Cross-lingual transfer for low-resource Arabic language understanding

Khadige Abboud[1], **Olga Golovneva**[*1,2], **Christopher DiPersio**[1]

[1]Alexa AI, Amazon, Cambridge, MA
[2]FAIR Labs, Meta, Washington, DC
abboudk@amazon.com, olggol@meta.com, dipersio@amazon.com

Abstract

This paper explores cross-lingual transfer learning in natural language understanding (NLU), with the focus on bootstrapping Arabic from high-resource English and French languages for domain classification, intent classification, and named entity recognition tasks. We adopt a BERT-based architecture and pretrain three models using open-source Wikipedia data and large-scale commercial datasets: monolingual:Arabic, bilingual:Arabic-English, and trilingual:Arabic-English-French models. Additionally, we use off-the-shelf machine translator to translate internal data from source English language to the target Arabic language, in an effort to enhance transfer learning through translation. We conduct experiments that finetune the three models for NLU tasks and evaluate them on a large internal dataset. Despite the morphological, orthographical, and grammatical differences between Arabic and the source languages, transfer learning performance gains from source languages and through machine translation are achieved on a real-world Arabic test dataset in both a zero-shot setting and in a setting when the models are further finetuned on labeled data from the target language.

1 Introduction

The fast growing interest in conversational AI-based voice assistants has increased the importance of finding ways to efficiently and rapidly expand these services to multiple new languages. One of the core components of virtual assistants is Natural Language Understanding (NLU), which is usually composed of three main tasks: domain classification (DC), intent classification (IC), and named entity recognition (NER). NLU tasks are responsible for classifying the domain and intent from the user's utterance and identifying and extracting entities from their requests through slot-filling.

Training an NLU model to support a new language requires a large amount of labeled utterances, which is costly and time-inefficient, particularly for low-resource languages. In recent years, a lot of success was shown through cross-lingual knowledge transfer on various NLU tasks for zero-shot transfer and few-shot transfer (Johnson et al., 2019; Ponti et al., 2021; Wang et al., 2021; Pires et al., 2019; Muller et al., 2021). This is made possible with the availability of multilingual pretrained language models such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020). However, cross-lingual transfer was shown to be more effective among similar languages (e.g., English to French) as opposed to distant languages (e.g., English to Arabic), especially for languages that differ in their script (Muller et al., 2021; Conneau et al., 2020; Johnson et al., 2019; Wu and Dredze, 2019). Efforts to reduce the distance between source and target languages include transliteration/romanization to Latin script (Muller et al., 2021; Johnson et al., 2019), and machine translation (Wang et al., 2021; Ponti et al., 2021). Although romanization was shown to be beneficial for languages that are not included in pretraining, it degraded performance on languages that are included in these large multilingual models like Arabic and Japanese (Muller et al., 2021). Driven by some of the shortcomings of pretrained multilingual models, several monolingual models have been trained and released in the past couple of years for multiple languages like Arabic (Antoun et al., 2020; Abdul-Mageed et al., 2021; Inoue et al., 2021), German (de Vries et al., 2019), and French (Martin et al., 2020). Whether multi-lingual or monolingual models are adopted, task-specific labeled data is still required for finetuning.

In this paper, we experiment with cross-lingual transfer from English and French, two high-resource languages with rich NLU labeled datasets for bootstrapping NLU model for the low-resource

[*]Work done during the author's tenure at Amazon.

Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP), pages 225 - 237

Arabic language, specifically for virtual assistant (VA) systems. To this end, we train three BERT models on a mix of open-source data and machine translated user inquiries: a monolingual - Arabic only, a bilingual Arabic-English and a trilingual Arabic - English - French models. Particulars of Arabic language such as orthographic inconsistencies in diacritized script and inflectional affixation are mitigated by preprocessing the data before training. We distill each of the BERT models to a smaller student model that better fit memory and latency requirements of commercial VA systems. We present experimental results on internally gathered real-world Arabic dataset that illustrate cross-lingual transfer through NLU knowledge transfer and machine translation (MT). Gains from transfer learning (TL) are achieved on the target Arabic dataset in both DC and joint IC-NER tasks in a zero-shot setting, few-shot setting, and in a setting with non-production Arabic labeled data included in finetuning.

2 Related Work

Cross-lingual transfer for low-resource language: There is a large body of research that shows successful cross-lingual transfer for a variety of tasks in both zero-shot setting, when the model is finetuned on data from the source language only, and in a regular setting, when the model is finetuned on the target language. (Johnson et al., 2019) explores cross-lingual transfer from English to Japanese, not only a morphologically dissimilar language, but also fundamentally different on the character and token level. Authors use a Bi-LSTM based model with word and character embeddings and finetune it for NER task. To increase the benefit of transfer learning, the authors propose to romanize Japanese characters to unify the character embedding space between the target and source languages.

The introduction of pretrained multilingual language models like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) has opened the doors for wider exploration of cross-lingual transfer learning (Wang et al., 2021; Libovickỳ et al., 2019; Muller et al., 2021; Wu and Dredze, 2019). (Muller et al., 2021) has shown that the reason why some languages do not benefit from these massive multilingual models is largely related to script differences; particularly for languages that have not been seen by mBERT. Experiments in (Muller et al., 2021) show that transliteration to Latin script for low-resource languages with different script does improve performance for part-of-speech tagging, dependency parsing, and NER tasks, however not for languages that are included in mBERT like Arabic and Japanese. Such findings are also echoed in (Wu and Dredze, 2019). Another way to bring distant languages closer is through machine translation. (Wang et al., 2021) introduces a step before finetuning on IC-NER task by retraining pretrained multilingual models (mBERT and XLM-R) for MT task. Authors show that performance gain with the proposed approach is larger between distant languages than that between similar languages. (Ponti et al., 2021) proposes an integrated translation – monolingual classifier system that exploits cross-lingual transfer through setting the translation as a latent variable between the target text and the labels (a translate-test approach). Using reinforcement learning, (Ponti et al., 2021) trains the integrated translation-classifier system with classification accuracy as the reward. This approach however, can be only applied to DC and IC tasks where the whole utterance is labeled with one class.

