Zero-resource Multi-dialectal Arabic Natural Language Understanding

Muhammad Khalifa[1], Hesham Hassan[2], Aly Fahmy[3]
Cairo University, Egypt

Abstract—**A reasonable amount of annotated data is required for fine-tuning pre-trained language models (PLM) on downstream tasks. However, obtaining labeled examples for different language varieties can be costly. In this paper, we investigate the zero-shot performance on Dialectal Arabic (DA) when fine-tuning a PLM on modern standard Arabic (MSA) data only — identifying a significant performance drop when evaluating such models on DA. To remedy such performance drop, we propose self-training with unlabeled DA data and apply it in the context of named entity recognition (NER), part-of-speech (POS) tagging, and sarcasm detection (SRD) on several DA varieties. Our results demonstrate the effectiveness of self-training with unlabeled DA data: improving zero-shot MSA-to-DA transfer by as large as $\sim$10% F_1 (NER), 2% accuracy (POS tagging), and 4.5% F_1 (SRD). We conduct an ablation experiment and show that the performance boost observed directly results from the unlabeled DA examples used for self-training. Our work opens up opportunities for leveraging the relatively abundant labeled MSA datasets to develop DA models for zero and low-resource dialects. We also report new state-of-the-art performance on all three tasks and open-source our fine-tuned models for the research community.**

Keywords—Natural language processing; natural language understanding; low-resource learning; semi-supervised learning; named entity recognition; part-of-speech tagging; sarcasm detection; pre-trained language models

I. Introduction

Neural language models [1], [2] with contextual word representations [3] have become dominant for a wide range of Natural Language Processing (NLP) downstream tasks. More precisely, contextual representations from transformer-based [4] language models [5], [6], pre-trained on large amounts of raw data and then fine-tuned on labeled tasks-specific data, has produced state-of-the-art performance on many tasks, even when using fewer labeled examples. Such tasks include question answering [7], text classification [6], named entity recognition (NER), and part-of-speech (POS) tagging [8], [9].

Typically, such language models see a huge amount of data during pre-training, which could mistakenly lead us to assume they have a strong generalization capability even in situations where the language varieties seen at test time are different from those the language model was fine-tuned on. To investigate this particular situation, we first study the impact of using a language model pre-trained on huge Arabic corpora for two popular sequence tagging tasks (NER and POS tagging) and one text classification task (sarcasm detection) when fine-tuned on available labeled data, regardless of language variety (Section VII-A). To test the model utility for tasks based on exclusively dialectal Arabic (DA), we then remove all dialectal data from the training splits and fine-tune a model only on MSA. Evaluating such a model in a *zero-shot* setting, i.e., on Egyptian (EGY), Gulf (GLF), and Levantine (LEV) varieties, we observe a significant performance drop. This shows the somewhat brittle ability of pre-trained language models without dialect-specific fine-tuning.

Unfortunately, the scarcity of labeled DA resources covering sufficient tasks and dialectal varieties has significantly slowed down research on DA [10]. Consequently, a question arises: "How can we develop models nuanced to downstream tasks in dialectal contexts without annotated DA examples?". We apply self-training, a classical semi-supervised approach where we augment the training data with confidently-predicted dialectal data. We empirically show that self-training is indeed an effective strategy, which proves to be useful in *zero-shot* (where no gold dialectal data are included in training set) independently as well as with self-training (Sections VII-B and VII-C, respectively).

Our experiments reveal that self-training is always a useful strategy that *consistently* improves over mere fine-tuning. In order to understand why this is the case (i.e., why combining self-training with fine-tuning yields better results than mere fine-tuning), we perform an extensive error analysis based on our NER data. We discover that self-training helps the model most with improving false positives (approximately 59.7%). This includes in cases involving DA tokens whose MSA orthographic counterparts [11] are either named entities or trigger words that frequently co-occur with named entities in MSA. Interestingly, such out-of-MSA tokens occur in highly dialectal contexts (e.g., interjections and idiomatic expressions employed in interpersonal social media communication) or ones where the social media context in which the language (DA) employed affords more freedom of speech and a platform for political satire. We present our error analysis in Section VIII.

We choose Arabic as our experimental playground since it affords a rich context of linguistic variation: In addition to the standard variety, MSA, Arabic also has several dialects, thus offering an excellent context for studying our problem. From a geopolitical perspective, Arabic also has a strategic significance. This is a function of Arabic being the native tongue of 400 million speakers in 22 countries, spanning across two continents (Africa and Asia). In addition, the three dialects of our choice (EGY, GLF, LEV) are popular dialects that are widely used online. This makes our resulting models highly useful in practical situations at scale. Pragmatically, ability to develop NLP systems on dialectal tasks with no-to-small labeled dialect data immediately eases a serious bottleneck. Arabic dialects differ among themselves and from

MSA at all linguistic levels, posing challenges to traditional NLP approaches. Having to develop annotated resources across the various dialects for the different tasks would be quite costly, and perhaps unnecessary. Therefore, zero-shot cross-dialectal transfer would be valuable when only some language varieties have the labeled resources. We also note that our method is language-independent, and we hypothesize it can be directly applied to other varieties of Arabic or in other linguistic contexts for other languages and varieties.

Our research contributions in this paper are 3-fold:

1) We study the problem of MSA-to-DA transfer in the context of sequence labeling and text classification and show, through experiments, that when training with MSA data only, a wide performance gap exists between testing on MSA and DA. That is, models fine-tuned on MSA generalize poorly to DA in zero-shot settings.
2) We propose self-training to improve zero- and few-shot MSA-to-DA transfer. Our approach requires little-to-no labeled DA data. We evaluate extensively on three different dialects across the three aforementioned tasks, and show that our method indeed narrows the performance gap between MSA and DA by a margin as wide as ~10% F_1 points. Moreover, we conduct an ablation experiment to evaluate the importance of using unlabeled DA rather than MSA data in the zero-shot setting, and we show that unlabeled DA data is indeed much more effective and necessary for adapting the model to DA data during testing.
3) We develop state-of-the-art models for the three tasks of (NER, POS tagging, and SRD), which we intend to publicly release for the research community.

We now review relevant literature.

II. Related Work

Classical machine learning techniques, including SVM and Conditional Random Fields (CRFs) [12] applied manually-extracted, hand-crafted word- and character-level features, were previously employed for various sequence labeling tasks including NER, POS tagging, chunking. More recently, however, neural architectures, have become the *defacto* approach for various tasks including sequence labeling. This usually includes an autoregressive architecture such as vanilla Recurrent Neural Networks (RNN) [13] or the more sophisticated Long Short-Term Memory networks (LSTM) [14]. The networks processes the input text in a word-by-word fashion, and the network is trained to predict the correct label for each word. In addition, more capacity can be given to such networks by adding an additional layer that processes the input in a right-to-left fashion [15], [16].

Neural approaches usually make use of both word- and character- features. Word-level features usually consist in semantic word embeddings, which are trained on a large raw corpus in a self-supervised fashion [17], [18]. Character-level features can be extracted through an additional network such as LSTM [19] or CNN [20]. Neural techniques has produced better or comparable results to classical approaches in addition to alleviating the need to manually hand-craft features.

In the context of Arabic NLP, the above neural techniques have also been applied to sequence tagging tasks including NER [21], [22], [23], [24], POS tagging [25], [26], and segmentation [27], outperforming classical rule-based approaches [28], [29], which certainly shows the promise of these techniques when applied to morphologically-rich languages such as Arabic.

With respect to **NER** but mostly in the context of MSA, due to lack of dialectal NER datasets. For example, [30] applied a CRF layer over n-gram features to perform NER. [31] combined a decision tree [32] with rule-based features. Other, but little, work has focused on NER in the context of social media data, where DA and MSA are usually mixed together. For instance, [29] used cross-lingual resources, namely English to improve Arabic NER. However, they obtained poor results when evaluating on social media data. More recently, [21] applied bi-directional LSTM networks on both character- and word-levels to perform NER on the Tweets dataset [29]. As for Egyptian dialect, specifically, [33] performed NER by applying a CRF tagger on a set of lexical, morphological, and gazetteer-based features. Their approach showed improvements over baselines but the performance on dialectal data was not on par with it on MSA data, showing the challenges brought by dialectal contexts. To the best of our knowledge, little attention has been given to NER on dialectal Arabic and no prior work has studied the performance when training on MSA data and evaluating on DA data, respectively.

As for **POS tagging** and similarly to NER, the performance of models trained on MSA drops significantly when used with DA [34], [25]. Initial systems for Arabic POS tagging relied on both statistical features and linguistic rules crafted by experts [35], [36] or combined machine learning techniques with rules [37]. More recent work adopted classical machine learning model such as SVM applied on n-gram features [38], [39]. Other work used n-gram features. RNNs and their variants were later adapted for the task [40], [25], [41].

Dialectal Arabic POS tagging has received some attention although usually limited to work individual dialects such as Gulf [42], [25] and Egyptian [43], [44]. [45] studied multi-dialectal POS tagging by proposing an annotated DA dataset from twitter spanning 4 different dialects, namely, Gulf, Egyptian, Levantine, and Maghrebi. While their results show a performance drop on DA when training on MSA only, no attempt was done to improve the DA performance in that case. We can see that despite both the difficulty and scarcity of annotated DA data for all of the different dialects and tasks, most previous work has focused on annotating uni-dialectal datasets attempting to leverage the already abundant MSA datasets. A classical work [43], who employed an MSA morphological analyzer with a minimal supervision to perform POS tagging on Egyptian data with unlabeled Egyptian and Levantine data.

