Assigment 1: ARABIC FINE-GRAINED DIALECT IDENTIFICATION

Step 1: Data Preprocessing:

We read the MADAR-Corpus both 26 and 6, and train and dev datasets. For all these files we follow the same procedure:

- We Split between sentences and labels.
- We use Camel Tools and some referenced functions for normalization, removing diacritics, punctuations and repeating characters to all sentences.
- We then apply Camel Tools simple Word tokenizer to the sentences set.
- We apply this process to both 26 and 6 and training and dev datasets.

Step 2: System Implementation:

1) Feature-Based Classification for Dialectal Arabic:

In this section we apply mainly two different feature-based classification methods (Word-gram and Character-gram features) by means of multiple experiments and variations. **We focus on MADAR-corpus 26 document for all cases in this section**:

1) Case 1: We fit a Tfidf Vectorizer for uni, bi and tri word-grams one at a time (one case uni, then other case bi, etc) with different number of maximum features. We apply the same vectorizer for both train and dev datasets.
2) Case 2: In this case we fit a Tfidf for a uni, bi and tri Word-gram all at once with different maximum number of features. We apply the same vectorizer for both train and dev datasets.
3) Case 3: We fit a Tfidf for 3, 4 and 5 Char-gram one at a time (one case tri, other 4, etc) with different number of max features and with Word Boundary considerations.
4) Case 4: We fit a Tfidf for the same as case 3, but without Word boundary considerations.
5) Case 5: We fit a Tfidf for a 3, 4 and 5 Char-gram but all at once with Word boundary consideration.
6) Case 6: Same as case 5 but without Word boundary consideration.

In all previous cases we apply the same vectorizer for both train and dev datasets and for different cases of maximum number of features.

We perform the multiclass clasiffication task with some classical machine learning algorithms that we have assesed have the best performance: Mainly Multilayer Perceptron and Random Forest.

We first show a table for the Multilayer Perceptrion in which we show the accuracy for different architectures. First number in brackets is the number of neurons in the first layer and second figure is the number of neurons in the second layer. Accuracy is shown for all 6 cases that were mentioned above (N-word, N-character with word boudary (wb) or without (No wb)).

	NN (100,)	NN(12,)	NN(12,2)
(1,1) Word	0,57057	0,397307	0,22
(2,2) Word	0,27269		
(1,2) Word	0,56		
(3,3) Word	0,09		
(1,3) Word	0,4525		
(2,2) Char wb	0,382		
(2,5) Char wb	0,5732		
(2,5) Char No wb	0,5444		
(2,2) Char No wb	0,394		

This is a preliminary study to filter the appropiate architecture and we decide that two layers do not improve performance and for one layer 100 neurons seems to work well as we did choose different number of neurons for one layer and 100 was approximately the best. We only report in the table the relevant cases for accuracy and the ones in blank are because they were much worse accuracies. To conclude we stick to a Neural Network with 100 neurons.

In the next table we look into more detail to the NN with 100 neurons architecture for all cases: (1,1) Word means uni-gran word, (3,3) means tri-gram word, (1,3) Word means from uni to tri gran word alltogether. Same logic for Character gran based with and without word boundary.

Cases	NN (100,)	Acc
1	(1,1) Word, max words = 20000	0,57057
2	(1,2) Word, max_words = None	0,56
3	(1,3) Word Max Word = 15000	0,51711
4	(2,2) Word Max Word = None	0,5532
5	(3,3) Word Max Word = None	0,09
6	(2,2) Char wb	0,36634
7	(1,3) Char wb	0,54307
8	(1,5) Char wb Max Word = None	0,55153
9	(2,2) Char No wb	0,3675
10	(1,3) Char No wb	0,5294
11	(1,4) Char No wb Max Word = None	0,5015

We end up with 11 cases, and next we report the evaluation metrics for the assigment from the MADAR-DIR-Scorer.py. We do not run this python file separately but we use the code inside this python file and pasted it into our own code for the assigment.

Next table shows MADAR-DIR-Scorer metrics for the previous 11 cases.