NLU models for Arabic: Non-deterministic NLU models for Arabic have not been extensively explored until recently; largely due to a lack of rich labeled datasets for the various NLU tasks. (Soliman et al., 2017) proposed Arabic specific word2vec embeddings. (Al-Smadi et al., 2020) utilized pretrained Multilingual Universal Sentence Encoder (MUSE) embedding and trained a bidirectional-gated recurrent neural network with a mix of average and max pooling layer for Arabic NER task using the WikiFANEGold dataset (Alotaibi and Lee, 2014) which classifies entities into eight classes only (person, location, organization, geopolitical, etc.).

Although mBERT includes Arabic, cross-lingual transfer did not show performance gains for Arabic as it did on Indo-European languages (Muller et al., 2021; Wu and Dredze, 2019). Motivated by the monolingual BERT models, (Antoun et al., 2020) trained AraBERT, a monolingual BERT-based language representation model for Arabic language on data that includes Arabic Wikipedia dumps, in addition to two publicly available large Arabic corpora: 5M (El-Khair, 2016) and 3.5M (Zeroual et al., 2019) articles, both extracted from Arabic news sources. Authors in (Antoun et al., 2020) also introduced a preprocessing step on the data prior

to using it for pre-training BERT, which used off-the-shelf Arabic Farasa tokenizer (Abdelali et al., 2016) for subword unit segmentation. Building on AraBERT, ArBERT (Abdul-Mageed et al., 2021) and CAMeLBERT (Inoue et al., 2021) have added additional Arabic datasets to pretraining a monolingual BERT that cover more topics and dialects. Because of the lack of rich labeled Arabic dataset, the NER task in (Helwe et al., 2020; Inoue et al., 2021; Abdul-Mageed et al., 2021) is limited to classifying nouns into three main classes only (person, location, organization)[1], a much simpler NER task than that needed to power a virtual assistant system, where user requests can span hundreds of entity labels.

In this paper, we propose a multilingual NLU model for Arabic language, targeted for commercial virtual assistant system. We explore cross-lingual transfer through MT and task-specific learning transfer from rich source languages (English and French) to Arabic. Despite the languages not being closely related, we show that multilingual models outperforms the monolingual model on large-scale Arabic traffic for both DC and IC-NER tasks. To our knowledge, this is the first Arabic model trained and evaluated for such complex NLU tasks required for virtual assistants which involves classifying 18 domains, 333 intents, and 268 entity labels.

3 Arabic NLU

3.1 Challenges in Arabic

Arabic differs from English and French morphologically, orthographically, and grammatically. Some of the differences can hinder cross-lingual transfer learning. These differences include:

- **Script**: Arabic script has opposite writing direction and does not use the Latin alphabet, instead it is written from right to left using the distinct *Abjad* writing system;

- **Inflectional morphology**: Unlike English, inflections in Arabic can be suffixes or prefixes (Shamsan and Attayib, 2015), and Arabic inflections have far more person, number, and gender distinctions than that in English;

- **Diacritics**: Some short vowels are included on Arabic text as diacritics, which are optional written symbols.

These are only a few of the differences that can complicate transfer learning to Arabic from resource-rich languages, usually Indo-European like English, Spanish, and French. The language complexity is further inflated in dialectal Arabic, due to the lack of writing standards resulting in orthographic inconsistencies (Kwaik et al., 2018). Modern standard Arabic (MSA) is only used for writing and is spoken mostly in official settings like news broadcasts and government announcements. In households, the common location for virtual assistants, dialectal Arabic is more likely to be used. Furthermore, to globalization and historical reasons, some of dialectal Arabic's loan-words and phrases come from other languages, particularly English and French[2].

Arabic has templatic and concatenative morphology where verbs and nouns are derived from 3,000 roots (El-Kishky et al., 2019) by applying templates to the roots to generate stems and then adding prefixes and suffixes. In Arabic, inflectional affixation is very common; the definite article ("the"), prepositions ("to", "in", "for"), conjunctions ("and", "then"), and pronouns ("you", "my", "our", etc.) are represented as affixes on words they modify. This poses a challenge for NER. For example, in the utterance "order **two** boxes of apples", the quantity to be ordered can be inferred from token "**two**". In Arabic, however, the quantity "**two**" would be a suffix to token "**box**", "اطلبي صندوقين من التفاح" (literal: "order box**Two** of apples"). Table 1 shows a few examples that illustrate the challenges of inflectional affixation in Arabic. In an effort to address this, we add a rule-based normalization step that splits affixes; however, we limit this to affixes that make a functional difference to the meaning (e.g., pronouns and quantity) as opposed to non-functional ones, e.g., definite article, and prepositions.

Although diacritics are used to disambiguate meaning, especially in the absence of context, we have decided to strip diacritics [3] from open-source data due to the following three reasons:

[1]The popular ANERcorp dataset (Benajiba and Paolo, 2008) has a total of 9 labels: the 3 main classes in addition to Other and IOB tagging.

[2]Although TL from French and English can particularly help dialectal Arabic due to natural code-switching, the specific impact on code-switching is out of scope of this paper.

[3]With the exception of Shadda diacritic.

- We conducted a study on internally localized and diacritized data that showed that diacritics in fact harm NLU model performance more than they help disambiguate words, and this is mainly due to inconsistencies in the use of diacritics when transcribing data. Details are in Appendix: A.2;
- Relying on diacritized text for NLU will further limit the available resources for Arabic, as most open-source datasets (e.g., Wikipedia) are not diacritized; and
- The use of DNN-based language models such as BERT heavily relies on context for predictions, which can help disambiguate words without the need for diacritics, similar to how Arabic speakers would use the surrounding context to infer the meanings of words.

would you turn it off?	*call my mum*	*play a song in the room*
أتطفئيها؟	إتصلي بأمي	شغلي أغنية بالغرفة
wouldYouTurnOffIt	call mumMy	play song InTheRoom

Table 1: Examples of inflectional affixation in Arabic. On the right, a 5-token English utterance can be written with a single token in Arabic, pronouns ("it", "my") are attached as a suffix, and the definite article ("the") and preposition ("in") can be attached as prefixes.

