

# Toward Micro-Dialect Identification in Diaglossic and Code-Switched Environments

Muhammad Abdul-Mageed Chiyu Zhang AbdelRahim Elmadany Lyle Ungar<sup>†</sup>

Natural Language Processing Lab, University of British Columbia

<sup>†</sup>Computer and Information Science, University of Pennsylvania

{muhammad.mageed, a.elmadany}@ubc.ca chiyuzh@mail.ubc.ca

<sup>†</sup>ungar@cis.upenn.edu

## Abstract

Although prediction of dialects is an important language processing task, with a wide range of applications, existing work is largely limited to coarse-grained varieties. Inspired by geolocation research, we propose the novel task of *Micro-Dialect Identification (MDI)* and introduce MARBERT, a new language model with striking abilities to predict a fine-grained variety (as small as that of a city) given a *single, short* message. For modeling, we offer a range of novel spatially and linguistically-motivated multi-task learning models. To showcase the utility of our models, we introduce a new, large-scale dataset of Arabic micro-varieties (low-resource) suited to our tasks. MARBERT predicts micro-dialects with 9.9%  $F_1$ ,  $\sim 76\times$  better than a majority class baseline. Our new language model also establishes new state-of-the-art on several external tasks.<sup>1</sup>

## 1 Introduction

Sociolinguistic research has shown how language varies across geographical regions, even for areas as small as different parts of the same city (Labov, 1964; Trudgill, 1974). These pioneering studies often used field work data from a handful of individuals and focused on small sets of carefully chosen features, often phonological. Inspired by this early work, researchers have used geographically tagged social media data from hundreds of thousands of users to predict user location (Paul and Dredze, 2011; Amitay et al., 2004; Han et al., 2014; Rahimi et al., 2017; Huang and Carley, 2019b; Tian et al., 2020; Zhong et al., 2020) or to develop language identification tools (Lui and Baldwin, 2012; Zubiaga et al., 2016; Jurgens et al., 2017a; Dunn and Adams, 2020). Whether it is possible at all to predict the *micro-varieties*<sup>2</sup> of the same general

language is a question that remains, to the best of our knowledge, unanswered. In this work, our goal is to investigate this specific question by introducing the *novel* task of *Micro-Dialect Identification (MDI)*. Given a single sequence of characters (e.g., a single tweet), the goal of MDI is to predict the particular micro-variety (defined at the level of a city) of the *community of users* to which the posting user belongs. This makes MDI different from geolocation in at least two ways: in geolocation, (1) a model consumes a *collection of messages* (e.g., 8-85 messages in popular datasets (Huang and Carley, 2019a) and (2) predicts the *location* of the posting user (i.e., user-level). In MDI, a model takes as input a *single message*, and predicts the *micro-variety* of that message (i.e., message-level).

While user location and micro-dialect (MD) are conceptually related (e.g., with a tag such as *Seattle* for the first, and *Seattle English*, for the second), they arguably constitute two *different* tasks. This is because location is an attribute of a *person* who authored a Wikipedia page (Overell, 2009) or posted on Facebook (Backstrom et al., 2010) or Twitter (Han et al., 2012), whereas MD is a characteristic of *language within a community of speakers* who, e.g., use similar words to refer to the same concepts in real world or pronounce certain sounds in the same way. To illustrate, consider a scenario where the same user is at different locations during different times. While a geolocation model is required to predict these *different* locations (for that *same* person), an MDI model takes as its target predicting the *same* micro-variety for texts authored by the person (regardless of the user location). After all, while the language of a person can, and does, change when they move from one region to another, such a change takes time.

Concretely, although to collect our data we use location as an initial *proxy* for user MD, we do not just exploit data where  $n$  number of posts (usually

<sup>1</sup>Our labeled data and models will be available at: <https://github.com/UBC-NLP/microdialects>.

<sup>2</sup>We use *micro-variety* and *micro-dialect* interchangeably.

$n=10$ ) came from the location of interest (as is usually the case for geolocation). Rather, to the extent it is possible, we take the additional *necessary* step of manually verifying that a user does live in a given region, and has not moved from a different city or country (at least recently, see Section 2). We hypothesize that, if we are able to predict user MD based on such data, we will have an empirical evidence suggesting MD does exist and can be detected. As it turns out, while it is almost impossible for humans to detect MD (see Section 2.3 for a human annotation study), our models predict varieties as small as those of cities surprisingly well (9.9%  $F_1$ ,  $\sim 76\times$  higher than a majority class baseline based on a single, short message) (Sections 6 and 7).

**Context.** MDI can be critical for multilingual NLP, especially for social media in global settings. In addition to potential uses to improve machine translation, web data collection and search, and pedagogical applications (Jauhiainen et al., 2019), MDI can be core for essential real-time applications in health and well-being (Paul and Dredze, 2011), recommendation systems (Quercia et al., 2010), event detection (Sakaki et al., 2010), and disaster response (Carley et al., 2016). Further, as technology continues to play an impactful role in our lives, access to nuanced NLP tools such as MDI becomes an issue of equity (Jurgens et al., 2017b). The great majority of the world’s currently known 7,111 living languages,<sup>3</sup> however, are not NLP-supported. This limitation also applies to closely related languages and varieties, even those that are widely spoken.

We focus on one such situation for the Arabic language, a large collection of similar varieties with  $\sim 400$  million native speakers. For Arabic, currently available NLP tools are limited to the standard variety of the language, Modern Standard Arabic (MSA), and a small set of dialects such as Egyptian, Levantine, and Iraqi. Varieties comprising dialectal Arabic (DA) differ amongst themselves and from MSA at various levels, including phonological and morphological (Watson, 2007), lexical (Salameh and Bouamor, 2018; Abdul-Mageed et al., 2018; Qwaider et al., 2018), syntactic (Benmamoun, 2011), and sociological (Bassiouney, 2020). Most main Arabic dialects, however, remain understudied. The situation is even more acute for MDs, where very limited knowledge (if

at all) currently exists. The prospect of research on Arabic MDs is thus large.

A major limitation to developing robust and equitable language technologies for Arabic language varieties has been absence of large, diverse data. A number of pioneering efforts, including shared tasks (Zampieri et al., 2014; Malmasi et al., 2016; Zampieri et al., 2018), have been invested to bridge this gap by collecting datasets. However, these works either depend on automatic geocoding of user profiles (Abdul-Mageed et al., 2018), which is not quite accurate, as we show in Section 2; use a small set of dialectal seed words as a basis for the collection (Zaghouani and Charfi, 2018; Qwaider et al., 2018), which limits text diversity; or are based on translation of a small dataset of sentences rather than naturally-occurring text (Salameh and Bouamor, 2018), which limits the ability of resulting tools. The recent Nuanced Arabic Dialect Identification (NADI) shared task (Abdul-Mageed et al., 2020a) aims at bridging this gap.

In this work, following Gonçalves and Sánchez (2014); Doyle (2014); Sloan and Morgan (2015), among others, we use *location as a surrogate for dialect* to build a very large scale Twitter dataset ( $\sim 6$  billion tweets), and automatically label a subset of it ( $\sim 500$ M tweets) with coverage for all 21 Arab countries at the nuanced levels of state and city (i.e., micro-dialects). In a departure from geolocation work, we then *manually* verify user locations, excluding  $\sim 37\%$  of users. We then exploit our data to develop highly effective hierarchical and multi-task learning models for detecting MDs.

Other motivations for choosing Arabic as the context for our work include that (1) Arabic is a *diaglossic* language (Ferguson, 1959; Bassiouney, 2020) with a so-called ‘High’ variety (used in educational and formal settings) and ‘Low’ variety (used in everyday communication). This allows us to exploit *diaglossia* in our models. In addition, (2) for historical reasons, different people in the Arab world *code-switch* in different foreign languages (e.g., English in Egypt, French in Algeria, Italian in Libya). This affords investigating the impact of code-switching on our models, thereby bringing yet another novelty to our work. Further, (3) while recent progress in *transfer learning* using language models such as BERT (Devlin et al., 2018) has proved strikingly useful, Arabic remains dependent on multilingual models such as mBERT trained on the restricted Wikipedia domain with limited

<sup>3</sup>Source: <https://www.ethnologue.com>.

data. Although an Arabic-focused language model, AraBERT (Baly et al., 2020), was recently introduced, it is limited to MSA rather than dialects. This makes AraBERT sub-optimal for social media processing as we show empirically. We thus present a novel Transformer-based Arabic language model, MARBERT, for MDs. Our new model exploits a massive dataset of 1B posts, and proves very powerful: It establishes new SOTA on a wide range of tasks. Given the impact self-supervised language models such as BERT have made, our work has the potential to be a key milestone in *all* Arabic (and perhaps multilingual) NLP.

To summarize, we offer the following contributions: (1) We collect a massive dataset from Arabic social media and exploit it to develop a large human-labeled corpus for Arabic MDs. (2) For modeling, we introduce a novel, *spatially* motivated hierarchical attention multi-task learning (HA-MTL) network that is suited to our tasks and that proves highly successful. (3) We then introduce *linguistically* guided multi-task learning models that leverage the diaglossic and code-switching environments in our social data. (4) We offer a new, powerful Transformer-based language model trained with self-supervision for Arabic MDs. (5) Using our powerful model, we establish new SOTA results on several external tasks.

The rest of the paper is organized as follows: In Section 2, we introduce our methods of data collection and annotation. Section 3 is about our experimental datasets and methods. We present our various models in Section 4 and our new micro-dialectal model, MARBERT, in Section 5. We investigate model generalization in Section 6, and the impact of removing MSA from our data in Section 7. Section 8 is a discussion of our findings. We compare to other works in Section 9, review related literature in Section 10, and conclude in Section 11.

## 2 Data Acquisition and Labeling

We first acquire a large user-level dataset covering the whole Arab world. We then use information in user profiles (available only for a subset of users) to automatically assign city, state, and country labels to each user. Since automatic labels can be noisy (e.g., due to typos in city names, use of different languages in user profiles), we manually fix resulting errors. To further account for issues with human mobility (e.g., a user from one country moving to another), we manually inspect user profiles, tweets,

and network behavior and verify assigned locations. Finally, we propagate city, state, and country labels from the user to the tweet level (each tweet gets the label assigned to its user). We now describe our data methods in detail.

### 2.1 A Large User-Level, Tagged Collection

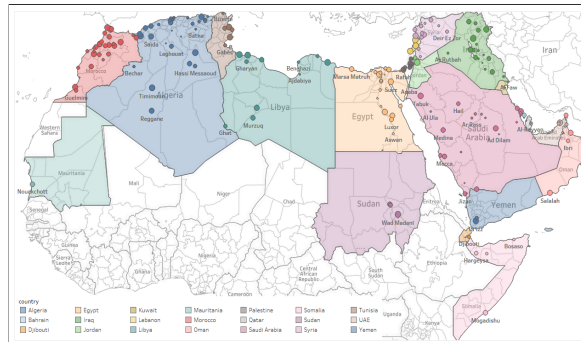


Figure 1: All 21 Arab countries in our data, with states demarcated in thin black lines within each country. All 319 cities from our user location validation study, in colored circles, are overlaid within respective states.