Cases	1	2	3	4	6
MACRO AVERAGE PRECISION SCORE	53,30%	58,45%	52,30%	56,05%	36,53%
MACRO AVERAGE RECALL SCORE	52,85%	57,92%	51,71%	55,33%	36,63%
MACRO AVERAGE F1 SCORE	52,94%	57,88%	51,74%	55,34%	36,39%
OVERALL ACCURACY	52,85%	57,92%	51,71%	55,33%	36,63%

7	8	9	10	11
48,95%	58,34%	36,70%	53,98%	50,62%
48,27%	55,15%	36,75%	52,94%	50,15%
47,47%	54,52%	36,50%	52,82%	49,25%
48,27%	55,15%	36,75%	52,94%	50,15%

We now show these metrics but for each label case:

	Case 1			Case 2			Case 3		
Labels	PRECISION SCORE	RECALL SCORE	F1 SCORE	PRECISION SCORE	RECALL SCORE	F1 SCORE	PRECISION SCORE	RECALL SCORE	F1 SCORE
ALE	56,65%	49,00%	52,55%	61,78%	48,50%	54,34%	57,80%	50,00%	53,62%
ALG	39,01%	35,50%	37,17%	50,31%	41,00%	45,18%	39,61%	41,00%	40,29%
ALX	57,21%	61,50%	59,28%	66,32%	64,00%	65,14%	63,35%	60,50%	61,89%
AMM	42,31%	44,00%	43,14%	53,66%	33,00%	40,87%	41,88%	33,50%	37,22%
ASW	46,24%	43,00%	44,56%	46,73%	50,00%	48,31%	45,30%	41,00%	43,04%

BAG	68,00%	68,00%	68,00 %	73,33%	77,00%	75,12 %	69,84%	66,00%	67,87 %
BAS	40,83%	44,50%	42,58 %	44,35%	51,00%	47,44 %	39,71%	41,50%	40,59 %
BEI	55,56%	52,50%	53,98 %	61,71%	54,00%	57,60 %	45,73%	45,50%	45,61 %
BEN	54,95%	55,50%	55,22 %	58,25%	56,50%	57,36 %	38,89%	59,50%	47,04 %
CAI	61,90%	58,50%	60,15 %	61,68%	66,00%	63,77 %	62,01%	55,50%	58,58 %
DAM	37,07%	38,00%	37,53 %	40,00%	49,00%	44,04 %	37,05%	46,50%	41,24 %
DOH	66,06%	54,50%	59,73 %	73,33%	60,50%	66,30 %	50,72%	52,50%	51,60 %
FES	44,94%	55,50%	49,66 %	50,23%	54,50%	52,28 %	48,15%	45,50%	46,79 %
JED	37,93%	38,50%	38,21 %	50,00%	46,50%	48,19 %	40,31%	39,50%	39,90 %
JER	61,49%	53,50%	57,22 %	58,42%	55,50%	56,92 %	52,80%	56,50%	54,59 %
KHA	59,05%	62,00%	60,49 %	69,57%	64,00%	66,67 %	66,86%	58,50%	62,40 %
MOS	60,50%	60,50%	60,50 %	66,33%	65,00%	65,66 %	58,59%	58,00%	58,29 %
MSA	57,78%	52,00%	54,74 %	56,35%	55,50%	55,92 %	48,69%	46,50%	47,57 %
MUS	46,41%	42,00%	44,09 %	42,19%	54,00%	47,37 %	53,05%	43,50%	47,80 %
RAB	50,48%	53,00%	51,71 %	59,38%	57,00%	58,16 %	51,92%	54,00%	52,94 %
RIY	53,14%	63,50%	57,86 %	64,53%	75,50%	69,59 %	65,12%	70,00%	67,47 %
SAL	37,97%	45,00%	41,19 %	47,75%	53,00%	50,24 %	42,78%	41,50%	42,13 %
SAN	61,58%	62,50%	62,03 %	61,76%	73,50%	67,12 %	51,84%	63,50%	57,08 %

SFX	51,60%	56,50%	53,94 %	56,11%	62,00%	58,91 %	59,14%	55,00%	56,99 %
TRI	78,33%	70,50%	74,21 %	79,68%	74,50%	77,00 %	62,95%	70,50%	66,51 %
TUN	58,92%	54,50%	56,62 %	65,99%	65,00%	65,49 %	65,77%	49,00%	56,16 %