3.2 Data

For training BERT models, we use two main sources of unlabeled data: internal data from a commercial VA system[4] and external open-source data from Wikipedia. For the latter, we collect Wikipedia dumps for Arabic (ar), English (en), and French (fr) and extract their content using WikiExtractor package (Attardi, 2015). For ar-Wikipedia data, in addition to the preprocessing described in the previous section, we split sentences based on full stop, along with semicolon and comma if the sentence length is greater than 25 tokens, because commas are commonly used in Arabic as a sentence delimiter, and the full stop is used at the end of a paragraph. The extracted Wikipedia data accounts for ≈ 6.3M, 98.5M, 34.2M sentences for ar, en, and fr, respectively, as listed in Table 2. Wikipedia and other open-source data are different from the nature of user inquiries to virtual assistants. We have found this to be particularly true for Arabic Wikipedia data, which overwhelmingly covers political and historical vocabulary and topics. To overcome this bias, we have opted to mix the data with commercial dataset from an NLU system. We use the rich and resource-heavy English and French data, accounting for 36.2M and 14.6M, respectively, and corresponding to users requests, i.e., unannotated utterance text. All user utterances have been de-identified and anonymized. We used AWS translate to translate English user requests into Arabic, and obtained an unannotated Arabic MT dataset of equal size to the English dataset (≈ 3.2M). For pretraining, we split the data randomly into 85:15 train:validation sets, and to balance the data across languages for the multilingual models, we follow (Conneau and Lample, 2019) and we sample sentences according to a multinomial distribution with probabilities $q_i = \sqrt{p_i}/(\sum_j^N \sqrt{p_j})$, $p_i = n_i/\sum_j^N n_j$ in which N is the total number of languages in the model and $n_i$is the total number of utterances in language i. For finetuning, we use annotated NLU data from a commercial VA system, representing user inquiries in English and French, two mature and high-resource languages. We sample equally 418,477 utterances from the two languages for finetuning the pretrained bilingual and trilingual models for DC and IC-NER tasks. In a zero-shot setting, only English and French labeled datasets are used in finetuning the models. Note that the bilingual model is pretrained on unlabeled Arabic and English datasets, it is finetuned only on labeled English data in a zero-shot setting. For comparison, we finetune a second set of models that we refer to as pre-production (pre-prod) models with an additional 369,485 annotated Arabic utterances added during finetuning. This dataset (forth row in Table3) is collected using Mechanical Turk (mTurk). We use the mTurk data to train a third set of few-shot models, by sampling only 10 utterances per intent and using that in training.We also explore transfer learning for NLU task through translation; we translate labeled English traffic using AWS Translate into Arabic. In order to enhance the quality of the MT dataset, we post-process the translated utterances automatically to reproject labels and recombine affix when split incorrectly, e.g.,

- **input**: <CallType> call </CallType> <ContactName>Ali</ContactName>
- **MT**: <CallType>الإتصال بـ</CallType> <ContactName>علي</ContactName>

[4]Details about the commercial virtual assistant system and the internal data are omitted to maintain authors anonymity.

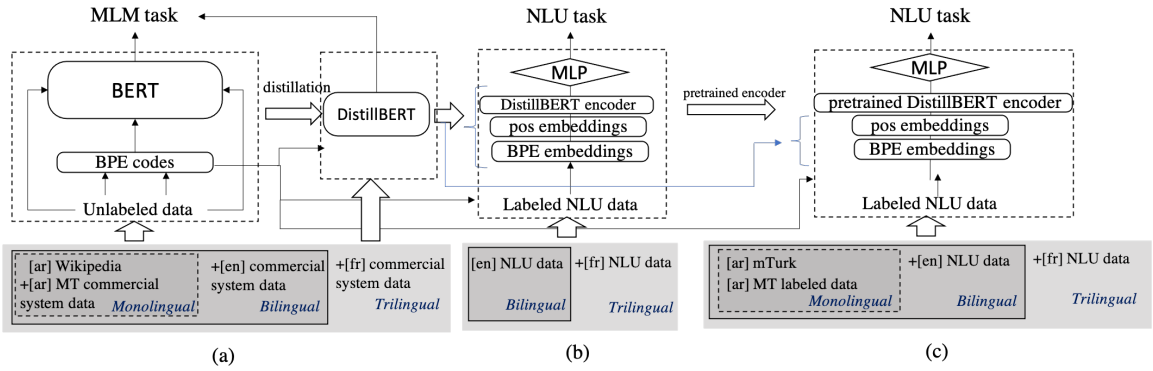

Figure 1: Schematic of monolingual and multilingual BERT training and distillation for Arabic NLU tasks: (a) BERT pretraining and distillation on unlabeled data; (b) Task-specific pretraining DistillBERT for NLU task on labeled data from resource rich source languages, English and French; (c) Finetuning on mix of Arabic, English, and French labeled data in addition to MT Arabic data.

- **postprocessed**: <CallType>الإتصال</CallType> <ContactName>بعلي</ContactName>
- **input**: Call </UserTrigger>my</UserTrigger> <NumberType>Phone</NumberType>
- **MT**: اتصل</UserTrigger> <NumberType>بهاتفي</NumberType></NumberType>
- **postprocessed**: اتصل</NumberType> بهاتف<NumberType> <UserTrigger>ي</UserTrigger>

We finetune another set of models for each of the zero-shot, few-shot, and pre-prod setting by adding a total of 417,895 utterances sampled from the MT labeled data during finetuning[5]. Having the MT labeled dataset enables the evaluation of the monolingual model in a zero-shot setting by finetuning only on the MT dataset. All models are tested on the same Arabic dataset consisting of a total 864,127 Arabic utterances annotated from real-world VA commercial system. This test dataset spans 18 domains, 333 intents, and 268 entity labels[6]

3.3 Model Training

3.3.1 Pretraining

We pretrain three BERT models, monolingual (Mono), bilingual (Bi) and trilingual (Tri), models using open-source Wikipedia data and unlabeled inquires to a commercial VA system together with the corresponding MT ones. We use **BERTbase** setting (Devlin et al., 2019) with 12 encoder layers, 768 hidden dimensions, 3072 hidden size, and 12 attention heads, and pretrain for a Masked Language Model (MLM) task for 40 epochs with 15% of tokens masked. We adopt Byte Pair Encoding (BPE) for subword tokenization of BERT pretraining in an effort to deal with inflectional affixation in Arabic. We use FastBPE (Sennrich et al., 2016; et al., 2015) for BPE extraction, learning 30K, 80K, 90K codes[7] from Wikipedia data for the monolingual, bilingual, and trilingual models, respectively. For run-time efficiency and inference speed, we further distill each model to a smaller student model during pretraining. The student model architecture is composed of 4 layers, 768 hidden dimensions, 1200 hidden size, and 12 attention head. This architecture, DistillBERT, is a slightly bigger model than TinyBERT (Jiao et al., 2020) but is 3x smaller and 4.7x faster than the original BERT. For knowledge distillation we use the same dataset used for training the teacher model and adopt logit matching method between teacher and student from (Hinton et al., 2015), where the student is trained to minimize two losses during training; the standard cross-entropy loss and the cross-entropy loss between the teacher and the student. We use the same datasets and BPE codes for distillation on the same MLM task. The pretraining step is illustrated in Figure 1(a).

