Post-hoc analysis of Arabic transformer models

Ahmed Abdelali[♢] **Nadir Durrani**[♢] **Fahim Dalvi**[♢] **Hassan Sajjad**[♣*]
[♢]Qatar Computing Research Institute, Hamad Bin Khalifa University, Qatar
[♣]Faculty of Computer Science, Dalhousie University, Canada
{aabdelali, ndurrani,faimaduddin}@hbku.edu.qa, hsajjad@dal.ca

Abstract

Arabic is a Semitic language which is widely spoken with many dialects. Given the success of pre-trained language models, many transformer models trained on Arabic and its dialects have surfaced. While there have been an extrinsic evaluation of these models with respect to downstream NLP tasks, no work has been carried out to analyze and compare their internal representations. We probe how linguistic information is encoded in the transformer models, trained on different Arabic dialects. We perform a layer and neuron analysis on the models using morphological tagging tasks for different dialects of Arabic and a dialectal identification task. Our analysis enlightens interesting findings such as: i) word morphology is learned at the lower and middle layers, ii) while syntactic dependencies are predominantly captured at the higher layers, iii) despite a large overlap in their vocabulary, the MSA-based models fail to capture the nuances of Arabic dialects, iv) we found that neurons in embedding layers are polysemous in nature, while the neurons in middle layers are exclusive to specific properties.

1 Introduction

Arabic is a linguistically rich language, with its structures realized using both concatenative and templatic morphology. The agglutinating aspect of the language adds to the complexity where a given word could be formed using multiple morphemes. For example, the word فأسقيناكموه (fOsqynAkmwh[1] – and we gave it to you to drink) combines a conjunction, a verb, and three pronouns. At another longitude, Arabic has three variants: Classical Arabic (CA), Modern Standard Arabic (MSA) and Dialectal Arabic (DA). While the MSA is traditionally considered as the de facto standard in the written medium and DA being the predominantly spoken counterpart, this has changed recently (Mubarak and Darwish, 2014; Zaidan and Callison-Burch, 2014; Durrani et al., 2014). Due to the recent influx of Social Media platforms, dialectal Arabic also enjoys a significant presence in the written medium.

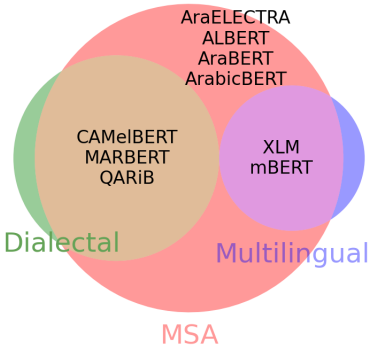


Figure 1: Data regimes of various pre-trained Transformer models of Arabic

Transfer learning using contextualized representations in pre-trained language models have revolutionized the arena of downstream NLP tasks. A plethora of transformer-based language models, trained in dozens of languages are uploaded every day now. Arabic is no different. Several researchers have released and benchmarked pre-trained Arabic transformer models such as AraBERT (Antoun et al., 2020), ArabicBERT (Safaya et al., 2020), CAMeLBERT (Inoue et al., 2021), MARBERT (Abdul-Mageed et al., 2020) and QARIB (Abdelali et al., 2021) etc. These models have demonstrated state-of-the-art performance on many tasks as well as their ability to learn salient features for Arabic. One of the main differences among these models is the genre and amount of Arabic data they are trained on. For example, AraBERT was trained only on the MSA (Modern Standard Arabic),

*The work was done while the author was at QCRI
[1]Using Safe Buckwalter Arabic (SBA) encoding.

ArabicBERT additionally used DA during training, and CAMelBERT-mix used a combination of all types of Arabic text for training. Multilingual models such as mBERT and XLM are mostly trained on Wikipedia and CommonCrawl data which is predominantly MSA (Suwaileh et al., 2016). Figure 1 summarizes the training data regimes of these models.

This large variety of Arabic pre-trained models motivates us to question **how their representations encode various linguistic concepts?** To this end, **we present the first work on interpreting deep Arabic models.** We experiment with nine transformer models including: five Arabic BERT models, Arabic ALBERT, Arabic Electra, and two multilingual models (mBERT and XLM). We analyze their representations using MSA and dialectal parts-of-speech tagging and dialect identification tasks. This allows us to compare the representations of Arabic transformer models using tasks involving different varieties of Arabic dialects.

We analyze representations of the network at layer-level and at neuron-level using diagnostic classifier framework (Belinkov et al., 2017; Hupkes et al., 2018). The overall idea is to extract feature vectors from the learned representations and train probing classifiers towards understudied auxiliary tasks (of predicting morphology or identifying dialect). We additionally use the *Linguistic Correlation Analysis* method (Dalvi et al., 2019a; Durrani et al., 2020) to identify salient neurons with respect to a downstream task. Our results show that:

Network and Layer Analysis

- Lower and middle layers capture word morphology
- The long-range contextual knowledge required to solve the dialectal identification is preserved in the higher layers

Neuron Analysis

- The salient neurons with respect to a property are well distributed across the network
- First (embedding) and last layers of the models contribute a substantial amount of salient neurons for any downstream task
- The neurons of embedding layer layer are polysemous in nature while the neurons of middle layers specializes in specific properties

MSA vs. Dialect

- Although dialects of Arabic are closely related to MSA, the pre-trained models trained using MSA only do not implicitly learn nuances of dialectal Arabic

2 Methodology

Our methodology is based on the class of interpretation methods called as the *Probing Classifiers*. The central idea is to extract the activation vectors from a pre-trained language model as static features. These activation vectors are then trained towards the task of predicting a property of interest, a linguistic task that we would like to probe the representation against. The underlying assumption is that if the classifier can predict the property, the representations implicitly encode this information. We train layer (Belinkov et al., 2020) and neuron probes (Durrani et al., 2022) using logistic-regression classifiers.

Formally, consider a pre-trained neural language model $\mathbf{M}$ with L layers: $\{l_1, l_2, \ldots, l_L\}$. Given a dataset $\mathbb{D} = \{w_1, w_2, ..., w_N\}$ with a corresponding set of linguistic annotations $\mathbb{T} = \{t_{w_1}, t_{w_2}, ..., t_{w_N}\}$, we map each word w_i in the data $\mathbb{D}$ to a sequence of latent representations: $\mathbb{D} \overset{\mathbb{M}}{\longmapsto} \mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$. The layer-wise probing classifier is trained by minimizing the following loss function:

$$\mathcal{L}(\theta) = -\sum_i \log P_\theta(t_{w_i}|w_i)$$

where $P_\theta(t_{w_i}|w_i) = \frac{\exp(\theta_l \cdot \mathbf{z}_i)}{\sum_{l'} \exp(\theta_{l'} \cdot \mathbf{z}_i)}$ is the probability that word i is assigned property t_{w_i}.

