hULMonA (حلمنا): The Universal Language Model in Arabic

Obeida ElJundi [(1)] **Wissam Antoun** [(1)] **Nour El Droubi** [(1)]
Hazem Hajj [(1)] **Wassim El-Hajj** [(2)] **Khaled Shaban** [(3)]
(1) American University of Beirut, Electrical and Computer Engineering Department
(2) American University of Beirut, Computer Science Department
Beirut, Lebanon
(3) Qatar University, Computer Science and Engineering Department, Doha, Qatar
{oae15;wfa07;ngd02;hh63;we07}@aub.edu.lb;khaled.shaban@qu.edu.qa

Abstract

Arabic is a complex language with limited resources which makes it challenging to produce accurate text classification tasks such as sentiment analysis. The utilization of transfer learning (TL) has recently shown promising results for advancing accuracy of text classification in English. TL models are pre-trained on large corpora, and then fine-tuned on task-specific datasets. In particular, universal language models (ULMs), such as recently developed BERT, have achieved state-of-the-art results in various NLP tasks in English. In this paper, we hypothesize that similar success can be achieved for Arabic. The work aims at supporting the hypothesis by developing the first Universal Language Model in Arabic (hULMonA - حلمنا meaning our dream), demonstrating its use for Arabic classifications tasks, and demonstrating how a pre-trained multilingual BERT can also be used for Arabic. We then conduct a benchmark study to evaluate both ULM successes with Arabic sentiment analysis. Experiment results show that the developed hULMonA and multi-lingual ULM are able to generalize well to multiple Arabic data sets and achieve new state of the art results in Arabic Sentiment Analysis for some of the tested sets.

1 Introduction

Transfer learning (TL) with universal language models (ULMs) have recently shown to achieve state of the art accuracy for several natural language processing (NLP) tasks (Devlin et al., 2018; Howard and Ruder, 2018; Radford et al., 2018). ULMs are trained unsupervised to provide an intrinsic representation of the language using large corpora that do not require annotations. These models can then be fine-tuned in a supervised mode with much smaller annotated training data to achieve a particular NLP task. The established success in English with limited data sets makes ULMs an attractive option for Arabic consideration since Arabic has limited amount of annotated resources. Early language models focused on vector embeddings for words and provided word-level vector representations (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017), sentence embeddings (Cer et al., 2018), and paragraph embeddings (Le and Mikolov, 2014; Kiros et al., 2015). These early models were able to achieve success comparable to models that were trained only on specific tasks. More recently, the language model representation was extended to cover a broader representation for text. BERT (Devlin et al., 2018), ULMFiT (Howard and Ruder, 2018), and OpenAI GPT (Radford et al., 2018) are examples of such new pre-trained language models and which were able to achieve state of the art results in many NLP tasks.

However, in the field of Arabic NLP, such ULMs have not been explored yet. The use of transfer learning in Arabic has been mainly focused on word embedding models (Dahou et al., 2016; Soliman et al., 2017). Among the recently, developed ULM models, BERT (Devlin et al., 2018) built a multilingual language version using 104 languages including Arabic but this model has only been tested on Arabic "sentence contradiction" task. One advantage of the multi-lingual BERT is that it can be used for many languages. However, one important limitation is that it was constrained to parallel multi-lingual corpora and did not take advantage of much larger corpora set

Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 68–77
Florence, Italy, August 1, 2019.

available for Arabic, making its intrinsic representation limited for Arabic. As a result, there is an opportunity to further improve the potential for ULM success by developing an Arabic specific ULM.

In this paper, we aim at advancing performance and generalization capabilities of Arabic NLP tasks by developing new ULMs for Arabic. We develop the first Arabic specific ULM model, called hULMonA. Furthermore, we show how pre-trained multi-lingual BERT can be fine tuned and applied for Arabic classification tasks. We also conduct a benchmark study to evaluate the success potentials for the ULMs with Arabic sentiment analysis. We consider several datasets in the evaluation and show the superiority of the methods' generalization handling both MSA and dialects. The results show the superiority of the models compared to state of the art. We further show that even though the multi-lingual BERT was not trained for dialects, it still achieves state of the art for some of the dialect data sets.