Case 6			Case 7			Case 8		
PRECISION SCORE	RECALL SCORE	F1 SCORE	PRECISION SCORE	RECALL SCORE	F1 SCORE	PRECISION SCORE	RECALL SCORE	F1 SCORE
43,62%	41,00%	42,27 %	51,55%	50,00%	50,76 %	60,56%	76,00%	67,41 %
28,49%	26,50%	27,46 %	38,40%	24,00%	29,54 %	54,78%	31,50%	40,00 %
34,02%	41,00%	37,19 %	45,61%	52,00%	48,60 %	86,63%	74,50%	80,11 %
30,60%	28,00%	29,24 %	55,07%	19,00%	28,25 %	55,50%	55,50%	55,50 %
39,85%	26,50%	31,83 %	29,50%	56,50%	38,77 %	48,89%	66,00%	56,17 %
47,06%	48,00%	47,52 %	56,02%	67,50%	61,22 %	67,12%	24,50%	35,90 %
32,52%	26,50%	29,20 %	34,52%	48,50%	40,33 %	59,69%	57,00%	58,31 %
27,03%	30,00%	28,44 %	42,79%	47,50%	45,02 %	50,53%	71,50%	59,21 %
37,67%	40,50%	39,04 %	49,19%	60,50%	54,26 %	46,23%	46,00%	46,12 %
42,51%	44,00%	43,24 %	55,26%	52,50%	53,85 %	49,65%	70,00%	58,09 %
28,08%	20,50%	23,70 %	36,81%	26,50%	30,81 %	84,96%	48,00%	61,34 %
42,66%	46,50%	44,50 %	59,13%	61,50%	60,29 %	70,81%	74,00%	72,37 %

36,16%	32,00%	33,95 %	48,33%	29,00%	36,25 %	47,16%	54,00%	50,35 %
26,70%	25,50%	26,09 %	44,20%	30,50%	36,09 %	66,67%	56,00%	60,87 %
39,79%	38,00%	38,87 %	45,69%	61,00%	52,25 %	58,96%	79,00%	67,52 %
41,90%	37,50%	39,58 %	52,34%	61,50%	56,55 %	46,70%	42,50%	44,50 %
42,29%	48,00%	44,96 %	61,54%	52,00%	56,37 %	34,03%	49,00%	40,16 %
23,86%	31,50%	27,16 %	33,33%	36,00%	34,62 %	56,80%	71,00%	63,11 %
24,00%	24,00%	24,00 %	45,45%	22,50%	30,10 %	40,78%	57,50%	47,72 %
44,72%	44,50%	44,61 %	49,16%	58,50%	53,42 %	73,28%	42,50%	53,80 %
44,75%	49,00%	46,78 %	60,53%	69,00%	64,49 %	36,57%	56,50%	44,40 %
32,54%	27,50%	29,81 %	45,23%	45,00%	45,11 %	51,38%	28,00%	36,25 %
37,45%	44,00%	40,46 %	58,88%	58,00%	58,44 %	71,43%	20,00%	31,25 %
36,11%	39,00%	37,50 %	52,66%	54,50%	53,56 %	69,86%	51,00%	58,96 %
49,17%	59,50%	53,85 %	70,75%	75,00%	72,82 %	49,45%	67,50%	57,08 %
36,22%	33,50%	34,81 %	50,69%	36,50%	42,44 %	78,31%	65,00%	71,04 %

Case 10			Case 11		
PRECISION SCORE	RECALL SCORE	F1 SCORE	PRECISION SCORE	RECALL SCORE	F1 SCORE
52,68%	29,50%	37,82%	33,16%	31,00%	32,04%
43,90%	45,00%	44,44%	42,15%	47,00%	44,44%