To develop a large scale dataset of Arabic varieties, we use the Twitter API to crawl up to 3,200 tweets from  $\sim 2.7$  million users collected from Twitter with bounding boxes around the Arab world. Overall, we acquire  $\sim 6$  billion tweets. We then use the Python geocoding library *geopy* to identify user location in terms of countries (e.g., Morocco) and cities (e.g., Beirut).<sup>4</sup> Out of the 2.7 million users, we acquired *both* ‘city’ and ‘country’ label for  $\sim 233$ K users who contribute  $\sim 507$ M tweets. The total number of cities initially tagged was 705, but we manually map them to only 646 after correcting several mistakes in results returned by *geopy*. *Geopy* also returned a total of 235 states/provinces that correspond to the 646 cities, which we also manually verified. We found all state names to be correct and to correspond to their respective cities and countries.<sup>5</sup>

### 2.2 Validation of User Location

Even after manually correcting location labels, it cannot be guaranteed that a user *actually* belongs to

<sup>4</sup>*Geopy* (<https://github.com/geopy>) is a client for several popular geocoding web services aiming at locating the coordinates of addresses, cities, countries, and landmarks across the world using third-party geocoders. In particular, we use the Nominatim geocoder for OpenStreetMap data (<https://nominatim.openstreetmap.org>). With Nominatim, *geopy* depends on user-provided geographic information in Twitter profiles such as names of countries or cities to assign user location.

<sup>5</sup>More information about manual correction of city tags is in Section A.1 in the Appendix.

(i.e., is a local of) the region (city, state, and country) they were assigned. Hence, we manually verify user location through an annotation task. To the extent it is possible, this helps us avoid assigning false MD labels to users whose profile information was captured rightly in the previous step but who indeed are *not* locals of automatically labeled places. Before location verification, we exclude *cities* that have  $< 500$  tweets and *users* with  $< 30$  tweets from the data. This initially gives us 417 cities. We then ask two native Arabic annotators to consider the automatic label for each task (city and country)<sup>6</sup> and assign one label from the set  $\{local, non-local, unknown\}$  per task for each user in the collection. We provide annotators with links to users’ Twitter profiles, and instruct them to base their decisions on each user’s network and posting content and behavior. As a result, we found that 81.00% of geopy tags for country are correct, but only 62.29% for city. This validates the need for the manual verification. Ultimately, we could verify a total of 3,085 users for *both* country and city from all 21 countries but only from 319 cities.<sup>7</sup> Figure 1 shows a map of all 21 Arab countries, each divided into its states with cities overlaid as small colored circles.

### 2.3 Can Humans Detect Micro-Dialect?

We were curious to know whether humans can identify micro-dialect from a *single message*, and so we performed a small annotation study. We extracted a random set of 1,050 tweets from our labeled collection and asked two native speakers from two non-neighboring Arab countries to tag each tweet with a *country* then (choosing from a drop-down menu) a *state* label. Annotators found the state-level task very challenging (or rather “impossible”, to quote one annotator) and so we did not complete it. Hence, we also did not go to the level of *city* since it became clear it will be almost impossible for humans. For country, annotators reported trying to identify larger regions (e.g. Western Arab world countries), then pick a specific country (e.g., Morocco). To facilitate the task, we asked annotators to assign an “unknown” tag when unsure. We calculated inter-annotator agreement and found it at Cohen’s (Cohen, 1960) Kappa ( $K$ )=0.16 (“poor” agreement). When we calculate the subset of data

<sup>6</sup>Note that we have already manually established the link between states and their corresponding cities and countries.

<sup>7</sup>More information about manual user verification is in Section A.2 of the Appendix.

where both annotators assigned an actual country label (i.e., rather than “unknown”;  $n=483$  tweets), we found the Kappa ( $K$ ) to increase to 0.47 (“moderate” agreement). Overall, the annotation study emphasizes challenges humans face when attempting to identify dialects (even at the level of country sometimes).

## 3 Datasets and Methods

### 3.1 Datasets

**Preprocessing.** To keep only high-quality data, we remove all retweets, reduce all consecutive sequences of the same character to only 2, replace usernames with  $\langle USER \rangle$  and URLs with  $\langle URL \rangle$ , and remove all tweets with less than 3 Arabic words. This gives  $\sim 277.4K$  tweets. We tokenize input text only lightly by splitting off punctuation.<sup>8</sup> Ultimately, we extract the following datasets for our experiments:

**Micro-Ara (Monolingual).** Extracted from our manually verified users, this is our core dataset for modeling. We randomly split it into 80% training (TRAIN), 10% development (DEV), and 10% test (TEST). To limit GPU time needed for training, we cap the number of tweets in our TRAIN in any given country at 100K. We describe the distribution of classes in Micro-Ara in Tables B.1 and B.2 in the Appendix. We note that Micro-Ara is reasonably balanced. Table 1 shows our data splits.

**CodSw (Code-Switching).** As explained in Section 1, Arabic speakers code-switch to various foreign languages. We hypothesize the distribution of foreign languages will vary across different regions (which proves to be true, as we show in Figure 2), thereby providing modeling opportunities that we capture in a multi-task setting (in Section 4.5). Hence, we introduce a code-switching dataset (CodSw) by tagging the non-Arabic content in all tweets in our wider collection with the langid tool (Lui and Baldwin, 2012). Keeping only tweets with at least 3 Arabic words and at least 4 non-Arabic words, we acquire  $\sim 934K$  tweets. CodSw is diverse, with a total of 87 languages. We split CodSw as is shown in Table 1.

**DiaGloss (Diaglossia).** We also explained in Section 1 that Arabic is a diaglossic language, with MSA as the “High” variety and dialects as the “Low” variety. MDs share various linguistic features (e.g., lexica) with MSA, but to varying degrees. We use an auxiliary task whose goal is to tease apart MSA from dialectal varieties. We

<sup>8</sup>For most DA varieties, there are no available tokenizers.

use existence of diacritics (at least 5) as a proxy for MSA,<sup>9</sup> and direct responses (vs. tweets or retweets) as a surrogate for dialectness. In each class, we keep 500K tweets, for a total of 1M tweets split as in Table 1. We refer to this dataset as DiaGloss.

Datasets	Train	Dev	Test
Micro-Ara	1,099,711	202,509	202,068
CodSw	747,173	93,431	93,565
DiaGloss	800K	100K	100K

Table 1: Splits of our datasets. **Micro-Ara:** City-verified dataset for MDs. **CodSw:** Code-switching dataset from our automatically-tagged collection. **DiaGloss:** MSA vs. DA data to approximate diaglossia.

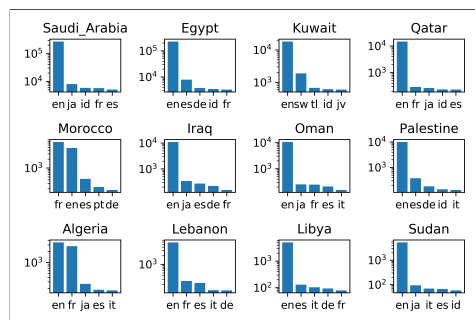


Figure 2: Code-switching over select countries (with different code-switching profiles) in CodSw.

### 3.2 Methods

**BiGRUs and BERT.** We perform dialect identification at the country, state, and city levels. We use two main neural network methods, Gated Recurrent Units (GRUs) (Cho et al., 2014), a variation of Recurrent Neural Networks (RNNs), and Google’s Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018). We model each task independently, but also under multi-task conditions.

**Multi-Task Learning.** *Multi-Task Learning (MTL)* is based on the intuition that many real-world tasks involve predictions about closely related labels or outcomes. For related tasks, MTL helps achieve inductive transfer between the various tasks by leveraging additional sources of information from some of the tasks to improve performance on the target task (Caruana, 1993). By using training signals for related tasks, MTL allows a learner to prefer hypotheses that explain more than one task (Caruana, 1997) and also helps regularize models. In some of our models, we leverage MTL by training a *single* network for our city, state, and country tasks where network layers are shared but with an independent output for each of the 3 tasks.

<sup>9</sup>Unlike dialect, MSA is usually diacritized.

## 4 Models

Here, we describe our baselines and present our MDI models. These are (i) our single- and multi-task BiGRU models (Sections 4.1, 4.2, and 4.3), (ii) single-task BERT (Section 4.4), and (iii) multi-task BERT (Section 4.5).

### 4.1 Baselines

For all our experiments, we remove diacritics from the input text. We use two baselines: the majority class in TRAIN (Baseline I) and a single-task BiGRU (Baseline II, described below). For all our experiments, we tune model hyper-parameters and identify best architectures on DEV. We run all models for 15 epochs (unless otherwise indicated), with early stopping ‘patience’ value of 5 epochs, choosing the model with highest performance on DEV as our best model. We then run each best model on TEST, and report *accuracy* and *macro  $F_1$  score*.<sup>10</sup>

**Single-Task BiGRUs.** As a *second baseline* (Baseline II), we build 3 independent networks (each for one of the 3 tasks) using the same architecture and model capacity. Each network has 3 hidden BiGRU layers, with 1,000 units each. More information about each of these networks and how we train them is in Section C.1 in the Appendix. Table 2 presents our results on TEST.

### 4.2 Multi-Task BiGRUs

With MTL, we design a single network to learn the 3 tasks simultaneously. In addition to our hierarchical attention MTL (HA-MTL) network, we design two architectures that differ as to how we endow the network with the *attention mechanism*. We describe these next. We provide illustrations of our MTL networks in Figures C.1 and C.2 in the Appendix.

**Shared and Task-Specific Attention.** We first design networks with attention at the same level in the architecture. Note that we use the same hyper-parameters as the single-task networks. We have two configurations:

**Shared Attention.** This network has 3 hidden BiGRU layers, each of which has 1,000 units per layer (500 in each direction).<sup>11</sup> All the 3 layers are shared across the 3 tasks, including the third layer.

<sup>10</sup>We include tables with results on DEV in Section C in the Appendix.

<sup>11</sup>Again, 4 hidden-layered network for both the *shared* and *task-specific* attention settings were sub-optimal and so we do not report their results here.

Only the third layer has attention applied. We call this setting *MTL-common-attn*.

**Task-Specific Attention.** This network is similar to the previous one in that the first two hidden layers are shared, but differs in that the third layer (attention layer) is task-specific (i.e., independent for each task). We call this setting *MTL-spec-attn*. This architecture will allow each task to specialize its own attention within the same network. As Table 2 shows, both *MTL-common-attn* and *MTL-spec-attn* improve over each of the two baselines (with first performing generally better).

### 4.3 Hierarchical Attention MTL (HA-MTL)

Instead of a ‘flat’ attention, we turn to hierarchical attention (spatially motivated, e.g., by how a smaller region is a part of a larger one): We design a single network for the 3 tasks but with supervision at different layers. Overall, the network has 4 BiGRU layers (each with a total of 1,000 units), the bottom-most of which has no attention. Layers 2, 3, and 4 each has dot-product attention applied, followed directly by one task-specific fully-connected layer with softmax for class prediction. In the two scenarios, state is supervised at the middle layer. These two architectures allow information flow with different granularity: While the city-first network tries to capture what is in the physical world a more fine-grained level (city), the country-first network does the opposite. Again, we use the same hyper-parameters as the single-task and MTL networks, but we use a dropout rate of 0.70 since we find it to work better. As Table 2 shows, our proposed HA-MTL models significantly outperform single-task and other BiGRU MTL models. They also outperform our Baseline II with 12.36%, 10.01%, and 13.22% acc on city, state, and country prediction respectively, thus demonstrating their effectiveness on the task.