In summary, our contributions are: 1. The development of hULMonA, the first Arabic specific ULM, 2. the fine tuning of multi-lingual BERT ULM for Araic sentiment analysis, and 3. the collection of a benchmark dataset for ULM evaluation with sentiment analysis

The rest of the paper is organized as follows: Section 2 provides a survey of previous work in language development for English and Arabic. Section 3 presents a description of the methodologies to develop the targeted ULMs and the description of the benchmark data set. Section 4 presents the experiment results. Finally, section 5 concludes the paper.

2 Related Work

This section describes the use of language models for NLP tasks. Historically, language models can be categorized into representations at word level and representation of larger units of text such as phrases, sentences, or documents. We will call the second sentence level representation.

2.1 Language Models for English

2.1.1 Word-level Models for English

The word-level language model is based on the use of pre-trained embedding vectors as additional features to the model. The most common embedding vectors used are word embeddings. With word embeddings, each word is linked to a vector representation in a way that captures semantic relationships (Mikolov et al., 2013). The most common word embeddings used in deep learning are word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and FastText (Bojanowski et al., 2017). Other embedding vectors have been also proposed for longer texts such as vectors at the sentence level (Cer et al., 2018) and at the paragraph level (Le and Mikolov, 2014; Kiros et al., 2015). The use of these embedding vectors has shown significant improvement compared to training models from scratch (Turian et al., 2010). One of the recent feature-based approaches is ELMo (Peters et al., 2018) which is based on the use of bidirectional LSTM models. Unlike the traditional word embedding representations mentioned previously, ELMo word embeddings are functions of the whole sentence which enables capturing context-related meanings. The use of these word embeddings was shown to improve the state-of-the-are results in six NLP tasks such as sentiment analysis and question answering.

2.1.2 Sentence-level Language Models for English

In contrast to word-level representation, sentence level representation develops language model which can then be fine-tuned for a supervised downstream task (Devlin et al., 2018). The advantage of these pre-trained language models is that very few parameters have to be learned from scratch. The use of the pre-trained language models has shown to result in a better performance than the use of the feature-based approach (Howard and Ruder, 2018). Several pre-trained language models have been proposed recently that were able to achieve state-of-the-art results in many NLP tasks. One of these language models is OpenAI GPT (Radford et al., 2018) which uses the Transformer network (Vaswani et al., 2017) that enables them to capture a long range of linguistic information. This is in contrast with ELMo (Peters et al., 2018) which uses the short-range LSTM models. OpenAI GPT was able to achieve state-of-the-art results in several sentence-level NLP tasks from the GLUE benchmark (Wang et al., 2018) such as question-answering and textual entailment.

Another proposed pre-trained language model is ULMFiT (Howard and Ruder, 2018) which is based on a three-layer LSTM architecture, called

AWD-LSTM (Merity et al., 2017). This language model was able to achieve state-of-the-art results in six text classification tasks with just a few task-specific fine-tuning.

In addition to these language models, one of the most recent and innovative pre-trained language models is BERT (Devlin et al., 2018). BERT is based on the use of the recently introduced Transformer attention networks (Vaswani et al., 2017). BERT uses the bidirectional part of the Transformer architecture which is the encoder which enabled the language model to capture both left and right context. This innovation enabled BERT to achieve remarkable improvements compared to previous models and to achieve state-of-the-art results in eleven NLP tasks with the addition of just one output layer.

2.2 Language Models for Arabic

Some word embedding models were built using multiple languages such as Polyglot (Al-Rfou et al., 2013) which was built using 117 languages including the Arabic language. This model was then tested in multilingual NLP tasks. In addition to that, building on the word embedding methods developed for English, several approaches were done to build word embeddings for MSA and dialectal Arabic. The first approach is AraVec (Soliman et al., 2017) which was built using a large Arabic corpus collected from Twitter, Internet, and Wikipedia articles. Another model was proposed by Dahou et al. (Dahou et al., 2016) in which Arabic word embeddings were built using a 3.4 billion words corpus.

For sentence-level representations, there has been a development of multi-lingual models using parallel corpora. As an example, multilingual BERT (Devlin et al., 2018) was built using 104 languages including Arabic. However, there has not been any Arabic only language models. Moreover, Bert was experimented on several NLP tasks, but sentiment analysis was not one of them.