[5]During finetuning, all data is mixed, with no particular order.

[6]Our evaluation data contains only 32.86% of the tokens labeled as *Other*. The combined training data in Table3 covers all 18 domains and a total of 235 intents out of which 225 intents are in the testset, and the remaining uncovered test intents are part of the tail 0.64% of the testsets.

[7]The reason we vary BPE code number across the three models is to account for the additional vocabulary from the added languages. Otherwise, either the smaller monolingual model will suffer from codes not generalizing to new vocabulary, or the larger trilingual model will suffer from codes being too granular.

Table 2: Unlabeled data for extracting BPE codes and BERT model pretraining and distillation.

Data source	Language	Size (sentence)
Wikipedia	Arabic (ar)	6,377,443
Wikipedia	English (en)	98,524,407
Wikipedia	French (fr)	34,248,312
VA system	English (en)	36,288,990
VA system	French (fr)	14,609,950
VA system	Machine-translated Arabic (ar-MT)	36,288,980

3.3.2 Task-specific Pretraining

Before the final-finetuning on NLU tasks, we leverage the rich English and French labeled data for a pre-finetuning step, in which we pretrain the encoders for the bilingual and trilingual models specifically on NLU tasks.In this task-specific pretraining, illustrated in Figure 1(b), we do not include any labeled data for the target language, Arabic, as we are testing how much of the NLU learning can be transferred from the source languages. Consequently, this step is excluded from the monolingual model.

Table 3: Labeled data for finetuning and evaluating NLU models for DC and IC-NER tasks. Only the first three datasets are used for the zero-shot experiments, the forth dataset is used for the few-shot experiment, the fifth dataset is added for finetuning the pre-prod models, and the last dataset is only used for evaluation.

Dataset	Language	Size (utterance)	
		Train	Test
en traffic	en	418,477	0
fr traffic	fr	418,477	0
ar-MT dataset	ar-MT	417,895	0
ar mTurk few shot	ar	2,547	0
ar mTurk data	ar	369,485	0
ar traffic	ar	0	864,127

3.3.3 Finetuning

In the final step, the three pretrained DistillBERT models are finetuned for NLU tasks on labeled internal data listed in Table 3 and illustrated in Figure 1(c). For each of the three models, we train three sets of models: zero-shot, few-shot, and pre-prod models. The only difference is the inclusion of the mTurk labeled data from the target Arabic language for the latter two experiments. In the few-shot setting we sample 10 utterances randomly per intent while maintaining a minimum of 40 utterances per domain. For each of these set of experiments, we also train a model with and without MT data, as a result we have a total of 17 models. We select the monolingual model with few-shots to be our baseline, and compare it to the bilingual and trilingual models[8].

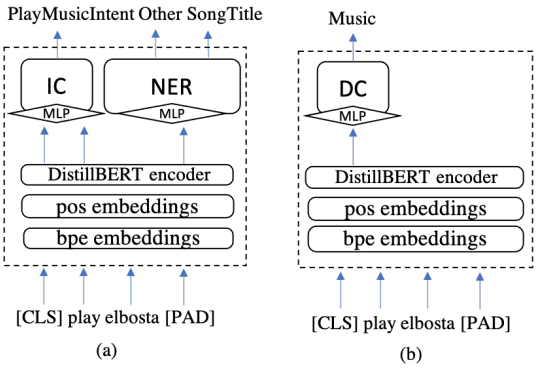

Figure 2: Schematics of the finetuning step for DC and IC-NER tasks.

- **BASELINE**: a monolingual DistillBERT model distilled from **BERTbase** model pretrained on Arabic unlabeled data
- **Bilingual**: a DistillBERT model distilled from BERTbase model pretrained on mix of unlabeled Arabic and English with task-specific pretraining on NLU labeled data from high-resource English language
- **Trilingual**: a DistillBERT model distilled from BERTbase model pretrained on mix of unlabeled Arabic, English, and French with task-specific pretraining on NLU labeled data from high-resource languages: English and French

The IC-NER model is trained for a joint-task objective with two-layer MLP for the IC task and two-layer MLP plus a CRF layer for the NER task as illustrated in Figure 2. For the DC task, we have the same DistillBERT architecture with the exception of the final two MLP layers for one-vs-all classification task.

4 Results and Discussion

We measure the performance of our models for DC and IC-NER tasks in terms of domain classification error rate (DCER) and semantic error

[8]Because the pretraining objective is targeted for MLM task, a different objective than the target NLU tasks, we do not have a monolingual zero-shot model, and therefore use the monolingual few-shot model as our baseline.

Table 4: Results relative to baseline (% change) for Monolingual (**Mono**) Bilingual (**Bi**) and Trilingual (**Tri**) models on IC-NER and DC tasks evaluated on 864,127 Arabic utterances. Average performance is across domains. Bold values indicate best performance for each setting (zero/few-shot/pre-prod).

		Zero-shot			Few-shot			Pre-prod		
Δ%SemER		**Mono**	**Bi**	**Tri**	**Mono**	**Bi**	**Tri**	**Mono**	**Bi**	**Tri**
Overall	w/o MT	-	0.29	-7.50	0	-15.06	-20.76	-49.32	-55.24	-52.49
	with MT	-10.64	-12.49	**-19.20**	-14.24	-20.47	**-23.21**	-56.60	**-57.85**	-57.31
Average	w/o MT	-	4.09	-6.16	0	-13.28	-22.55	-41.46	-44.49	-44.68
	with MT	-6.30	-11.80	**-17.88**	-8.69	-18.51	**-24.03**	-47.42	-46.48	**-47.76**
Δ%DCER										
Overall	w/o MT	-	-8.89	**-22.65**	0	-27.33	**-30.21**	-53.61	-61.59	-59.48
	with MT	-12.93	-18.12	-21.53	-14.73	-28.03	-29.25	-58.99	**-62.92**	-61.63
Average	w/o MT	-	-2.48	-24.15	0	-23.50	-32.70	-47.53	-53.44	-51.71
	with MT	-13.32	-18.90	**-24.31**	-15.62	-30.08	**-33.20**	-51.33	-53.88	**-53.91**

rate (SemER), respectively. DCER is calculated by $\frac{\#domain\ errors}{\#total\ utterances}$. The semantic error measures how many mistakes are done in entity recognition and slot filling, and is calculated by $SemER = \frac{D+I+S}{C+D+S}$ (Su et al., 2018), where D=deletion, I=insertion, S=substitution and C=correct-slots. An IC error is counted as a substitution. All models are evaluated on the same testset and performance is reported as a percentage difference (%Δ) to the baseline few-shot monolingual model.