Sarcasm Detection (SRD) is the task of identifying sarcastic utterances where the author intends a different meaning than what is being literally enunciated [46]. Sarcasm detection is crucial for NLU as neglecting sarcasm can easily lead to the misinterpretation of the intended meaning, and therefore significantly degrade the accuracy of tasks such as sentiment classification, emotion recognition, and opinion mining. Much research effort has addressed Sarcasm detection in English, where abundant resources exist [47], [48], [49], [50]. Earlier methods employed linguistic rules [51] or classical machine learning models [49], [52]. More recent methods used neural networks [53], [54], [55], [56], [57], [58] or pre-trained language models [59], [60], [61], [62].

With respect to Arabic Sarcasm Detection, the majority of research has focused on detecting sarcastic tweets. The author in [63] used Random Forests to identify sarcastic political tweets. [64] proposed a shared task on irony detection in Arabic Tweets. The submitted systems to the shared task varied in their approaches from classical models with count-based features [65], [66] to deep models [67], [68]. [69] highlighted the connection between sentiment analysis and sarcasm detection, by showing how sentiment classifiers fail with sarcastic inputs. They also proposed the largest publicly available Arabic sarcasm detection dataset, ArSarcasm, which we use in this work. We can see that so far, sarcasm detection methods have been applied to social media data collectively, with no effort made to study the zero-shot performance across dialects of state-of-the-art methods.

Pre-trained Language Models. Sequential transfer learning, where a network is first pre-trained on a relevant task before fine-tuning on the target task, originally appeared in domain of computer vision, and has recently been adapted in NLP. The author in [70] proposed to pre-train a LSTM network for language modeling and then fine-tune for classification. Similarly, ELMO [3] leveraged contextual representations obtained from a network pretrained for language modeling to perform many NLP tasks. Similar approaches were proposed such as BERT [5] that relied not on RNNs, but on bidirectional Transformers [4], and on a different pre-training objective, namely masked language modeling. Other variations appeared including RoBERTa [6], MASS [71], and ELECTRA [72]. Fine-tuning these pre-trained models on task-specific data has produced state-of-the-art performance, especially in cases when sufficiently large labeled data does not exist. They have been applied to several tasks, including text classification, question answering, named entity recognition [9], and POS tagging [8].

Cross-lingual Learning. Cross-lingual learning (CLL) refers to using labeled resources from resource-rich languages to build models for data-scarce languages. In a sense, knowledge learned about language structure and tasks is *transferred* to low-resource languages Cross-lingual learning is of particular importance due to the scarcity of labeled resources in many of the world's languages, some of which are spoken by millions of people (Marathi and Gondi, for example). While our work can be better described as cross-dialectal, the techniques used for cross-lingual learning can easily be adapted for settings such as ours. In this work, Modern Standard Arabic (MSA) and Arabic dialects (DA) represent the high-resource and low-resource languages, respectively.

Many techniques were proposed for CLL, including using cross-lingual word embeddings [73], [74], [75], [76], where the two monolingual vector spaces are mapped into the same shared space. While cross-lingual word embeddings enable comparing meaning across languages [73], they typically fail when we do not have enough data to train good monolingual embeddings. In addition, adversarial learning [77] has played an important role in cross-lingual learning where an adversarial objective is employed to learn language-independent representations [78], [79], [80], [81]. As a result, the model learns to rely more on general language structure and commonalities between languages, and therefore can generalize across languages. Multilingual extensions of pre-trained language models have emerged through joint pre-training on several languages. Examples include mBERT [5], XLM [82] and XLM-RoBERTa [9]. During pre-training on multiple languages, the model learns to exploit common structure among pre-training languages even without explicit alignment [83]. These models have become useful for few-shot and zero-shot cross-lingual settings, where there is little or no access to labeled data in the target language. For instance [9] evaluate a cross-lingual version of RoBERTa [6], namely XLM-RoBERTa, on cross-lingual learning across different tasks such as question answering, text classification, and named entity recognition.

Semi-supervised Learning. Several methods were proposed for leveraging unlabeled data for learning including co-training [84], graph-based learning [85], tri-training [86], and self-training [87]. A variety of semi-supervised learning methods have been successfully applied to a number of NLP tasks including NER [88], [89], POS tagging [90], parsing [91], word sense disambiguation [92], and text classification [93], [94]. Self-training has been applied in cross-lingual settings where gold labels are rare in the target language. For example, [95] proposed a combination of Active learning and self-training for cross-lingual sentiment classification. [96] made use of self-training for named entity tagging and linking across 282 different languages. [97] used self-training for cross-lingual word mapping to create additional word pairs for training. [98] employed self-training to improve zero-shot cross-lingual sentiment classification with mBERT [5]. With English as their source language, they improved performance on 7 languages by self-training using unlabeled data in their target languages. Lastly, [99] used the self-labeled examples produced by self-training to create adversarial examples in order to improve robustness and generalization.

We now introduce our tasks.

III. Tasks

Named Entity Recognition (NER) is defined as the information extraction task that attempts to locate, extract, and automatically classify named entities into predefined classes or types in unstructured texts [100]. Typically, NER is integrated into more complex tasks, where, for example, we might need to handle entities in a special way. For instance, when translating the Arabic sentence "حقق كرم فضية المصارعه" to English, it would be useful to know that "كرم" is a person name, and therefore should not be be translated into the word "generosity". Similarly, NER can be useful for other tasks question answering, information retrieval and summarization.

Part-of-Speech (POS) tagging is the task of assigning a word in a context to its part-of-speech tag. Such tags include adverb (ADV), adjective (ADJ), pronoun (PRON), and many others. For example, given an input sentence "أنا أحب كرة القدم", our goal is to tag each word as follows: أنا (PRON) أحب (VERB) كرة (NOUN) ال (DET) قدم (NOUN). POS tagging is an essential NLU task with many applications in speech recognition, machine translation, and information retrieval. Both NER and POS tagging are sequence labeling tasks, where we assign a label to each word in the input context.

Sarcasm Detection is the task of identifying sarcastic utterances where the author intends a different meaning than what is being literally enunciated [46]. Sarcasm detection is crucial for NLU as neglecting to detect sarcasm can easily lead to the misinterpretation of the intended meaning, and therefore significantly degrade the accuracy of tasks such as sentiment classification, emotion recognition, and opinion mining [69]. For example the word "سعيد" in the utterance "أنا سعيد جدا بهذا الجوال البطيء" can erroneously lead sentiment classifiers into positive sentiment, although the sentiment has negative sentiment. Sarcasm Detection is typically treated as a binary classification task, where an utterance is classified as either sarcastic or not.

IV. Method

In this work, we show that models trained on MSA for NER, POS tagging, and Sarcasm Detection generalize poorly to dialect inputs when used in zero-shot-settings (i.e., no annotated DA data used during training). Across the three tasks, we test how self-training would fare as an approach to leverage unlabeled DA data to improve performance on DA. Self-training involves training a model using its own predictions on a set of unlabeled data identical from its original training split. Next, we formally describe our algorithm. The notation used in this section to describe our algorithm is directed towards sequence labeling (since we experiment with 2 sequence labeling tasks out of 3). However, it should be straightforward to adapt it to the context of text classification as in [98].

A. Self-training for Sequence Labeling

For sequence labeling, our proposed self-training procedure is given two sets of examples: a labeled set L and an unlabeled set U. To perform zero-shot MSA-to-DA transfer, MSA examples are used as the labeled set, while unlabeled DA examples are the unlabeled set. As shown in Fig. 1, each iteration of the self-training algorithm consists mainly in three steps. First, a pre-trained language model is fine-tuned on the labeled MSA examples L. Second, for every unlabeled DA example u_i, we use the model to tag each of its tokens to obtain a set of predictions and confidence scores for each token $p_{u_i} = (l_1^{(i)}, c_1^{(i)}), (l_2^{(i)}, c_2^{(i)}), ...(l_{|u_i|}^{(i)}, c_{|u_i|}^{(i)})$, where $(l_j^{(i)}, c_j^{(i)})$ are the label and confidence score (Softmax probability) for the j-th token in u_i. Third, we employ a selection mechanism to identify examples from U that are going to be added to L for the next iteration.

For a selection mechanism, we experiment with both a thresholding approach and a fixed-size [98] approach. In the thresholding method, a threshold τ is applied on the minimum confidence per example. That is, we only add an example u_i to L if $\min_{(l_j^{(i)}, c_j^{(i)}) \in p_{u_i}} c_j^{(i)} \geq \tau$. See Algorithm 1. The fixed-size approach involves, at each iteration, the selection of the top S examples with respect to the minimum confidence score $\min_{(l_j^{(i)}, c_j^{(i)}) \in p_{u_i}} c_j^{(i)}$, where S is a hyper-parameter. We experiment with both approaches and report results in Section VII.

Algorithm 1: MSA-to-DA Self-Training for Sequence Labeling

```
Given set $L$ of labeled MSA examples, set $U$ of
  unlabeled DA examples, $\tau$ parameter for probability
  threshold selection.
repeat
    Fine-tune model $M$ for $K$ epochs on labeled
      MSA examples $L$;
    for $u_i \in U$ do
        Obtain prediction $p_{u_i}$ on unlabeled DA
          example $u_i$ using model $M$;
        if $\min_{(l_j^{(i)}, c_j^{(i)}) \in p_{u_i}} c_j^{(i)} \geq \tau$ then
            remove $u_i$ from $U$ and add it to $L$;
    end
until stopping criterion satisfied
```

B. Self-training for Classification

For sarcasm detection, we follow [98] who select an equivalent number of examples from each class, which we will refer to as *class balancing*. In other words, let c_{u_i} be the confidence of the most probable class assigned to example u_i. Then we sort the unlabeled examples in a descending order according to their confidence and select the top $\lfloor S/C \rfloor$ examples from each class such that we have a total of S examples, where C is the number of classes.