For neuron analysis, we use *Linguistic Correlation Analysis* (LCA) as described in (Dalvi et al., 2019a). LCA is also based on the probing classifier paradigm. However, they used elastic-net regularization (Zou and Hastie, 2005) that enables the selection of both focused and distributed neurons. The loss function is as follows:

$$\mathcal{L}(\theta) = -\sum_i \log P_\theta(t_{w_i}|w_i) + \lambda_1\|\theta\|_1 + \lambda_2\|\theta\|_2^2$$

The regularization parameters λ_1 and λ_2 are tuned using a grid-search algorithm. The classifier assigns weight to each feature (neuron) which serves as their importance with respect to a class like Noun. We ranked the neurons based on the

absolute weights for every class. We select salient neurons for the task such as POS by iteratively selecting top neurons of every class.

A minimum set of neurons is identified by iteratively selecting top neurons that achieves classification performance comparable (within a certain threshold) to the *Oracle* – accuracy of the classifier trained using all the features in the network.

Data	Size	Tokens	Vocab	Type
AraBERT	23GB	2.7B	64K	MSA
ArabicBERT	95GB	8.2B	32K	MSA
CAMeLBERT	167B	17.3B	30K	MSA/CA/DA
MARBERT	128GB	15.6B	100K	MSA/DA
mBERT	-	1.5B	110K	MSA
QARiB	127GB	14.0B	64K	MSA/DA
AraELECTRA	77GB	8.6B	64K	MSA
ALBERT	-	4.4B	30K	MSA
XLM	2.5TB	-	250K	MSA

Table 1: Pretrained Models data and statistics.

3 Experimental Setup

In this section, we describe our experimental setup including the Arabic transformer models, probing tasks that we have used to carry the analysis and the classifier settings.

3.1 Pre-trained Models

We select a number of Arabic transformer models, trained using various varieties of Arabic and based on different architectures. Table 1 provides a summary of these models. In the following, we describe each model and the dataset used for their training.

AraBERT was trained using a combination of 70 million sentences from Arabic Wikipedia Dumps, 1.5B words Arabic Corpus (El-khair, 2016) and the Open Source International Arabic News Corpus (OSIAN) from (Zeroual et al., 2019). The final corpus contained mostly MSA news from different Arab regions.

ArabicBERT Safaya et al. (2020) pretrained a BERT model using a concatenation of Arabic version of OSCAR (Ortiz Suárez et al., 2019), a filtered subset from Common Crawl and a dump of Arabic Wikipedia totalling to 8.2B words.

CAMeLBERT Inoue et al. (2021) combined a mixed collection of MSA, Dialectal and Classical Arabic texts with a total of 17.3B tokens. They used the data to pre-train CAMeLBERT-Mix model.

MARBERT Abdul-Mageed et al. (2020) combined a dataset of 1B tweets that covering mostly Arabic dialects and Arabic Gigaword 5th Edition,[2] OSCAR (Ortiz Suárez et al., 2019), OSIAN (Zeroual et al., 2019) and Wikipedia dump totally up to 15.6B tokens.

QARIB Abdelali et al. (2021) combined Arabic Gigaword Fourth Edition,[3] 1.5B words Arabic Corpus (El-khair, 2016), the Arabic part of Open Subtitles (Lison and Tiedemann, 2016) and 440M tweets collected between 2012 and 2020. The data was processed using Farasa (Abdelali et al., 2016).

ALBERT used a subset of OSCAR (Ortiz Suárez et al., 2019) and a dump of Wikipedia, selecting around 4.4 Billion words (Safaya, 2020). The model differs from BERT using factorized embedding and repeating layers which results in a small memory footprint (Lan et al., 2020).

AraELECTRA ELECTRA, model Clark et al. (2020) is trained to distinguish "real" vs "fake" input tokens generated by another neural network. The Arabic ELECTRA was trained on 77GB of data combining OSCAR dataset, Arabic Wikipedia dump, the 1.5B words Arabic Corpus, the OSIAN Corpus and Assafir news articles (Antoun et al., 2021). Different than other models, AraELECTRA uses a hidden layer size of 256 while all other models have 768 neurons per layer.

Multilingual BERT Google research released BERT multilingual base model pretrained on the concatenation of monolingual Wikipedia corpora from 104 languages with a shared word piece vocabulary of 110K.

XLM Conneau et al. (2020) is a multilingual version of RoBERTa, trained on 2.5TB CommonCrawl data. The model is trained on 100 different languages.

3.2 Probing Tasks

We consider morphological tagging on a variety of Arabic dialects and dialectal identification tasks to analyze and compare the models. Below we describe the task details.

POS Tagging on Arabic Treebank (ATB): The Arabic Treebank Part1 v2.0 and Part3 v1.0 with a total of 515k tokens labeled at the segment level with POS tags. The data is a combination of

[2]LDC Catalogue LDC2011T11
[3]LDC Catalogue LDC2009T30

ATB	Text	واعرب بيرسول عن دهشته ازاء تطور مستواه المتواصل ،
	Labels	VBD NNP IN NN NN NN NN DT+JJ PUNC
	SBA	wAErb byrswl En dhcth AzAC tTwr mstwAh AlmtwASl ,
	Gloss.	And Peirsol expressed his surprise at the continuous development of his level ,
CRS	Text	نط أخوي الصغير وجبلي مي !
	Labels	VERB NOUN+PRON DET+ADJC PREP+VERB+PREP+PRON NOUN PUNC
	SBA	nT Oxwy AlSgyr wjbly my !
	Gloss.	My little brother jumped and brought me water !
DIA	Text	في ناس مناح ما في متلن ، و في ناس منيح اللي ما في متلن
	Labels	ADV NOUN ADJ PART ADV NOUN PUNC CONJ ADV NOUN ADJ PART PART ADV NOUN
	SBA	fy nAs mnAH mA fy mtln , w fy nAs mnyH Ally mA fy mtln
	Gloss.	There are good people who are unparalleled, and there are people that it is good they are unparalleled.
DID	Text	له مجددا أقول مفقوسة أوي !
	Labels	lang1 lang1 ambiguous lang2 lang2 other
	SBA	lh mjddA Oqwl mfqwsp Owy
	Gloss.	For him again I say (I am) very upset !
GMR	Text	سيف : انا ما غلطت قلت صدق
	Labels	NOUN_PROP:MS PUNC:- PRON:1S PART:- VERB:P1S VERB:P1S NOUN:MS
	SBA	syf : AnA mA glTt qlt Sdq
	Gloss.	Saif: I wasn't wrong, I said the truth.

