَّرِيقِ (Continue on the road) or اِستَمَرَّ عَلَى الطَّرِيقِ (He continued on the road). This results in the model choosing the much more common perfect or imperfect forms in the majority of cases that should be imperative.

6.5 WER vs Word Frequency

MTL improves learning across all word frequencies: see Table 9. The biggest improvements are seen for words seen once and 2-4 times in training, with relative improvements of 43.5% and 45.4%, respectively.

7 Manual Analysis

To better understand the performance of our MTL model, we manually annotate all differences between our model prediction and the gold test set for a randomly selected 20% of the 1246 sentences in the test set that contain at least one disagreement.

We find that approximately 66% of the disagreements between the gold test set and the model are the result of model errors, which we denote as "true errors". The majority of these errors are due to case markings being either incorrect (38.6% of all true errors) or missing (16.5% of all true errors), while the rest of the word is correct.

However, we find that in approximately 32% of disagreements the model output is, in fact, correct. We denote such cases as "false errors." About half (50.3%) of the false errors were due to the test set missing diacritics and another 31.2% of all false errors were due to errors in the test set diacritics. 10.7% of the false errors were the result of valid variations which did not change the meaning of the sentence in any way (e.g. يُكْشِفْ vs يَكْشِفْ and الدُّوَلِي vs الدَّوْلِي). Another 4.4% of false errors were the result of valid variations that changed the meaning of the sentence while still resulting in a plausible meaning. A very small number of words (3.4%

	Count	Baseline WER	MTL WER	Rel. imprv.	Examples
Noun: Proper	5969	18.24%	**14.09%**	22.8%	مَرِيَم (Mary); أَحمَد (Ahmed)
Noun: Numeric	1609	3.29%	**2.11%**	35.8%	عَشَرَة (ten); أَربَعَة (four)
Noun: Quantity	451	10.42%	**5.32%**	48.9%	أَيَّة (any; fem); بَعض (some)
Noun: Other	22795	8.43%	**5.03%**	40.3%	يَوم (day); دُوَيلَة (small country)
Pronoun: Possessive	1681	11.42%	**6.60%**	42.2%	كِتابِيَ (my book); كِتابُكُن(your book; fem)
Pronoun: Demonstrative	601	**0.00%**	0.17%	-	هٰذا (this; male singular); هاتان (these, fem dual)
Pronoun: Other	1154	1.04%	**0.52%**	50.0%	شاهَدَتنِي (she saw me); أَنتَ (you; male singular)
Verb: Inflected, Perfect	3273	9.53%	**4.89%**	48.7%	ذَهَبَ (he went); قُبِلَ (it was accepted)
Verb: Inflected, Imperfect	2598	13.55%	**7.89%**	41.8%	يَذهَبُ (he goes); تُقبَلُ (it is accepted)
Verb: Inflected, Imperative	18	83.33%	**72.22%**	13.3%	اِذهَب (go; male); قِفِي (stop; fem)
Adverb	260	**0.00%**	0.38%	-	مَتَى (when); حِينَذاك (then)
Adjective: Cardinal	348	7.18%	**4.31%**	40.0%	القَرن (19th century); الأَوَّلُون (the first)
Adjective: Comparative	264	16.67%	**9.85%**	40.9%	أَحرَصُ (more cautious); الأَحسَن (the best)
Adjective: Other	6106	10.87%	**6.49%**	40.4%	تارِخِيٌّ (historic); يَهُودِيٌّ (Jewish)
Determiner	15337	8.72%	**5.85%**	32.9%	التُونسِي (the Tunisian); اليَومُ (the day)

Table 8: WER for our baseline and our main MTL model, for various parts of speech, and their associated count in the test set. Note: many words have more than one POS and contribute to 2+ categories (see §6.4).

# Occur in PATB-train	Baseline	Multitask Learning ∗→Ar	Ar→∗	Both
0	30.93%	26.30%	23.20%	**21.92%**
1	17.63%	12.46%	10.33%	**9.95%**
2-4	11.94%	8.32%	7.56%	**6.51%**
5-16	8.78%	6.83%	6.50%	**5.67%**
17-64	7.80%	5.81%	5.50%	**4.86%**
65-256	6.33%	4.97%	4.55%	**3.76%**
257-1024	4.34%	3.28%	3.16%	**2.94%**
>1024	0.30%	**0.20%**	0.29%	0.22%

Table 9: WER vs number of times a word occurs in PATB-train (ignoring diacritics), for all four models trained in this work.

of false errors) had trivial diacritic variations that do not change meaning or pronunciation (e.g. one having a sakun while the other had no diacritic, or one having a fatha before an alif while the other did not).

Finally, about 2% of the disagreements are cases where the input to the model is not a real word, making the correct output undefined.