Table 4, shows the zero-shot and the few-shot performance for the three models with and without MT Arabic data added to finetuning. The multilingual models outperform the baseline monolingual model with the exception of slight 0.29% in SemER in Bi zero-shot model. In the few-shot models, NLU models benefit from a reduction of 15.06% SemER from English alone, and an additional 5.7% reduction from French data with respect to baseline. Table 4, to the right, compares the overall performance of pre-prod models. The impact of cross-lingual transfer learning does not fade even when development Arabic labeled data is added to the model, both multilingual models still outperform the monolingual one. However, adding the mTurk data to finetuning overshadows the impact of French data and the Bi model slightly outperfroms the Tri model. Notice for pre-prod models the benefit of cross-lingual transfer reduces significantly with the addition of MT data. Table 4 demonstrates the transfer learning through translation. By simply using an off-the-shelf machine translator, we can boost the NLU performance on a low-resource target language by 12.79% and 11.7% for the bilingual and the trilingual models, respectively. Adding few-shots and full MTurk data reduces the benefit of MT data to 2-4.8% and 2.7-5.4% for the Bi and Tri models, respectively. For the sake of comparison, we repeat the Bi and Tri experiments on a distilled version of mBERT: distilmBERT (Sanh et al., 2019) (details in Appendix A.3). Results in Table A.2 illustrate the importance of utilizing unlabeled utterances from VA system in pretraining, particularly in early stages of bootstrapping NLU model for a new language, where our model achieves up to 25.1 SemER improvement over distilmBERT in zero-shot setting. Nevertheless, similar TL gains are obtained on distilmBERT with the Tri model outperforming the monolingual model in all settings.

In addition to the overall, i.e., where all utterances have equal contribution to performance (micro-average), Table 4 also reports the average performance per domain, where each domain has equal weight despite its size (macro-average). Considering the average performance and the overall performance, the best performing model in terms of SemER is the trilingual model finetuned on a mix of labeled English, French and Arabic MT data in all zero-shot, and few-shot setting. Although the Bi model beats the Tri model overall in pre-prod setting, the Tri model is still better on average per domain. This suggests that the trilingual model is improving performance for the smaller domains on the target Arabic language. In fact, the Tri model outperforms on average all other models in zero-shot, few-shot, and pre-prod setting. For the latter model setting, we further investigated whether adding English/French data hurt specific domains. We looked at top large domains that did not benefit from adding English and French in Table A.4: AlarmsAndNotifications, SmartHome, and CallingAndCommunication domains with performance reduction of 2.94%, 0.53% and 9.93%.

CallingAndCommunication domain consistently under performed in the Tri model when compared to the Mono model, in all zero-shot, few-shot, and pre-prod models. In these domains, there were issues related to language differences. For example, the top failing utterances in SmartHome were requests to turn off/on appliances. In Arabic turn off/on is a single token (طفّي/أضيئي/شغّلي/سكّري), while in English it is two tokens. Similarly, utterances in the CallingAndCommunication domain are related to finishing the call, in English that would be "hang up", but in Arabic it is again a single token (اقطعي/سكّري/أنهي/أقفلي). This causes imbalance in carrier phrases and a change in the distribution of label sequence for these domains, e.g., compare the two label sequence in the two languages: "turn|Action on|Action light|Device" with "اشغّلي|Action الإضاءة|Device". This can be mitigated by down-sampling English data for these domains, which is left for future experimentation. Overall, even without MT data, the multilingual pre-prod models beat the monolingual model 14 out of 18 domains on the DC task and in 13 out of 18 domains for IC-NER task, clearly showing the effect of cross-lingual transfer of NLU learning from rich English and French source languages to the low-resource Arabic language, despite being linguistically very different.

5 Conclusion

In this paper, we addressed the problem of bootstrapping an NLU model for Arabic from two high-resource Indo-European languages. We presented two multilingual BERT-based models, pretrained and distilled in-house, and compared them to a monolingual Arabic baseline model to explore cross-lingual transfer learning. In an effort to tackle the unique challenges in Arabic language, we adopted a preprocessing step in which we dediacritize the text to reduce the variance and inconsistencies in the data for an already low-resource language. We also split functional affixes and adopt BPE encoding to deal with inflectional affixation in Arabic. Furthermore, in order to reduce the distance between the target language and the source languages we used off-the-shelf machine translator to pretrain and finetune the models, in addition to large-scale open-source Wikipedia and internal datasets. Transfer learning performance gains on the target Arabic language showed a reduction of up to 20.76% in semantic error rate for the IC-NER task and 30.21% in classification error for the DC task for the trilingual model in few-shot setting. Similar cross-lingual learning gains were achieved in a zero-shot setting and pre-prod setting with the improvement gap between monolingual and multilingual models narrowing as data from MT and the Arabic target language is added to finetuning the models.

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. ACL.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105. ACL.

Mohammad Al-Smadi, Saad Al-Zboon, Yaser Jararweh, and Patrick Juola. 2020. Transfer learning for Arabic named entity recognition with deep neural networks. *IEEE Access*, 8:37736–37745.

Fahd Alotaibi and Mark Lee. 2014. A hybrid approach to features representation for fine-grained Arabic named entity recognition. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 984–995. Dublin City University and 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.

Giusepppe Attardi. 2015. Wikiextractor. https://github.com/attardi/wikiextractor.

Yassine Benajiba and Rosso Paolo. 2008. Anercorpdataset. https://camel.abudhabi.nyu.edu/anercorp/.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. ACL.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A dutch BERT model. *arXiv preprint arXiv:1912.09582*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL HLT*, pages 4171–4186.

Ibrahim Abu El-Khair. 2016. 1.5 billion words Arabic corpus. *arXiv preprint arXiv:1611.04033*.

