ALUE: Arabic Language Understanding Evaluation

Haitham Seelawi Ibraheem Tuffaha Mahmoud Gzawi Wael Farhan
Bashar Talafha Riham Badawi Zyad Sober Oday Al-Dweik
Abed Alhakim Freihat Hussein T. Al-Natsheh
Mawdoo3 Ltd, Amman, Jordan
{haitham.selawi,ibraheem.tuffaha,mahmoud.gzawi,wael.farhan,
bashar.talafha,riham.badawi,zyad.sober,oday.aldweik,
abedalhakim.freihat,h.natsheh}
@mawdoo3.com

Abstract

The emergence of Multi-task learning (MTL) models in recent years has helped push the state of the art in Natural Language Understanding (NLU). We strongly believe that many NLU problems in Arabic are especially poised to reap the benefits of such models. To this end, we propose the Arabic Language Understanding Evaluation Benchmark (ALUE), based on 8 carefully selected and previously published tasks. For five of these, we provide new privately held evaluation datasets to ensure the fairness and validity of our benchmark. We also provide a diagnostic dataset to help researchers probe the inner workings of their models. Our initial experiments show that MTL models outperform their singly trained counterparts on most tasks. But in order to entice participation from the wider community, we stick to publishing singly trained baselines only. Nonetheless, our analysis reveals that there is plenty of room for improvement in Arabic NLU. We hope that ALUE will play a part in helping our community realize some of these improvements. Interested researchers are invited to submit their results to our online, and publicly accessible leaderboard.

1 Introduction

Historically, research into the wide spectrum of problems in Natural Language Processing (NLP) and Understanding (NLU), has been highly compartmentalized, with each line of research attempting to tackle every single problem on its own, irrespective of the rest. However in recent years, the view has been shifting towards re-examining the whole field of NLP under a multitasking lens. This has manifested itself in the development of Multi-Task Learning (MTL) models, which are trained to optimize multiple losses, each for a different task, simultaneously.

This shift in paradigm was brought about by a confluence of various elements from the wide landscape of NLP research. For one, most core NLP tasks have been researched extensively with a significant slowdown in improvements under the banner of "singlism"! Another is the recent advances in contextual word embeddings, which was brought about in turn by the advent of a whole new class of neural network architectures, namely, the transformer, as described in the seminal paper of Vaswani et al. (2017).

We believe that the community of Arabic NLP is particularly poised to reap significant benefits from adopting this shift in paradigm. To this end, we seek to present a collection of 8 different Arabic-specific tasks, as part of a collective benchmark which we refer to as the Arabic Language Understanding Evaluation benchmark (ALUE). Similar to the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019b), the tasks we present in ALUE, are already available online and have featured in previous competitive workshops. To ensure fair use of the benchmark, we provide our privately held evaluation datasets for five of these tasks, in which we follow the respective original authors' annotation processes to a tee. In addition, we present an expert-constructed diagnostic dataset, to help researchers probe the inner workings of their models, and diagnose their shortcomings against a wide range of linguistic phenomena. We also present an automated and publicly published leaderboard on the internet[1], open to any researcher to directly submit the results of their models to. Last but not least, we include baseline results for several publicly available pre-trained Arabic models.

This paper aims to introduce our work to the Arabic NLP community. We hope it will provide an impetus that aids the research and development ef-

[1] www.alue.org

forts related to Arabic MTL approaches, and leads to wider collaboration as well as healthy competition. In Section 2, we discuss related work, both from the point of view of MTL models and datasets. In Section 3, we discuss the tasks comprising the ALUE benchmark, and their respective datasets. Section 4 focuses on the diagnostic dataset, and the way it was constructed, and the rationale behind it. An overview of our selected baselines can be found in Section 5. Results and discussions of our work can be found under Section 6, followed by our conclusions in section 7.

2 Related Work

The idea of MTL is relatively new in NLP. One of the earliest oft-cited works in this regard was by Collobert et al. (2011), in which they trained a multi-layered neural model on different sentence tagging tasks, with a common sentence encoder shared between them all, achieving solid performance on all of the tasks in the process.

The shift in paradigm towards MTL, requires a shift in terms of benchmarking as well. The General Language Understanding Evaluation (Wang et al., 2019b) is one of the most widely used benchmarks for comparing MTL models in the English language. It consists of nine tasks that are based on previously published benchmarks/datasets, mainly focusing on NLU. A subsequent iteration of the benchmark, named SuperGlue (Wang et al., 2019a), extends the scope of focus of the original benchmark to more challenging tasks, including question answering and co-reference resolution, while including a human performance baseline. For both benchmarks, the organizers provide an automated leaderboard that serves to compare and showcase the latest advancements in the field.