Table 2: Examples of Arabic annotated text and their corresponding labels for each task.

newswire text from An-Nahar and Agence France Presse corpus (Maamouri et al., 2004). The data is labeled with 42 distinct tags.

Gumar POS Tagging on Gulf Arabic (GMR): Khalifa et al.(2020) compiled a collection of 15,225 sentences from eight different novels written in the Emirati Arabic dialect from the Gumar Corpus (Khalifa et al., 2018). The data was manually annotated for tokenization, part-of-speech, lemmatization, spelling adjustment, English glosses and sentence level dialect identification, using 169 tags.

Curras POS Tagging on Palestinian Arabic dialect (CRS): Jarrar et al.(2017) collected around 5K sentences written in Palestinian Arabic dialect from web blogs, Twitter and Facebook comments and transcripts from a TV Shows Watan Aa Watar. The sentences were manually annotated for part-of-speech (POS), stem, prefix, suffix, lemma, and gloss using 260 tags.

POS Multidialects (DIA): A total of 1.4k tweets from four Arabic dialects, namely Egyptian (EGY), Levantine (LEV), Gulf (GLF), and Maghrebi (MGR). The tweets were morphologically tagged (Samih et al., 2017) using a reduced subset of 22 tags.

Dialect IDentification (DID): This task is related to code switching and language identification (LID) between MSA and Egyptian dialect on social media content. The data comprises intrasentential code switched sentences (mixing languages between utterances) used for the Second Shared Task on Language Identification in Code-Switched Data. The data contains over 11k sentences, where each token in the sentences is labeled with one of the eight labels:lang1, lang2, fw, mixed, unk, ambiguous, other and named entities (ne) (Molina et al., 2016).

Figure2 shows examples for each of the probing tasks with their respective labels.

3.3 Post-hoc Classifier

We used the NeuroX toolkit (Dalvi et al., 2019b) to perform our analysis. Our probe is a linear classifier with categorical cross-entropy loss, optimized by Adam (Kingma and Ba, 2014). For neuron-analysis, the classifier additionally used the elastic-net regularization (Zou and Hastie, 2005). The regularization weights are trained using grid-search algorithm. Training is run with shuffled mini-batches of size 512 and stopped after 10 epochs. Linear classifiers are a popular choice in analyzing deep NLP models due to their better interpretability (Qian et al., 2016). Hewitt and Liang (2019) have also shown linear probes to have higher *Selectivity*, a property deemed desirable for more interpretable probes. We perform control task experiments to ensure that our probes are reflective of the linguistic knowledge that representations capture. For sub-word based models, we use the average activation value (Durrani et al., 2019) to be the representative of the word. We additionally

Task	ATB		CRS		DIA		DID		GMR		Avg. Acc.
Model	Acc.	Sel.	Acc.	Sel.	Acc.	Sel.	Acc.	Sel.	Acc.	Sel.	
AraBERT	93.9	48.1	77.3	22.1	79.0	58.1	84.7	37.4	90.4	06.4	85.06
ArabicBERT	95.2	48.2	80.5	24.7	83.6	50.4	91.2	34.7	91.6	05.1	88.43
CAMelBERT	**95.8**	39.2	82.9	23.6	**86.0**	37.2	92.0	21.3	93.0	05.6	89.94
MARBERT	95.6	51.4	**84.2**	27.5	84.8	48.8	93.1	33.9	**93.4**	07.3	90.22
QARiB	**95.8**	50.6	84.0	28.9	85.4	45.0	**93.3**	28.8	93.3	06.7	**90.38**
mBERT	94.4	48.8	73.7	22.7	77.6	58.4	81.7	36.3	88.0	04.4	83.08
AraELECTRA	94.4	46.9	72.7	28.4	79.0	56.2	87.9	34.3	89.1	08.1	84.64
ALBERT	95.2	40.9	77.0	28.3	82.1	39.8	88.3	27.1	90.2	09.4	86.56
XLM	95.7	43.7	75.0	20.6	78.9	42.1	86.7	29.3	88.2	06.0	84.90

Table 3: Classifier performance on Test sets using **top layers**

Task	ATB		CRS		DIA		DID		GMR	
Threshold δ	5%		10%		10%		7%		5%	
Model	Acc.	Sel.	Acc.	Sel.	Acc.	Sel.	Acc.	Sel.	Acc.	Sel.
AraBERT	93.4	48.8	82.1	33.5	79.3	39.9	86.0	21.5	89.9	18.1
ArabicBERT	94.0	50.8	83.6	31.6	83.3	44.9	90.1	26.3	91.0	12.5
CAMelBERT	94.9	51.1	86.1	37.0	85.1	47.5	91.0	27.2	92.6	22.6
MARBERT	94.5	51.6	86.2	30.0	84.2	48.5	91.6	29.2	92.3	15.0
QARiB	95.0	52.7	86.1	31.0	83.6	46.7	91.7	30.0	92.4	11.6
mBERT	94.1	48.5	78.4	33.1	77.5	37.7	83.2	17.0	87.6	13.4
AraELECTRA	91.2	53.3	79.0	33.9	79.4	45.0	88.2	25.6	87.6	13.1
ALBERT	94.7	56.8	80.7	33.7	81.8	47.4	88.5	25.9	89.8	12.0
XLM	95.3	51.8	78.5	27.9	79.0	44.5	86.5	21.7	88.0	12.4

Table 4: Classifier performance on Test sets using **top neurons** as features

normalize the embeddings using znorm as it has shown to provide better ranking of neurons with respect to a property (Sajjad et al., 2021a).

4 Analysis and Discussion

Our goal is to carry out a comparative investigation of the knowledge encoded in different Arabic transformer models. First we compare the representations in terms of how much linguistic information is preserved in the network using the overall accuracy on the understudied auxiliary tasks. Then we analyze how such information is preserved across individual layers of the model. Lastly, we analyze the distribution of neurons across the model with respect to these tasks.