For example if $S = 100$ and $C = 2$ i.e we have 2 classes, we will select the top 50 confident examples that were classified as positive and the top 50 confident examples classified as negative. Similarly to [98], we observe the positive effect of class balancing on the performance of self-training in sarcasm detection[1] and we compare class balancing against selecting the top S confident example regardless of their predicted class. See Section VII-C.

V. Pretrained Language Model

In this work, we turn our attention to fine-tuning pre-trained language models (PLMs) on our three tasks. While self-training can basically be applied to many types of other models such as LSTM networks [14], we select PLMs for two reasons. First, PLMs have been shown to outperform models trained from scratch on a wide variety of tasks [5], [70], [82]. Second, we aim to show that even state-of-the-art models still perform poorly in certain low-resource settings asserting that we still need methods to handle such scenarios.

Pre-trained language models make use As a pre-trained language model, we use XLM-RoBERTa [9] (XML-R for short). XLM-R is a cross-lingual model, and we choose it since it is reported to perform better than mBERT, the multilingual

[1] We do not use class balancing with sequence labeling tasks since each example contains a set of tokens, each assigned to a possibly different class, which makes it very difficult to guarantee that an equal number of examples are selected for each class.

Algorithm 2: MSA-to-DA Self-Training for Classification

```
Given set $L$ of labeled MSA examples, set $U$ of
  unlabeled DA examples, $S$ total number of unlabeled
  examples to add to the training data every iteration,
  $C$ the number of classes.
repeat
    Fine-tune model $M$ for $K$ epochs on labeled
      MSA examples $L$;
    Obtain class predictions and confidences on all
      unlabeled DA examples $u_i$ using model $M$;
    Sort all unlabeled examples $u_i$ in descending
      order by the confidence of their most probable
      class $c_{u_i}$;
    Select the top $\lfloor S/C \rfloor$ examples from each class,
      remove them from $U$, and add them to $L$;
until stopping criterion satisfied
```

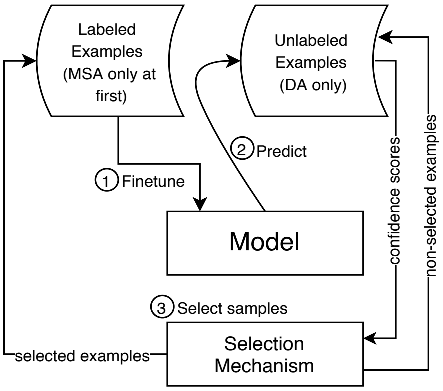

Fig. 1: MSA-to-DA Self-Training Transfer.

model from Google [5]. XLM-R also uses Common Crawl for training, which is more likely to have dialectal data than Wikipedia Arabic (used in mBERT), making it more suited to our work. We now introduce our experiments.

VI. Experiments

We begin our experiments with evaluating the standard fine-tuning performance of XLM-R models on NER, POS tagging, and SRD against strong baselines. We then use our best models from this first round to investigate the MSA-to-DA zero-shot transfer, showing a significant performance drop even when using pre-trained XLM-R. Consequently, we evaluate self-training in zero- (NER, POS tagging, SRD) and few-shot (POS tagging) settings, showing substantial performance improvements in both cases. We now introduce our datasets.

A. Datasets

NER: For our work on NER, we use four datasets: ANERCorp [101];ACE 2003 [102] BNews (BN-2003);ACE 2003 Newswire (NW-2003); and Twitter [29]. Named entity types in all datasets are *location (LOC)*, *organization (ORG)*, and *person (PER)*.

POS Tagging: There are a number of Arabic POS tagging datasets, mostly on MSA [103] but also on dialects such as EGY [104]. To show that the proposed approach is able to work across multiple dialects, we ideally needed data from more than one dialect. Hence, we use the multi-dialectal (MD) dataset from [45], comprising 350 tweets from various Arabic dialects including MSA, Egyptian (EGY), Gulf (GLF), and Levantine (LEV). This dataset has 21 POS tags, some of which are suited to social media (since it is derived from Twitter). We show the POS tag set from [45] in Table XIII (in the Appendix). We further evaluate fine-tuning XLMR for POS tagging on a Classical Arabic dataset, namely the Quranic Arabic Corpus (QAS). [105].

Sarcasm Detection: We use the Ar-Sarcasm dataset provided by [69], which has a total of 10,547 example split into training and test sets. Each example in this dataset is labeled by its dialect and sarcasm label. For our experiments, we set aside 20% of the training data as a development set. Table I shows sizes of the datasets used. We now introduce our baselines.

B. Baselines

For the **NER task**, we use the following baselines:

- **NERA [31]**: A hybrid system of rule-based features and a decision tree classifier.
- **WC-BiLSTM [21]**: A character- and a word-level Bi-LSTM with a conditional random fields (CRF) layer.
- **WC-CNN [22]**: A character- and a word-level CNN with a CRF layer.
- **mBERT [5]**: A fine-tuned multilingual BERT-Base-Cased (110M parameters), pre-trained with a masked language modeling objective on the Wikipedia corpus of 104 languages (including Arabic). For fine-tuning, we find that (based on experiments on our development set) a learning rate of 6×10^{-5} works best with a dropout of 0.1.

In addition, we compare to the published results in [28], AraBERT [106], and CAMel [107] for the ANERCorp dataset. We also compare to the published results in [22] for the 4 datasets.

For the **POS tagging task**, we compare to our own implementation of WC-BiLSTM (since there is no published research that uses this method on the task, as far as we know) and run mBERT on our data. We also compare to the CRF results published by [45]. In addition, for the Gulf dialect, we compare to the BiLSTM with compositional character representation and word representations (CC2W+W) published results in [25].

For the **Sarcasm Detection task**:

- **Word-level BiLSTM**: A bidirectional LSTM on the word level. We use the same hyper-parameters as in [69].
- **Word-level CNN [108]**: the network is has one convolutional layer of 10 filters of sizes 3, 5, and 7.
- **mBERT [5]**: mBERT fine-tuned for SRD. Here, we find that a different learning rate of 5×10^{-6} performs best.

C. Experimental Setup

Our main models are XLM-R$_{\text{BASE}}$ $(L = 12, H = 768, A = 12,$ 270M params) and XLM-R$_{\text{LARGE}}$ $(L = 24, H = 1024, A = 16,$ 550M params), where L is number of

TABLE I: Datasets used for Each of the 3 Tasks Studied

Task	Dataset	Size
NER	ANERCorp [101]	∼150K tokens
	ACE 2003-BNews [102]	∼15K tokens
	ACE 2003-News Wire [102]	∼27K tokens
	Twitter [29]	∼81K tokens
POS Tagging	Multi-dialectal (MD) - MSA [45]	∼26K tokens
	Multi-dialectal (MD) - EGY [45]	∼23K tokens
	Multi-dialectal (MD) - GLF [45]	∼21K tokens
	Multi-dialectal (MD) - LEV [45]	∼23K tokens
	Quranic Arabic Corpus (QAC)	∼134K tokens
Sarcasm Detection	Ar-Sarcasm [69]	∼10K sentences

layers, H is the hidden size, A is the number of self-attention heads. For XLM-R experiments, we use Adam optimizer with $1e^{-5}$ learning rate, batch size of 16. We typically fine-tune for 20 epochs, keeping the best model on the development set for testing. We report results on the test split for each dataset, across the two tasks. For all BiLSTM experiments, we use the same hyper-parameters as [22].

For all the self-training experiments, we use the dialect subset of the Arabic online news commentary (AOC) dataset [109], comprising the EGY, GLF, and LEV varieties limiting to equal sizes of 9K examples per dialect (total =27K) [2]. We use the split from [110] of AOC, removing the dialect labels and just using the comments themselves for our self-training. Each iteration involved fine-tuning the model for $K = 5$ epochs. As a stopping criterion, we use early stopping with patience of 10 epochs. Other hyper-parameters are set as listed before. For selecting confident samples, we experiment with a fixed number of top samples $S = [50, 100, 200]$ and selection based on a probability threshold $\tau = [0.80, 0.90, 0.95]$ (softmax values) [3]. For all evaluations, we use the *seqeval* toolkit [4].

VII. Results

A. Fine-tuning XLM-R

We start by showing the result of fine-tuning XLM-R on the **NER task**, on each of the four Arabic NER (ANER) datasets listed in Section VI-A. Table II shows the test set macro F_1 score on each of the four ANER datasets. Clearly, the fine-tuned XLM-R models outperform other baselines on all datasets, except on the NW-2003 where WC-CNN [22] performs slightly better than XLM-R$_{\text{LARGE}}$.