Setting	City		State		Country	
	acc	F1	acc	F1	acc	F1
Baseline I	1.31	0.01	3.11	0.03	9.19	0.80
Baseline II	1.65	0.25	6.13	1.92	31.14	15.84
MTL (common-attn)	2.86	0.74	5.12	1.01	26.51	12.41
MTL (spec-attn)	2.40	0.68	4.60	0.90	27.04	10.98
HA-MTL (city 1 <sup>st</sup> )	<b>14.01</b>	<b>14.02</b>	<b>16.14</b>	<b>15.90</b>	<b>44.36</b>	32.14
HA-MTL (cntry 1 <sup>st</sup> )	13.23	13.06	15.84	15.40	44.17	<b>32.37</b>
mBERT	19.33	19.45	<b>21.24</b>	21.67	47.74	38.12
MTL-mBERT (DiGls)	<b>19.88</b>	<b>20.11</b>	21.04	<b>21.69</b>	48.30	<b>38.34</b>
MTL-mBERT (CodSw)	19.47	19.86	20.76	21.47	<b>48.61</b>	38.20

Table 2: Performance on TEST. **Baseline I:** majority in TRAIN. **Baseline II:** single task Attn-BiGRU. **MTL-mBERT (DiGls):** Multi-Task BERT with diaglossia. **CodSw:** code-switching.

### 4.4 Single-Task BERT

We use the BERT-Base, Multilingual Cased model released by the authors.<sup>12</sup> For fine-tuning, we use a maximum sequence length of 50 words and a batch size of 32. We set the learning rate to 2e-5. We train for 15 epochs, as mentioned earlier. As Table 2 shows, BERT performs consistently better on the three tasks. It outperforms the best of our two HA-MTL networks with an acc of 5.32% (city), 5.10% (state), 3.38% (country). To show how a small network (and hence deployable on machines with limited capacity with quick inference time) can be trained on knowledge acquired by a bigger one, we distill (Hinton et al., 2015; Tang et al., 2019) BERT representation (big) with a BiGRU (small). We provide related results in Section C.2 in the Appendix.

### 4.5 Multi-Task BERT

We investigate two linguistically-motivated auxiliary tasks trained with BERT, as follows:

**Exploiting Diaglossia.** As Table 2 shows, a diaglossia-based auxiliary task improves over the single task BERT for both city (0.55% acc) and country (0.56% acc).

**Exploiting Code-Switching.** We run 4 experiments with our CodSw dataset, as follows: (1) with the two tasks supervised at the city level, (2) at the country level, (3 & 4) with the levels reversed (city-country vs. country-city). Although the code-switching dataset is automatically labeled, we find that when we supervise with its country-level labels, it helps improve our MDI on city (0.14% acc) and on country (0.87% acc). Related results are shown in Table 2. We now describe our new language model, MARBERT.

## 5 MARBERT: A New Language Model

We introduce MARBERT, a new language model trained with self-supervision on 1B tweets from from our unlabeled Twitter collection (described in Section 2). We train MARBERT on 100K word-Piece vocabulary, for 14 epochs with a batch size of 256 and a maximum sequence length of 128. Training took 14 days on 8 Google Cloud TPUs. We use the same network architecture as mBERT, but without the next sentence prediction objective since tweets are short. MARBERT has much larger token count than BERT (Devlin et al., 2018) (15.6B

<sup>12</sup><https://github.com/google-research/bert/blob/master/multilingual.md>.

vs. 3.3B), and is trained on  $5\times$  bigger data than AraBERT (Baly et al., 2020) (126GB vs. 24GB of text). Unlike AraBERT, which is focused on MSA, MARBERT has diverse coverage of dialects in addition to MSA. As Table 3 shows, MARBERT significantly outperforms *all* other models across the 3 tasks, with improvements of 4.13% and 3.54% acc over mBERT and AraBERT respectively on the country task. We also run a set of **MTL experiments** with MARBERT, fine-tuning it with a diagglossia auxiliary task, a code-switching auxiliary task, and *both* diagglossia and code-switching as auxiliary tasks. As Table 3 shows, MTL does not bring acc improvements, and helps the country task only slightly (0.30% acc gain with CodSw). This reflects MARBERT’s already-powerful representation, with little need for MTL. We provide an *error analysis* of MARBERT’s MDI in Section E.1 of the Appendix.

Setting Eval Metric	City		State		Country	
	acc	F1	acc	F1	acc	F1
mBERT (Devlin et al. (2018))	19.33	19.45	21.24	21.67	47.74	38.12
AraBERT (Baly et al. (2020))	18.82	18.73	20.73	20.87	48.33	38.09
MARBERT (Ours)	<b>20.78</b>	<b>20.41</b>	<b>22.97</b>	<b>22.58</b>	<b>51.87</b>	<b>42.17</b>
MTL-MARBERT (DiaGloss)	20.19	<b>20.60</b>	<b>23.22</b>	22.97	51.53	41.75
MTL-MARBERT (CodSw)	<b>20.77</b>	20.56	23.21	<b>23.16</b>	51.78	<b>42.36</b>
MTL-MARBERT (CSD)	20.76	20.23	23.18	23.15	<b>52.17</b>	42.27

Table 3: MARBERT (ours) outperforms mBERT and AraBERT (TEST results). **CSD**: DiaGloss+CodSw.

## 6 Model Generalization

For the experiments reported thus far, we have split our Micro-Ara (monolingual) dataset randomly at the tweet level. While this helped us cover the full list of our 319 cities, including cities from which we have as few as a single user, this split does not prevent tweets from the *same* user to be divided across TRAIN, DEV, and TEST. In other words, while the tweets across the splits are unique (not shared), users who posted them are not unique. We hypothesize this may have the consequence of allowing our models to acquire knowledge about user identity (idiolect) that interact with our classification tasks. To test this hypothesis, we run a set of experiments with different data splits where users in TEST are *never* seen in TRAIN. To allow the model to see enough users during training, we split the data only into TRAIN and TEST and use no DEV set. We use the same hyper-parameters identified on previous experiments. An exception is the number of epochs, where we report the best epoch identified on TEST. To alleviate the concern about absence of a DEV set, we run each experiment 3 times. Each time we choose a different TEST set,

and we average out performance on the 3 TEST sets. This is generally similar to cross-validation.

For this set of experiments, we first remove all cities from which we have only one user (79 cities) and run experiments across 3 different settings (**narrow**, **medium**, and **wide**).<sup>13</sup> We provide a description of these 3 settings in Section D.1 in the Appendix. For the *narrow* setting only, we also run with the same code-switching and diagglossic auxiliary tasks (individually and combined) as before. We use mBERT fine-tuned on each respective TRAIN as our baseline for the current experiments.

As Table 4 shows, MARBERT significantly ( $p < 0.01$ ) outperforms the strong mBERT baseline across the 3 settings. With the *narrow* setting on MDs, MARBERT reaches 8.12% F<sub>1</sub> (61 cities). These results drop to 5.81% (for *medium*, 116 cities) and 3.59% (for *wide*, 240 cities). We also observe a positive impact<sup>14</sup> from the combined code-switching and diagglossic auxiliary tasks. All results are also *several folds better* than a majority class city baseline (city of Abu Dhabi, UAE; not shown in Table 4). For example, results acquired with the two (combined) auxiliary tasks are  $4.7\times$  better in acc and  $229\times$  better for F<sub>1</sub> than the majority class.

Importantly, although results in Table 4 are not comparable to those described in Section 5 (due to the different data splits), these results suggest that our powerful transformer models in Section 5 may have made use of user-level information (which may have caused inflated performance). To further investigate this issue in a reasonably comparable set up, we apply the models based on the narrow, medium and wide settings *and* our single task MARBERT model (shown in Table 3) *all* to a completely new test set. This additional evaluation iteration, which we describe in Section D.2 of the Appendix, verifies the undesirable effect of sharing users between the data splits.<sup>15</sup> For this reason, we strongly advise against sharing users across data splits for tweet-level tasks *even* if the overall dataset involves several thousand users.

## 7 Impact of MSA

Our efforts thus far focused on teasing apart tweets regardless of their (degree of) dialectness. Our

<sup>13</sup>We give each setting a name that reflects its respective geographical coverage. For example, *wide* has wider coverage than *medium*, which in turn has wider coverage than *narrow*.

<sup>14</sup>But not significant ( $p < 0.07$  for city-level with *narrow*), since MARBERT is already powerful.

<sup>15</sup>Relevant results are in Appendix Table D.1.

Setting	City		State		Country	
	acc	F1	acc	F1	acc	F1
mBERT-wide	4.58	3.18	7.88	4.05	35.39	24.73
MARBERT-wide	<b>5.06</b>	<b>3.59</b>	<b>8.92</b>	<b>4.77</b>	<b>39.49</b>	<b>28.77</b>
mBERT-medium	6.39	4.92	9.33	6.11	38.48	28.16
MARBERT-medium	<b>7.07</b>	<b>5.81</b>	<b>10.53</b>	<b>7.22</b>	<b>42.51</b>	<b>32.37</b>
mBERT-narw	9.60	6.63	12.08	8.55	51.32	33.87
MARBERT-narw	<b>11.66</b>	<b>8.12</b>	<b>14.62</b>	<b>10.09</b>	<b>57.27</b>	<b>39.93</b>
MTL-MARBERT-narw (DiaGloss)	11.76	8.35	14.37	10.23	56.74	39.68
MTL-MARBERT-narw (CodSw)	11.59	8.33	15.45	10.27	57.78	40.32
MTL-MARBERT-narw (CSD)	<b>11.78</b>	<b>8.44</b>	<b>15.25</b>	<b>10.34</b>	<b>57.75</b>	<b>40.46</b>

Table 4: Performance on TEST sets with unique users (i.e., users with no data in TRAIN). Setting names are suffixed to each model. **CSD**: DiaGloss+CodSw.

dataset comprises posts either solely in MSA or in MSA mixed with dialectal content. Since MSA is shared across different regions, filtering it out is likely to enhance system ability to distinguish posts across our 3 classification tasks. We test (and confirm) this hypothesis by removing MSA from both TRAIN and TEST in our *narrow* setting data (from Section 6) and fine-tuning MARBERT on the resulting (‘dialectal’) data only.<sup>16</sup> As Table 5 shows, this procedure results in higher performance across the 3 classification tasks.<sup>17</sup> For micro-dialects, performance is at 14.09% acc. and 9.87% F<sub>1</sub>. Again, this performance is much better (3.3× better acc and 75.9× better F<sub>1</sub>) than the majority class baseline (city of Abu Dhabi, UAE, in 2 of our 3 runs).

Setting	City		State		Country	
	acc	F1	acc	F1	acc	F1
mBERT-narrow-DA	11.72	8.45	15.41	10.95	69.30	45.46
MARBERT-narrow-DA	<b>14.09</b>	<b>9.87</b>	<b>17.37</b>	<b>12.90</b>	<b>75.35</b>	<b>51.03</b>

Table 5: Performance on predicted dialectal data.