8 Conclusion

We demonstrate that training a diacritics model to both diacritize and translate substantially outperforms a model trained on the diacritization task alone. Adding translation data substantially increases the amount of training data seen by the model, addressing data sparsity issues in diacritization. The translation task also injects semantic and linguistic knowledge into the model, helping the model resolve ambiguities in diacritization.

Our method achieves a new state-of-the-art word error rate of 4.79% on the Penn Arabic Treebank datasets, using the standard data splits of Diab et al. (2013).

Finally, we present extensive manual and automatic analysis which provides insight into our method and highlights several challenges that still remain in Arabic diacritization, including proper nouns, female word forms, and case endings.

References

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

Ikbel Hadj Ali, Zied Mnasri, and Zied Lachiri. 2020. Dnn-based grapheme-to-phoneme conversion for arabic text-to-speech synthesis. *International Journal of Speech Technology*, 23(3):569–584.

Sawsan Alqahtani, Hanan Aldarmaki, and Mona Diab. 2019a. Homograph disambiguation through selective diacritic restoration. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 49–59, Florence, Italy. Association for Computational Linguistics.

Sawsan Alqahtani and Mona Diab. 2019. Investigating input and output units in diacritic restoration. In

2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), pages 811–817.

Sawsan Alqahtani, Ajay Mishra, and Mona Diab. 2019b. Efficient convolutional neural networks for diacritic restoration. 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 1442–1448, Hong Kong, China. Association for Computational Linguistics.

Sawsan Alqahtani, Ajay Mishra, and Mona Diab. 2020. A multitask learning approach for diacritic restoration. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8238–8247, Online. Association for Computational Linguistics.

Sankaranarayanan Ananthakrishnan, Shrikanth Narayanan, and Srinivas Bangalore. 2005. Automatic diacritization of arabic transcripts for automatic speech recognition. In *Proceedings of the 4th International Conference on Natural Language Processing*, pages 47–54.

Fadi Biadsy, Nizar Habash, and Julia Hirschberg. 2009. Improving the Arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 397–405, Boulder, Colorado. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language resources and evaluation*, 49(2):375–395.

Fathi Debili, Hadhémi Achour, and Emna Souissi. 2002. La langue arabe et l'ordinateur: de l'étiquetage grammatical à la voyellation automatique. *Correspondances*, 71:10–28.

Mona Diab, Mahmoud Ghoneim, and Nizar Habash. 2007. Arabic diacritization in the context of statistical machine translation. In *Proceedings of MT-Summit*. Citeseer.

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

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2021. *Ethnologue: Languages of the World*, 24th edition. SIL International.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*.

Ali Fadel, Ibraheem Tuffaha, Mahmoud Al-Ayyoub, et al. 2019. Arabic text diacritization using deep neural networks. In *2019 2nd international conference on computer applications & information security (ICCAIS)*, pages 1–7. IEEE.

Nizar Habash, Anas Shahrour, and Muhamed Al-Khalil. 2016. Exploiting Arabic diacritization for high quality automatic annotation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4298–4304, Portorož, Slovenia. European Language Resources Association (ELRA).

Osama Hamed. 2019. Automatic diacritization as prerequisite towards the automatic generation of Arabic lexical recognition tests. In *Proceedings of the 3rd International Conference on Natural Language and Speech Processing*, pages 100–106, Trento, Italy. Association for Computational Linguistics.

Osama Hamed and Torsten Zesch. 2018. The role of diacritics in increasing the difficulty of Arabic lexical recognition tests. In *Proceedings of the 7th workshop on NLP for Computer Assisted Language Learning*, pages 23–31, Stockholm, Sweden. LiU Electronic Press.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions*

of the Association for Computational Linguistics, 5:365–378.

Jakub Náplava, Milan Straka, and Jana Straková. 2021. Diacritics Restoration using BERT with Analysis on Czech language. *The Prague Bulletin of Mathematical Linguistics*, 116:27–42.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Han Qin, Guimin Chen, Yuanhe Tian, and Yan Song. 2021. Improving Arabic diacritization with regularized decoding and adversarial training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 534–542, Online. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. 2021. Zen 2.0: Continue training and adaption for n-gram enhanced text encoders. *arXiv preprint arXiv:2105.01279*.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, Florence, Italy. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020a. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020b. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.

Jörg Tiedemann and Yves Scherrer. 2019. Measuring semantic abstraction of multilingual NMT with paraphrase recognition and generation tasks. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 35–42, Minneapolis, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Dimitra Vergyri and Katrin Kirchhoff. 2004. Automatic diacritization of Arabic for acoustic modeling in speech recognition. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, pages 66–73, Geneva, Switzerland. COLING.

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

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.

Taha Zerrouki and Amar Balla. 2017. Tashkeela: Novel corpus of arabic vocalized texts, data for auto-diacritization systems. *Data in brief*, 11:147.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.