2.3 Arabic Sentiment Analysis

In (Abdul-Mageed and Diab, 2014), a large-scale, multi-genre, multi-dialect lexicon named SANA was built for the sentiment analysis of Arabic dialects. This lexicon covers the MSA, the Egyptian dialect, and the Levantine Arabic. SANA has several features which are the part of speech (POS) tagger and diacritics, number, gender, and rationality. Despite this lexicons coverage, it was still not complete, and many terms were not present. In (Abdul-Mageed and Diab, 2012), Abdul Majeed et al. worked on expanding a polarity lexicon which was built on MSA using existing English polarity lexica. The problems faced with this lexicon was that many terms that existed in social media were not found in the lexicon. Hence, the coverage of dialectical Arabic was poorly achieved using this lexicon.

In the work of Duwairi (Duwairi, 2015), sentiment analysis was done on tweets where dialectical Arabic words were present. This work used both the supervised and unsupervised approaches to build the model. To deal with dialectical words, a dialect lexicon was created in which two annotators mapped each dialectical word to its corresponding Modern Standard Arabic word. Two classifiers were used to train the model which are the Naive Bayes (NB) and the Support Vector Machines (SVM). The model was then tested using a dataset of 22,550 tweets written in Arabic and that contain dialectical Arabic words. Testing was done on the dataset when the dialect lexicon was used and when it was not used. Results showed some improvement on the Macro-Recall when the dialect lexicon was used on the NB classifier. However, the improvement was negligible on the SVM classifier and the precision and the recall were even negatively affected when classifying the negative and the Neutral classes using both classifiers.

Recently, deep learning models were the main focus of Arabic NLP researchers (Badaro et al., 2019). The first deep learning attempt was conducted by (Al Sallab et al., 2015) who explored four deep learning models, namely Deep Neural Network (DNN), Deep Believe Network (DBN), Deep Auto Encoder (DAE), and RAE. The sentiment lexicon ArSenL (Badaro et al., 2014) was utilized to represent the text vector space. In a follow up work, (Al-Sallab et al., 2017) proposed a recursive deep learning model for opinion mining in Arabic (AROMA) to address some limitations of using RAE for Arabic. To address the morphological richness and orthographic ambiguity of the Arabic language, (Baly et al., 2017) proposed the first Arabic Sentiment Treebank (ARSENTB) and trained RNTN to outperform AROMA. AraVec word embeddings (Soliman et al., 2017) were utilized by (Badaro et al., 2018) to win SemEval 2018 (Mohammad et al., 2018). (Dahou et al.,

2016) and (Dahou et al., 2019) investigated a CNN architecture similar to (Kim, 2014) trained on locally trained word embeddings to achieve significant results.

Despite all this emerging progress in Arabic sentiment analysis, transfer learning was utilized by only using a single layer of weights - usually the first layer - known as embeddings. However, typical neural network architecture consists of several layers, and utilizing transfer learning for only the first layer was clearly just scratching the surface of what is possible.

3 Methodology

In this section, we describe how we constructed hULMonA and how we then tuned both hULMonA and the multi-lingual BERT ULM for Arabic classification tasks.

The high-level architecture for using a ULM model is shown in Figure 1. The complete model consists of the combination of a pre-trained ULM model and additional task-specific layers for the desired tasks. Once a ULM model is developed, the learning process becomes limited to learning the parameters of the additional layers. This transfer learning process is referred to as fine-tuning with ULM and this is the main benefit of using ULMs.

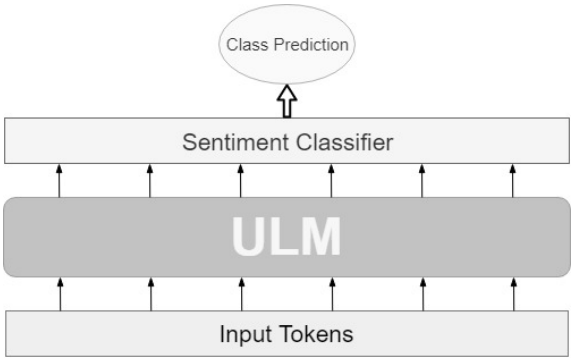


Figure 1: High Level Architecture for ULM Tranfer Learning

Below, we describe the data pre-processing step required for Arabic and the fine tuning process for the additional layers.