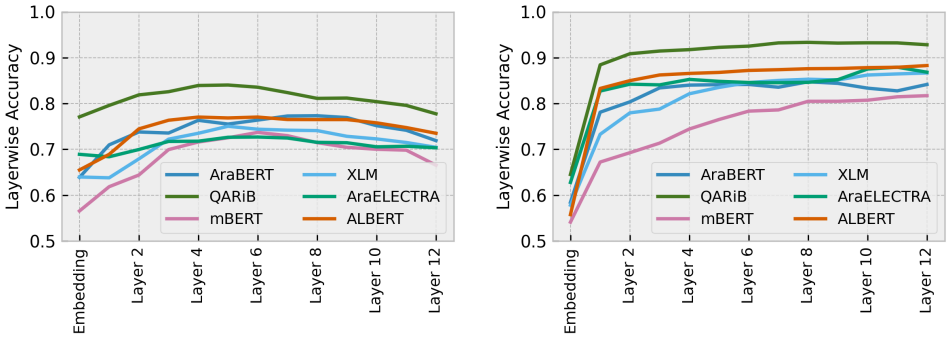
4.1 Network Analysis

We use the feature vectors[4] generated from different dialects of Arabic to train post-hoc classifiers towards the task of predicting morphology in these dialects or predicting the dialect themselves. Table 3 gives accuracy of the classifiers on different dialectal tasks. Firstly, the high accuracy numbers show that the representations learn non-trivial linguistic knowledge. We found all the models to do well on the task of predicting MSA morphology unsurprisingly, since all these models have been trained on a large amount of MSA data. Contrastingly, the performance varied a lot on the dialectal tasks with different models giving optimal performance on different dialects. Note that the models that were trained only using MSA performed much worse despite the fact that MSA and dialect have a significant vocabulary overlap. This shows that **to capture specific dialectal nuances these transformer models need to train on dialectal data.** Comparing the models, we found dialectal models (QARiB, MARBERT and CAMeLBERT) to perform considerably well across all the tasks. Lastly the high selectivity numbers in Table 3 validate the fact that our classifiers are not memorizing the tasks and are a true reflection of the knowledge captured within the underlying representations.

4.2 Layer-wise Analysis

We now analyze how the understudied linguistic knowledge is distributed across the layers. We train a classifier for each probing task using representations of individual layers as features. The

[4]We concatenated the features from all layers of the network to train the classifier.



(a) CRS (b) DID

Figure 2: Layer-wise accuracy for different selected tasks.

performance of the classifier serves as a proxy to the amount of task knowledge learned in each layer representation. Figure 2 provides per-layer accuracy for the CRS (morphological tagging for Palestinian dialect) and DID (Dialect Identification) tasks.[5] We found that **the word morphology is captured predominantly in the lower layers of the model**, retained in the middle layers before declining in the final layers. **The higher layers are reserved for complex phenomenon such as capturing non-local dependencies.** This is confirmed from our DID results. Identifying dialect requires learning non-local dependencies and sentence level phenomenon to accurately predict the dialect. For example, a lexical form can belong to two different dialects depending on the context to disambiguate the dialect of the word. For example, حاجة "HAjp" (thing or need) is MSA in the context: لست في حاجة لأن "lst fy HAjp lOn" (I am not in **need** to) or Egyptian: مفيش حاجة أصعب من "mfyc HAjp OSEb mn" (there is no **thing** difficult than). The contextual knowledge is essential to disambiguate in such cases.

4.3 Neurons Analysis

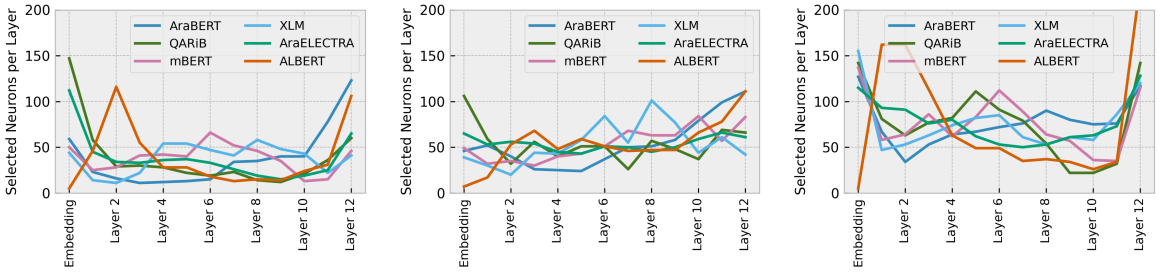
We now study how the information is spread across neurons instead of layer by carrying a fine-grained analysis. We discover neurons that learn a particular linguistic property using LCA (Dalvi et al., 2019a) and analyze: i) how many neurons can sufficiently capture a concept, ii) how these neurons are distributed across the layers. LCA provides a ranked list of neurons with respect to the understudied property. We select a minimumal set of top neurons from the ranked list that yield close to the oracle performance.[6]

Minimal Neurons: We found 5% neurons to be optimal for ATB, and GMR; while 10% for both CRS and DIA tasks, due to their more granular tag-set. For the DID task, we found 7% neurons to be optimal (Table 4 shows results – please also see Appendix for a more detailed result using different neuron thresholds). Our results show that **a small subset of features can achieve close to oracle performance.** This entails that re-trainable features are available in the network as also shown by Dalvi et al. (2020). Such a finding entails interesting frontier in efficient feature-based transfer learning, which is considered as a viable alternative to the traditional fine-tuning based transfer learning (Peters et al., 2019; Durrani et al., 2021; Alrowili and Shanker, 2021).

Neuron Distribution: Let us now turn our attention towards how these neurons are distributed in the network. In Figure 3 we plot salient neurons across the layers (See Appendix for all the tasks). A dominant pattern that we observed was that the **embedding and final layers of the model consistently contribute the most number of salient neurons**. This entails that while the neurons in middle layers capture intricate details of the task, the input and output layers of the model that are closer to the actual words possess most lexical information required to for accurate predictions. The model uses the embedding layer to focus on more localized information and final layers to capture contextual dependencies. An exception to this overall pattern was the ALBERT

[5] We limit the presentation to fewer models for clarity purposes. Our observations consistently hold for all dialectal tasks. See Appendix for complete results.

[6] Accuracy when using the entire network or best layer, whichever is higher.