For **POS Tagging**, Table III shows test set word accuracy of the XLM-R models compared to baselines on the Quranic Arabic Corpus (QAC) and four different subsets from the multi-dialectal dataset [45]. Again, XLM-R models (both base and large) outperform all other models. A question arises why XLM-R models outperform both mBERT and AraBERT. As noted before, for XLM-R vs. mBERT, XLM-R was pre-trained on much larger data: CommonCrawl for XLM-R vs. Wikipedia for mBERT. Hence, the *larger dataset* of XLM-R is giving it an advantage over mBERT. For comparison with AraBERT, although the pre-training data for XLM-R and AraBERT may be comparable, even the smaller XLM-R model (XLM-R$_{\text{BASE}}$) has more than twice the number of parameters of the BERT$_{\text{BASE}}$ architecture on which AraBERT and mBERT are built (270M v. 110M). Hence, XLM-R model *capacity* gives it another advantage. We now report our experiments with zero-shot transfer from MSA to DA.

For **Sarcasm Detection**, we fine-tune XLM-R$_{\text{BASE}}$ and XLM-R$_{\text{LARGE}}$ on the full Ar-Sarcasm dataset and compare their performance against three different baselines in Table IV. Worst performance is given by CNN, which can be attributed to the way CNNs work; by capturing local n-gram features, the CNN filters fail to learn the wide contextual features required to detect sarcasm. Clearly, mBERT is performing very well compared to BiLSTM and CNN but XLM-R$_{\text{BASE}}$ and XLM-R$_{\text{LARGE}}$ outperfrom all other baselines on the task with 69.83% and 74.07% macro F1 points, respectively, achieving new state-of-the-art on the Ar-Sarcasm dataset.

B. MSA-DA Zero-Shot Transfer

As before, we start by the discussion of **NER experiments**. To evaluate the utility of approach, we obviously need DA data labeled for NER. We observed that the dataset from [29] contains both MSA and DA examples (tweets). Hence, we train a binary classifier to distinguish DA data from MSA[5]. We then extract examples that are labeled with probability $p > 0.90$ as either DA or MSA. We obtain 2,027 MSA examples (henceforth, *Darwish-MSA*) and 1,695 DA examples (henceforth, *Darwish-DA*), respectively. We split these into development and test sets with 30% and 70% ratios. As **for POS Tagging**, we already have MSA data for training and the three previously used DA datasets, namely EGY, GLF and LEV, for evaluation. We use those for the zero-shot setting by omitting their training sets and using only the development and test sets.

We first study how well models trained for NER and POS tagging on MSA data only will generalize to DA inputs during test time. We evaluate this zero-shot performance on both the XLM-R$_{\text{BASE}}$ and XLM-R$_{\text{LARGE}}$ models. **For NER**, we train on ANERCorp (which is pure MSA) and evaluate

[2] We note that our approach could be scaled with an even bigger unlabeled dataset, given the performance gains we report with self-training in this work.

[3] It is worth noting that our S values are similar to those used in [98]. We also experimented with other values for τ and S, but found them sub-optimal and hence we report performance only for the listed values of these two hyper-parameters here.

[4] https://github.com/chakki-works/seqeval

[5] The classifier is XLM-R$_{\text{BASE}}$ fine-tuned on the AOC data. The fine-tuned model achieved development and test accuracies of 90.3% and 89.4 %, respectively, outperforming the best results in [110].

TABLE II: Test Set Macro F_1 Scores for NER

Model	ANERCorp	BN-2003	NW-2003	NW-2004	Twitter
NERA [31]	88.77	–	–	–	–
CAMeL [107]	85.00	–	–	–	–
Hybrid [28]	90.66	–	–	–	–
WC-BiLSTM [21]	88.56	94.92	90.32	89.62	64.93
WC-CNN [22]	88.77	94.12	**91.20**	**91.47**	65.34
mBERT (ours)	85.86	89.52	87.19	88.58	58.92
AraBERT [106]	84.2	–	–	–	–
XLM-R_{BASE} (ours)	87.75	95.35	85.25	89.61	60.39
XLM-R_{LARGE} (ours)	**91.43**	**97.33**	91.10	90.78	**68.91**

TABLE III: Test Set Accuracy for POS Tagging using Several Baselines

Model	QAC	MD-MSA	MD-EGY	MD-GLF	MD-LEV
BiLSTM (CC2W + W) [25]	–	–	–	89.7	–
CRF [45]	–	93.6	92.9	87.8	87.9
WC-BiLSTM (ours)	91.65	94.63	93.41	88.79	86.13
mBERT (ours)	94.83	90.57	92.88	87.85	72.30
XLM-R_{BASE} (ours)	**96.70**	96.30	94.70	92.18	89.98
XLM-R_{LARGE} (ours)	96.59	**98.21**	**97.00**	**94.41**	**93.19**

TABLE IV: Macro F1 Scores with Several Baselines for Sarcasm Detection. Dataset used is Ar-Sarcasm [69]

Model	DEV	TEST
BiLSTM [69]	63.51	62.19
CNN	59.7	58.50
mBERT	68.87	69.51
XLM-R_{BASE} (ours)	73.22	69.83
XLM-R_{LARGE} (ours)	**73.72**	**74.07**

on both Darwish-MSA and Darwish-DA. While for POS tagging, we train on the MSA subset [45] and evaluate on the corresponding test set for each dialect. As shown in Table V, For NER, a significant generalization gap of around 20 % F_1 points exists between evaluation on MSA and DA using both models. While for **POS tagging**, the gap is as large as 18.13 % accuracy for the LEV dialect with XLM-R_{BASE}. The smallest generalization gap is on the GLF variety, which is perhaps due to the high overlap between GLF and MSA [25].

For Sarcasm Detection, Since Ar-Sarcasm is labeled by dialect, it is trivial to extract the MSA examples for training. Similarly to what was done with the NER data, we split all[6] the remaining DA examples into development and test sets with 30% and 70% ratios, respectively for evaluation. Finally, we obtain 4506 MSA training, 1202 DA development, and 2268 DA test examples. As shown in Table V, a performance gap of around 8 macro F1 points with both XLM-R_{BASE} and XLM-R_{LARGE}, showing poor generalization on DA in context of text classification, as well. In the next section, we evaluate the ability of self-training to close this MSA-DA performance gap.

C. Zero-shot Self-Training

Here, **for NER**, similar to Section VII-B, we train on ANERCorp (pure MSA) and evaluate on Darwish-MSA and Darwish-DA. Table VI shows self-training NER results employing the selection mechanisms listed in Section IV, and with different values for S and τ. The best improvement is achieved with the thresholding selection mechanism with a $\tau = 0.90$, where we have an F_1 gain of 10.03 points. More generally, self-training improves zero-shot performance in all cases albeit with different F_1 gains. Interestingly, we find that self-training also improves test performance on MSA with the base XLM-R model. This is likely attributed to the existence of MSA content in the unlabeled AOC data. It is noteworthy, however, that the much higher-capacity large model deteriorates on MSA if self-trained (dropping from 68.32% to 67.21%). This shows the ability of the large model to learn representations very specific to DA when self-trained. It is also interesting to see that the best self-trained base model achieving 50.10% F_1, outperforming the large model before the latter is self-trained (47.35% in the zero-shot setting). This shows that a base self-trained model, suitable for running on terminal machines with less computational capacity, can (and in our case does) improve over a large (not-self-trained) model that needs significant computation. The fact that, when self-trained, the large model improves 15.35% points over the base model in the zero-shot setting (55.42 vs. 40.07) is remarkable.

As **for POS tagging**, we similarly observe consistent improvements in zero-shot transfer with self-training (Table VII). The best model achieves accuracy gains of 2.41% (EGY), 1.41% (GLF), and 1.74% (LEV). Again, this demonstrates the utility of self-training pre-trained language models on the POS tagging task even in absence of labeled dialectal POS data (zero-shot).

For **Sarcasm Detection**, we follow [98] in balancing the examples selected in each self-training iteration through selecting an equal number of examples from each class (sarcastic and non-sarcastic). Without the balancing step, we find that

[6] Without this, we had only 528 and 698 development and test examples, respectively and it resulted in high variance in the results obtained. So we had to increase the sizes of the development and test sets by sacrificing the DA training data.

TABLE V: Zero-shot Transfer Results on the DA Test Sets. Metrics used are Macro F_1 for NER and Sarcasm Detection, and Accuracy for POS Tagging. Models are Trained on MSA only and Evaluated on DA. Datasets used are: Darwish-MSA and Darwish-DA [29] (NER), Multi-Dialectal [45] (POS Tagging), and Ar-Sarcasm [69] (Sarcasm Detection). As shown, a Significant Performance Drop Exists when Training on MSA and Evaluating on DA

Model	**NER**		**POS Tagging**				**Sarcasm Detection**	
	MSA	**DA**	**MSA**	**EGY**	**GLF**	**LEV**	**MSA**	**DA**
XLM-R_{BASE}	60.42	40.07	96.30	78.38	83.72	78.17	68.68	60.17
XLM-R_{LARGE}	68.32	47.35	98.21	82.28	85.95	81.24	71.55	62.90

TABLE VI: Test Set Macro F_1 in the Zero-Short Setting for NER. Training was Done on MSA Data Only. **ST** Stands for Self-Training. Models were Trained on ANERCorp (Pure MSA) and Evaluated on Darwish-MSA and Darwish-DA Extracted from the Twitter Dataset [45]. Self-training Boosts the Performance on DA Data by 10% Macro F1 Points with XLM-R_{BASE} and $\tau = 0.90$

Model	**Darwish-MSA**	**Darwish-DA**
XLM-R_{BASE}	61.88	40.07
XLM-R_{BASE}, ST, S=50	60.98	43.88
XLM-R_{BASE}, ST, S=100	61.13	42.01
XLM-R_{BASE}, ST, S=200	61.46	43.49
XLM-R_{BASE}, ST, $\tau = 0.80$	**63.36**	46.97
XLM-R_{BASE}, ST, $\tau = 0.90$	61.02	**50.10**
XLM-R_{BASE}, ST, $\tau = 0.95$	62.25	47.91
XLM-R_{LARGE}	**68.32**	47.35
XLM-R_{LARGE} + ST, $\tau = 0.90$	67.21	**55.42**

the selected examples come from the most frequent class (non-sarcastic), which hurts performance since the model is learning only one class. The results for sarcasm detection are shown in Table VIII, where we see that self-training adds 3% and 2.5% (for XLM-R_{BASE}) and 5.9% and 4.5% (for XLM-R_{LARGE}) macro F1 points on the development and test sets, respectively using the best settings for self-training ($S = 100$ with class balancing). We also find that selection based on probability thresholds performs much worse than fixed-size selection, hence we omit these results.