3.1 Arabic Specific ULM: hULMonA

Transfer Learning implies that training a model which already has some language knowledge performs better, converges faster, and requires less data for new task when comparing to training from raw text. Language modeling is considered the ideal task to obtain general understanding of a particular language due to its ability of capturing many aspects of language relevant for downstream tasks, such as long-term dependencies (Linzen et al., 2016), hierarchical relations (Gulordava et al., 2018), and sentiment orientation (Radford et al., 2017).

Inspired by the Universal Language Model Fine-tuning (ULMFiT) (Howard and Ruder, 2018), we propose, develop, and make available for public[1], the first ULM in Arabic (hULMonA - حلمنا) that is trained on large general-domain Arabic corpus and can be fine-tuned on any target task to achieve significant results. hULMonA, illustrated in Figure 2, consists of three main stages: 1. pretraining the state-of-the-art language model AWD-LSTM (Merity et al., 2017) on a huge Wikipedia corpus (section 3.1.1), 2. fine-tuning the pretrained language model on a target dataset (section 3.1.2), 3. and adding a classification layer on top of the fine-tuned language model for the purpose of text classification (section 3.1.3).

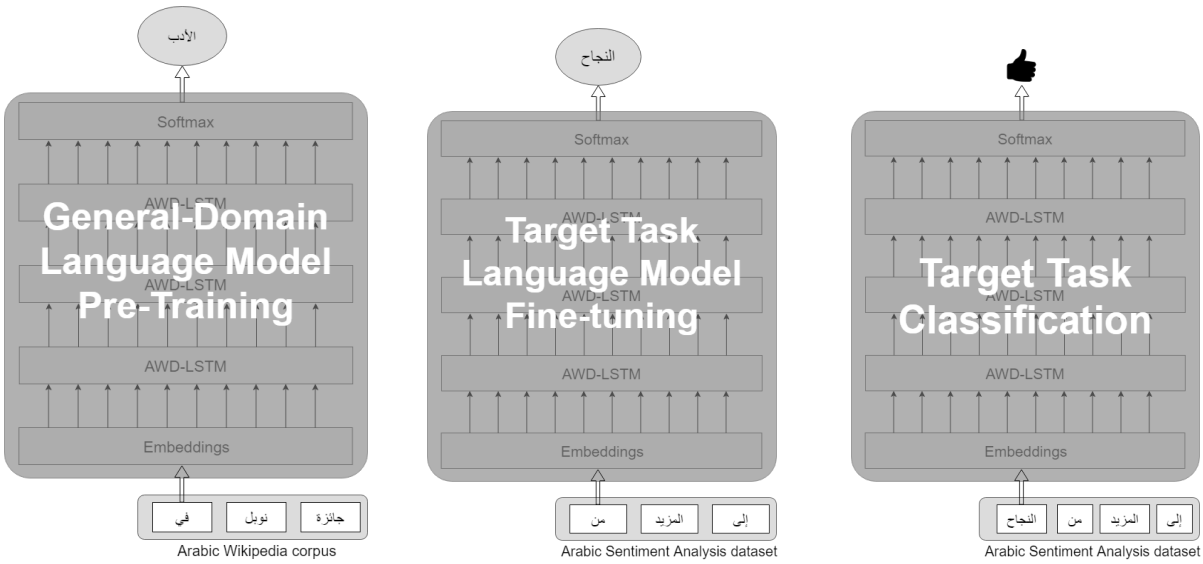
3.1.1 General domain huLMonA pretraining

To capture the various properties of a language, we constructed a large scale Arabic language modeling dataset by extracting text from Arabic Wikipedia. The 600K Wikipedia articles were used to train a three layers of the start-of-the-art language model architecture, namely AWD-LSTM (Merity et al., 2017). The output of this stage is the model weights and the distributional representations of each word in the constructed corpus, also know as word embeddings. Although Wikipedia text is mainly in MSA, the resultant pretrained model can be fine-tuned later on different text text genres (e.g., tweets) and Arabic dialects to outperform training from scratch. Due to the huge amount of text and model parameters, especially at the last softmax layer which has as many neurons as the vocabulary size, the pretraining stage consumes much time and computational power. Fortunately, pretraining is done once, and the resultant model is made available to the community.