One of the earliest successful employment of MTL in Arabic NLP was by (Zalmout and Habash, 2019). By using MTL in an Adversarial learning setting, they reported state of the art results in cross-dialectal morphological tagging. This was mainly achieved by learning dialect invariant features between MSA (high resource dialect), and Egyptian Arabic (low resource dialect). This, they argue, helps in knowledge-transfer from the former to the latter, thereby sidestepping the issue of resource scarcity that plagues many Arabic variants. They also note that the gain from such knowledge transfer approach is more significant the smaller the datasets are.

Another paper by Baniata et al. (2018) employs the idea of MTL in the context of Neural Machine Translation. For translation from dialectical Arabic to Modern Standard Arabic (MSA), the authors use a sequence-to-sequence architecture, where each source language has its own encoder. However, for the target language, only a single shared decoder is used. Using this setup they report better results. Using this setup, they report better results and are able to efficiently use smaller datasets.

Freihat et al. (2018) manually curated an Arabic corpus of 2 million words, that was simultaneously annotated for POS, NER, and segmentation. Using an MTL model trained to perform the three aforementioned tasks, the authors were able to achieve state of the art results on said tasks, and show that such a model can greatly simplify and enhance downstream tasks, such as lemmatization.

Using this setup, they report better results and are able to efficiently use smaller datasets.

3 Tasks and Datasets

ALUE incorporates a total of 8 tasks covering a wide spectrum of Arabic dialects and NLP/NLU problems. Below, we provide a brief description of each task; the nature of the problem, its original workshop, and the evaluation metrics used. If the task is one of the five we have provided our own private dataset for, then we will also discuss the annotation process we followed to generate said private dataset.

It is worth mentioning that some of these tasks were subtasks in their respective workshops, such as the **Emotion Classification** and **Sentiment Intensity Regression** subtasks from **SemEval-2018 Task 1**, and the **Offensive** and **Hate Speech** subtasks from **OSACT4 Shared Task on Offensive Language Detection**. However, for the purposes of ALUE, these will be treated as independent tasks rather than subtasks. Nevertheless, in the discussion below, we are going to list them under the name of the original workshop task they featured in first.

3.1 IDAT@FIRE2019 Irony Detection Task (FID)

The shared task of Irony Detection in Arabic Tweets (Ghanem et al., 2019) is based on a dataset of around 5,000 tweets. Each tweet is labeled with a "1" when it is ironic, holds satire, parody, sarcasm, or if the intended meaning is the contrary of

the literal one. A label of "0" is given otherwise. This task will be evaluated using the F1 score.

3.2 MADAR Shared Task Subtask 1 (MDD)

This task is based on the MADAR Shared Task on Arabic Fine-Grained Dialect Identification (Bouamor et al., 2019). Each sentence is exclusively classified into one of 25 labels, corresponding to one city out of 25 predefined Arab cities. A 26th label is added for MSA. The data is sourced from the Basic Traveling Expression Corpus (Takezawa et al., 2007), with the same 2,000 sentences translated to the spoken dialect in each of the cities and MSA (Corpus-26). The metric of choice for this task is the F1-score.

3.3 NSURL-2019 Shared Task 8 (MQ2Q)

The Semantic Question Similarity in Arabic task (Seelawi et al., 2019) was presented in the First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019). In this task, a pair of questions is assumed to be semantically similar if they have the same exact answer and meaning, which is denoted with a label of "1". A label of "0" is given otherwise. For this task, we develop a new evaluation dataset. We start by clustering a dataset of Arabic article titles into clusters of similar semantic meaning. From these, we select headlines that have a question format. Then by pairing questions from similar clusters, we obtain 29,254 similar question pairs. Non-similar question pairs were generated by pairing questions from clusters that are close but not similar in semantic meaning. This is done to ensure that the resultant dataset is challenging. We then select 4,000 of these pairs, with an equal representation of "0"s and "1"s. A final round of human validation on those question pairs is conducted to ensure the quality of the resulting dataset. The evaluation of this task is performed using an F1-score.

3.4 OSACT4 Shared Task on Offensive Language Detection (OOLD & OHSD)

The offensive Language Detection shared task (Mubarak et al., 2020) is based on a dataset that contains a total of 10,000 tweets, with 2 subtasks. The first is the Offensive Task **(OOLD)** where a tweet is labeled offensive if it consists of inappropriate language or imply insults or attacks against other people, and not offensive otherwise. The second is the HateSpeech Task **(OHSD)** in which offensive tweets from the subtask above are also considered hate speech if they are attacking a certain group based on nationality, ethnicity, gender, political or sports affiliation, religious belief, or other related characteristics. Otherwise, they are labeled as not hate speech.