50,23%	55,00%	52,51%	44,94%	60,00%	51,39%
42,41%	54,50%	47,70%	41,89%	46,50%	44,08%
59,12%	53,50%	56,17%	58,67%	57,50%	58,08%
54,21%	51,50%	52,82%	60,95%	32,00%	41,97%
69,54%	68,50%	69,02%	64,73%	67,00%	65,85%
45,99%	43,00%	44,44%	36,00%	22,50%	27,69%
33,53%	58,00%	42,49%	53,00%	26,50%	35,33%
44,74%	42,50%	43,59%	50,00%	40,50%	44,75%
70,00%	59,50%	64,32%	55,78%	55,50%	55,64%
41,82%	34,50%	37,81%	41,33%	31,00%	35,43%
48,00%	36,00%	41,14%	36,54%	28,50%	32,02%
74,10%	51,50%	60,77%	60,94%	58,50%	59,69%
41,10%	45,00%	42,96%	47,40%	36,50%	41,24%
47,00%	47,00%	47,00%	50,65%	39,00%	44,07%
52,57%	71,50%	60,59%	60,56%	64,50%	62,47%
53,23%	53,50%	53,37%	45,86%	61,00%	52,36%
66,84%	64,50%	65,65%	53,56%	71,50%	61,24%
65,26%	62,00%	63,59%	73,58%	39,00%	50,98%
58,38%	57,50%	57,93%	47,86%	61,50%	53,83%
62,81%	76,00%	68,78%	67,87%	75,00%	71,26%
49,42%	63,50%	55,58%	50,88%	57,50%	53,99%
50,64%	39,50%	44,38%	42,75%	57,50%	49,04%
60,27%	66,00%	63,01%	47,76%	80,00%	59,81%
65,75%	48,00%	55,49%	47,30%	57,00%	51,70%

Because all these tables contain a lot of information and is difficult to infer we show some statistics in the next table for the previous results.

We show the average of the Precision, Recall and F1 Scores for all previous cases and for each Dialect so we can infer which dialect is better classified. We also report the standard deviation, so we can infer which dialect is more sensitive to the specific case. Remember, we show again the cases in the next table:

1	(1,1) Word, max words = 20000
2	(1,2) Word, max_words = None
3	(1,3) Word Max Word = 15000
4	(2,2) Word Max Word = None
5	(3,3) Word Max Word = None
6	(2,2) Char wb
7	(1,3) Char wb
8	(1,5) Char wb Max Word = None
9	(2,2) Char No wb
10	(1,3) Char No wb
11	(1,4) Char No wb Max Word = None

Dialect	Average	std
ALE	49,32%	12%
ALG	39,03%	8%
ALX	57,21%	13%
AMM	41,81%	10%
ASW	47,72%	10%
BAG	57,56%	14%
BAS	49,32%	13%
BEI	45,58%	12%
BEN	47,98%	10%

CAI	53,47%	9%
DAM	45,49%	16%
DOH	53,70%	13%
FES	43,21%	9%
JED	46,55%	14%
JER	51,76%	10%
KHA	52,81%	10%
MOS	56,41%	9%
MSA	48,82%	12%
MUS	46,66%	14%
RAB	54,48%	9%
RIY	58,16%	10%
SAL	48,50%	15%
SAN	54,65%	13%
SFX	51,96%	8%
TRI	66,09%	10%
TUN	54,63%	12%

Form this table we can infer the TRI dialect is the best classified by far followed by the ALX. The worst classified are ALG by far. The rest are more or less in line with +/-10% in average of metrics scores. Dialects DAM and SAL are by far the most sensitive ones to each case (uni, bi word or char gran, etc), whereas ALG and SFX are the least sensitives.

Remember the average is calculated by summing for each dialect all the three metrics for all the eleven previous cases. The standard deviation is computed for this data aswell. The average gives an idea of how good is that dialect for classification purposes and the std gives an idea of how sensitive is the dialect accuracy to the choose of specific NLP configuration.