## 8 Discussion

As we showed in Sections 6 and 7, our models are able to predict variation at the city-level significantly better than *all* competitive baselines. This is the case even when we do not remove MSA content, but better results are acquired after removing it. A question arises as to whether the models are indeed capturing micro-linguistic variation between the different cities, or simply depending on different topical and named entity distributions in the data. To answer this question, we visualize attention in  $\sim 250$  examples from our TEST set using one of our MARBERT-narrow models reported

<sup>16</sup>To filter out MSA, we apply an in-house MSA vs. dialect classifier ( $acc = 89.1\%$ ,  $F_1 = 88.6\%$ ) on the data, and remove tweets predicted as MSA. More information about the MSA vs. dialect model is in Section D.3 of the Appendix. We cast more extensive investigation of the interaction between dialects and MSA vis-a-vis our classification tasks, including based on manually-filtered MSA, as future research.

<sup>17</sup>MARBERT is significantly better than mBERT with  $p < 0.03$  for city,  $p < 0.01$  for state, and  $p < 0.0004$  for country.

in Table 4.<sup>18</sup> Our analysis reveals that the model *does* capture micro-dialectal variation. We provide two example visualizations in Section E in the Appendix demonstrating the model’s micro-dialectal predictive power. Still, we also observe that the model makes use of especially names of places. For this reason, we believe future research should control for topical and named entity cues in the data.

## 9 Comparisons and Impact

**Comparisons to Other Dialect Models.** In absence of similar-spirited nuanced language models, we compare our work to existing models trained at the country level. These include the tweet-level 4-country SHAMI (Qwaider et al., 2018) which we split into TRAIN (80%), DEV (10%), and TEST (10%) for our experiments, thus using less training data than Qwaider et al. (2018) (who use cross-validation). We also compare to (Zhang and Abdul-Mageed, 2019), the winning team in the the 21-country MADAR Shared Task-2 (Bouamor et al., 2019b). Note that the shared task targets user-level dialect based on a collection of tweets, which our models are not designed to directly predict (since we rather take a single tweet input, making our task harder).<sup>19</sup> For the purpose, we train two models, one on MADAR data (shared tasks 1 and 2) and another on our Micro-Ara+MADAR data. We also develop models using the 17-country Arap-Tweet (Zaghouani and Charfi, 2018), noting that authors did not preform classification on their data and so we include a unidirectional 1-layered GRU, with 500 units as a baseline for Arap-Tweet. Note that we do not report on the dataset described in Abdul-Mageed et al. (2018) since it is automatically labeled, and so is *noisy*. We also do not compare to the dataset in Salameh and Bouamor (2018) since it is small, *not naturally occurring* (only 2,000 translated sentences per class), and the authors have already reported linear classifiers outperforming a deep learning model due to small data size. As Table 7 shows, our models achieve new SOTA on all three tasks with a significant margin. Our results on MADAR show that if we have up to 100 messages from a user, we can detect their MD at 80.69% acc.

**Impact on External Tasks:** We further demon-

<sup>18</sup>Namely, we use the model fine-tuned in split B in Table D.2.

<sup>19</sup>We follow (Zhang and Abdul-Mageed, 2019) in assigning a user-level label based on message-level majority class.



Dataset	#cls	Model	acc	F1
Arap-TWT	17	GRU-500	38.79	39.17
		mBERT	54.67	55.07
		MARBERT	<b>57.00</b>	<b>57.21</b>
SHAMI	4	Qwaider et al. (2018)	70.00	71.00
		mBERT	86.07	85.46
		MARBERT	<b>91.20</b>	<b>87.70</b>
		Zhang and Abdul-Mageed (2019)	77.40	71.70
MADAR (User-level)	21	mBERT (MST)	76.40	68.47
		MARBERT (MST)	76.39	70.61
		MARBERT (Micro-Ara+MST)	<b>80.69</b>	<b>74.45</b>

Table 6: Results on external data. **MST**: MADAR task 1 and 2. MARBERT (ours) sets new SOTA on *all* tasks.

strate the impact of our newly-developed model, MARBERT, by fine-tuning it on a range of text classification tasks. These involve 4 sentiment analysis datasets: ArSAS (Elmadany et al., 2018), ASTD (Nabil et al., 2015), SemEval-2017 task 4-A benchmark dataset (Rosenthal et al., 2017), and Arabic sentiment collection (ASC) in Abdul-Mageed et al. (2020b); and a benchmark for offensive language (OFF) from OSACT4 Arabic Offensive Language Detection Shared Task (Mubarak et al., 2020). More information about each dataset can be found in the respective sources. We compare to the SOTA on each dataset, using the same metrics for each respective systems: For ArSAS, ASTD, and SemEval, we use  $F_1^{PN}$ .<sup>20</sup> And for OFF and ASC, we use *macro*  $F_1$ . As Table 7 shows, our models set new SOTA on *all* 5 datasets.

	ArSAS	ASTD	SemEv	ASC	OFF
Farha and Magdy (2019) (Mazaj)	90.00	72.00	63.00	—	—
Obeid et al. (2020) (mBERT)	89.00	66.00	60.00	—	—
Obeid et al. (2020) (AraBERT)	92.00	73.00	69.00	—	—
Hassan et al. (2020) (AraBERT)	—	—	—	—	90.51
Abdul-Mageed et al. (2020b) (mBERT)	—	—	—	76.67	—
MARBERT (Ours)	<b>92.50</b>	<b>78.50</b>	<b>70.50</b>	<b>90.86</b>	<b>91.47</b>

Table 7: Evaluation of MARBERT on external tasks.

## 10 Related Work

**Dialectal Arabic Data and Models.** Much of the early work on Arabic varieties focused on collecting data for main varieties such as Egyptian and Levantine (Diab et al., 2010; Elfardy and Diab, 2012; Al-Sabbagh and Girju, 2012; Sadat et al., 2014; Zaidan and Callison-Burch, 2011). Many works developed models for detecting 2-3 dialects (Elfardy and Diab, 2013; Zaidan and Callison-Burch, 2011, 2014; Cotterell and Callison-Burch, 2014). Larger datasets, mainly based on Twitter, were recently introduced (Mubarak and Darwish, 2014; Abdul-Mageed et al., 2018; Zaghouni and Charfi, 2018; Bouamor et al., 2019a). Our dataset is orders of magnitude larger than other datasets,

<sup>20</sup> $F_1^{PN}$  was defined by SemEval-2017 as the *macro*  $F_1$  over the positive and negative classes only while neglecting the neutral class.

more balanced, and more diverse. It is also, by far, the most fine-grained.

**Geolocation, Variation, and MTL.** Research on geolocation is also relevant, whether based on text (Roller et al., 2012; Graham et al., 2014; Han et al., 2016; Do et al., 2018), user profile (Han et al., 2013), or network-based methods (Miura et al., 2017; Ebrahimi et al., 2018). Models exploiting network information, however, do not scale well to larger datasets (Rahimi et al., 2015). (Eisenstein, 2012) exploit geotagged Twitter data to study how words spread geographically. (Bamman et al., 2014) uses representations based on geolocation to improve semantic similarity. Dunn (2019) studies syntactic variations in 7 languages based on geolocated data. Hovy et al. (2020) visualizes regional variation across Europe using Twitter. Dunn and Adams (2020) find that Twitter data are representative of actual population. MTL has been successfully applied to many NLP problems, including MT and syntactic parsing (Luong et al., 2015), sequence labeling (Søgaard and Goldberg, 2016; Rei, 2017), and text classification (Liu et al., 2016).

## 11 Conclusion

We introduced the novel task of MDI and offered a large-scale, manually-labeled dataset covering 319 city-based Arabic micro-varieties. We also introduced several novel MTL scenarios for modeling MDs including at hierarchical levels, and with linguistically-motivated auxiliary tasks inspired by diatopic and code-switching environments. We have also exploited our own data to train MARBERT, a very large and powerful masked language model covering all Arabic varieties. Our models establish new SOTA on a wide range of tasks, thereby demonstrating their value. Ultimately, we hope our work can open up new horizons for studying MDs in various languages.

## Acknowledgment

We thank the anonymous reviewers for their valuable feedback, and Hassan Alhuzali for initial help with data. We gratefully acknowledge support from the Natural Sciences and Engineering Research Council of Canada, the Social Sciences Research Council of Canada, Compute Canada ([www.computecanada.ca](http://www.computecanada.ca)), and UBC ARC-Sockeye (<https://doi.org/10.14288/SOCKEYE>).

## References

- Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. 2018. You tweet what you speak: A city-level dataset of Arabic dialects. In *LREC*, pages 3653–3659.
- Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020a. NADI 2020: The First Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop (WANLP 2020)*, Barcelona, Spain.
- Muhammad Abdul-Mageed, Chiyu Zhang, Azadeh Hashemi, and El Moatez Billah Nagoudi. 2020b. AraNet: A deep learning toolkit for Arabic social media. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 16–23.
- Rania Al-Sabbagh and Roxana Girju. 2012. Yadac: Yet another dialectal Arabic corpus. In *LREC*, pages 2882–2889.
- Einat Amitay, Nadav Har’El, Ron Sivan, and Aya Soffer. 2004. Web-a-where: geotagging web content. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–280. ACM.
- Lars Backstrom, Eric Sun, and Cameron Marlow. 2010. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web*, pages 61–70. ACM.
- Fady Baly, Hazem Hajj, et al. 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.
- David Bamman, Chris Dyer, and Noah A Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 828–834.
- Reem Bassiouney. 2020. *Arabic sociolinguistics: Topics in diglossia, gender, identity, and politics*. Georgetown University Press.
- Elabbas Benmamoun. 2011. Agreement and cliticization in Arabic varieties from diachronic and synchronic perspectives. *al-‘Arabiyya*, pages 137–150.
- Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019a. [The MADAR shared task on Arabic fine-grained dialect identification](#). In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207, Florence, Italy. Association for Computational Linguistics.
- Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019b. Madar Twitter user dialect identification (shared task 2). In *Proceedings of the The Fourth Workshop for Arabic Natural Language Processing (WANLP2019)*, pages 00–00.
- Kathleen M Carley, Momin Malik, Peter M Landwehr, Jürgen Pfeffer, and Michael Kowalchuck. 2016. Crowd sourcing disaster management: The complex nature of Twitter usage in padang indonesia. *Safety science*, 90:48–61.
- R. Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the 10th International Conference on Machine Learning*.
- Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- Ryan Cotterell and Chris Callison-Burch. 2014. A multi-dialect, multi-genre corpus of informal written Arabic. In *LREC*, pages 241–245.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba. 2010. Colaba: Arabic dialect annotation and processing. In *Lrec workshop on semitic language processing*, pages 66–74.
- Tien Huu Do, Duc Minh Nguyen, Evaggelia Tsiliogianni, Bruno Cornelis, and Nikos Deligiannis. 2018. Twitter user geolocation using deep multi-view learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6304–6308. IEEE.
- Gabriel Doyle. 2014. Mapping dialectal variation by querying social media. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 98–106.
- Jonathan Dunn. 2019. Global syntactic variation in seven languages: Towards a computational dialectology. *Frontiers in Artificial Intelligence*, 2:15.
- Jonathan Dunn and Ben Adams. 2020. Mapping languages and demographics with georeferenced corpora. *arXiv preprint arXiv:2004.00809*.