For both of these tasks, we develop our own evaluation dataset, using the Abusive Language Detection on Arabic Social Media corpus (Al Jazeera) (Mubarak et al., 2017), which contains 32,000 comments. We refine this corpus with 8 multi-labeled fine-grained classes, namely: toxic, insult, threat, identity hate, sexual, racial, blasphemy, and politically incorrect. Each label of these is denoted with either "1" if the class applies, or "0" otherwise.

We then annotate these comments using the following guidelines: (i) offensive if 3 or more classes of insult, toxic, threat, identity hate, and/or sexual are present, (ii) hate speech if the same previous conditions were satisfied, but with the additional requirement of the racial class having a label of "1" too (iii) comments with 0 values across all the classes are labeled as not offensive nor hate speech (iv) anything else that fails to satisfy any of the previous conditions is discarded.

We select 1,000 sentences from the resultant dataset, with special care to achieve a similar distribution of the original one. Finally, a round of human validation is conducted to ensure the quality of the overall evaluation dataset. Both of these tasks are evaluated using the F1-score.

3.5 SemEval-2018 Task 1 - Affect in Tweets (SVREG & SEC)

The Affect in Tweets dataset (Mohammad et al., 2018) was introduced in the 2018 SemEval workshop. The task consists of five subtasks. For our purposes, we will only include two of these. The first is the Emotion Classification task **(SEC)** in which a tweet is classified using one or more of eleven possible labels that best capture the emotions expressed by it. These labels are anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, and trust. The second is the Sentiment Intensity Regression task **(SVREG)** in which participants are expected to predict the "valence" of a given tweet, using a real-valued score between "0" and "1", with "0" representing the most negative sentiment possible, while "1" being the most positive sentiment possible.

For both of these tasks, we develop privately held evaluation datasets. For SEC, our annota-

tion process follows the convention described in SemEval-2018 (Mohammad et al., 2018). First, we select certain keywords, which collectively capture the 11 emotions. This is accomplished using various morphological forms and sub-phrases. A total of 18,000 tweets are then crawled using said keywords, each of which is subsequently labeled by four experienced annotators. A given emotion is then labeled as present with a "1", if two or more annotators agree. Otherwise, the emotion is labeled with a "0". 1,000 tweets are selected in a manner that resembles the distribution captured by the original dataset.

Our VREG evaluation dataset is based on the same 1,000 tweets selected for SEC. It is annotated using the Best-Worst Scaling (BWS) annotation methodology as described by Kiritchenko and Mohammad (2016). We combine these tweets and group them in tuples of four tweets each, according to the following set of rules: (i) No two 4-tuples should have the same four tweets. (ii) No duplicated tweets should exist within the same 4-tuple. (iii) All the tweets should have equal representation during the annotation process. As noted by the original authors, somewhere between 1,500 and 2,000 BWS questions should be sufficient to obtain reliable scores. Each of our tweets is present in 8 different 4-tuples, making a total of 2,000 BWS questions. Those tuples are then presented to the annotators who pick the tweet with the highest sentiment, as well as the lowest sentiment out of a given 4-tuple. Each 4-tuple is annotated by two different annotators.

Regression values are then obtained using the equation below:

$$V_i = 0.5 + 0.5((B_i - W_i)/T_i)$$

Where *Vi* is the regression value for the tweet *i*, *Bi* is the number of times that tweet *i* was voted as having the highest sentiment in a 4-tuple, and *Wi* representing the number of times it was voted as having the lowest sentiment. *Ti* is the number of times the tweet appears throughout all of the 4-tuples. It is worth noting that the *Vi* value obtained using the above equation will be between 0 and 1, inclusive.

For evaluation, we use the Jaccard similarity score and the Pearson correlation coefficient for SEC and VREG, respectively.

3.6 Cross-lingual Sentence Representations (XNLI)

This task is based on a dataset that was first presented by Conneau et al. (2018). It contains 7,500 textual entailment sentence pairs, each representing a hypothesis and a premise. These sentence pairs are labeled with either one of the following logical relationship labels: entailment, contradictory, or neutral. The data was originally labeled in the English language and then translated into 15 other languages including Arabic. It is split into 2,500 for development, and 5,000 for testing. The training data is meant to be the Multi-Genre Natural Language Inference (MultiNLI) corpus (Williams et al., 2018), which is available only in English. However, for the purposes of ALUE, we consider the 5,000 test dataset to be the training dataset, and the 2,500 development dataset to be the test dataset. The Evaluation metric for this task is accuracy.

3.7 How to Participate

Similar to GLUE, researchers interested in submitting their results to our leaderboard need to download and run their models, only using the training and, where available, the validation datasets, for each and every task. Once the results are obtained, and proper formatting and naming are adhered to, they can be submitted to our website[2]. The leaderboard shows each accepted team's submission performance per task. The final ranking is based on the unweighted average across all the tasks, not including the diagnostic dataset. Appendix B can be consulted for more details on the technology stack powering our leaderboard.