3.1.2 Target task huLMonA fine-tuning

Regardless of the diversity of the general-domain data, the target task data will likely come from

[1] http://www.oma-project.com/



(a) LM pre-training (b) LM fine-tuning (c) classification

Figure 2: Three-step Process for Creating hULMonA

a different distribution. Although the general-domain LM is trained on MSA, most Arabic datasets and social media platforms contains dialects. Unlike MSA, dialects have no standard or codified form and are influenced by region specific slang. Thus, fine-tuning the pretrained general-domain LM on the target task data is necessary for the LM to adapt to the new textual properties. One difference though is that fine-tuning utilizes different learning rates for different layers, which is referred to as discriminative fine-tuning. This is crucial since different layers capture different types of information (Yosinski et al., 2014). Discriminative fine-tuning updates the model parameters as follows:

$$\theta_t^l = \theta_{t-1}^l - \eta^l \cdot \nabla_{\theta^l} J(\theta)$$

where θ^l is the model parameters of layer l, and η^l is the learning rate of layer l.

3.1.3 Augmenting hULMonA with target task classification layers

Finally, two fully connected layers are added to the LM for classification with ReLU and Softmax activations respectively. At first, the two fully connected layers are trained from scratch, while previous layers are frozen. After each epoch, the next lower frozen layer is unfrozen and fine-tuned until convergence. This is known as gradual unfreezing, and it is essential to avoid catastrophic forgetting of the information captured during language modeling.

3.2 Multi-lingual BERT ULM for Arabic tasks

3.2.1 Data Pre-processing

The ULM BERT model requires a special format for the data before feeding the model. A special token, called [CLS], is added at the beginning of every sentence and a special token, called [SEP] is added at the end of every sentence. For Arabic tokenization, we chose WordPiece(Wu et al., 2016) tokenizer as it was also used during the pre-training of BERT. Figure 3 presents a sentence before and after going through the BERT tokenizer.

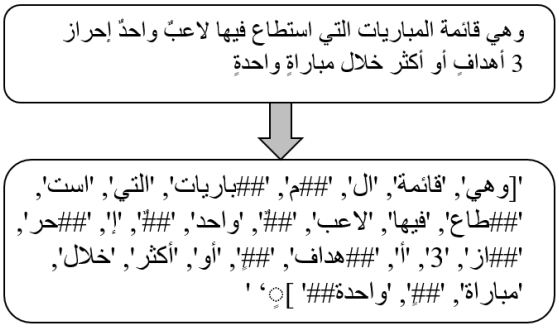


Figure 3: BERT Tokenizer Results

The tokenizer splits sentences into WordPiece tokens separated by ##. After tokenization, each word is mapped to an index using a 110k token vocabulary file that is provided by BERT for all the languages.

3.2.2 Model Fine Tuning

For sentiment analysis, or other Multi-label classification problems, a linear (fully-connected) layer with a standard softmax activation function is added to the last hidden state of the first token (the [CLS] token) as shown in Figure 4. With a hidden state vector $C \in R^H$ where H is the dimension of the hidden state and a fully-connected classification layer with weights $W \in R^{K \times H}$ where K is the number of classification labels, the label probability after applying the softmax function is then $P = softmax(CW^T)$.

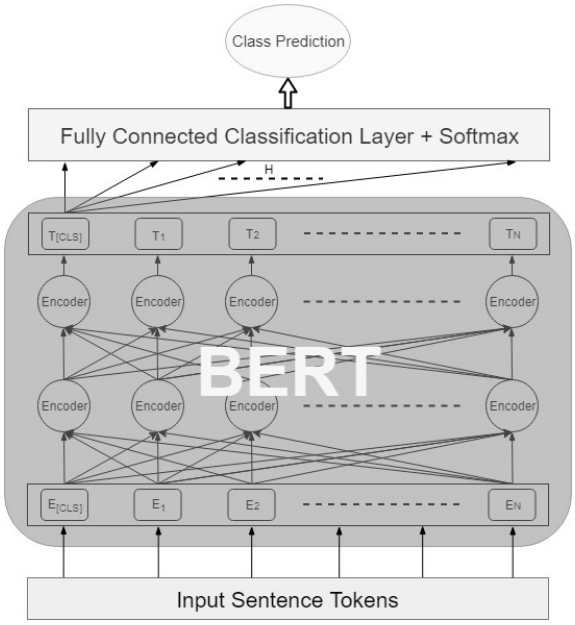


Figure 4: BERT Fine-Tuning Model Architecture

3.3 Benchmark Dataset for ULM Evaluation with Sentiment Analysis

To provide credible evaluation for the performance of the two ULM's, we catalog a benchmark dataset for Arabic which can also be used for future research benchmark evaluations. The data sets vary in size allowing us to demonstrate the ULM's abilities to fine tune with little data and achieve high performance. The benchmark data set is summarized in table 1 along with statistics on its content.