We now show the same results for the Random Forest Cases. Clearly the Random Forest is worse than the Neural Network overall but performs better for specific cases. We only report the relevant cases:

Cases	RF(100,50)	
12	(1,1) Word max words = 20000	0,41
13	(1,2) Word max words = 50000	0,4186
14	(1,3) Word Max Word = 50000	0,4163
15	(2,2) Word Max Word = None	0,175
16	(3,3) Word Max Word = None	0,0669
17	(1,2) Char wb	0,3479
18	(1,3) Char wb	0,4027
19	(1,5) Char wb Max Word = None	0,4265
20	(1,2) Char No wb	0,3419
21	(1,3) Char No wb	0,4038
22	(1,4) Char No wb Max Word = None	0,41

Cases	12	13	18	21
MACRO AVERAGE PRECISION SCORE	0,4768	0,4675	0,4036	0,4091
MACRO AVERAGE RECALL SCORE	0,41	0,4187	0,4027	0,4038
MACRO AVERAGE F1 SCORE	0,4252	0,4277	0,3978	0,4011
OVERALL ACCURACY	0,41	0,4187	0,4027	0,4038

	Architecture 12			Architecture 18			Architecture 21		
Labels	PRECISION SCORE	RECALL SCORE	F1 SCORE	PRECISION SCORE	RECALL SCORE	F1 SCORE	PRECISION SCORE	RECALL SCORE	F1 SCORE
ALE	42,77%	35,50%	38,80%	41,22%	30,50%	35,06%	41,84%	29,50%	34,60%
ALG	41,61%	57,00%	48,10%	43,48%	55,00%	48,57%	45,26%	52,50%	48,61%
ALX	61,83%	40,50%	48,94%	38,31%	38,50%	38,40%	33,33%	38,00%	35,51%
AMM	54,25%	57,50%	55,83%	49,78%	56,00%	52,71%	48,46%	55,00%	51,52%
ASW	53,29%	40,50%	46,02%	38,18%	42,00%	40,00%	42,79%	49,00%	45,69%
BAG	76,88%	61,50%	68,33%	64,48%	59,00%	61,62%	65,90%	57,00%	61,13%
BAS	43,46%	51,50%	47,14%	42,26%	50,50%	46,01%	38,33%	46,00%	41,82%
BEI	31,13%	47,00%	37,45%	34,20%	46,00%	39,23%	29,25%	37,00%	32,67%
BEN	45,08%	66,50%	53,74%	45,04%	63,50%	52,70%	47,48%	66,00%	55,23%
CAI	47,70%	41,50%	44,39%	36,59%	50,50%	42,44%	39,86%	55,00%	46,22%
DAM	14,13%	57,00%	22,64%	41,58%	42,00%	41,79%	45,76%	40,50%	42,97%
DOH	50,00%	47,50%	48,72%	46,67%	49,00%	47,80%	43,04%	51,00%	46,68%
FES	46,15%	27,00%	34,07%	40,82%	20,00%	26,85%	39,81%	21,50%	27,92%
JED	56,50%	50,00%	53,05%	54,55%	45,00%	49,32%	61,83%	40,50%	48,94%
JER	33,77%	26,00%	29,38%	33,33%	22,00%	26,51%	33,57%	24,00%	27,99%
KHA	34,97%	28,50%	31,40%	31,21%	27,00%	28,95%	25,93%	28,00%	26,92%
MOS	42,95%	33,50%	37,64%	33,47%	39,50%	36,24%	33,94%	37,50%	35,63%
MSA	35,47%	36,00%	35,73%	34,92%	33,00%	33,93%	35,60%	34,00%	34,78%
MUS	42,24%	24,50%	31,01%	33,56%	25,00%	28,65%	33,12%	26,00%	29,13%
RAB	54,26%	35,00%	42,55%	26,69%	35,50%	30,47%	27,20%	35,50%	30,80%
RIY	44,88%	28,50%	34,86%	35,53%	27,00%	30,68%	34,92%	33,00%	33,93%
SAL	61,58%	54,50%	57,82%	50,00%	49,50%	49,75%	54,64%	53,00%	53,81%
SAN	32,75%	28,00%	30,19%	32,45%	24,50%	27,92%	31,69%	29,00%	30,29%
SFX	58,04%	41,50%	48,40%	37,31%	37,50%	37,41%	39,34%	36,00%	37,60%
TRI	50,61%	41,50%	45,60%	34,48%	35,00%	34,74%	42,33%	34,50%	38,02%
TUN	59,22%	30,50%	40,26%	49,16%	44,00%	46,44%	48,52%	41,00%	44,44%