Ahmed El-Kishky, Xingyu Fu, Aseel Addawood, Nahil Sobh, Clare Voss, and Jiawei Han. 2019. Constrained sequence-to-sequence semitic root extraction for enriching word embeddings. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 88–96.

Guillaume Lample et al. 2015. C++ implementation of Neural Machine Translation of Rare Words with Subword Units, with Python API. https://github.com/glample/fastBPE.

Chadi Helwe, Ghassan Dib, Mohsen Shamas, and Shady Elbassuoni. 2020. A semi-supervised BERT approach for Arabic named entity recognition. In *Proceedings of the Fifth Arabic NLP Workshop*, pages 49–57. ACL.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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. ACL.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4163–4174.

Andrew Johnson, Penny Karanasou, Judith Gaspers, and Dietrich Klakow. 2019. Cross-lingual transfer learning for Japanese named entity recognition. In *Proceedings of NAACL HLT (Industry Papers)*, pages 182–189.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnika. 2018. A lexical distance study of Arabic dialects. *Procedia computer science*, 142:2–13.

Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual BERT? *arXiv preprint arXiv:1911.03310*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219. ACL.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462. ACL.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001. ACL.

Edoardo Maria Ponti, Julia Kreutzer, Ivan Vulić, and Siva Reddy. 2021. Modelling latent translations for cross-lingual transfer. *arXiv preprint arXiv:2107.11353*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

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 ACL*, pages 1715–1725. ACL.

Muayad Abdul-Halim Ahmad Shamsan and Abdulmajeed Attayib. 2015. Inflectional morphology in arabic and english: a contrastive study. *International Journal of English Linguistics*, 5(2):139.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. 2017. Aravec: A set of Arabic word embedding models for use in Arabic NLP. *Procedia Computer Science*, 117:256–265.

Chengwei Su, Rahul Gupta, Shankar Ananthakrishnan, and Spyros Matsoukas. 2018. A re-ranker scheme for integrating large scale nlu models. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 670–676. IEEE.

Chao Wang, Judith Gaspers, Thi Ngoc Quynh Do, and Hui Jiang. 2021. Exploring cross-lingual transfer learning with unsupervised machine translation. In *ACL-IJCNLP 2021*, pages 2011–2020. ACL.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 833–844. ACL.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international arabic news corpus-preparation and integration into the CLARIN-infrastructure. In *Proceedings of the 4th Arabic NLP Workshop*, pages 175–182.

A Appendix

A.1 Limitations

The task-specific knowledge transfer proposed in this paper is dependent on the availability of annotated data in high-resource languages for the same NLU tasks: domain/intent classification and NER. That is, although the training data is not in the target language, it still covers the same domains and majority of the intents in the test sets. The ability of the model to generalize to new domains and intents (out-of-domain) in the target language needs further assessment and experimentation. The affix-splitting and de-diacritization preprocessing step proposed in this paper works only for languages with templatic and concatenative morphology, like Arabic and other Semitic languages (e.g., Hebrew). Additionally, the transfer learning gains obtained with machine translation can be limited by the quality of the adopted translator itself. The experiments conducted in this paper uses only a single machine translator for both pretraining and finetuning. Exploring different off-the-shelf machine translators and the impact of the translation quality on NLU tasks needs further experimentation and requires large GPU resources, particularly for pretraining.

Table A.1: Average performance difference (Δ) between models with and without diacritics in 5-fold experiments on NLU tasks (+ve values in favor of model without diacritics).

Δ	DC accuracy	IC accuracy	Slot F1	Frame accuracy
Avg.	0.07	0.38	0.97	2.91
fold1	2.59	1.45	1.34	3.44
fold2	-0.99	1.69	0.24	2.07
fold3	-0.92	0.46	0.86	3.37
fold4	-0.23	-0.31	0.23	1.46
fold5	-0.08	-1.38	2.17	4.21

A.2 Diacritics harm NLU model

In Arabic, short vowels are indicated on letters as diacritics and are used to disambiguate the meaning of the word. Full diacritization is used in classical Arabic, but are often omitted from written texts in MSA. As a result, Arabic has many homographs, that can be distinguished from the context. We conducted a limited-scope study to assess the impact of diacritics on NLU model performance using a set of 1,306 utterances fully diacritized and annotated internally, the utterances cover 12 of the 18 domains used in this paper. We performed a 5-fold cross-validation experiment on the 1,306 set with and without diacritics. We created 5 folds of train-test splits, stratified per domain. Then we duplicate these sets and strip the diacritics. Finally for each of these 10 sets we train a statistical NLU model and evaluate its performance. In addition to training 10 models corresponding to 5 folds of data splits with and without diacritics, each fold was trained and tested 5 times to average the variations in stochastic model performance.

Table A.1 above represents performance averaged across 25 runs for each of the models (with and without diacritics). T-test on domain accuracy, overall intent accuracy and slot F1 showed no significant difference in the means. Overall frame accuracy is slightly better in the model without diacritics, with p=0.01 in two-sample two-tailed t-Tests. To investigate the difference in performance, we further looked at the tokens in the broken utterances in the model with diacritics with respect to the model without diacritics (i.e., utterances that are correctly recognized in the model without diacritics but not in the model with diacritics). We found that on average, the coverage percentage of the tokens in the broken utterances by the training data reduced by 6.59% when adding diacritics. This suggests that diacritics is adding noise through annotation inconsistencies and increasing out-of-vocabulary data, thus reducing model performance.

A.3 Comparison to open-source distilmBERT (Sanh et al., 2019)

We repeat the multilingual experiments on distilmBERT (Sanh et al., 2019), a distilled version of mBERT pretrained and distilled on concatenation of Wikipedia data from 104 languages including English, French, and our target language Arabic. distilmBERT is slightly larger than our distilled model with 6 layers, 768 dimension and 12 heads, compared to our 4-layer distillBERT described in Subsection3.3.1. Because distilmBERT is multilingual, we only run the bilingual and trilingual ver-

Table A.2: SemER and DCER performance of DistilmBERT (Sanh et al., 2019) relative to our monolingual baseline (% change) for Bilingual (**Bi**) and Trilingual (**Tri**) on IC-NER and DC tasks evaluated on 864,127 Arabic utterances. Average performance is across domains. Bold represents best performance within the same setting (zero-shot/few-shot/pre-prod).