3.3.1 HARD data set

The Hotel Arabic Reviews Dataset (HARD) (Elnagar et al., 2018) is a dataset of hotel reviews written in Modern Standard Arabic and Arabic dialect classified into positive and negative. The dataset consists of a corpus of 93,700 hotel reviews which are equally divided into 46,850 positive reviews and 46.850 negative reviews. The dataset is structured in columns containing the number of the review, the name of the hotel, the rating given by the user, the type of the user, the type of the room, the number of nights stayed, and the review. Reviews have been classified into positive and negative according to the rating given by the user. A negative review is defined by a rating of 1 or 2 and a positive review is defined by a rating of 4 or 5. Neutral reviews of rating 3 were ignored in this dataset.

3.3.2 ASTD data set

The Arabic Sentiment Tweets Dataset (ASTD) (Nabil et al., 2015) is a corpus of 10,000 tweets written in MSA and Egyptian dialect. The unbalanced dataset has been manually annotated and structured in columns containing the tweet and its sentiment whether it is objective, neutral, positive, or negative. The dataset consists of 777 positive tweets, 1,642 negative tweets, 805 neutral tweets, and 6,466 objective tweets. A balanced version, called ASTD-B, is created as well taking into account positive and negative tweets only.

3.3.3 ArSenTD-Lev

(Baly et al., 2018) developed The Arabic Sentiment Twitter Dataset for LEVantine dialect (ArSenTD-Lev), a corpus of 4,000 tweets collected from Levantine countries (Palestine, Jordan, Syria, and Lebanon) and annotated for sentiment, topic, target, etc.

4 Experiments and Results

In this section, we discuss in detail the experiments that were conducted to evaluate the development of hULMonA, fine-tuning of hULMonA and BERT, and and testing the performance of the models with sentiment analysis. The benchmark data set was used to fine tune both models and provide different evaluations.

4.1 Experimental Setup

We evaluate our work on four widely-studied Arabic sentiment analysis datasets, with varying numbers of sentences and dialects. All used datasets are described in details in section 3.3, and datasets statistics are shown in table 1. Following previous works, 20% of the data was held out for testing for some datasets, while other datasets were tested on 10%.

Dataset	Resource	# samples	# classes	MSA || Dialect
HARD (Elnagar et al., 2018)	Hotel reviews (www.booking.com)	93,700	2	MSA & Gulf
ASTD (Nabil et al., 2015)	Twitter	10,000	4	MSA & Egyptian
ASTD-B (Nabil et al., 2015)	Twitter	1,600	2	MSA & Egyptian
ArSenTD-Lev (Baly et al., 2018)	Twitter	4,000	5	Levantine Dialect

Table 1: Datasets statistics

Initial tokens	Generated sequence
الدكتور (Doctor)	الدكتور احمد الحسن ، كاتب وباحث سعودي ، ولد في يونيو (Doctor Ahmad Al Hassan is a Saudi writer and researcher. He was born in June)
لاعب كره قدم (football player)	لاعب كره قدم امريكي يلعب كلاعب وسط (American football player plays as midfield)
وتقع دوله (The country is located)	وتقع دوله الامارات العربيه المتحده في الشرق الاوسط (United Arab Emirates is located in the middle east)

Table 2: generating text using the pretrained Arabic language model

4.2 hULMonA Model Training

hULMonA was constructed by first extracting and preprocessing all Arabic Wikipedia articles up to March of 2019. Articles images, links, and HTML were removed using an online tool[2], and articles with less than 100 characters were excluded resulting in 600,559 Arabic articles consisting of 108M words, 4M of which were unique.