- Mohammad Ebrahimi, Elaheh ShafieiBavani, Raymond Wong, and Fang Chen. 2018. A unified neural network model for geolocating Twitter users. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 42–53.
- Jacob Eisenstein. 2012. Mapping the geographical diffusion of new words. *arXiv preprint arXiv:1210.5268*, 1:13.
- Mohamed Elaraby and Muhammad Abdul-Mageed. 2018. [Deep models for Arabic dialect identification on benchmarked data](#). In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 263–274, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Heba Elfardy and Mona T Diab. 2012. Simplified guidelines for the creation of large scale dialectal Arabic annotations. In *LREC*, pages 371–378.
- Heba Elfardy and Mona T Diab. 2013. Sentence level dialect identification in Arabic. In *ACL (2)*, pages 456–461.
- AbdelRahim Elmadany, Hamdy Mubarak, and Walid Magdy. 2018. ArSAS: An Arabic speech-act and sentiment corpus of tweets. *OSACT*, 3:20.
- Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198.
- Charles A Ferguson. 1959. Diglossia. *word*, 15(2):325–340.
- Bruno Gonçalves and David Sánchez. 2014. Crowdsourcing dialect characterization through Twitter. *PloS one*, 9(11).
- Mark Graham, Scott A Hale, and Devin Gaffney. 2014. Where in the world are you? geolocation and language identification in Twitter. *The Professional Geographer*, 66(4):568–578.
- Bo Han, Paul Cook, and Timothy Baldwin. 2012. Geolocation prediction in social media data by finding location indicative words. In *Proceedings of COLING 2012*, pages 1045–1062.
- Bo Han, Paul Cook, and Timothy Baldwin. 2013. A stacking-based approach to Twitter user geolocation prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12.
- Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based Twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49:451–500.
- Bo Han, Afshin Rahimi, Leon Derczynski, and Timothy Baldwin. 2016. Twitter geolocation prediction shared task of the 2016 workshop on noisy user-generated text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 213–217.
- Sabit Hassan, Younes Samih, Hamdy Mubarak, Ahmed Abdelali, Ammar Rashed, and Shammur Absar Chowdhury. 2020. Alt submission for osact shared task on offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 61–65.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Dirk Hovy, Afshin Rahimi, Timothy Baldwin, and Julian Brooke. 2020. Visualizing regional language variation across europe on Twitter. *Handbook of the Changing World Language Map*, pages 3719–3742.
- Binxuan Huang and Kathleen M Carley. 2019a. A hierarchical location prediction neural network for Twitter user geolocation. 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 4734–4744.
- Binxuan Huang and Kathleen M Carley. 2019b. A large-scale empirical study of geotagging behavior on Twitter. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2019*. ACM.
- Tommi Sakari Jauhiainen, Heidi Annika Jauhiainen, Bo Krister Johan Linden, et al. 2019. Discriminating between mandarin chinese and swiss-german varieties using adaptive language models. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2019)*. Association for Computational Linguistics.
- David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017a. [Incorporating dialectal variability for socially equitable language identification](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 51–57, Vancouver, Canada. Association for Computational Linguistics.
- David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017b. Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 51–57.
- Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- William Labov. 1964. *The social stratification of English in New York City*. Ph.D. thesis, Columbia university.
- Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

- Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30. Association for Computational Linguistics.
- Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.
- Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between similar languages and Arabic dialect identification: A report on the third dsl shared task. *VarDial 3*, page 1.
- Yasuhide Miura, Motoki Taniguchi, Tomoki Taniguchi, and Tomoko Ohkuma. 2017. Unifying text, metadata, and user network representations with a neural network for geolocation prediction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1260–1272.
- Hamdy Mubarak and Kareem Darwish. 2014. Using Twitter to collect a multi-dialectal corpus of Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 1–7.
- Hamdy Mubarak, Kareem Darwish, Walid Magdy, Tamer Elsayed, and Hend Al-Khalifa. 2020. Overview of osact4 Arabic offensive language detection shared task. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 48–52.
- Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519.
- Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhil Eryani, Alexander Erdmann, and Nizar Habash. 2020. Camel tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 7022–7032.
- Simon E Overell. 2009. *Geographic information retrieval: Classification, disambiguation and modelling*. Ph.D. thesis, Citeseer.
- Michael J Paul and Mark Dredze. 2011. You are what you tweet: Analyzing Twitter for public health. In *Fifth International AAAI Conference on Weblogs and Social Media*.
- Daniele Quercia, Neal Lathia, Francesco Calabrese, Giusy Di Lorenzo, and Jon Crowcroft. 2010. Recommending social events from mobile phone location data. In *2010 IEEE international conference on data mining*, pages 971–976. IEEE.
- Chatrine Qwaider, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. Shami: A corpus of Levantine Arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.
- Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2015. Twitter user geolocation using a unified text and network prediction model. *arXiv preprint arXiv:1506.08259*.
- Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2017. A neural model for user geolocation and lexical dialectology. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 209–216.
- Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. *arXiv preprint arXiv:1704.07156*.
- Stephen Roller, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldrige. 2012. Supervised text-based geolocation using language models on an adaptive grid. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1500–1510. Association for Computational Linguistics.
- Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 502–518.
- Fatiha Sadat, Farnazeh Kazemi, and Atefeh Farzindar. 2014. Automatic identification of Arabic language varieties and dialects in social media. *Proceedings of SocialNLP*, page 22.
- Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM.
- Mohammad Salameh and Houda Bouamor. 2018. Fine-grained Arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344.
- Luke Sloan and Jeffrey Morgan. 2015. Who tweets with their location? understanding the relationship between demographic characteristics and the use of geoservices and geotagging on Twitter. *PloS one*, 10(11).
- Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–235.

- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.
- Hechan Tian, Meng Zhang, Xiangyang Luo, Fenlin Liu, and Yaqiong Qiao. 2020. Twitter user location inference based on representation learning and label propagation. In *Proceedings of The Web Conference 2020*, pages 2648–2654.
- Peter Trudgill. 1974. Linguistic change and diffusion: Description and explanation in sociolinguistic dialect geography. *Language in society*, 3(2):215–246.
- Janet CE Watson. 2007. *The phonology and morphology of Arabic*. Oxford University Press.
- Wajdi Zaghouani and Anis Charfi. 2018. Arap-tweet: A large multi-dialect Twitter corpus for gender, age and language variety identification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.
- Omar F Zaidan and Chris Callison-Burch. 2011. The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 37–41. Association for Computational Linguistics.
- Omar F Zaidan and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.
- Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, et al. 2018. Language identification and morphosyntactic tagging: The second vardial evaluation campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects*. Association for Computational Linguistics.
- Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A report on the dsl shared task 2014. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 58–67.
- Chiyu Zhang and Muhammad Abdul-Mageed. 2019. No army, no navy: Bert semi-supervised learning of Arabic dialects. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 279–284.
- Ting Zhong, Tianliang Wang, Fan Zhou, Goce Trajcevski, Kunpeng Zhang, and Yi Yang. 2020. Interpreting Twitter user geolocation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 853–859, Online. Association for Computational Linguistics.
- Arkaitz Zubiaga, Inaki San Vicente, Pablo Gamallo, José Ramon Pichel, Inaki Alegria, Nora Aranberri, Aitzol Ezeiza, and Víctor Fresno. 2016. Tweetlid: a benchmark for tweet language identification. *Language Resources and Evaluation*, 50(4):729–766.

# **Toward Micro-Dialect Identification in Diaglossic and Code-Switched Environments**

Muhammad Abdul-Mageed Chiyu Zhang AbdelRahim Elmadany Lyle Ungar<sup>†</sup>  
Natural Language Processing Lab, University of British Columbia  
<sup>†</sup>Computer and Information Science, University of Pennsylvania  
{muhammad.mageed, a.elmadany}@ubc.ca chiyuzh@mail.ubc.ca  
<sup>†</sup>ungar@cis.upenn.edu

# Appendices

## A Data Acquisition and Labeling

### A.1 Correction of City and State Tags

**City-Level.** Investigating examples of the geolocated data, we observed geopy made some mistakes. To solve the issue, we decided to manually verify the information returned from geopy on all the 705 assumed ‘cities’. For this purpose of manual verification, we use Wikipedia, Google maps, and web search as sources of information while checking city names. We found that geopy made mistakes in 7 cases as a result of misspelled city names in the queries we sent (as coming from user profiles). We also found that 44 cases were not assigned the correct city name as the first ‘solution’. Geopy provided us with a maximum of 7 solutions for a query, with best solutions sometimes being names of hamlets, villages, etc., rather than cities. In many cases, we found the correct solution to fall between the 2nd and 4th solutions. A third problem was that some city names (as coming from user profiles) were written in non-Arabic (e.g., English or French). We solved this issue by requiring geopy to also return the English version of a city name, and *exclusively* using that English version. Ultimately, we acquired a total of 646 cities.

**State-Level.** As explained, geopy returned to us a total of 235 states/provinces that correspond to the 646 identified (manually fixed) cities. We also manually verified all the state names and their correspondence to the cities and countries. We found no issues with state tags.

### A.2 Validation of User Location

We trained the two annotators and instructed them to examine the profile information of each user on Twitter, providing a link to the profile. We asked them to consider various sources of information as a basis for their decisions, including (1) the profile picture, (2) profile textual description (including user-provided location), (3) the actual name of the user (if available), (4) at least 10 tweets, (5) the followers and followees of the user, and (5) user’s network behavior such as the ‘likes’.

Each annotator was responsible for  $\sim 50\%$  of the usernames and was given a random sample of 100<sup>21</sup> users for each city along with the Twitter handles and the automatically assigned *city* and

<sup>21</sup>But we note that some cities had less than 100 users.

*country* labels. We asked the users to label the first 10 accounts in each city, and only add more if the city proves specially challenging (as we observed to be the case in a pilot analysis of a few cities). Annotators ended up labeling a total of 4,953 accounts ( $\sim 11.88$  users per city), of whom 4,012 users were verified for country and 3,085 for *both* country and city locations. We found that 81.00% of geopy tags for country are correct, but only 62.29% for city (which reduced our final city count to 319). As a final sanity check, a third annotator reviewed the labels for a random sample of 20 users from each annotator and agreed fully.

## B Datasets

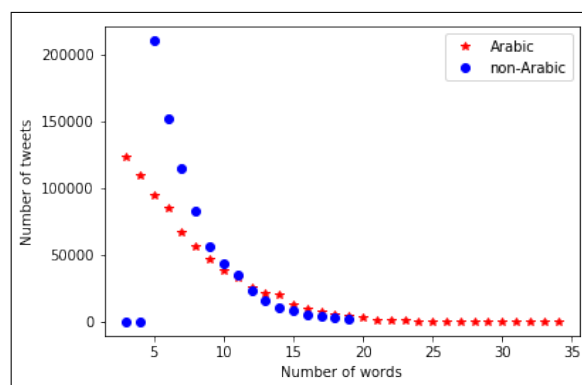


Figure B.1: Word frequency distribution in our CS-21 (code-switching) dataset.