		Zero-shot		**Few-shot**		**Pre-prod**	
$\Delta\%$ SemER		**Bi**	**Tri**	**Bi**	**Tri**	**Bi**	**Tri**
Overall	w/o MT	19.8	17.6	3.3	-3.6	-52.1	-55.2
	with MT	-8.5	**-11.3**	-18.4	**-20.5**	-59.4	**-59.9**
Average	w/o MT	34.8	22.1	-0.5	-10.8	-42.1	-44.2
	with MT	-11.9	**-16.3**	-20.9	**-24.1**	-50.0	**-50.9**
$\Delta\%$Overall DCER							
Overall	w/o MT	59.7	41.8	10.8	3.4	-55.7	-55.4
	with MT	-8.3	**-11.1**	**-17.4**	**-17.4**	-61.2	**-61.7**
Average	w/o MT	74.3	46.1	-9.0	-16.0	-48.5	-48.3
	with MT	-19.6	**-24.6**	-30.0	**-32.8**	-54.9	**-55.2**

sions of it, i.e., models finetuned on task-specific annotated data from English, French, and/or MT data. For each model, we finetune different versions of the model one with MT data and one without (w/o) MT data in each of the settings: zero-shot, few-shot, and pre-prod using the same data described in Table3, resulting a total of 12 models. TableA.2 shows the performance of Bi and Tri models using pretrained distilmBERT evaluated on our internally gathered real-world Arabic dataset. The reported SemER and DCER error rates in are relative to our baseline model, so that the values can be compared to our results reported in Table 4. The zero-shot performance w/o MT shows the power of pretraining our in-house models on unlabeled data from a VA system combined with Wikipedia data. Overall, our Tri model beats the corresponding distilmBERT model by 25.1 SemER reduction and 64.45 DCER reduction relative to baseline. However, the gap in performance reduces to 2.71 SemER point reduction in few-shot setting to Tri distilmBERT slightly beating our model with 2.59 SemER in pre-prod setting. This could be attributed to the larger model distilmBERT uses. Nevertheless, a similar trend in the gains obtained from transferring the NLU task-specific knowledge and through MT from English and French in distilmBERT, this generalizes our conclusion that a multilingual model, and particularly the Tri one, outperforms a monolingual model for early stage bootstrapping NLU model for Arabic as seen in Zero-shot, Few-shot and Pre-prod setting.

Table A.3: Zero- and few-shot performance relative (%Δ) to baseline for DC and IC-NER tasks on Arabic.

ΔSemER				**Zero-shot**					**Few-shot**		
Domain	# **Test Utterances**	**Bi**	**Tri**	**Mono + MT**	**Bi + MT**	**Tri + MT**	**Bi**	**Tri**	**Mono + MT**	**Bi + MT**	**Tri + MT**
Overall	864127	0.29	-7.5	-10.64	-12.49	**-19.2**	-15.06	-20.76	-14.24	-20.47	**-23.21**
Average	48007	4.09	-6.16	-6.3	-11.8	**-17.88**	-13.28	-22.55	-8.69	-18.51	**-24.03**
Music	202589	-3.48	-19.67	-25.17	-16.91	**-35.63**	-15.8	-26.54	-28.3	-25.28	**-32.25**
Knowledge	137882	**-79.26**	-64.66	-48.69	-56.14	-53.56	**-61.13**	-47.94	-45.02	-55.02	-54.02
General	131709	42.24	35.17	18.08	16.18	**12.14**	-0.04	**-8.5**	20.11	0.66	-2.7
AlarmsAndNotifications	110817	64.82	57.91	27.87	**13.99**	19.95	5.44	**-2.84**	9.59	**-2.84**	-1.06
SmartHome	68787	14.58	-1.24	14.34	3.7	**-6.69**	2.7	**-17.38**	9.03	-5.7	-14.24
CallingAndCommunication	56787	4.66	6.72	-3.65	**-5.22**	-1.72	-6.04	2.07	-4.18	**-10.36**	-1.05
ToDos	42428	35.44	26.58	14.4	22.41	**13.68**	21.51	**7.41**	8.45	11.58	9.63
Weather	23422	-22.14	-15.67	-14.31	-13.4	**-23.93**	**-33.79**	-28.13	-10.24	-20.31	-23.62
Calendar	23157	-7.79	-27.53	-38.56	-36.95	**-40.68**	-22.35	-31.9	-45.92	-43.44	**-45.93**
Video	17285	-0.68	-12.74	-3.43	-21.46	**-25.45**	-24.2	**-27.1**	-6.05	-21.12	-27.0
AssistantGeneratedContent	16870	174.99	99.59	**71.06**	87.71	85.94	**41.32**	75.49	66.07	72.19	75.58
Apps	8887	-17.85	**-47.81**	24.54	7.35	-29.67	-12.66	**-48.93**	19.05	-33.62	-47.69
Books	8748	-7.76	-28.27	-27.99	**-34.83**	-30.8	-24.33	-36.41	-31.51	-33.24	**-38.33**
Help	7727	31.02	28.79	**5.34**	13.58	6.06	13.91	**1.61**	4.28	10.39	3.33
News	4448	-25.07	-25.07	-42.05	-44.85	**-45.29**	-27.98	-37.12	-43.79	-30.13	**-49.08**
Shopping	2121	12.82	1.09	-10.69	**-17.54**	-15.86	-9.44	-14.71	-8.95	-18.03	**-18.89**
MovieShowTimes	374	-39.31	-48.25	-42.18	**-50.3**	-49.83	-40.82	**-56.11**	-47.51	-44.51	-51.94
Sports	89	21.16	-21.16	**-34.61**	23.08	7.7	30.78	-25.01	**-30.76**	13.48	-28.83
ΔDCER											
Overall	864127	-8.89	**-22.65**	-12.92	-18.13	-21.52	-27.32	**-30.21**	-14.73	-28.04	-29.25
Average	48007	-2.48	-24.15	-13.32	-18.9	**-24.31**	-23.5	-32.7	-15.62	-30.08	**-33.2**
Music	202589	-0.07	-25.45	-28.71	4.3	**-36.43**	-39.98	-57.1	-60.82	**-68.58**	-67.29
Knowledge	137882	**-82.25**	-71.46	-54.14	-61.49	-58.45	-61.96	-59.44	-79.83	-78.99	**-80.83**
General	131709	67.37	39.34	-10.9	**-23.51**	15.14	-74.88	**-82.13**	-76.33	-76.81	-75.85
AlarmsAndNotifications	110817	266.49	229.82	193.52	**139.7**	201.5	11.9	18.81	37.92	**10.89**	21.11
SmartHome	68787	-11.36	**-32.1**	33.32	-14.94	-16.44	3.02	-3.31	-7.96	-4.75	**-29.93**
CallingAndCommunication	56787	26.82	**26.5**	48.31	37.22	28.65	-22.6	**-62.32**	12.77	-51.95	-61.02
ToDos	42428	157.05	57.58	**46.96**	77.85	59.61	-19.11	**-25.62**	32.78	-12.45	13.47
Weather	23422	0.54	-1.5	7.89	6.46	**-13.89**	10.9	**-2.32**	1.05	4.82	-1.83
Calendar	23157	-58.25	**-84.35**	-81.31	-67.32	-76.38	-27.47	**-30.26**	22.14	-21.85	-25.23
Video	17285	7.34	-17.83	-5.47	-19.97	**-22.64**	**-62.12**	-51.18	-50.89	-59.94	-58.97
AssistantGeneratedContent	16870	314.18	120.79	116.59	115.87	**114.9**	-0.02	-27.45	-34.02	-2.87	**-35.15**
Apps	8887	-31.77	**-63.78**	11.48	-14.4	-42.7	40.72	45.03	112.88	**16.99**	88.94
Books	8748	-19.04	-52.47	-62.12	**-72.2**	-65.31	**77.01**	100.26	109.78	113.02	108.13
Help	7727	22.6	21.17	5.12	9.18	**3.79**	-8.97	-8.97	0.11	**-20.8**	-12.13
News	4448	8.3	16.78	-6.95	-18.85	**-23.74**	6.67	**-56.67**	**-56.67**	-16.67	**-56.67**
Shopping	2121	24.14	9.78	0.11	**-13.97**	-9.99	33.23	**22.08**	34.89	29.25	31.19
MovieShowTimes	374	-67.63	-68.6	-71.5	**-75.85**	-69.57	**-31.32**	-27.03	-4.02	-16.72	-24.68
Sports	89	3.33	**-56.67**	-50.0	6.67	-26.67	-19.84	**-29.11**	10.73	-4.84	-16.08