TABLE VII: Test Set Accuracy in the Zero-Shot Setting for POS Tagging. **ST** Stands for Self-Training. Models were Trained on the MSA Data of the When Training on MSA Only. Self-Training Boosts Performance of XLMR$_{BASE}$ by Around 2% Accuracy Points on Different Dialects with the Best Setting of $S = 50$

Model	**MSA**	**EGY**	**GLF**	**LEV**
XLM-R_{BASE}	96.30	78.38	83.72	78.17
XLM-R_{BASE}, ST, S=50	–	**80.79**	**85.13**	**79.91**
XLM-R_{BASE}, ST, S=100	–	80.43	84.74	79.16
XLM-R_{BASE}, ST, S=200	–	78.75	84.21	79.40
XLM-R_{BASE}, ST, τ = 0.90	–	79.52	83.97	79.21
XLM-R_{BASE}, ST, τ = 0.85	–	78.97	83.53	79.06
XLM-R_{BASE}, ST, τ = 0.80	–	78.88	83.72	78.50
XLM-R_{LARGE}	98.21	82.28	85.95	81.24
XLM-R_{LARGE} + ST, S=50	–	**82.65**	**87.76**	**83.70**

D. Ablation Experiment

Here, we conduct an ablation experiment with the NER task in order to verify our hypothesis that the performance boost primarily comes from using unlabeled DA data for self-training. By using a MSA dataset with the same size as our unlabeled DA one[7], we can compare the performance of the self-trained model in both settings: MSA and DA unlabeled data. We run three different self-training experiments using 3 different values for τ using each type of unlabeled data. Results are shown in Table IX. While we find slight performance boost due to self-training even with MSA unlabeled data, the average F1 score with unlabeled DA is better by 2.67 points, showing that using unlabeled DA data for self-training has helped the model adapt to DA data during testing.

VIII. Error Analysis

A. NER

To understand why self-training the pre-trained language model improves over mere fine-tuning, we perform an error analysis. For the error analysis, we focus on the NER task where we observe a huge self-training gain. We use the development set of Darwish-DA (see Section VII-C) for the error analysis. We compare predictions of the standard fine-tuned XLM-R_{BASE} model (FT) and the best performing self-training ($\tau = 0.9$) model (ST) on the data. The error analysis leads to an interesting discovery: The greatest benefit from the ST model comes mostly from reducing *false positives* (see Table X). In other words, self-training helps regularize the model predictions such that tokens misclassified by the original FT model as a named entities are now correctly tagged as *unnamed entity* "O".

To understand why the ST model improves false positive rate, we manually inspect the cases it correctly identifies that were misclassified by the FT model. We show examples of these cases in Table XIV (in the Appendix). As the table shows, the ST model is able to identify dialectal tokens whose equivalent MSA forms can act as trigger words (usually followed by a PER named entity). We refer to this category as ***false trigger words***. An example is the word نبي "prophet" (row 1 in Table XIV). A similar example that falls within this category is in row (2), where the model is confused by the token الي ("who" in EGY, but "to" in MSA and hence the wrong prediction as LOC). A second category of errors is caused by ***non-standard social media language***, such as use of letter repetitions in interjections (e.g., in row (3) in

[7] We use a set of MSA tweets from the AOC dataset mentioned before.

TABLE VIII: Macro F$_1$ in the Zero-Shot Setting for Sarcasm Detection on the Ar-Sarcasm [69] Dataset. Training was Done on MSA Data Only. **ST:** Stands for Self-Training. An Obvious Performance Boost Occurs when using Self-Training in the Best Setting with $S = 100$ and Class Balancing

Model	**MSA**		**DA**	
	DEV	**TEST**	**DEV**	**TEST**
XLM-R$_{\text{BASE}}$	65.64	68.68	61.66	60.17
XLM-R$_{\text{BASE}}$ + ST, S=50,	–	–	62.53	60.82
XLM-R$_{\text{BASE}}$ + ST, S=100	–	–	61.15	59.46
XLM-R$_{\text{BASE}}$ + ST, S=200	–	–	62.57	60.25
XLM-R$_{\text{BASE}}$ + ST, S=50, class balancing	–	–	62.49	59.34
XLM-R$_{\text{BASE}}$ + ST, S=100, class balancing	–	–	**64.72**	**62.66**
XLM-R$_{\text{BASE}}$ + ST, S=200, class balancing	–	–	62.89	59.46
XLM-R$_{\text{LARGE}}$	67.81	71.55	62.28	62.90
XLM-R$_{\text{LARGE}}$ + ST, S=100, class balancing	–	–	**68.21**	**67.43**

TABLE IX: Ablation Experiment with MSA Unlabeled Data for Zero-Shot NER. Development Set Macro F1 is Shown when Using Both Unlabeled MSA and DA Data with the Same Size. Average Performance with DA Unlabeled Data is Higher Showing the Effect of Unlabeled DA on the Model Final Performance

Setting	**Unlbl. MSA**	**Unlbl. DA**
XLM-R$_{\text{BASE}}$, ST, $\tau = 0.80$	43.88	44.46
XLM-R$_{\text{BASE}}$, ST, $\tau = 0.90$	44.69	47.83
XLM-R$_{\text{BASE}}$, ST, $\tau = 0.95$	43.43	46.87
Avg	43.67	**46.34**

Table XIV). In these cases, the FT model also assigns the class PER, but the ST model correctly identifies the tag as "O". A third class of errors arises as a result of ***out-of-MSA*** vocabulary. For example, the words in rows (4-6) are all out-of-MSA where the FT model, not knowing these, assigns the most frequent named entity label in train (PER). A fourth category of errors occurs as a result of a token that is usually part of a named entity in MSA, that otherwise functions as part of an ***idiomatic expression*** in DA. Row (7) in Table XIV illustrates this case.

We also investigate errors shared by both the FT and ST models (errors which the ST model also could not fix). Some of these errors result from the fact that often times both MSA and DA use the same word for both person and location names. Row (1) in Table XV (in the Appendix) is an example where the word "Mubarak", name of the ex-Egypt President, is used as LOC. Other errors include *out-of-MSA* tokens mistaken as named entities. An example is in row (3) in Table XV, where بأمارة ,("proof" or "basis" in EGY) is confused for بإمارة ("emirate", which is a location). *False trigger words*, mentioned before, also play a role here. An example is in row (7) where يابطل is confused for PER due to the trigger word يا "Hey!" that is usually followed by a person name. ***Spelling mistakes*** cause third source of errors, as in row (4). We also note that even with self-training, detecting ORG entities is more challenging than PER or LOC. The problem becomes harder when such organizations are not seen in training such as in rows (8) الاخوان المسلمين, (9) قناة العربية and (10) المجلس العسكري, all of which do not occur in the training set (ANERCorp).

Here we investigate the false negatives produces by the self-trained models observing a number of named entities that were misclassified by the self-trained model as unnamed ones. See Table XVI (in the Appendix). As an example, we take the last name الجنزوري which was classified both correctly and incorrectly in different contexts by the self-trained model. Context of correct classification is "هاش تاج لكمال الجنزوري ", while it is "ماسك على الناس كلها سي دي الا الجنزوري ماسك عليه فلوبي" for the incorrect classification. First, we note that الجنزوري is not a common name (zero occurrences in the MSA training set). Second, we observe that in the correct case, the word was preceded by the first name كمال which was correctly classified as PER, making it easier for the model to assign PER to the word afterwards as a surname.

TABLE X: Comparison of Error Categories in Percentage between the Fine-Tuned Model (FT) and the Model Combining Fine-Tuned+self-trained (ST) Model for NER. The Values are based on the Dialectal Part of the Development Set

Measure	**FT**	**ST**	**% improvement**
True Positives	155	165	+6.5 %
False Positive	159	64	+59.7 %
False Negatives	162	168	-3.7 %
True Negatives	5,940	6,035	+1.5 %

TABLE XI: **NER task.** Sample False Negatives Produced by Self-Training

no.	**Word**	**Gold**	**FT**	**ST**
(1)	الاخوان	ORG	ORG	O
(2)	للبرادعي	PER	PER	O
(3)	مجدي الجلاد	PER	PER	O
(4)	فان ديزل	PER	PER	O
(5)	الجنزوري	PER	PER	O
(6)	زين يسون	PER	PER	O

B. Sarcasm Detection

We also conduct an error analysis on Sarcasm Detection comparing the predictions of XLM-R$_{\text{BASE}}$ with and without self-training. For that we use the best model on the development set (XLM-R$_{\text{BASE}}$, S=100 with class balancing). Our analysis with SRD yields a similar observation to NER, where the performance boost driven by self-training is mostly due to the alleviation of false positives or the improvement of true negatives[8]. Table XII compares performance measures between the two models. However, we can see that, unlike NER, false negatives increase by as much as 44%, which is likely due to the self-training regularization effect mentioned earlier.