(a) ATB (b) DIA (c) DID

Figure 3: Distribution of selected neurons across the layers

model, where the embedding layer has close to zero contribution in the salient neurons and relatively higher number of neurons from the initial contextualized layers. Recall that ALBERT has a different architecture where parameters are shared across the encoder layers. Moreover the model factorizes the embedding layer. These architectural choices perhaps explain the difference of neuron distribution pattern. A detailed analysis of word embedding layers using lexical tasks such as word similarity and word relatedness is required to fully understand this.

Property Distribution: We have seen how salient neurons distribute across the network. Now we analyze how these neurons distribute across sub-properties within a task. A morphological tagging task for example is composed of different properties such as Noun, Verb, Adjective etc. In Figure 4 we plot the number of salient neurons required to capture different properties on the task of predicting classes in the ATB task. We observed that **closed class categories such as personal pronoun (PRP) are localized to fewer neurons, where as the open-class words such as past-tense verbs (VBD) that exhibit a variety of roles in different contexts require a large number of neurons.** We found this observation to be true for all the models across different dialectal tasks (Please see Appendix for more results).

Layer-wise Property Distribution: We also analyzed **how individual properties are encoded across the layers in the network**, Do they have similar neuron distribution pattern or are the specific properties learned more on higher layers than lower layers and vice versa? Figure 5 shows the distribution of selected neurons of ALBERT, AraBERT and QARiB for a few properties. We observed a very consistent pattern to the overall neuron distribution that we saw in Figure 3. For most of the properties salient neurons were contributed from the embedding and final layers, and middle layers contributed less than 20 neurons. Another interesting pattern to be noted is that noun neurons were more prevalent in the embedding layer (layer 1-2 for ALBERT) but verb neurons were dominantly found in the final layers. Verbs are considered to be structural center in linguistic theories as they connect to all other syntactic units in a sentence (Hudson, 2010). This further reinforces our result that **the higher layers of the model capture long distance dependencies**.

Polysemous Neurons: Neurons are multi-variate in nature and may capture multiple concepts. For example Bau et al. (2019) discovered switch neurons that activate positively for present-tense verbs and negatively for the past-tense verbs in LSTM encoders. We also analyzed the overlaps between salient neurons that learn different linguistic properties in an attempt to discover polysemous neurons. Figure 6 shows the overlap of neurons across properties in different layers in the QARiB model. The zeros means that none of the top neurons between the properties overlap. Note that there is a high concentration of overlapping neurons between determiners (DT), adjectives (JJ) and nouns (NN) or between determiners and verbs. The intersection was around 54% in the case of Determiner "DT" and Noun "NN". We believe this is an artifact of concatenative morphology that Arabic exhibits, where it is common for affixes such as preposition or determiner to join with nouns or adjectives to form composite constructions. We also observed that the number of polysemous neurons exist more dominantly in the embedding layer. Higher layers (Exp. Figure 6e and 6f) show less shared and overlapping neurons.

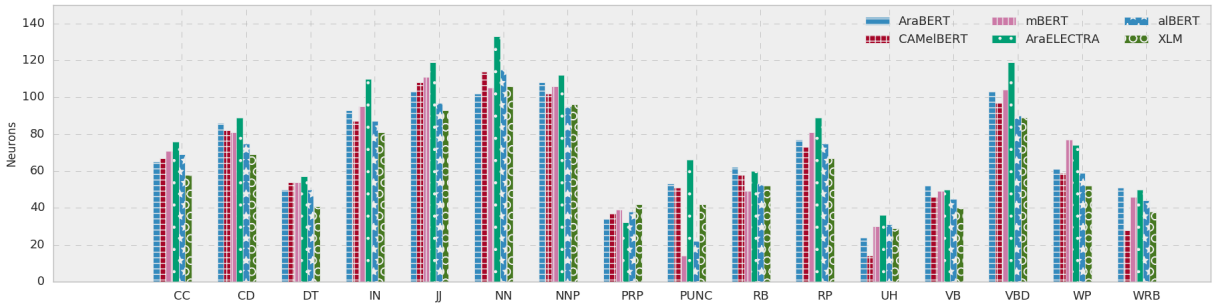
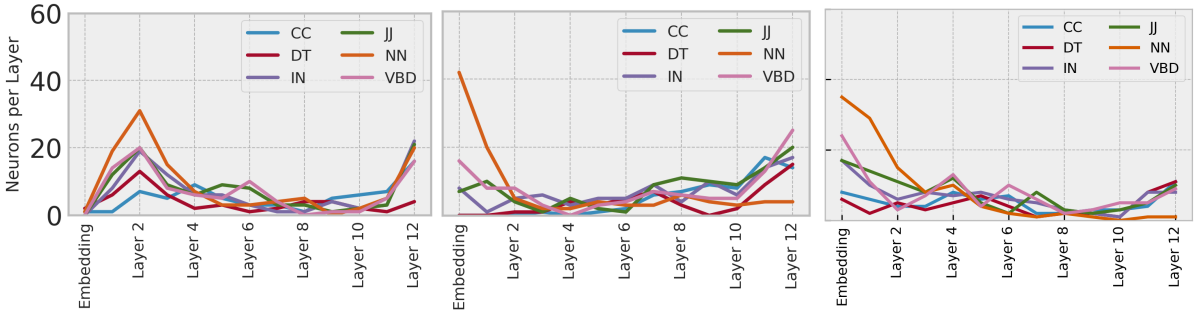
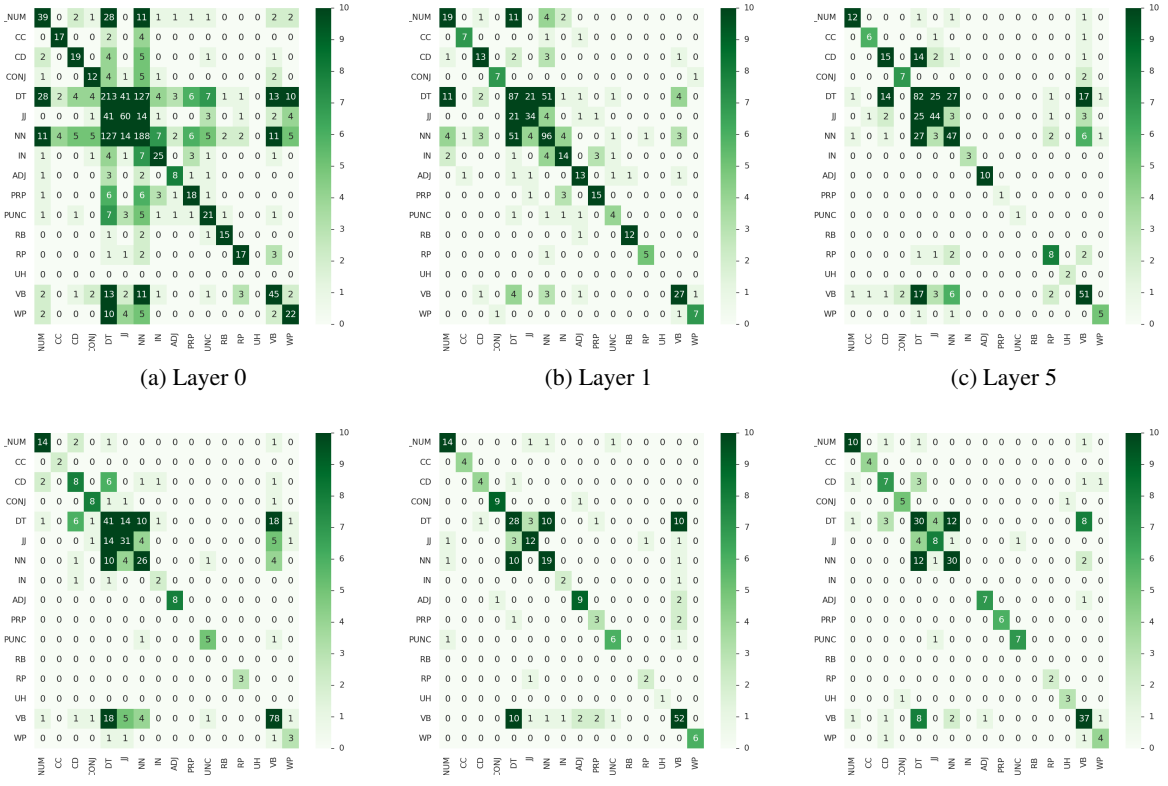