4 Diagnostics

Understanding a sentence depends on the capacity of the model to capture multiple underlying linguistic representations; changes in the logical, semantic, syntactic, and/or morphological features of a sentence can alter its meaning. These alterations can be described in terms of logical relationships (Bos and Markert, 2005). The Single-Task Training Benchmark by Wang et al. (2019b) shows varying weaknesses between different models when it comes to capturing the linguistic phenomena involved. For example, double negation is especially difficult for Contextualized Word Vectors such as Cove (McCann et al., 2017). Nonetheless, this seems to be somewhat ameliorated by

[2] www.alue.org

deep learning-based Contextualized Word representations such as ELMo (Peters et al., 2018). Yet, ELMo seems to struggle with Morphological Negation when compared to Cove.

The use of a manually annotated evaluation dataset that captures the diversity of such linguistic phenomena can help model designers better understand their models' generalization behavior, and work on improving them in the process. For this purpose, we create the ALUE diagnostic dataset; a high-quality, hand-labeled evaluation dataset, inspired by its GLUE counterpart (Wang et al., 2019b). It is composed of around 1,100 Arabic pairs of hypothesis and premise sentences. Each of these is labeled with their respective entailment relationship (entailment, contradiction, or neutral) and tagged with one or more category representing the host of linguistic phenomena they represent.

For the purposes of the evaluation of a given model's performance on the diagnostic dataset, we opted to follow GLUE's lead, and use the R3 metric (Gorodkin, 2004), for similar reasons (i.e. unbalanced class distribution in the diagnostic dataset labels). We also believe that the R3 readily helps with the investigation of systematic errors.

4.1 Annotation Process

We begin with the set of categories introduced in GLUE (Wang et al., 2019b)[3]. Each class describes a linguistic phenomenon that is important for NLU models to capture. Given that many of them are well described in the linguistics literature, they can be matched using syntactic/semantic patterns. Next, we look up for sentences expressing these patterns in a syntactically tagged Arabic corpus, with the help of WordNet relations. From these sentences, we construct sentence pairs, mostly by modifying the sub-phrases of a premise to produce a hypothesis reflecting the same linguistic phenomena in GLUE. For this purpose, we use three main sources, namely: Arabic Wikipedia, UN Multilingual Corpus V1.0 (Ziemski et al., 2016) and the corpus from the Arabic Linguistic Tool (ALP) (Freihat et al., 2018). This latter source is composed of syntactically annotated texts of various genres (e.g. news items, prose, literature, dialogues and TV podcasts) to ensure high coverage of spoken and written MSA. In addition, we manually translate the 330 artificially created examples from GLUE, which mostly express complex linguistic phenomena, and included them as part of our diagnostic dataset.

It must be noted that languages do not always describe similar linguistic phenomena in the same manner. This can lead to different entailments for equivalent translations. For instance, the agent of a verb in a passive construction is often mentioned in English, and when combined with Negation the entailment always yields a contradiction label: *John didn't break the vase* vs *The vase was broken by John*. The Arabic language tends to hide the agent of the verb in passive constructions which leads to a neutral label[4]: *John didn't break the vase* vs *The vase was broken*. With this in mind, during the annotation process, we took care of such peculiarities and, hence, ended up adding other categories. See appendix C.1 for more details.

5 Baselines

All of our baselines build on publicly released pre-trained word embeddings or models. These were carefully selected to represent a temporal cross-section of progress in Arabic NLP during the past few years; we start with fixed word-embeddings (i.e. AraVec, Soliman et al., 2017 and Fasttext, Joulin et al., 2017), and end with masked language models (i.e. Arabic-BERT, Safaya et al., 2020 and Multilingual-BERT, Turc et al., 2019), by way of sentence representations (i.e. Large Multilingual Universal Sentence Encoder, Yang et al., 2019) and the early starts of contextual embeddings (i.e. ELMoForManyLangs, Che et al., 2018; Fares et al., 2017).

For BERT based models, we use Huggingface (Wolf et al., 2019) for implementation, whereas for all other models, we use TensorFlow 2.0 (Abadi et al., 2015). However since the ELMoForManyLangs model is implemented in PyTorch, we decided to use its contextual embeddings without fine-tuning, as otherwise, we would have needed to implement it using PyTorch (Paszke et al., 2019). This would have effectively required from us to use three different deep-learning frameworks to implement all of our baselines. This, we reckon, would make the process of reproduction of our baselines much more complex for other researchers; as a matter of fact, as we trawled through the literature in search for Arabic pre-trained and publicly re-

[3]The full details of the annotation process for the diagnostics data are too long to be included here. Therefore we opted to provide them on our website.