Country	%vld_cntry	%vld_city	#tweets
Algeria	77.49	69.74	185,854
Bahrain	83.95	39.51	25,495
Djibouti	68.42	68.42	3,939
Egypt	92.66	64.02	463,695
Iraq	51.50	37.61	59,287
Jordan	83.61	54.10	17,958
KSA	96.37	62.88	353,057
Kuwait	84.30	34.88	65,036
Lebanon	92.42	56.06	37,273
Libya	75.48	72.03	128,152
Maurit.	45.00	35.00	3,244
Morocco	75.59	62.42	140,341
Oman	90.25	77.97	108,846
Palestine	87.50	82.35	87,446
Qatar	85.00	77.50	29,445
Somalia	52.73	45.45	9,640
Sudan	56.88	41.28	23,642
Syria	76.28	71.63	79,649
Tunisia	78.95	75.94	26,300
UAE	85.31	82.49	129,264
Yemen	72.41	56.32	47,450
<b>Avg/Total</b>	<b>81.00</b>	<b>62.29</b>	<b>2,025,013</b>

Table B.2: Our gold data, from manually verified users.

Countries		#Users	#Tweets			#States	#Cities
Name	Code		Collected	Retweets	Normalized		
Algeria	dz	1,960	3,939,411	2,889,447	2,324,099	47	200
Bahrain	bh	1,080	2,801,399	1,681,337	1,385,533	4	4
Djibouti	dj	6	11,901	9,173	8,790	1	1
Egypt	eg	42,858	92,804,863	61,264,656	47,463,301	27	56
Iraq	iq	4,624	7,514,750	4,922,553	4,318,523	18	62
Jordan	jo	3,806	7,796,794	5,416,413	4,209,815	4	5
KSA	sa	136,455	297,264,647	177,751,985	165,036,420	11	31
Kuwait	kw	4,466	11,461,531	7,984,758	6,628,689	4	14
Lebanon	lb	1,364	3,036,432	1,893,089	1,160,167	6	19
Libya	ly	2,083	4,227,802	3,109,355	2,655,180	21	32
Mauritania	mr	102	209,131	148,261	129,919	4	4
Morocco	ma	1,729	3,407,741	2,644,733	1,815,947	17	117
Oman	om	4,260	8,139,374	4,866,813	4,259,780	8	17
Palestine	ps	2,854	6,004,791	4,820,335	4,263,491	2	12
Qatar	qa	5,047	11,824,490	7,891,425	6,867,304	2	2
Somalia	so	78	168,136	131,944	104,946	8	9
Sudan	sd	1,162	2,348,325	1,522,274	1,171,866	14	27
Syria	sy	1,630	2,992,106	2,184,715	1,889,455	12	19
Tunisia	tn	227	460,268	362,806	239,769	10	10
UAE	ae	14,923	36,121,319	23,309,788	18,484,296	7	15
Yemen	ye	2,391	4,783,144	3,368,262	3,013,517	8	8
<b>Total</b>		<b>233,105</b>	<b>507,318,355</b>	<b>318,174,122</b>	<b>277,430,807</b>	<b>235</b>	<b>664</b>

Table B.1: Statistics of our data representing 233,105 users from 664 cities and 21 countries. We process more than half a billion tweets, from a larger pool of  $\sim 6$  billion tweets, to acquire our final dataset. Note that the number of states and cities is further reduced after our manual user verification. Eventually, we acquire data for 319 cities, belonging to 192. The data represent all 21 Arab countries.

Country	TRAIN	DEV	TEST
Algeria	100,000	18,700	18,572
Bahrain	20,387	2,556	2,552
Djibouti	3,158	408	373
Egypt	100,000	46,136	46,325
Iraq	47,395	5,903	5,989
Jordan	14,413	1,826	1,719
KSA	100,000	35,312	35,106
Kuwait	52,127	6,416	6,493
Lebanon	29,821	3,641	3,811
Libya	100,000	12,847	12,803
Maurit.	2,579	338	327
Morocco	100,000	14,118	13,862
Oman	87,048	10,807	10,991
Palestine	69,834	8,668	8,944
Qatar	23,624	2,968	2,853
Somalia	7,678	1,023	939
Sudan	18,929	2,334	2,379
Syria	63,668	7,987	7,994
Tunisia	21,164	2,599	2,537
UAE	100,000	13,089	12,768
Yemen	37,886	4,833	4,731
Total	1,099,711	202,509	202,068

Table B.3: Distribution of classes in our data splits.

## C Models

### C.1 Single Task BiGRUs (Second Baseline)

As mentioned in Section 4.1, our *second baseline* (Baseline II), is comprised of 3 independent networks (each for one of the 3 tasks) using the same architecture and model capacity. Each net-

work has 3 hidden BiGRU layers,<sup>22</sup> with 1,000 units each (500 units from left to right and 500 units from right to left). We add dot-product attention *only* to the third hidden layer. We trim each sequence at 50 words,<sup>23</sup> and use a batch size of 8. Each word in the input sequence is represented as a vector of 300 dimensions that are learned directly from the data. Word vectors weights  $W$  are initialized with a normal distribution, with  $\mu = 0$ , and  $\sigma = 0.05$ , i.e.,  $W \sim N(0, 0.05)$ . For optimization, we use Adam (Kingma and Ba, 2014) with a fixed learning rate of  $1e - 3$ . For regularization, we use dropout (Srivastava et al., 2014) with a value of 0.5 on each of the 3 hidden layers.

### C.2 Distill BERT

We distill mBERT knowledge in out HA-MTL BiGRUs. In other words, we use the output of the mBERT logit layer as input to our city-first and country-first HA-MTL BiGRUs to optimize a mean-squared error objective function, but not a cross-entropy function (following equation 3 in

<sup>22</sup>We also ran single-task networks with 4 hidden layers, but we find them to overfit quickly even when we regularize with dropout at 0.7 on all layers.

<sup>23</sup>In initial experiments, we found a maximum sequence of 30 words to perform slightly worse.



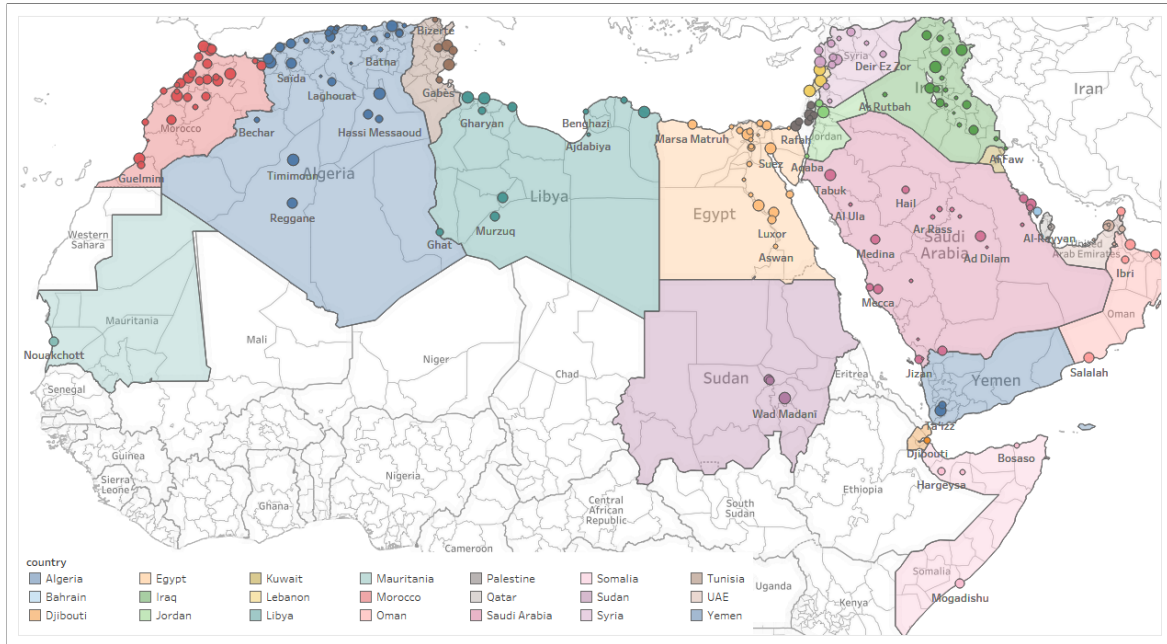


Figure B.2: A bigger-sized map of all 21 Arab countries. States are demarcated in thin black lines within each country. A total of 319 cities (from our user location validation study, in colored circles) are overlaid within corresponding countries.

Tang et al. (2019)).<sup>24</sup> As Table C.1, both of these networks (HA-MTL-Dist-city 1<sup>st</sup> and HA-MTL-Dist-country 1<sup>st</sup> in the table) acquire sizeable improvements over the equivalent, non-distilled BiGRUs. Although these distillation models are still less than BERT, the goal behind them is to yield as closer-as-possible performance to BERT albeit with a smaller network that can be deployed in machines with limited capacity and offer quicker inference. Concretely, a HA-MTL-BiGRU model learns the 3 tasks of city, state, and country together compared to the single task BERT where 3 different models are needed for these 3 tasks. In terms of the number of parameters, this means the multi-task BiGRU distillation model has  $11.6\times$  fewer parameters. HA-MTL-BiGRU is also  $1.7$  times faster at inference.<sup>25</sup>

### C.3 Multi-Task BiGRUs

## D Model Generalization

### D.1 Different Data Splits

**Narrow, Medium, and Wide Settings.** For (1) **narrow**, we select data from cities where we have at least 16 users, dividing users randomly into 3 in

<sup>24</sup>The network architecture of the HA-MTL BiGRU is otherwise similar as before, but we train them for 20 epochs rather than 15.

<sup>25</sup>We perform model inference on the DEV set with a batch size of 128 on a single NVIDIA V100 GPU.

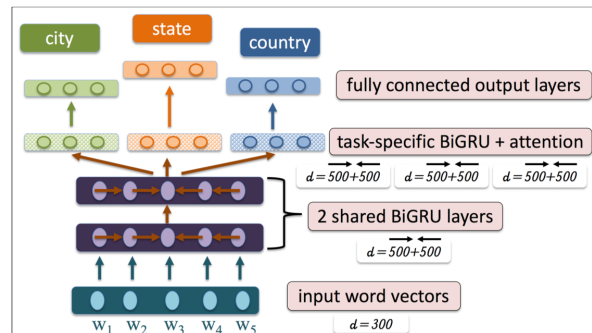


Figure C.1: Illustration of MTL (spec-attn) network for city, state, and country. The three tasks share 2 hidden layers, with each task having its independent attention layer.

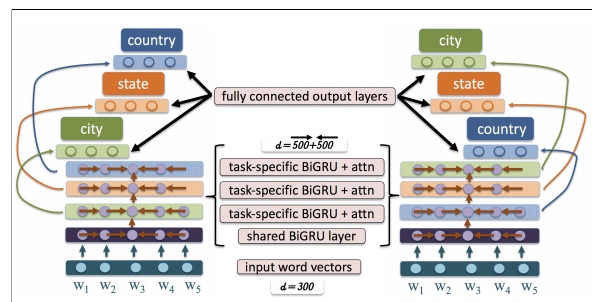


Figure C.2: Hierarchical Attention MTL of city, state, and country. All models share one BiGRU layer of 1,000 units. Layers 2-4 are also BiGRU layers, with dot-product attention. **Left:** City network supervised at layer 2, state at layer 3, and country at layer 4. **Right:** Supervision is reversed from left network.