Figure 4: Distribution of neurons per property (ATB)



(a) ALBERT (b) AraBERT (c) QARiB

Figure 5: Property-wise distribution of neurons across the layers in ATB



(a) Layer 0 (b) Layer 1 (c) Layer 5

(d) Layer 6 (e) Layer 11 (f) Layer 12

Figure 6: QARiB: Neurons overlap across the ATB properties

5 Related Work

Work done on interpreting deep NLP models can be broadly classified into *Concept Analysis* and *Attribution Analysis*. The former thrives on post-hoc decomposability, where we analyze representations to uncover linguistic (and non-linguistic) phenomenon that are captured as the network is trained towards any NLP task (Conneau et al., 2018; Liu et al., 2019; Tenney et al., 2019; Sajjad et al., 2022; Dalvi et al., 2022) and the latter characterize the role of model components and input features towards a specific prediction (Linzen et al., 2016; Gulordava et al., 2018; Marvin and Linzen, 2018). Our work falls into the former category. We carry out a layer and neuron-wise analysis on the Arabic transformer models. We used Diagnostic classifiers (Belinkov et al., 2017) to train layer and neuron-wise probes towards predicting linguistic properties of interest. **To the best of our knowledge this is the first work on analyzing Arabic transformer models.** Suau et al. (2020) used *max-pooling* to identify relevant neurons (aka *Expert units*) in pre-trained models, with respect to a specific concept (for example word-sense). Mu and Andreas (2020) proposed a *Masked-based Corpus Selection* method to determine important neurons with respect to a concept. See Sajjad et al. (2021b) for a comprehensive survey of these techniques. We used the *Linguistic Correlation Analysis* of Dalvi et al. (2019a) to perform neuron analysis.

6 Conclusion and Future Work

In this paper we carry out a post-hoc analysis on a number of Arabic transformer models using five linguistic tasks. Our results enlighten interesting insights: i) neural networks learn non-trivial amount of linguistic knowledge with lower and middle layers capturing word morphology and higher layers learning more universal phenomenon, ii) we found that salient neurons are distributed across the network, but some layers contribute more salient neurons towards a task, iii) we found some neurons to be polysemous in nature while other capturing very specialized properties, iv) lastly we showed that MSA-based models do not capture dialectal nuances despite having a large overlap with dialects. For future work, we aim to expand this analysis to include more tasks and explore related languages such as the families of Semitic, Germanic or Latin languages.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California. Association for Computational Linguistics.

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. Arbert & marbert: Deep bidirectional transformers for arabic.

Sultan Alrowili and Vijay Shanker. 2021. ArabicTransformer: Efficient large Arabic language model with funnel transformer and ELECTRA objective. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1255–1261, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Araelectra: Pre-training text discriminators for arabic language understanding.

D. Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and Controlling Important Neurons in Neural Machine Translation. *arXiv preprint arXiv:1811.01157*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do Neural Machine Translation Models Learn about Morphology? In *Proceedings of ACL*, Vancouver. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2020. On the linguistic representational power of neural machine translation models. *Computational Linguistics*, 46(1):1–52.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of ACL*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, D. Anthony Bau, and James Glass. 2019a. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI, Oral presentation)*.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in BERT. In *International Conference on Learning Representations*.

Fahim Dalvi, Avery Nortonsmith, D. Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, and James Glass. 2019b. NeuroX: A toolkit for analyzing individual neurons in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Honolulu, US.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 EMNLP (EMNLP)*, pages 4908–4926, Online. Association for Computational Linguistics.

Nadir Durrani, Yaser Al-Onaizan, and Abraham Ittycheriah. 2014. Improving egyptian-to-english SMT by mapping egyptian into MSA. In *Computational Linguistics and Intelligent Text Processing - 15th International Conference, CICLing 2014, Kathmandu, Nepal, April 6-12, 2014, Proceedings, Part II*, volume 8404 of *Lecture Notes in Computer Science*, pages 271–282. Springer.

Nadir Durrani, Fahim Dalvi, and Hassan Sajjad. 2022. Linguistic correlation analysis: Discovering salient neurons in deepnlp models.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. In *Proceedings of the 2019 Conference of the NAACL-HLT, Volume 1 (Long and Short Papers)*, pages 1504–1516, Minneapolis, Minnesota. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, and Fahim Dalvi. 2021. How transfer learning impacts linguistic knowledge in deep NLP models? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4947–4957, Online. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 EMNLP (EMNLP)*, pages 4865–4880, Online. Association for Computational Linguistics.