[4]The logical inference is not centered on the agent of the verb, but the whole event.

leased models, we couldn't but notice the dearth of publicly released Arabic models especially those that capitalize on the latest advancements in NLP. We hope that our contributions through ALUE will help in this regard, by being a part of a conducive environment for the Arabic NLP community, to develop and publicly release state of the art MTL models.

We also note the lack of large corpora dedicated in full to Arabic and its varients. The majority of our selected baselines are pre-trained on dumps of Arabic Wikipedia and the Common Crawl (both of which are predominantly MSA in nature), while a few of them are trained on unreleased crawls with larger coverage of the various dialects of Arabic (i.e. ArabicBERT and AraVec). This makes our baselines a little harder to compare directly, but we hope that this highlights the peculiar challenges that Arabic NLP faces in this regard.

For all of our models, we embed our sentences into fixed-size vectors which in turn are fed into a feed-forward network that produces the final prediction. For the Universal Sentence Encoder (USE), this is readily achieved. However, for our AraVec, Fasttext and ELMO based models, in order to achieve the same step, we first consume the word embeddings using a BiLSTM. For the BERT based models, this is achieved somewhat indirectly, via the [CLS] token, which serves as a surrogate for sentence embeddings. More details on the exact architecture for each model per task, can be obtained via appendix A and the github repo where we release all our code[5]. The parameters used for each model might differ slightly, as we attempted to bring out the best performance possible from each to make our comparisons between them a little more fair.

6 Results and Discussion

Several key points in our work are worth analyzing, namely: (i) the baseline scores and the comparison of the different approaches, (ii) the analysis of the performance of our baseline models on the diagnostic dataset, and (iii) the comparative analysis between private and public evaluation datasets. This section goes through each of these key points in the same order as presented above.

6.1 Benchmark Results

Each one of our models was trained in a single-task fashion. To ensure reproducibility we used a random seed. While we strongly advocate for MTL models, we strategically eschewed training any such model for our baselines. This is mainly because our initial experiments show that it significantly boosts the performance on our benchmark, and as such, with the end goal of encouraging researchers to submit their results to ALUE, we decided to keep our baselines competitive enough to entice participation from the wide community, but simple enough to surpass. At the end of the day, we believe that baselines, as their name suggests, belong at the base of a leaderboard. The results obtained for each of our baselines are outlined in Table 1.

We note that, expectedly, BERT based models outperform all others by a large margin, with ArabicBERT's performance being the best across all tasks. This can be attributed in part, to the fact that it is trained on a large corpus composed of MSA as well as dialectical variants of Arabic. The importance of this factor is evident in tasks that are heavily skewed towards dialectical Arabic (i.e. OOLD and OHSD), where a simple model using twitter-based word representations such as AraVec (i.e. which has a heavy representation of dialectical words) outclassed Multi-lingual BERT, which was only trained on MSA. This strongly highlights the importance of pre-training data that covers the wide spectrum of Arabic variants in this time and age.

Interestingly, the difference in performance between Multi-BERT cased and uncased is somewhat negligible across all tasks except for those that require strong syntactical performance (i.e. MQ2Q and XNLI). This indicates that orthographic normalization in Arabic might impede a model's ability to achieve good syntactical modeling. We also suspect that the cased Multi-BERT model is indirectly benefiting from preserving the case for the other languages in the shared WordPiece vocabulary space it learns.

In a similar vein, the USE model performs very competitively on XNLI and MQ2Q. This is due to the fact the Natural Language Inference (NLI) is part of the pre-training method for said model.

[5]We release the code for our baselines publicly for reproducibility at the following GitHub repo: https://github.com/hseelawi/alue_baselines

Model	FID	MDD	MQ2Q	OOLD	OHSD	SVREG	SEC	XNLI	Avg	DIAG
ArabicBERT	**82.18**	59.66	**85.69**	**89.47**	**78.72**	**55.12**	**25.13**	60.96	**67.12**	**19.60**
ML-BERT Cased	81.61	**61.26**	83.24	80.33	70.54	33.85	14.02	**63.09**	60.99	19.00
ML-BERT Uncased	81.01	57.98	75.79	79.85	70.64	32.01	13.81	57.91	58.63	15.10
USE	76.90	23.40	76.50	76.30	68.20	36.50	14.80	57.10	53.71	13.90
ML-ELMo	77.00	52.10	70.50	71.60	62.88	24.90	14.40	50.00	52.92	09.60
AraVec	76.70	48.40	62.60	85.60	73.80	32.20	18.05	47.70	55.63	10.00
FastText	77.10	50.80	66.80	79.70	60.40	37.00	15.30	52.70	54.98	03.50

Table 1: Evaluation scores for our baseline models on the various ALUE tasks, with Pearson Correlation and Jaccard Index scores for SVREG and SEC tasks respectively, Matthews Correlation Coefficient for the Diagnostic dataset (DIAG), Accuracy for XNLI, and F1-score for the rest. Note that DIAG is not included in the average as it is not designed for direct model comparison.