We also analyze sample errors that were fixed by the self-trained model. See Table XVII (in the Appendix). The first four examples represent false negatives, where the fine-tuned model assumed to be non-sarcastic. We can see that in such dialectal contexts, the fine-tuned model suffers from many unseen words during training on MSA. More specifically, words such as بيه and غساله in example (1), or عاهات in (2), عبيط in (4), or an idiom such as حاميها حراميها n e(3), or ما كانش حد غلب in (5), or ياترى in (6), all of which represent dialect-specific language that is not encountered in MSA contexts, and therefore represents a significant challenge in zero-shot settings.

In addition, we show sample errors shared between the fine-tuned and the self-training models. See Table XVIII (in the Appendix). As to why the self-trained model has not corrected these errors, we can hypothesize that it may be due to that the vocabulary used in these inputs was not seen during self-training. In other words, this vocabulary was either not selected by the self-training selection mechanism to be added to the training data or not existing at all in the unlabeled examples used for self-training. As a result, the model was not adapted sufficiently to handle these or similar contexts. We assume the performance on these inputs could improve with larger and more diverse unlabeled examples used for self-training.

TABLE XII: Comparison of Error Categories in Percentage between the Fine-Tuned Model (FT) and the Model Combining Fine-Tuned+Self-Trained (ST) Model for Sarcasm Detection, based on the Dialectal Part of the Development Set

Measure	**FT**	**ST**	**% improvement**
True Positives	737	688	-6.6 %
False Positive	230	185	+19.7 %
False Negatives	111	160	-44.1%
True Negatives	124	169	+36.29 %

IX. Conclusion

Even though pre-trained language models have improved many NLP tasks, they still need a significant amount of labeled data for high-performance fine-tuning. In this paper, we proposed to self-train pre-trained language models by using unlabeled Dialectal Arabic (DA) data to improve zero-shot performance when training on Modern Standard Arabic (MSA) data only. Our experiments showed substantial performance gains on two sequence labeling tasks (NER and POS), and one text classification task (sarcasm detection) on different Arabic varieties. Our method is dialect- and task-agnostic, and we believe it can be applied to other tasks and dialectal varieties. We intend to test this claim in future research. Moreover, we evaluated the fine-tuning of the recent XLM-RoBERTa language models, establishing new state-of-the-art results on all of the three tasks studied.

References

[1] W. Xu and A. Rudnicky, "Can artificial neural networks learn language models?" in *Sixth international conference on spoken language processing*, 2000.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[3] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, M. A. Walker, H. Ji, and A. Stent, Eds. Association for Computational Linguistics, 2018, pp. 2227–2237. [Online]. Available: https://doi.org/10.18653/v1/n18-1202

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," pp. 4171–4186, 2019. [Online]. Available: https://doi.org/10.18653/v1/n19-1423

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[7] A. Yang, Q. Wang, J. Liu, K. Liu, Y. Lyu, H. Wu, Q. She, and S. Li, "Enhancing pre-trained language representations with rich knowledge for machine reading comprehension," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2346–2357.

[8] H. Tsai, J. Riesa, M. Johnson, N. Arivazhagan, X. Li, and A. Archer, "Small and practical BERT models for sequence labeling," pp. 3630–3634, 2019. [Online]. Available: https://doi.org/10.18653/v1/D19-1374

[9] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," pp. 8440–8451, 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.747/

[10] K. Darwish, M. Attia, H. Mubarak, Y. Samih, A. Abdelali, L. Màrquez, M. Eldesouki, and L. Kallmeyer, "Effective multi dialectal arabic pos tagging," *Natural Language Engineering*, vol. 1, no. 1, p. 18, 2020.

[11] K. Shaalan, "A survey of arabic named entity recognition and classification," *Computational Linguistics*, vol. 40, no. 2, pp. 469–510, 2014.

[12] H. M. Wallach, "Conditional random fields: An introduction," *Technical Reports (CIS)*, p. 22, 2004.

[13] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, 2001.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[16] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," *CoRR*, vol. abs/1508.01991, 2015. [Online]. Available: http://arxiv.org/abs/1508.01991

[8] We can see that in binary classification, every false positive removed is a true negative added

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[18] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[19] Q. Wang and M. Iwaihara, "Deep neural architectures for joint named entity recognition and disambiguation," pp. 1–4, 2019. [Online]. Available: https://doi.org/10.1109/BIGCOMP.2019.8679233

[20] X. Ma and E. H. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," 2016. [Online]. Available: https://doi.org/10.18653/v1/p16-1101

[21] M. Gridach, "Character-aware neural networks for arabic named entity recognition for social media," in *Proceedings of the 6th workshop on South and Southeast Asian natural language processing (WS-SANLP2016)*, 2016, pp. 23–32.

[22] M. Khalifa and K. Shaalan, "Character convolutions for arabic named entity recognition with long short-term memory networks," *Computer Speech & Language*, vol. 58, pp. 335–346, 2019.

[23] M. Al-Smadi, S. Al-Zboon, Y. Jararweh, and P. Juola, "Transfer learning for arabic named entity recognition with deep neural networks," *IEEE Access*, vol. 8, pp. 37 736–37 745, 2020.

[24] I. El Bazi and N. Laachfoubi, "Arabic named entity recognition using deep learning approach." *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 9, no. 3, 2019.

[25] R. Alharbi, W. Magdy, K. Darwish, A. AbdelAli, and H. Mubarak, "Part-of-speech tagging for arabic gulf dialect using bi-lstm," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[26] W. AlKhwiter and N. Al-Twairesh, "Part-of-speech tagging for arabic tweets using crf and bi-lstm," *Computer Speech & Language*, vol. 65, p. 101138, 2021.

[27] Y. Samih, M. Attia, M. Eldesouki, A. Abdelali, H. Mubarak, L. Kallmeyer, and K. Darwish, "A neural architecture for dialectal arabic segmentation," in *Proceedings of the Third Arabic Natural Language Processing Workshop*, 2017, pp. 46–54.

[28] K. Shaalan and M. Oudah, "A hybrid approach to arabic named entity recognition," *Journal of Information Science*, vol. 40, no. 1, pp. 67–87, 2014.

[29] K. Darwish, "Named entity recognition using cross-lingual resources: Arabic as an example," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2013, pp. 1558–1567.

[30] A. Abdul-Hamid and K. Darwish, "Simplified feature set for Arabic named entity recognition," in *Proceedings of the 2010 Named Entities Workshop*. Association for Computational Linguistics, 2010, pp. 110–115.

[31] S. Abdallah, K. Shaalan, and M. Shoaib, "Integrating rule-based system with classification for arabic named entity recognition," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2012, pp. 311–322.

[32] Y.-Y. Song and L. Ying, "Decision tree methods: applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

[33] A. Zirikly and M. Diab, "Named entity recognition for arabic social media," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015, pp. 176–185.

[34] A. Pasha, M. Al-Badrashiny, M. T. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. Roth, "Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic." in *LREC*, vol. 14, no. 2014. Citeseer, 2014, pp. 1094–1101.

[35] S. Khoja, "Apt: Arabic part-of-speech tagger," in *Proceedings of the Student Workshop at NAACL*. Citeseer, 2001, pp. 20–25.

[36] S. Alqrainy, "A morphological-syntactical analysis approach for arabic textual tagging," 2008.

[37] Y. Tlili-Guiassa, "Hybrid method for tagging arabic text," *Journal of Computer science*, vol. 2, no. 3, pp. 245–248, 2006.

[38] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic tagging of arabic text: From raw text to base phrase chunks," in *Proceedings of HLT-NAACL 2004: Short papers*, 2004, pp. 149–152.

[39] J. H. Yousif and T. M. T. Sembok, "Arabic part-of-speech tagger based support vectors machines," in *2008 International Symposium on Information Technology*, vol. 3. IEEE, 2008, pp. 1–7.

[40] K. Darwish, H. Mubarak, A. Abdelali, and M. Eldesouki, "Arabic pos tagging: Don't abandon feature engineering just yet," in *Proceedings of the Third Arabic Natural Language Processing Workshop*, 2017, pp. 130–137.

[41] K. Alrajhi and M. A. ELAffendi, "Automatic arabic part-of-speech tagging: Deep learning neural lstm versus word2vec," *International Journal of Computing and Digital Systems*, vol. 8, no. 03, pp. 307–315, 2019.

[42] S. Khalifa, S. Hassan, and N. Habash, "A morphological analyzer for gulf arabic verbs," in *Proceedings of the Third Arabic Natural Language Processing Workshop*, 2017, pp. 35–45.

[43] K. Duh and K. Kirchhoff, "Pos tagging of dialectal arabic: a minimally supervised approach," in *Proceedings of the acl workshop on computational approaches to semitic languages*, 2005, pp. 55–62.

[44] R. Al-Sabbagh and R. Girju, "Yadac: Yet another dialectal arabic corpus." in *LREC*, 2012, pp. 2882–2889.

[45] K. Darwish, H. Mubarak, A. Abdelali, M. Eldesouki, Y. Samih, R. Alharbi, M. Attia, W. Magdy, and L. Kallmeyer, "Multi-dialect arabic pos tagging: a crf approach," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[46] C. Van Hee, E. Lefever, and V. Hoste, "Semeval-2018 task 3: Irony detection in english tweets," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 39–50.