Ibrahim Abu El-khair. 2016. 1.5 billion words arabic corpus.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of NAACL-HLT, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Shilan Hameed. 2018. Filter-wrapper combination and embedded feature selection for gene expression data. *International Journal of Advances in Soft Computing and its Applications*, 10:90–105.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 EMNLP-IJCNLP*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Richard Hudson. 2010. *An Introduction to Word Grammar*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online). Association for Computational Linguistics.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. Curras: an annotated corpus for the palestinian arabic dialect. *Lang. Resour. Evaluation*, 51(3):745–775.

Salam Khalifa, Nizar Habash, Fadhl Eryani, Ossama Obeid, Dana Abdulrahim, and Meera Al Kaabi. 2018. A morphologically annotated corpus of Emirati Arabic. In *Proceedings of LREC 2018*, Miyazaki, Japan. European Language Resources Association (ELRA).

Salam Khalifa, Nasser Zalmout, and Nizar Habash. 2020. Morphological analysis and disambiguation for Gulf Arabic: The interplay between resources and methods. In *Proceedings of LREC 2020*, pages 3895–3904, Marseille, France. European Language Resources Association.

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

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.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of LREC'16*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the NAACL-HLT, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics.

M. Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The penn arabic treebank : Building a large-scale annotated arabic corpus. *NEMLAR Conference on Arabic Language Resources and Tools*.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 EMNLP*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas. Association for Computational Linguistics.

Jesse Mu and Jacob Andreas. 2020. Compositional explanations of neurons. *CoRR*, abs/2006.14032.

Hamdy Mubarak and Kareem Darwish. 2014. Using Twitter to collect a multi-dialectal corpus of Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 1–7, Doha, Qatar. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, United Kingdom. Leibniz-Institut für Deutsche Sprache.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, Florence, Italy. Association for Computational Linguistics.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. Analyzing Linguistic Knowledge in Sequential Model of Sentence. In *Proceedings of the 2016 EMNLP*, pages 826–835, Austin, Texas. Association for Computational Linguistics.

Ali Safaya. 2020. Arabic-albert, zenodo, 10.5281/zenodo.4718724.

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 Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Hassan Sajjad, Firoj Alam, Fahim Dalvi, and Nadir Durrani. 2021a. Effect of post-processing on contextualized word representations. *CoRR*, abs/2104.07456.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2021b. Neuron-level interpretation of deep NLP models: A survey. *CoRR*, abs/2108.13138.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Khan, and Jia Xu. 2022. In *Proceedings of the 2022 Conference of NAACL-HLT*, pages 3082–3101, Seattle, United States. Association for Computational Linguistics. [link].

Younes Samih, Mohamed Eldesouki, Mohammed Attia, Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak, and Laura Kallmeyer. 2017. Learning from relatives: Unified dialectal Arabic segmentation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 432–441, Vancouver, Canada. Association for Computational Linguistics.

Xavier Suau, Luca Zappella, and Nicholas Apostoloff. 2020. Finding experts in transformer models. *CoRR*, abs/2005.07647.

Reem Suwaileh, Mucahid Kutlu, Nihal Fathima, Tamer Elsayed, and Matthew Lease. 2016. Arabicweb16: A new crawl for today's arabic web. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 673–676, New York, NY, USA. Association for Computing Machinery.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of ACL*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Omar F. Zaidan and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182, Florence, Italy. Association for Computational Linguistics.

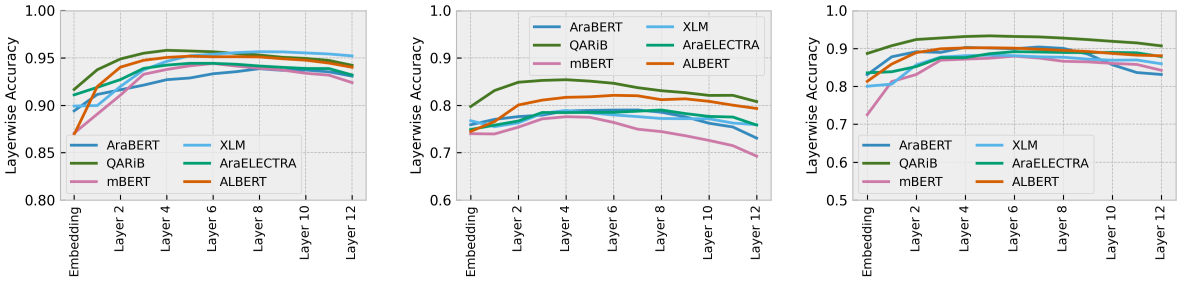
Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320.

A Appendix

Table 5 shows the performance loss for different thresholds. Highlighted thresholds were selected based on the 1% average performance loss. For the case of DIA, some overfitting is noticeable. Such case is reported in literature where the classifiers with large contextualized vectors tend to overfit when supervised data is insufficient (Hameed, 2018).