Nonetheless, the model's very poor performance on dialect detection (i.e. MDD) reveals the inherent issues that sentence embedding models face in tasks where lexical information is important, as it tends to be discarded, in the process of embedding the full sentence into a single fixed-size vector. For such tasks, we can make the observation that models which use subword embeddings or some form of word morphological based tokenization tend to perform well, for the exact opposite reason, even for those that have been trained on MSA only (i.e. Multi-BERT).

The benefits of using subwords can be generalized to dialect-heavy tasks too, as can be seen in the case of the Multi-BERT and Fasttext models. This might be explained by the fact that subwords make better use of cognates across the different forms of Arabic. Additionally, there is no denying that the use of subwords mitigates the effects of non-standard orthography, amongst and across the various dialects. Nonetheless, their contributions might become less pronounced when dialectical Arabic is strongly present in the pre-training corpus. This is very evident in the case of the AraVec model, which, as alluded to above, outperforms all the models on three out of five of those tasks (i.e. SEC, OOLD, and OHSD), except for ArabicBERT, which even then, is heavily pre-trained on dialectical data.

6.2 Diagnostic Results Analysis

We report the performance of our models on the diagnostic dataset in Table 2.

For **coarse-grained** categories, ArabicBERT outperforms, on average, all other models with a considerable difference, although the overall performance across all models is low. The highest scores fall under the Predicate-Argument Structures category, with an average of 15, whereas the scores on Lexical Semantics seem to be the lowest with an 8.1 average. Interestingly, both versions of MultiBERT outperform ArabicBERT in world-knowledge, which depends on extra-linguistic information. This, perhaps, has something to do with the fact that they were trained on Wikipedia dumps of many languages. USE results seem to be high in World Knowledge compared to other coarse-grained categories as well.

For **fine-grained** categories, here again we can see the strong correlation between USE and Multi-BERT performance on Named Entities, which perhaps is underpinned by the same factors for their solid performance on World Knowledge. All of our models seem to exhibit very poor performance on Double Negation and Conditionals. This is probably due to the very rich diversity of tool-words used in Arabic to describe such phenomena, which makes it difficult for the models to make adequate generalizations. Of note is the fact that FastText seems to work especially well on Restrictivity, where all other models seem to struggle.

6.3 Private vs Public Set Analysis

Here we preform a comparative analysis between our private evaluation datasets and the original ones, to provide a better understanding of the involved baselines and datasets. First, we compute a correlation score between both, private and public results as displayed in Table 3. As expected, all of these scores are positively correlated but some datasets are more so than others. For instance, the strong correlation between the public and private evaluation datasets for MQ2Q can be explained by the fact that the only difference is in the diversity of the topics covered. On the other hand, while both the public and private evaluation datasets for SVREG and SEC are from the same source (i.e. tweets), they were collected at different points in

	Coarse-Grained					Fine-Grained					
Model	All	LS	PAS	L	K	Quant	2N	Cond	Rest	Nom	NE
ArabicBert	**19.6**	15	**28**	13.1	15	**48**	-17	-06	16	15	10
ML-BERT cased	19	**17**	20	12	**19**	45	-20	00	00	**34**	**23**
ML-BERT uncased	15.1	07	15	**13.5**	**19**	36	-12	-14	-10	10	06
USE	13.9	07	14	10	15	34	-30	-14	00	20	22
Elmo	9.6	14	10	09	10	08	-26	**10**	-12	14	00
AraVec	10	00	10	09	06	07	-37	-01	00	10	10
FastText	3.5	-02	08	02	-03	14	-37	-24	**30**	09	-06
AVG		8.1	**15**	9.8	11.6						

Table 2: The R3 results of our different baseline models on the diagnostic dataset. Scores are scaled by 100. The "All" score is the average of coarse-grained categories. Abbreviations are: Lexical Semantics (LS), Predicate-Argument Structure (PAS), Logic (L), World knowledge (k), Quantifiers (Quant), Double Negation (2N), Conditional (Cond), Restrictivity (Restr), Nominalization (Nom) and Named entities (NE). Here we only report results on the fine-grained classes that we find to be the most interesting.