Setting	City		State		Country	
	acc	F1	acc	F1	acc	F1
Baseline I	1.31	0.01	3.11	0.03	9.19	0.80
Baseline II	1.65	0.25	6.13	1.92	31.14	15.84
MTL (common-attn)	2.86	0.74	5.12	1.01	26.51	12.41
MTL (spec-attn)	2.40	0.68	4.60	0.90	27.04	10.98
HA-MTL (city 1 <sup>st</sup> )	<b>14.01</b>	<b>14.02</b>	<b>16.14</b>	<b>15.90</b>	<b>44.36</b>	32.14
HA-MTL (cntry 1 <sup>st</sup> )	13.23	13.06	15.84	15.40	44.17	<b>32.37</b>
mBERT(Devlin et al. (2018))	<b>19.33</b>	<b>19.45</b>	<b>21.24</b>	<b>21.67</b>	<b>47.74</b>	<b>38.12</b>
HA-MTL-Dist (city 1 <sup>st</sup> )	15.69	15.95	18.33	18.72	46.37	36.27
HA-MTL-Dist (cntry 1 <sup>st</sup> )	15.73	15.58	18.21	18.46	46.34	36.13
MTL-mBERT (DiaGloss)	<b>19.88</b>	<b>20.11</b>	21.04	<b>21.69</b>	48.30	<b>38.34</b>
MTL-mBERT (CodSw)	19.47	19.86	20.76	21.47	<b>48.61</b>	38.20
AraBERT (Baly et al. (2020))	18.82	18.73	20.73	20.87	48.33	38.09
MARBERT (Ours)	<b>20.78</b>	<b>20.41</b>	<b>22.97</b>	<b>22.58</b>	<b>51.87</b>	<b>42.17</b>
MTL-MARBERT (DiaGloss)	20.19	<b>20.60</b>	<b>23.22</b>	22.97	51.53	41.75
MTL-MARBERT (CodSw)	<b>20.77</b>	20.56	23.21	<b>23.16</b>	51.78	<b>42.36</b>
MTL-MARBERT (CSD)	20.76	20.23	23.18	23.15	<b>52.17</b>	42.27

Table C.1: Performance on TEST. Baseline I: majority in TRAIN. Baseline II: single task Attn-BiGRU. HA-MTL-Dist: BiGRU distilling of mBERT knowledge. MTL-mBERT/MARBERT (CodSw): Code-switching at “country” level. MTL-MARBERT (CSD): Two auxiliary tasks, code-switching supervised at country level and diglossia.

Setting	City		State		Country	
	acc	F1	acc	F1	acc	F1
Baseline I	1.31	0.01	3.11	0.03	9.19	0.80
Baseline II	1.72	0.26	6.08	1.92	30.94	15.84
MTL (common-attn)	2.90	0.74	5.07	1.04	26.52	12.44
MTL (spec-attn)	2.48	0.70	4.57	0.90	27.04	10.98
HA-MTL (city 1 <sup>st</sup> )	<b>14.08</b>	<b>14.29</b>	<b>16.10</b>	<b>16.00</b>	<b>44.14</b>	<b>33.14</b>
HA-MTL (cntry 1 <sup>st</sup> )	13.31	13.20	15.91	15.78	44.06	32.79
mBERT	<b>19.56</b>	<b>19.82</b>	<b>21.20</b>	<b>21.67</b>	<b>47.57</b>	<b>38.30</b>
HA-MTL-Dist (city 1 <sup>st</sup> )	15.79	15.96	18.38	18.69	46.11	36.50
HA-MTL-Dist (cntry 1 <sup>st</sup> )	15.85	15.76	18.28	18.39	46.22	36.21
MTL-mBERT (DiaGloss)	<b>19.92</b>	<b>20.57</b>	20.89	21.58	48.16	38.43
MTL-mBERT (CodSw)	19.56	20.25	<b>21.09</b>	<b>21.92</b>	<b>48.57</b>	<b>38.66</b>

Table C.2: Performance on DEV. Baseline I: majority in TRAIN. Baseline II: single task Attn-BiGRU. MTL-mBERT (CodSw): Code-switching at “country” level. HA-MTL-Dist: BiGRU distilling of mBERT knowledge.

Setting	City		State		Country	
	acc	F1	acc	F1	acc	F1
mBERT (Devlin et al. (2018))	19.56	19.82	21.20	21.67	47.57	38.30
AraBERT (Baly et al. (2020))	18.77	18.69	20.63	21.16	48.18	38.53
MARBERT (Ours)	<b>20.81</b>	<b>20.29</b>	<b>23.05</b>	<b>22.92</b>	<b>51.73</b>	<b>42.59</b>
MTL-MARBERT (DiaGloss)	<b>20.86</b>	<b>20.87</b>	23.15	23.17	51.34	42.07
MTL-MARBERT (CodSw)	20.85	20.78	23.13	23.03	51.60	42.37
MTL-MARBERT (CSD)	20.76	20.36	<b>23.23</b>	<b>23.24</b>	<b>52.06</b>	<b>42.52</b>

Table C.3: Performance on DEV. MTL-MARBERT (CSD): Two auxiliary tasks, code-switching supervised at country level and diglossia.

TEST and the rest (13 or more) in TRAIN. This gives us 61 cities, 48 states, and 11 countries. Our (2) **medium** setting is similar to *narrow*, but we sample from cities with at least 13 users instead of 16. We use 3 users for TEST and the rest (10 users or more) for TRAIN. This results in 116 cities, 90 states, and 17 countries. (3) **Wide** has data from a single user from a given city in TEST and the rest of users from the same city in TRAIN. This setting allows more coverage (240 cities, 158 states, and all the 21 countries), at the cost of having as

few as only two users for a given city in TRAIN. Figure D.1 shows the distribution of users over the 3 data settings. In addition, Table D.2 shows the data sizes of the TRAIN and TEST sets in each of the 3 runs, across each of the 3 settings.

## D.2 Comparison of Models on a Completely New TEST Set.

As mentioned in Section 6, we evaluate our models from the narrow, medium and wide settings *and* our single task MARBERT model (shown in Ta-

Model	DEV						TEST					
	acc	F1	acc@161	acc@80.5	mean(K)	median(K)	acc	F1	acc@161	acc@80.5	mean(K)	median(K)
MARB-319	3.23	2.88	12.32	7.07	1,509.41	1,140.17	3.12	2.53	15.11	7.83	1,446.83	1,022.29
Wide	3.90	3.70	12.71	7.50	1,438.68	1,070.05	3.49	2.97	15.27	7.91	1,370.23	936.60
Medium	5.24	4.69	15.85	9.70	1,171.39	734.98	5.43	4.56	19.91	10.58	1,151.77	652.06
Narrow	7.66	6.85	17.95	12.58	1,036.58	678.04	6.56	5.49	20.69	12.25	1,052.54	624.09

Table D.1: Evaluation of our MARBERT-based models on a new TEST set (GeoAra). **MARB-319**: our single task MARBERT model trained on 319 cities (reported on Table 3). **Narrow**: MARBERT-narrow, **Medium**: MARBERT-medium; **Wide**: MARBERT-wide. **Acc@80.5**: Accuracy at 80.5 kilometers (=50 miles). **Acc@161**: Accuracy at 161 kilometers (=100 miles). **Mean(K)**: Mean distance in kilometers. **Median(K)**: Median distance in kilometers.

ble 3) all on a completely new test set. This allows a more direct comparison between these models, including to test the impact of *sharing users* across the various data splits (as is the case of single task MARBERT) or *lack thereof* (as is the case for the narrow, medium and wide settings models). We now introduce GeoAra, our new evaluation dataset.

**GeoAra Dataset.** GeoAra is a dataset of tweets with city labels from 20 Arab countries.<sup>26</sup> To build GeoAra, we run a crawler on each of the 319 cities in our gold data for a total of 10 month (Jan. 2019 - Oct. 2019). We acquire a total of 4.7M tweets from all the cities. We collect Twitter user ids from users who posted consistently from a single location over the whole 10 months ( $n=390,396$ ), and crawl the timeline of 148K users.<sup>27</sup> Note that MicroAra (our monolingual dataset) is collected in 2016 and 2017. This means GeoAra involves data from a period significantly different (more recent) than MicroAra (2 years later). We then *only* keep users who posted at least 10 tweets. This leaves us with 101,960 users from 147 cities. From GeoAra, we create a DEV set from a random sample of 100K tweets (908 from users) and a TEST set from a random sample of 97,834 tweets (from 1,053 users).<sup>28</sup> We *do not* share users between the TRAIN, DEV, and TEST splits.

As Table D.1 shows, although all the 4 models degrade on GeoAra, single task MARBERT (MARB-319 in the table) suffers most. This further suggests, that this particular model has captured user-level knowledge that may have allowed it to perform much higher on the TEST set in Table 3

<sup>26</sup>These are the same countries as in our MicroAra, with the exception of Djibouti.

<sup>27</sup>We note that this is more conservative than previous geolocation works (e.g., (Han et al., 2012)) that take the majority class city of a user who posted 10 tweets as the label.

<sup>28</sup>The two splits do not identically match since we also needed to create a specific TRAIN split from the same dataset. The TRAIN split is not part of the current work and so we leave it out.

than what it would if user data were not shared across the various splits. In addition, even though our *narrow setting* model covers only 61 cities, it is the one that performs best on both the DEV and TEST GeoAra splits. This might be the case because this model is trained on the most number of users (at least 13 users for each city), which allows it to generalize well on these cities. An error analysis may reveal more information on performance of these particular models on GeoAra. We cast further investigation of this issue as future research.

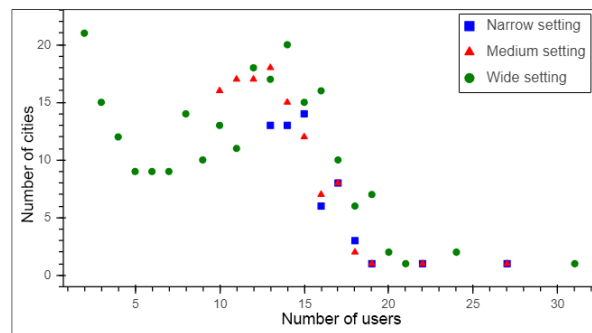


Figure D.1: Distribution of users over cities across our 3 model generalization settings.

	split	TRAIN	TEST	#cities	#states	#countries
Narrow	A	444,838	87,540	61	48	11
	B	445,297	87,081			
	C	453,451	78,927			
Medium	A	807,590	176,337	116	90	17
	B	808,667	175,260			
	C	810,471	173,456			
Wide	A	1,308,640	124,834	240	158	21
	B	1,309,705	123,769			
	C	1,305,098	128,376			

Table D.2: TRAIN and TEST data sizes and label distribution (in city, state, and country) across the 3 splits for each of the *narrow*, *medium*, and *wide* settings.