The large number of unique words requires more parameters to be learnt and is more prone to overfitting. This problem is called lexical sparsity, and it is a well-known challenge in Arabic NLP. Therefore, text was preprocessed by replacing numbers by a special token, normalizing Alif and Ta-marbota, separating punctuations from words by a white space, and removing diacritics and non-Arabic tokens. Moreover, MADAMIRA (Pasha et al., 2014), an Arabic morphological analyzer and disambiguator, was utilized to separate words prefixes, such as Al-taareef (the), and suffixes, such as possessive pronouns, resulting in words stems, thus, reducing lexical sparsity. Table 3 shows the number of unique words before and after preprocessing Arabic text using MADAMIRA. Finally, tokens that appeared less than 5 times were replaced by a special token.

The preprocessed text was then fed to train a three layers AWD-LSTM for 4 epochs to predict next token given current sequence of tokens. Each epoch took around 200 minutes on an i7 CPU with 32 GB of RAM and Nvidia GTX 1080 GPU. We used a dropout of 0.1 with learning rate of 3e-3, and to account for GPU VRAM limitations, we were limited with batch sizes equal to 32. 10% of the data was held out for testing. Table 2 demonstrates the capabilities of the pretrained language model of generating Arabic sequence based on initial tokens. The Arabic language model dataset, code, and pre-trained weights are publicly available through the Opinion Mining for Arabic (OMA) website[3].

	Example	Unique tokens
Before	الماء مادةٌ شفافةٌ عديمة اللون والرائحة	4.1M
After	ال+ ماء ماده شفافه عديمه ال+ لون و+ ال+ راءحه	9.1K

Table 3: preprocessing reduces lexical sparsity

4.3 hULMonA Evaluation for Arabic Sentiment Analysis

To perform sentiment analysis, we fine-tuned the pretrained ULMs on a target dataset; meaning we

[2] https://github.com/attardi/wikiextractor

[3] http://www.oma-project.com/

Dataset	SOTA Results	hULMonA	BERT
HARD	93.1-93.2 (Elnagar et al., 2018)	**95.7-95.7**	**95.7-95.7**
ASTD	62.0-68.7 (Nabil et al., 2015)	**67.7-69.9**	**67.0-77.1**
ASTD-B	82.5-82.4 (Dahou et al., 2019)	**85.8-86.5**	80.0-80.1
ArSenTD-Lev	50.0-51.0 (Baly et al., 2018)	**51.1-52.4**	51.0-51.0

Table 4: Comparison of results (F1-Accuracy) obtained using hULMonA and other state-of-the-art models

resume training the language model to predict the next token but with a sentiment dataset instead of Wikipedia. Fine-tuning improved the model by adapting to new words (e.g., dialects) or words that may convey several meanings. Fine-tuning was done on each of the data sets in the aforementioned benchmark data separately and utilizing different learning rates for different layers, ranging from 2e-5 to 1e-3. Finally, after adding a classification layer, the network was trained by unfreezing one layer after each epoch, starting from the output layer. Results are reported in table 4. Note that hULMonA outperformed the state-of-the-art in four Arabic sentiment analysis datasets, demonstrating the benefit of transferring knowledge from a large corpus into small and dialectal datasets.

4.4 BERT ULM Model Fine Tuning for Arabic Sentiment Analysis

BERT was fine-tuned on the different datasets independently. The learning rate and number of epochs used for each dataset are shown in table 5. Batch size was also fixed for BERT at 32 due to our hardware memory limitations. Fine-tuning took 90 ~100 seconds for every 3000 data-point on Google's Colaboratory TensorFlow environment with GPU acceleration. BERT Base Multilingual Cased used as it is recommended in BERT's github repository[4] and the pre-trained weights were downloaded from TensorFLow's Hub[5].

Dataset	Learning Rate	# of Epochs
HARD	10^{-5}	3
ASTD	10^{-5}	5
ASTD-B	10^{-5}	5
AJGT	2×10^{-5}	6
ArSenTD-Lev	2×10^{-5}	5

Table 5: Learning rate and number of epochs used for training each dataset

[4] https://github.com/google-research/bert/blob/master/multilingual.md

[5] https://tfhub.dev/f/google

4.5 BERT ULM Evaluation for Arabic Sentiment Analysis

The results obtained are compared to state-of-the-art models and presented in Table 4. Even-though BERT achieved state-of-the-art results on two benchmark datasets, during the evaluation, we noticed that the BERT multilingual tokenizer failed to tokenize Arabic sentences as seen in Figure 3. This tokenizer could have limited the model's accuracy and compromised the model's Arabic pre-training.