Model	MQ2Q		SVREG		SEC		OOLD		OHSD	
	Private	Public	Private	Public	Private	Public	Private	Public	Private	Public
ArabicBERT	0.8569	0.9523	0.5512	0.8376	0.2513	0.5422	0.8947	0.9583	0.7872	0.9820
MultiBERT-cased	0.8324	0.9573	0.3385	0.7261	0.1402	0.4802	0.8033	0.9439	0.7054	0.9715
MultiBERT-uncased	0.7579	0.9389	0.3201	0.7274	0.1381	0.4739	0.7985	0.9453	0.7064	0.9772
USE	0.7650	0.8451	0.3650	0.7224	0.1480	0.2705	0.7630	0.7264	0.6820	0.5501
ELMo	0.7050	0.9018	0.2490	0.5550	0.1440	0.2850	0.7160	0.6119	0.6288	0.4930
FastText	0.6680	0.8869	0.3700	0.6480	0.1530	0.3221	0.7970	0.7059	0.6040	0.4770
AraVec	0.6260	0.8096	0.3220	0.6402	0.1805	0.3552	0.8560	0.7439	0.7380	0.6145
Correlation	**0.7757**		**0.8492**		**0.5012**		**0.6229**		**0.7151**	

Table 3: The evaluation scores for our baseline models on both, the private and public evaluation datasets. The last row shows the Pearson correlation coefficient for both sets across all the models for the corresponding task.

time. For SVREG, the scores on both evaluation datasets are highly correlated, yet there is a drastic gap between the results. This can be explained by the comparative nature of the BWS annotation process. Each tweet is labeled with a floating number that evaluates how its sentiment compares to the average or norm of the entire dataset. From the results, we can deduce that the overall sentiment of the public evaluation dataset is more positive than ours. The results on the SEC datasets seem to be the least correlated. This is because both versions of MultiBERT (cased and otherwise) achieve high scores on the original evaluation dataset, yet, curiously, they score the lowest on our evaluation dataset. Interestingly, in the case of OOLD and OHSD, the public and private evaluation datasets for both cases, have a strong positive correlation despite the fact that they were collected from two different sources; the original one is from Twitter while our dataset is from Aljazeera comments.

7 Conclusion

In this paper, we introduce the ALUE benchmark, with the purpose of providing a platform for researchers interested in pushing the state of the art in Arabic NLU. It consists of 8 previously published tasks, with 5 privately curated evaluation datasets to ensure the validity of the leaderboard. We evaluated the correctness of these 5 evaluation sets, finding a positive correlation between the ones we developed and the original ones. In addition, we built a novel diagnostic dataset that helps analyze the results of models against a comprehensive range of linguistic phenomena. Our initial experiments show that MTL approaches outperform their single-model-per-task counterparts, but to keep our leaderboard lucrative for participation, we decided to only use single-task models as our baselines. Our BERT baselines seem to outperform all other models, and especially so, when the pertraining data is not confined to MSA-dominant corpuses, but contain dialectical varieties of Arabic as well. For the diagnostic dataset, we found that our baselines struggle to capture many of the linguistic phe-

nomena represented by the dataset itself, which suggests that there is plenty of room for improvement in the state of art for Arabic NLU. We hope that ALUE will be an integral part of the efforts to push said state of the art in the coming few years.

8 Acknowledgement

We would like to express our gratitude to our partners in this project, the MIND Lab at the American University of Beirut (AUB), and the CAMeL Lab at New York University Abu Dhabi. Their outstanding support and ongoing guidance was invaluable to the work presented in this paper. We would also like to thank the team members of Mawdoo3 AI data Annotation and Linguistics departments for their contributions in annotating the privately held evaluation datasets; namely: Manar Salous, Yasmin Al Momani, Othman Abu Saa', Mohammad Saleh and Mariam Arnaout. In addition, we thank Abdallah Abu Sham and Wael Gaith from Mawdoo3 for help in developing and deploying our leaderboard website. Finally, special thanks goes to Mawdoo3 Ltd. for supporting our research efforts and for making the evaluation datasets publicly available for further research on Arabic NLP/NLU.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Laith H Baniata, Seyoung Park, and Seong-Bae Park. 2018. A neural machine translation model for arabic dialects that utilizes multitask learning (mtl). *Computational intelligence and neuroscience*, 2018.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 628–635, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The madar shared task on arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 271–276, Gothenburg, Sweden. Association for Computational Linguistics.

Abed Alhakim Freihat, Gabor Bella, Hamdy Mubarak, and Fausto Giunchiglia. 2018. A single-model approach for arabic segmentation, pos tagging, and named entity recognition. In *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pages 1–8. IEEE.

Bilal Ghanem, Jihen Karoui, Farah Benamara, Véronique Moriceau, and Paolo Rosso. 2019. Idat at fire2019: Overview of the track on irony detection in arabic tweets. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 10–13.