[47] F. Barbieri, H. Saggion, and F. Ronzano, "Modelling sarcasm in twitter, a novel approach," in *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2014, pp. 50–58.

[48] G. Abercrombie and D. Hovy, "Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations," in *Proceedings of the ACL 2016 student research workshop*, 2016, pp. 107–113.

[49] A. Joshi, V. Sharma, and P. Bhattacharyya, "Harnessing context incongruity for sarcasm detection," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, pp. 757–762.

[50] M. Bouazizi and T. O. Ohtsuki, "A pattern-based approach for sarcasm detection on twitter," *IEEE Access*, vol. 4, pp. 5477–5488, 2016.

[51] S. K. Bharti, K. S. Babu, and S. K. Jena, "Parsing-based sarcasm sentiment recognition in twitter data," in *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2015, pp. 1373–1380.

[52] S. Saha, J. Yadav, and P. Ranjan, "Proposed approach for sarcasm detection in twitter," *Indian Journal of Science and Technology*, vol. 10, no. 25, pp. 1–8, 2017.

[53] S. Porwal, G. Ostwal, A. Phadtare, M. Pandey, and M. V. Marathe, "Sarcasm detection using recurrent neural network," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2018, pp. 746–748.

[54] Y. Ren, D. Ji, and H. Ren, "Context-augmented convolutional neural networks for twitter sarcasm detection," *Neurocomputing*, vol. 308, pp. 1–7, 2018.

[55] P. K. Mandal and R. Mahto, "Deep cnn-lstm with word embeddings for news headline sarcasm detection," in *16th International Conference on Information Technology-New Generations (ITNG 2019)*. Springer, 2019, pp. 495–498.

[56] D. Jain, A. Kumar, and G. Garg, "Sarcasm detection in mash-up language using soft-attention based bi-directional lstm and feature-rich cnn," *Applied Soft Computing*, vol. 91, p. 106198, 2020.

[57] A. Kumar, V. T. Narapareddy, V. A. Srikanth, A. Malapati, and L. B. M. Neti, "Sarcasm detection using multi-head attention based bidirectional lstm," *IEEE Access*, vol. 8, pp. 6388–6397, 2020.

[58] S. He, F. Guo, and S. Qin, "Sarcasm detection using graph convolutional networks with bidirectional lstm," in *Proceedings of the 2020 3rd International Conference on Big Data Technologies*, 2020, pp. 97–101.

[59] A. Baruah, K. Das, F. Barbhuiya, and K. Dey, "Context-aware sarcasm detection using bert," in *Proceedings of the Second Workshop on Figurative Language Processing*, 2020, pp. 83–87.

[60] H. Srivastava, V. Varshney, S. Kumari, and S. Srivastava, "A novel hierarchical bert architecture for sarcasm detection," in *Proceedings of the Second Workshop on Figurative Language Processing*, 2020, pp. 93–97.

[61] A. Kumar, V. T. Narapareddy, P. Gupta, V. A. Srikanth, L. B. M. Neti, and A. Malapati, "Adversarial and auxiliary features-aware bert for sarcasm detection," in *8th ACM IKDD CODS and 26th COMAD*, 2021, pp. 163–170.

[62] R. A. Potamias, G. Siolas, and A.-G. Stafylopatis, "A transformer-based approach to irony and sarcasm detection," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17 309–17 320, 2020.

[63] J. Karoui, F. B. Zitoune, and V. Moriceau, "Soukhria: Towards an irony detection system for arabic in social media," *Procedia Computer Science*, vol. 117, pp. 161–168, 2017.

[64] B. Ghanem, J. Karoui, F. Benamara, V. Moriceau, and P. Rosso, "Idat at fire2019: Overview of the track on irony detection in arabic tweets," in *Proceedings of the 11th Forum for Information Retrieval Evaluation*, 2019, pp. 10–13.

[65] M. Khalifa and N. Hussein, "Ensemble learning for irony detection in arabic tweets." in *FIRE (Working Notes)*, 2019, pp. 433–438.

[66] H. A. Nayel, W. Medhat, and M. Rashad, "Benha@ idat: Improving irony detection in arabic tweets using ensemble approach." in *FIRE (Working Notes)*, 2019, pp. 401–408.

[67] T. Ranasinghe, H. Saadany, A. Plum, S. Mandhari, E. Mohamed, C. Orasan, and R. Mitkov, "Rgcl at idat: deep learning models for irony detection in arabic language," 2019.

[68] C. Zhang and M. Abdul-Mageed, "Multi-task bidirectional transformer representations for irony detection," in *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019*, ser. CEUR Workshop Proceedings, P. Mehta, P. Rosso, P. Majumder, and M. Mitra, Eds., vol. 2517. CEUR-WS.org, 2019, pp. 391–400. [Online]. Available: http://ceur-ws.org/Vol-2517/T4-2.pdf

[69] I. A. Farha and W. Magdy, "From arabic sentiment analysis to sarcasm detection: The arsarcasm dataset," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 2020, pp. 32–39.

[70] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," pp. 328–339, 2018. [Online]. Available: https://www.aclweb.org/anthology/P18-1031/

[71] K. Song, X. Tan, T. Qin, J. Lu, and T. Liu, "MASS: masked sequence to sequence pre-training for language generation," vol. 97, pp. 5926–5936, 2019. [Online]. Available: http://proceedings.mlr.press/v97/song19d.html

[72] K. Clark, M. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: pre-training text encoders as discriminators rather than generators," 2020. [Online]. Available: https://openreview.net/forum?id=r1xMH1BtvB

[73] S. Ruder, I. Vulić, and A. Søgaard, "A survey of cross-lingual word embedding models," *Journal of Artificial Intelligence Research*, vol. 65, pp. 569–631, 2019.

[74] O. Adams, A. Makarucha, G. Neubig, S. Bird, and T. Cohn, "Cross-lingual word embeddings for low-resource language modeling," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 937–947.

[75] D. Wang, N. Peng, and K. Duh, "A multi-task learning approach to adapting bilingual word embeddings for cross-lingual named entity recognition," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2017, pp. 383–388.

[76] J. Xie, Z. Yang, G. Neubig, N. A. Smith, and J. G. Carbonell, "Neural cross-lingual named entity recognition with minimal resources," pp. 369–379, 2018. [Online]. Available: https://doi.org/10.18653/v1/d18-1034

[77] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[78] A. V. M. Barone, "Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders," pp. 121–126, 2016. [Online]. Available: https://doi.org/10.18653/v1/W16-1614

[79] J.-K. Kim, Y.-B. Kim, R. Sarikaya, and E. Fosler-Lussier, "Cross-lingual transfer learning for pos tagging without cross-lingual resources," in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 2832–2838.

[80] X. Chen, Y. Sun, B. Athiwaratkun, C. Cardie, and K. Weinberger, "Adversarial deep averaging networks for cross-lingual sentiment classification," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 557–570, 2018.

[81] P. Keung, Y. Lu, and V. Bhardwaj, "Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and NER," 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 2019, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 1355–1360. [Online]. Available: https://doi.org/10.18653/v1/D19-1138

[82] A. Conneau and G. Lample, "Cross-lingual language model pretraining," pp. 7057–7067, 2019. [Online]. Available: http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining

[83] S. Wu and M. Dredze, "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 2019, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 833–844. [Online]. Available: https://doi.org/10.18653/v1/D19-1077

[84] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 92–100.

[85] M. Culp and G. Michailidis, "Graph-based semisupervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 174–179, 2007.

[86] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.

[87] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proceedings of the ninth international conference on Information and knowledge management*, 2000, pp. 86–93.

[88] Z. Kozareva, B. Bonev, and A. Montoyo, "Self-training and co-training applied to spanish named entity recognition," in *Mexican International conference on Artificial Intelligence*. Springer, 2005, pp. 770–779.

[89] C. Helwe and S. Elbassuoni, "Arabic named entity recognition via deep co-learning," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 197–215, 2019.

[90] W. Wang, Z. Huang, and M. Harper, "Semi-supervised learning for part-of-speech tagging of mandarin transcribed speech," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4. IEEE, 2007, pp. IV–137.

[91] K. Sagae, "Self-training without reranking for parser domain adaptation and its impact on semantic role labeling," in *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, 2010, pp. 37–44.

[92] R. Mihalcea, "Co-training and self-training for word sense disambiguation," in *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, 2004, pp. 33–40.

[93] S. Kiritchenko and S. Matwin, "Email classification with co-training," in *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*. Citeseer, 2001, p. 8.

[94] V. V. Asch and W. Daelemans, “Predicting the effectiveness of self-training: Application to sentiment classification,” *CoRR*, vol. abs/1601.03288, 2016. [Online]. Available: http://arxiv.org/abs/1601.03288

[95] M. S. Hajmohammadi, R. Ibrahim, A. Selamat, and H. Fujita, “Combination of active learning and self-training for cross-lingual sentiment classification with density analysis of unlabelled samples,” *Information sciences*, vol. 317, pp. 67–77, 2015.

[96] X. Pan, B. Zhang, J. May, J. Nothman, K. Knight, and H. Ji, “Cross-lingual name tagging and linking for 282 languages,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1946–1958.

[97] N. Garneau, M. Godbout, D. Beauchemin, A. Durand, and L. Lamontagne, “A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings: Making the method robustly reproducible as well,” *CoRR*, vol. abs/1912.01706, 2019. [Online]. Available: http://arxiv.org/abs/1912.01706

[98] X. L. Dong and G. de Melo, “A robust self-learning framework for cross-lingual text classification,” 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)*, 2019, pp. 6307–6311.