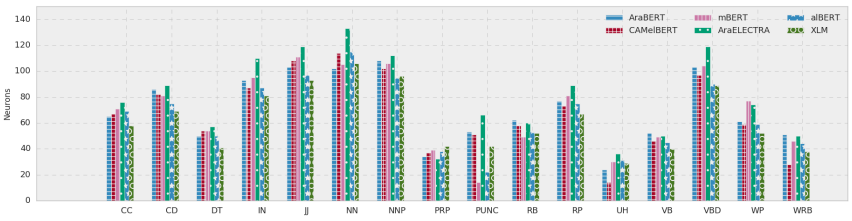
Task	Threshold	AraBERT	ArabicBERT	CAMelBERT	MARBERT	QARiB	mBERT	AraELECTRA	ALBERT	XLM
ATB	3.00%	0.914	0.915	0.929	0.916	0.924	0.924	0.868	0.935	0.938
	5.00%	0.934	0.940	0.949	0.945	0.950	0.941	0.912	0.947	0.953
	7.00%	0.939	0.949	0.957	0.952	0.957	0.945	0.934	0.953	0.957
	10.00%	0.943	0.953	0.960	0.957	0.961	0.947	0.945	0.954	0.960
	20.00%	0.945	0.956	0.960	0.958	0.962	0.948	0.954	0.953	0.961
	50.00%	0.940	0.953	0.955	0.954	0.958	0.941	0.957	0.948	0.955
	100.00%	0.937	0.954	0.957	0.955	0.955	0.938	0.954	0.947	0.953
CRS	3.00%	0.769	0.767	0.803	0.784	0.787	0.725	0.658	0.763	0.714
	5.00%	0.798	0.809	0.834	0.831	0.828	0.757	0.723	0.791	0.755
	7.00%	0.791	0.811	0.842	0.840	0.845	0.755	0.789	0.782	0.761
	10.00%	0.821	0.836	0.861	0.862	0.861	0.784	0.790	0.807	0.785
	20.00%	0.822	0.844	0.868	0.864	0.866	0.796	0.824	0.809	0.797
	50.00%	0.804	0.827	0.858	0.857	0.861	0.776	0.825	0.792	0.792
	100.00%	0.788	0.824	0.839	0.845	0.847	0.763	0.816	0.779	0.780
DIA	3.00%	0.753	0.780	0.798	0.766	0.783	0.732	0.683	0.779	0.753
	5.00%	0.774	0.812	0.835	0.809	0.820	0.748	0.747	0.808	0.767
	7.00%	0.788	0.831	0.847	0.830	0.834	0.757	0.776	0.815	0.783
	10.00%	0.793	0.833	0.851	0.842	0.836	0.775	0.794	0.818	0.790
	20.00%	0.794	0.840	0.857	0.850	0.851	0.768	0.809	0.814	0.806
	50.00%	0.784	0.832	0.840	0.844	0.847	0.752	0.814	0.798	0.799
	100.00%	0.770	0.818	0.831	0.826	0.829	0.734	0.803	0.790	0.776
DID	3.00%	0.829	0.876	0.879	0.879	0.885	0.809	0.840	0.864	0.833
	5.00%	0.854	0.892	0.897	0.907	0.908	0.821	0.868	0.881	0.860
	7.00%	0.860	0.901	0.910	0.916	0.917	0.832	0.882	0.885	0.865
	10.00%	0.872	0.905	0.914	0.918	0.920	0.837	0.887	0.892	0.878
	20.00%	0.880	0.908	0.917	0.922	0.923	0.846	0.900	0.893	0.878
	50.00%	0.876	0.902	0.909	0.915	0.915	0.840	0.906	0.888	0.871
	100.00%	0.864	0.892	0.896	0.903	0.903	0.823	0.906	0.877	0.858
GMR	3.00%	0.881	0.891	0.913	0.907	0.912	0.856	0.833	0.885	0.856
	5.00%	0.899	0.910	0.926	0.923	0.924	0.876	0.876	0.898	0.880
	7.00%	0.913	0.925	0.929	0.936	0.934	0.892	0.892	0.909	0.891
	10.00%	0.908	0.920	0.931	0.930	0.929	0.890	0.901	0.905	0.897
	20.00%	0.907	0.920	0.926	0.929	0.925	0.891	0.914	0.904	0.898
	50.00%	0.899	0.909	0.918	0.919	0.915	0.876	0.911	0.889	0.884
	100.00%	0.890	0.900	0.910	0.909	0.908	0.865	0.901	0.880	0.878

Table 5: Performance per models using different threshold δ

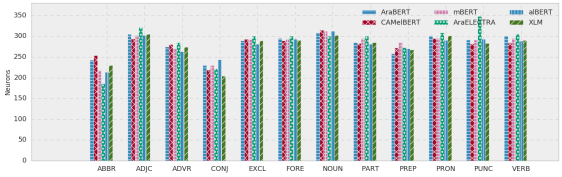


(a) ATB (b) DIA (c) GMR

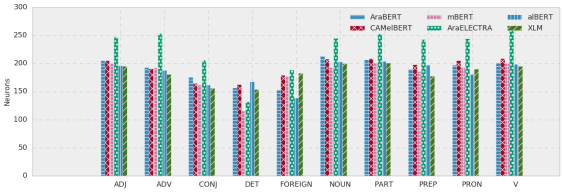
Figure 7: Layer-wise accuracy for ATB, DIA, GMR tasks.



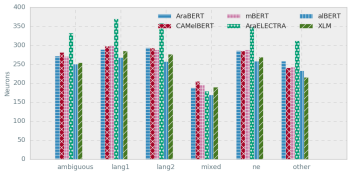
(a) ATB



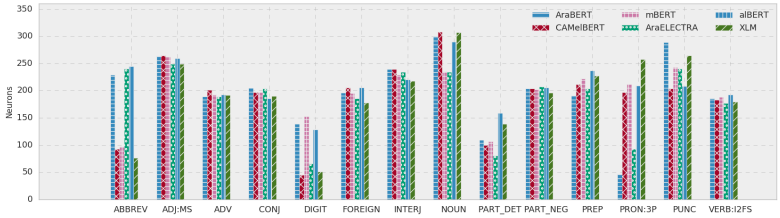
(b) CRS



(c) DIA

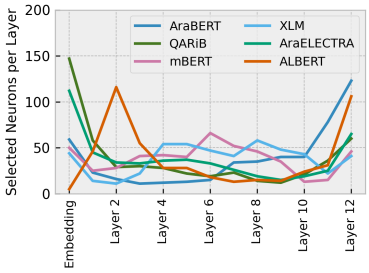


(d) DID

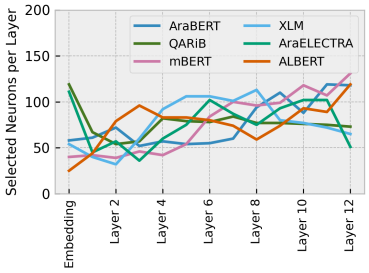


(e) GMR

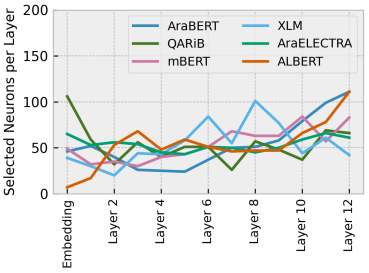
Figure 8: Distribution of neurons per property



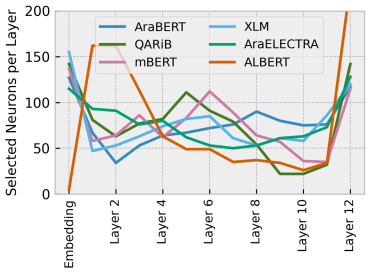
(a) ATB



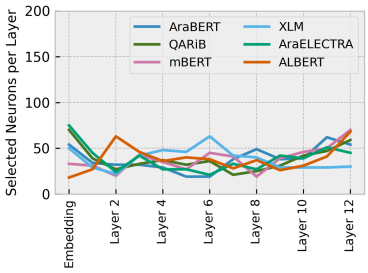
(b) CRS



(c) DIA



(d) DID



(e) GMR

Figure 9: Distribution of selected neurons across the layers