Jan Gorodkin. 2004. Comparing two k-category assignments by a k-category correlation coefficient. *Computational biology and chemistry*, 28(5-6):367–374.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Svetlana Kiritchenko and Saif M Mohammad. 2016. Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of NAACL-HLT*, pages 811–817.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*.

Haitham Seelawi, Ahmad Mustafa, Wael Farhan, and Hussein T. Al-Natsheh. 2019. NSURL-2019 task 8: Semantic question similarity in arabic. In *Proceedings of the First Workshop on NLP Solutions for Under Resourced Languages*, NSURL '19, Trento, Italy.

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

Toshiyuki Takezawa, Genichiro Kikui, Masahide Mizushima, and Eiichiro Sumita. 2007. Multilingual spoken language corpus development for communication research. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 12, Number 3, September 2007: Special Issue on Invited Papers from ISCSLP 2006*, pages 303–324.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019b. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2019. Multilingual universal sentence encoder for semantic retrieval. *arXiv*, pages arXiv–1907.

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

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

A Additional Benchmark Details

Each of our baselines is adapted to each ALUE task as needed. As such, extra trainable layers are added to each model to consume its outputs, which in turn is trained to make task specific predictions.

Please consult the subsections below in addition to the publicly available github repo[6] for exact details on the fine-tuning process for each model.

A.1 BERT

All of our BERT models weights are fine-tuned per task, using the [CLS] token, along with a dropout layer followed by a linear output layer that is suitable for each task.

A.2 USE

The produced sentence representations are fed into a feed-forward network that is designed and trained independently for each task.

A.3 ELMo

Given that the size of embeddings produced by this model is 1024, we use a BiLSTM for each task with a hidden size of 1024 in both directions. The hidden states of the last token are then fed to the appropriate feed-forward network for a given task. No fine tuning of the ELMo model itself is done.

A.4 AraVec

We use the skip-gram model with 300 vector size of unigram type that is trained on 66.9M Arabic tweets. This is mainly because skip-gram embeddings provide better representations for less frequent words compared to continuous bag of words. The embedding of each token in a given sentence is fed into a BiLSTM and a feed-forward network, which are trained on each task separately, similar to ELMo.

A.5 FastText

These are pre-trained word embeddings that use subword information to avoid out-of-vocab issues. The Arabic model is mainly trained on Arabic Wikipedia dumps and Arabic content from the Common Crawl. They come in a fixed size of 300. We use these embeddings to train a BiLSTM and a feed-forward network for each task similar to ELMo and AraVec.

[6] https://github.com/hseelawi/alue_baselines

B Benchmark Website Details

Our online leaderboard is powered by CodaLab[7], but is self-hosted on our own servers. Many high caliber Arabic NLP/NLU workshops are typically hosted on Codalab, and as such many researchers in the field are already familiar with its interface. This was a deciding factor in selecting it to power our leaderboard.

C Additional Diagnostics Details

The Diagnostic dataset is composed of sentence pairs. Each, has a premise and hypothesis sentences, and tagged with an entailment label : entailment, neutral or contradiction, in addition to the linguistic phenomena involved in the relationship. Linguistic phenomena tags are divided into 4 coarse-grained categories: Lexical semantics (LS), predicate-argument structure (PAS), logic (L) and world-knowledge (K).

When applicable, coarse-grained categories are tagged with fine-grained categories. Each sentence-pair is followed by its inverse form to establish a confusion of the entailment.

C.1 Diagnostics Categories

The original GLUE diagnostic dataset uses 4 coarse-grained categories further divided into 33 fine-grained categories. For an explanation of these phenomena, please consult the related sections in the original GLUE paper by Wang et al. (2019b). In addition to these, we added the following fine-grained categories, some of which are specific to Arabic:

Lexical Implication: a verb Y is entailed by X if by doing X you must be doing Y:

- *Saeed is snoring* **entails but is not entailed by** *Saeed is a sleep*.

Topicalization: a syntactic movement where an argument is moved to the beginning of the sentence to put emphasis:

- *The city of Amman is located in Jordan* **entails and is entailed by** *The city of Amman, it is located in Jordan*.

Reciprocity: an alternation resulting in the realization of the object as a part of a conjoined subject.

[7] https://codalab.org/

- *Reptiles fight each other by biting and scratching* **entails and is entailed by** *Reptiles fight (each other) by biting and scratching*.

Causative/Inchoative: an alternation that expresses a change of state leading to the omission of the agent. This case is marked by the addition of an inchoative *n* morpheme in Arabic.

- *Saeed broke the vase* **entails but is not entailed by** *The vase broke*.

Adjectivation: use of relational adjectives instead of the entity they describe i.e. China/Chinese.

- *These technologies will help strengthen the Chinese social security system* **entails and is entailed by** *These technologies will help strengthen the social security system of China.*