Table A.4: Pre-prod DCER and SemER performance relative (%Δ) to baseline for DC and IC-NER tasks on Arabic.

ΔSemER **Domain**	# **Test** **Utterances**	**Mono**	**Bi**	**Tri**	**Monol** **+ MT**	**Bi** **+ MT**	**Tri** **+ MT**
Overall	864127	-49.32	-55.24	-52.49	-56.6	**-57.85**	-57.31
Average	48007	-41.46	-44.49	-44.68	-47.42	-46.48	**-47.76**
Music	202589	-46.72	-58.09	-51.21	-58.67	-60.5	**-62.93**
Knowledge	137882	-42.29	-55.68	-57.66	-52.81	**-61.29**	-59.64
General	131709	-50.01	-47.1	**-55.04**	-50.54	-46.71	-53.02
AlarmsAndNotifications	110817	-68.49	-66.66	-67.28	**-69.93**	-66.64	-66.99
SmartHome	68787	-54.05	-68.49	-56.93	-60.1	**-69.29**	-59.57
CallingAndCommunication	56787	-55.56	-45.78	-41.64	**-56.86**	-53.13	-46.93
ToDos	42428	-43.56	-41.27	-34.58	**-52.05**	-40.41	-39.96
Weather	23422	-58.86	-52.36	-56.84	**-63.35**	-45.63	-52.72
Calendar	23157	-57.59	-54.83	-60.57	-63.86	-62.73	**-64.23**
Video	17285	-15.78	-31.34	-31.09	-25.94	-33.63	**-35.72**
AssistantGeneratedContent	16870	**-65.39**	-63.49	-51.95	-55.44	-61.86	-43.44
Apps	8887	-64.95	-67.54	-72.14	-69.82	-67.88	**-74.16**
Books	8748	-10.65	-18.31	-16.92	-16.18	-21.07	**-21.65**
Help	7727	5.86	10.24	5.17	**2.6**	7.91	3.29
News	4448	-47.58	-51.96	-49.69	-49.77	-52.32	**-54.26**
Shopping	2121	-8.83	-21.15	-18.89	-18.28	-24.94	**-27.57**
MovieShowTimes	374	-57.75	-59.38	**-63.89**	-61.71	-60.0	-60.89
Sports	89	-19.23	-9.6	-21.16	**-38.46**	-9.6	-32.68
ΔDCER							
Overall	864127	-53.6	-61.58	-59.48	-58.99	**-62.91**	-61.62
Average	48007	-47.53	-53.44	-51.71	-51.33	-53.88	**-53.91**
Music	202589	-66.53	-70.47	-67.01	-69.06	-68.77	**-73.53**
Knowledge	137882	-48.61	-61.49	-62.96	-58.19	**-65.64**	-65.11
General	131709	-30.69	-27.65	-41.26	-34.41	-35.52	**-42.22**
AlarmsAndNotifications	110817	-71.58	-75.34	-66.29	-73.34	**-77.13**	-68.65
SmartHome	68787	-63.99	**-80.65**	-64.55	-70.78	-78.4	-64.69
CallingAndCommunication	56787	-64.15	**-69.59**	-52.2	-67.32	-68.56	-49.49
ToDos	42428	-17.45	**-28.54**	-26.88	-20.87	-20.15	-26.9
Weather	23422	-67.58	-64.27	-65.91	**-70.16**	-50.76	-57.58
Calendar	23157	-89.16	-88.84	-89.53	-89.58	-90.47	**-91.34**
Video	17285	-6.21	-18.02	-20.16	-8.84	-17.7	**-22.75**
AssistantGeneratedContent	16870	**-75.52**	-63.42	-45.24	-39.51	-72.86	-27.67
Apps	8887	-76.62	-77.19	**-83.24**	-79.24	-77.12	-82.92
Books	8748	-8.17	-17.89	-19.55	-14.96	-23.41	**-26.38**
Help	7727	-5.92	-0.44	-5.46	-7.8	-3.83	**-9.63**
News	4448	-29.78	-48.92	-35.68	-46.33	-54.39	**-56.69**
Shopping	2121	-6.32	**-22.32**	-11.11	-9.79	-16.0	-19.67
MovieShowTimes	374	-80.19	-86.47	-85.02	-84.06	**-87.92**	-84.06
Sports	89	-56.67	-56.67	-56.67	**-63.33**	-53.33	**-63.33**