[99] X. Dong, Y. Zhu, Y. Zhang, Z. Fu, D. Xu, S. Yang, and G. de Melo, “Leveraging adversarial training in self-learning for cross-lingual text classification,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1541–1544.

[100] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[101] Y. Benajiba, P. Rosso, and J. M. Bened\'\iruiz, “Anersys: An arabic named entity recognition system based on maximum entropy,” in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2007, pp. 143–153.

[102] A. Mitchell, S. Strassel, M. Przybocki, J. Davis, G. Doddington, R. Grishman, A. Meyers, A. Brunstain, L. Ferro, and B. Sundheim, “Tides extraction (ace) 2003 multilingual training data,” *LDC2004T09, Philadelphia, Penn.: Linguistic Data Consortium*, 2003.

[103] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, “The penn arabic treebank: Building a large-scale annotated arabic corpus,” in *NEMLAR conference on Arabic language resources and tools*, vol. 27. Cairo, 2004, pp. 466–467.

[104] M. Maamouri, A. Bies, S. Kulick, M. Ciul, N. Habash, and R. Eskander, “Developing an egyptian arabic treebank: Impact of dialectal morphology on annotation and tool development.” in *LREC*, 2014, pp. 2348–2354.

[105] K. Dukes and N. Habash, “Morphological annotation of quranic arabic.” in *Lrec*, 2010.

[106] W. Antoun, F. Baly, and H. Hajj, “Arabert: Transformer-based model for arabic language understanding,” *arXiv preprint arXiv:2003.00104*, 2020.

[107] O. Obeid, N. Zalmout, S. Khalifa, D. Taji, M. Oudah, B. Alhafni, G. Inoue, F. Eryani, A. Erdmann, and N. Habash, “Camel tools: An open source python toolkit for arabic natural language processing,” in *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020, pp. 7022–7032.

[108] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, A. Moschitti, B. Pang, and W. Daelemans, Eds. ACL, 2014, pp. 1746–1751. [Online]. Available: https://doi.org/10.3115/v1/d14-1181

[109] O. Zaidan and C. Callison-Burch, “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*, 2011, pp. 37–41.

[110] M. Elaraby and M. Abdul-Mageed, “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)*, 2018, pp. 263–274.

APPENDIX

A. POS Tag Set

B. Error Analysis

The “regularizing” effect caused by self-training and discussed in section VIII can sometimes produce false negatives as shown in Table XI. We see a number of named entities that were misclassified by the self-trained model as unnamed ones. As an example, we take the last name الجنزوري which was classified both correctly and incorrectly in different contexts by the self-trained model. Context of correct classification is “هاش تاج لكمال الجنزوري”, while it is “ماسك على الناس كلها سي دي الا الجنزوري ماسك عليه فلوبي” for the incorrect classification. First, we note that الجنزوري is not a common name (zero occurrences in the MSA training set). Second, we observe that in the correct case, the word was preceded by the first name كمال which was correctly classified as PER, making it easier for the model to assign PER to the word afterwards as a surname.

TABLE XIII: The POS Tag Set in [45]

Tag	Description	Tag	Description
ADV	adverb	ADJ	adjective
CONJ	conjunction	DET	determiner
NOUN	noun	NSUFF	noun suffix
NUM	number	PART	particle
PUNC	punctuation	PRON	pronoun
PREP	preposition	V	verb
ABBREV	abbreviation	VSUFF	verb suffix
FOREIGN	non-Arabic	FUT_PART	future particle
PROG_PART	progressive particle	EMOT	Emoticon/Emoji
MENTION	twitter mention	HASH	Hashtag
URL	URL	–	–

TABLE XIV: **NER task.** Bigger Sample False Positives Mitigated by Self-Training. These were Correctly Predicted as the Unnamed Entity “O” by the Self-Trained Model

no.	Token	Eng.	MSA	Context/Explanation	FT Pred.
(1)	نبي	we want	نريد	نبي نعرف من… (*we want to know who*)	PER
(2)	ماكانوا	wasn’t	لم يكونوا	أغلب الي ماكانوا مصدقين (*most of those who wasn’t believing*)	LOC
(3)	لووول	LOL	ضحك	لووول… (interjection)	PER
(4)	عشان	for	لكي	تبي بطاريات عشان تلعب (*she wants batteries to play*)	LOC
(5)	دلوقتي	now	الآن	..اقنعوه ينزل دلوقتي (*convince him to move now*)	PER
(6)	ايش	what	ماذا	ايش رأيك (*what do you think?*)	PER
(7)	قادر	capable	قادر	وبقدرة قادر… (*magically*; idiomatic expression)	PER
(8)	المشين	shameful	المشين	المشين طنطاوي (*shameful Tantawy*; Playful for *General Tant.*)	PER
(9)	ايديكوا	your hands	أيديكم	ابوس ايديكوا اقنعوه… (*I entreat you to convince h*im)	PER
(10)	اسالك	I ask you	أسألك	ودي اسالك شنهي (*I ask you what*)	ORG
(11)	مين	who	مين	صوتك مع مين البدوي (*who do you vote for, Badawi*)	PER
(12)	فلوبي ديسك	floppy disk	قرص مرن	ماسك عليه فلوبي ديسك (*holds a floppy disk against him*)	PER
(13)	لحبايب	loved ones	الأحباء	تعال علم يف لحبايب (*come teach your loved ones*)	LOC
(14)	ماي	water	ماء	جبت لهم ماي (*brought them water*)	PER
(15)	ريتويت	retweet	إعادة تغريد	لو قرفان دوس ريتويت (*if depressed click retweet*)	PER

TABLE XV: **NER task.** Sample Errors that are Not Fixed by Self-Training (Shared with the Mere Fine-Tuned Model)

no.	Token(s)	Context/Explanation	Gold	FT	ST
(1)	بالمبارك	بالمبارك عاد احنا *(We are still in Mubarak)*	LOC	PER	O
(2)	محشش	محشش دخل المحاضرة *(a drunk entered the lecture)*	O	PER	PER
(3)	بأمارة	بأمارة ايه وفين *(what is the evidence/sign and where?)*	O	LOC	LOC
(4)	لمستفشي	لمستفشي قصر الدوباره *(to Qasr AlDobara Hospital)*	LOC	O	O
(5)	كنتاكي	عند كنتاكي *(by Kentucky [resturant])*	LOC	O	O
(6)	داون تاون	مشروع داون تاون بطنطا *(a down town Tanta project)*	LOC	O	O
(7)	يابطل	مبروك يابطل *(Congratulations, hero!)*	O	PER	PER
(8)	الاخوان	نختلف مع الاخوان *(we disagree with the Muslim brotherhood)*	ORG	O	O
(9)	قناة العربية	شفت قناة العربية *(watched Al Arabya Channel)*	ORG	O	O
(10)	المجلس العسكري	اللي عمله المجلس العسكري *(what the military council did)*	ORG	O	O

TABLE XVI: **NER task.** Sample False Negatives Produced by Self-Training

no.	Word	Gold	FT	ST
(1)	الاخوان	ORG	ORG	O
(2)	للبرادعي	PER	PER	O
(3)	مجدي الجلاد	PER	PER	O
(4)	فان ديزل	PER	PER	O
(5)	الجنزوري	PER	PER	O
(6)	زين يسون	PER	PER	O

TABLE XVII: **SRD task.** Sample Errors that were Fixed by Self-Training

no	Example	FT	ST
(1)	لن يفهما غير العباقره كنت عامل بيه غساله	Non-Sarcastic	sarcastic
(2)	هذا وضعكم بدون ميسي يا تافهين يا عاهات	Non-sarcastic	sarcastic
(3)	حاميها حراميها	Non-sarcastic	sarcastic
(4)	ايز لكل رجل قبيله عايزها متقسمه حاصح عبيط رسمي نظمي	Non-sarcastic	sarcastic
(5)	للعلم ده مهرجان درجه اولي ما لو بالفلوس ما كانش حد غلب	Sarcastic	Non-sarcastic
(6)	الاستاذه ميريام فارس ليها اغاني حلوه فشخ مش واخده حقها	Sarcastic	Non-sarcastic
(7)	يا تري بين هيلاري كلنتون ودونالد ترامب مين نختار	Sarcastic	Non-sarcastic
(8)	دعوه الست قصده هيلاري كلنتون مثلا	Sarcastic	Non-sarcastic

TABLE XVIII: **SRD task.** Sample Errors that Were not Fixed by Self-Training (Shared with the Mere Fine-Tuned Model)

no	Example	Prediction	Gold
(1)	عارفصوره وجدتها علي فيسبوك	sarcastic	non-sarcastic
(2)	فضيعه	sarcastic	non-sarcastic
(3)	بغنيلا وبدئلا وغير بحبك مابئلا	sarcastic	Non-sarcastic
(4)	انت الغالي اللي بقالي سنين بهواه	Sarcastic	Non-sarcastic
(5)	هه دنتم مسخره ياراقل هونو علي أنفسكم يامجن فرنسا	non-sarcastic	sarcastic
(6)	يعني حيكون زي اللورد دارث فيدرر هه	non-sarcastic	sarcastic
(7)	يا جماعه هذا بوكيمون ماحدا عرف يصطاده ويطعميه للجرذان	non-sarcastic	sarcastic
(8)	حضرتك مفيش فكه تاخد بالباقي ريتويتس	non-sarcastic	sarcastic