5 Conclusion

This works aims at utilizing transfer learning to develop the first Arabic universal language model, hULMonA, that can be fine-tuned for almost any Arabic text classification task. Language knowledge learnt unsupervisedly from general-domain dataset is transferred to target task to improve overall performance and generalization. We show that hULMonA outperforms several state-of-the-art Arabic sentiment analysis datasets, and we make hULMonA available for the community. In addition, we evaluate another ULM, BERT, and compare results.

As a future work, we aim at utilizing hULMonA to improve more Arabic NLP tasks such as emotion recognition, cyberbullying detection, question answering, etc. Moreover, we plan to develop Arabic specific BERT by improving its limited tokenizer and training on Arabic only instead of multiple languages at once.

References

Muhammad Abdul-Mageed and Mona Diab. 2012. Toward building a large-scale arabic sentiment lexicon. In *Proceedings of the 6th international global WordNet conference*, pages 18–22.

Muhammad Abdul-Mageed and Mona T Diab. 2014. Sana: A large scale multi-genre, multi-dialect lexicon for arabic subjectivity and sentiment analysis. In *LREC*, pages 1162–1169.

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*.

Ahmad Al-Sallab, Ramy Baly, Hazem Hajj, Khaled Bashir Shaban, Wassim El-Hajj, and Gilbert Badaro. 2017. Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):25.

Ahmad Al Sallab, Hazem Hajj, Gilbert Badaro, Ramy Baly, Wassim El Hajj, and Khaled Bashir Shaban. 2015. Deep learning models for sentiment analysis in arabic. In *Proceedings of the second workshop on Arabic natural language processing*, pages 9–17.

Gilbert Badaro, Ramy Baly, Hazem Hajj, Wassim El-Hajj, Khaled Bashir Shaban, Nizar Habash, Ahmad Al-Sallab, and Ali Hamdi. 2019. A survey of opinion mining in arabic: A comprehensive system perspective covering challenges and advances in tools, resources, models, applications, and visualizations. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(3):27.

Gilbert Badaro, Ramy Baly, Hazem Hajj, Nizar Habash, and Wassim El-Hajj. 2014. A large scale arabic sentiment lexicon for arabic opinion mining. In *Proceedings of the EMNLP 2014 workshop on arabic natural language processing (ANLP)*, pages 165–173.

Gilbert Badaro, Obeida El Jundi, Alaa Khaddaj, Alaa Maarouf, Raslan Kain, Hazem Hajj, and Wassim El-Hajj. 2018. Ema at semeval-2018 task 1: Emotion mining for arabic. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 236–244.

Ramy Baly, Hazem Hajj, Nizar Habash, Khaled Bashir Shaban, and Wassim El-Hajj. 2017. A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in arabic. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):23.

Ramy Baly, Alaa Khaddaj, Hazem Hajj, Wassim El-Hajj, and Khaled Shaban. 2018. Arsentd-lev: A multi-topic corpus for target-based sentiment analysis in arabic levantine tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Abdelghani Dahou, Mohamed Abd Elaziz, Junwei Zhou, and Shengwu Xiong. 2019. Arabic sentiment classification using convolutional neural network and differential evolution algorithm. *Computational Intelligence and Neuroscience*, 2019.

Abdelghani Dahou, Shengwu Xiong, Junwei Zhou, Mohamed Houcine Haddoud, and Pengfei Duan. 2016. Word embeddings and convolutional neural network for arabic sentiment classification. In *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers*, pages 2418–2427.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rehab M Duwairi. 2015. Sentiment analysis for dialectical arabic. In *2015 6th International Conference on Information and Communication Systems (ICICS)*, pages 166–170. IEEE.

Ashraf Elnagar, Yasmin S Khalifa, and Anas Einea. 2018. Hotel arabic-reviews dataset construction for sentiment analysis applications. In *Intelligent Natural Language Processing: Trends and Applications*, pages 35–52. Springer.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. *arXiv preprint arXiv:1803.11138*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519.

Arfath Pasha, Mohamed Al-Badrashiny, Mona T Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *LREC*, volume 14, pages 1094–1101.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

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.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

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, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.