### D.3 MSA vs. Dialect Classifier

As described in Section 7, we apply an MSA vs. dialect in-house classifier on our *narrow* data setting, to remove MSA. Our in-house classifier fine-tuned MARBERT on the MSA-dialect TRAIN split described in Elaraby and Abdul-Mageed (2018). This

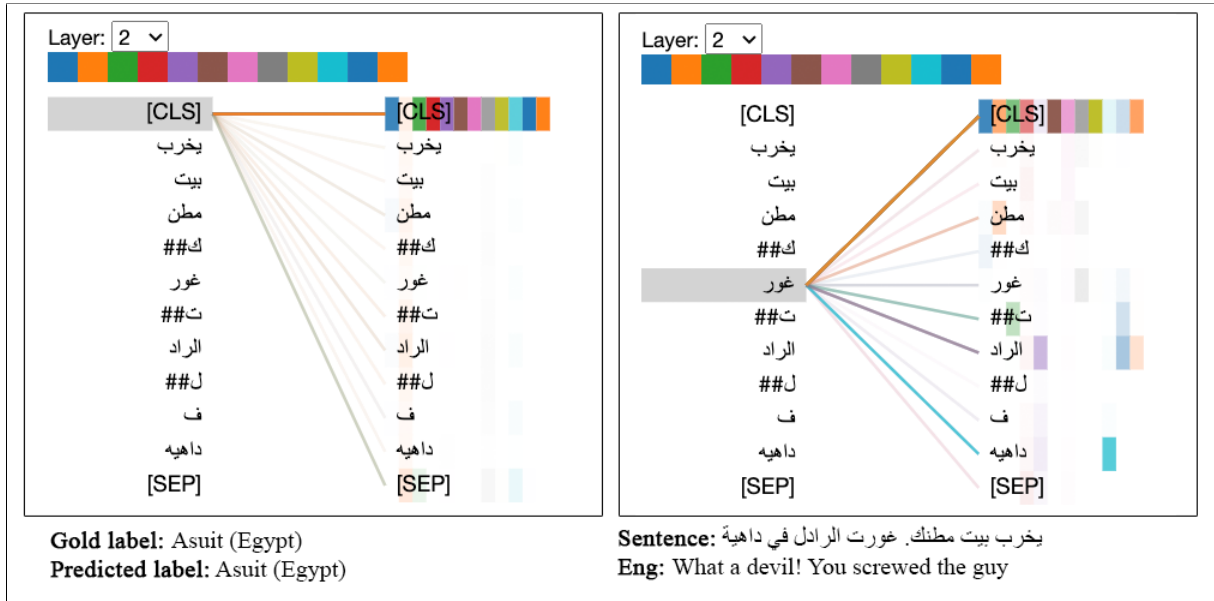


Figure D.2: An example sentence where the model correctly predicted the gold city (Asuit, Egypt), clearly laying attention on relevant micro-dialectal tokens.

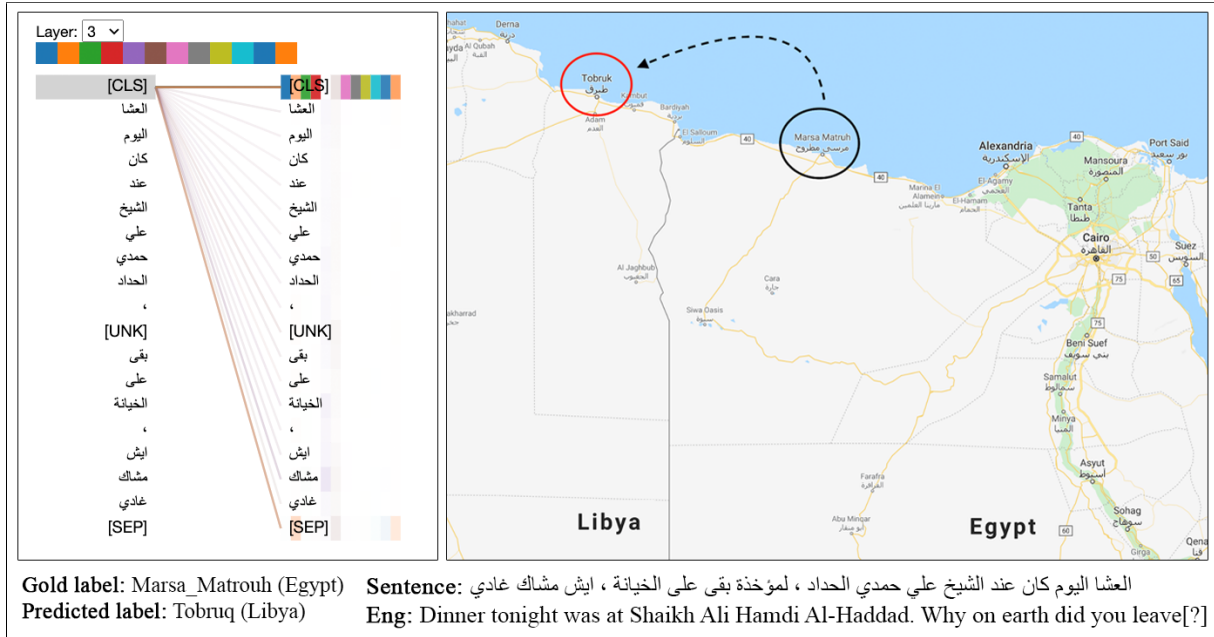


Figure D.3: An example of model error. The model confuses the city of Marsa Matrouh (Egypt, black circled) with that of Tobruq (Libya, red circled). The visualization illustrates how the model is capable of relating language across country boundaries, suggesting it does possess micro-dialectal predictive power.

split	Dialect		MSA	
	TRAIN	TEST	TRAIN	TEST
A	218,231	44,641	226,607	42,899
B	220,697	42,175	224,600	44,906
C	225,737	37,135	227,714	41,792

Table D.3: Distribution of MSA and DA over TRAIN and TEST of *narrow* setting.

binary classifier performs at 89.1% accuracy, and 88.6%  $F_1$  on Elaraby and Abdul-Mageed (2018)

MSA-dialect TEST set. Running this model on our *narrow* setting data, gives us the TRAIN and TEST splits with predicted labels described in Table D.3.

## E Discussion

As discussed in Section 8, we visualize attention in  $\sim 250$  examples from our TEST set using our MARBERT-narrow model fine-tuned in split B in Table D.2. We provide visualizations from two

examples here.

**Example 1:** Figure D.2 shows a visualization of a sentence from the city of Asuit, Egypt, that the model correctly predicted. **Left:** Attention layer #3 of the model<sup>29</sup> has several heads attending to lexical micro-dialectal cues related to Asuit. Most notably, tokens characteristic of the language of the correct city are attended to. Namely, the word **مطنك** (part of the metaphorical expression meaning “what a devil”) receives attention in heads 1-3, and the word **الرادل** “man” in city of Asuit) receives attention in head 2. These cues usually co-occur with the token **غور** (“you screwed [somebody]”), which is also characteristic of the Southern Egyptian region, and the city of Asuit. This is clear in the image in the **right** where the token **غور** attends to other micro-dialectal cues in the sequence.

**Example 2:** Figure D.3 shows a visualization of a sentence from the city of Marsa Matrouh, Egypt, that was incorrectly predicted as Tobruq, Libya. Even though the model makes a prediction error here, its error is meaningful in that it chooses a city that is located in the vicinity of that of the gold city. This means, interestingly, that the city-level model can pick a city close-by to gold in a different country rather than a far city in the same country. This reflects how micro-dialects paint a more nuanced (and linguistically plausible) picture. This also suggests that country-level dialect models are based on arbitrary assumptions, by virtue of being dependent on political boundaries which are not always what defines language variation.

### E.1 Brief Error Analysis

We provide a brief error analysis of single Task MARBERT (described in Table 3 of the paper) in Table E.1.

---

<sup>29</sup>Layer counting starts from zero.

City	Country	Avg Error Dist	Countries with Confused Cities
Beni_Malek	Morocco	4491.90	Oman 39.08, UAE 33.19, Morocco 18.07, Saudi Arabia 4.62, Libya 1.26
Jabria	Morocco	4331.06	UAE 22.63, Kuwait 20.44, Oman 10.95, Saudi Arabia 8.76, Libya 4.38
Nouakchott	Mauritania	4324.06	Oman 14.36, Libya 13.81, Palestine 8.29, Algeria 8.29, Syria 7.18
Casablanca	Morocco	3946.58	Oman 33.69, UAE 20.22, Morocco 7.68, Algeria 6.87, Libya 4.45
Bordj_El_Kiffan	Algeria	3454.71	Kuwait 21.64, UAE 20.15, Bahrain 9.95, Oman 8.46, Saudi Arabia 6.97
Ain_Taya	Algeria	3318.77	Kuwait 17.24, Saudi Arabia 16.55, Oman 11.03, UAE 7.59, Tunisia 5.52
Murzuq	Libya	3255.02	Oman 40.0, UAE 28.57, Morocco 19.29, Libya 3.57, Syria 1.43
Ouillen	Algeria	3222.48	Saudi Arabia 20.55, UAE 13.7, Oman 12.33, Iraq 8.22, Yemen 8.22
Laayoune	Morocco	3146.83	Morocco 33.53, UAE 8.76, Algeria 8.16, Saudi Arabia 7.85, Oman 7.25
Atar	Mauritania	3109.41	Algeria 14.58, Morocco 14.58, Libya 10.42, Syria 10.42, Bahrain 6.25
Ben_Allez	Algeria	3027.69	Oman 26.0, Libya 14.0, Saudi Arabia 14.0, Algeria 10.0, Morocco 10.0
Beni_Mellal	Morocco	3019.40	Palestine 12.09, UAE 12.09, Morocco 10.99, Oman 9.89, Algeria 8.79
Hargeysa	Somalia	3011.22	Algeria 30.47, Saudi Arabia 10.16, UAE 10.16, Iraq 8.59, Morocco 7.81
Dakhla	Morocco	2924.22	Algeria 19.27, Morocco 15.27, Oman 9.09, Libya 8.36, Palestine 8.36
Mogadishu	Somalia	2918.91	Saudi Arabia 18.98, Kuwait 11.92, Oman 9.98, Libya 7.54, Syria 7.54
Bab_Ezzouar	Algeria	2907.83	Iraq 46.67, Algeria 29.33, Somalia 10.67, Saudi Arabia 5.33
Timimoun	Algeria	2788.69	Algeria 22.89, Oman 13.25, Lebanon 7.23, Saudi Arabia 7.23, UAE 7.23
Mohammedia	Morocco	2745.87	Morocco 16.81, Algeria 15.93, Oman 9.73, UAE 7.96, Libya 7.08
Bni_Oulid	Morocco	2702.39	Libya 32.35, Saudi Arabia 11.76, Morocco 8.82, Yemen 8.82, Oman 5.88
El_Jadida	Morocco	2693.21	Algeria 16.67, Morocco 11.11, Oman 11.11, Bahrain 11.11, Egypt 11.11

Table E.1: Top wrongly predicted cities in our DEV based on mBERT. For each gold city, we provide the average distance from the cities with which they were confused (we call it *avg. error distance*), countries to which confused cities belong, followed by percentage in which cities of each country were confused with the gold city. In the future, we also plan to carry out a more extensive (including manual) error analysis based on the tweets involved.