	Neural Network		Random Forest	
Dialect	Average	std	Average	std
ALE	49,32%	12%	36,64%	4,83%
ALG	39,03%	8%	48,90%	5,15%
ALX	57,21%	13%	41,48%	8,76%
AMM	41,81%	10%	53,45%	3,05%
ASW	47,72%	10%	44,16%	4,82%
BAG	57,56%	14%	63,98%	5,94%
BAS	49,32%	13%	45,22%	4,23%
BEI	45,58%	12%	37,10%	6,20%
BEN	47,98%	10%	55,03%	8,56%
CAI	53,47%	9%	44,91%	5,65%
DAM	45,49%	16%	38,71%	12,73%
DOH	53,70%	13%	47,82%	2,31%
FES	43,21%	9%	31,57%	9,11%
JED	46,55%	14%	51,08%	6,31%
JER	51,76%	10%	28,51%	4,34%
KHA	52,81%	10%	29,21%	2,85%
MOS	56,41%	9%	36,71%	3,12%
MSA	48,82%	12%	34,83%	1,00%
MUS	46,66%	14%	30,36%	5,54%
RAB	54,48%	9%	35,33%	8,63%
RIY	58,16%	10%	33,70%	5,16%
SAL	48,50%	15%	53,84%	4,00%
SAN	54,65%	13%	29,64%	2,62%
SFX	51,96%	8%	41,46%	7,26%
TRI	66,09%	10%	39,64%	5,79%
TUN	54,63%	12%	44,84%	7,78%

In the previous statistics table we report both NN and Random Forest for comparison. Clearly the RF is much less sensitive to the specific case (NLP configuration) than the Neural Net as can be seen by the much lower std. On the other hand the average score is lower in the RF which infers that Neural Nets are better for the dialect classification task. In the RF case Dialect BAG has the best performance for classification by far with 63.98%, and specifically with a 76.88% in Precision Score (NN has 58% on BAG), very close to the 66.09% of the Neural Net on TRI, specifically Neural Net has a 78.66% on TRI in Precision Score on case 1 (RF has 39,64% on TRI). RF has the poorest performer in SAN and KHA.

CASE 6 DIALECTS:

We do not report evaluation metrics for the 6 dialect case as we have focused on the 26 case which is much more challenging. Anyway we show a couple of cases for the 6 dialect case to show that it is a much easier classification problem and therefore you can obtain much better evaluation metrics.

We show accuracies for the Random Forest cases, same as described above for all the 26 dialect cases but now for 6 dialects.

Architectures	RF(100,50)	
12	(1,1) Word max words = 20000	0,773
13	(1,2) Word max words = 50000	0,786
14	(1,3) Word Max Word = 50000	0,746
15	(2,2) Word Max Word = None	0,68
16	(3,3) Word Max Word = None	0,15
17	(1,2) Char wb	0,68
18	(1,3) Char wb	0,79
19	(1,5) Char wb Max Word = None	0,81
20	(1,2) Char No wb	0,69
21	(1,3) Char No wb	0,82
22	(1,4) Char No wb Max Word = None	0,85

2) LSTM Deep Network

In this section we use an LSTM Network archtecture with AraVec pre-trained Word embeddings. We also focus on different experiments for this setup. We focus on MADAR Corpus 26 for these experiments. We have tested the difference between using the tokenizer from keras and applying the same preprocessing techniques from previous section (normalization, removing repeated words, diacritics, punctuations, single word tokrnizer from Camel Tools) and there is only 0.063% different tokens with one or the other so the difference is negligible and we stick to the one with Keras for this section.

Important: In all cases, 80% of MADAR_Corpus_26_train is for training and 20% is for validation. **Full MADAR_Corpus_26_dev is for test**

Case 1:

We first focus on "full_grams_cbow_300_twitter.mdl" , we Split the data in 80/20 for training and validation sets and we use the MADAR Corpus 26 Dev Dataset as our test set for performance evaluation. So in this case the embedding vectors have 300 dimensions.

We use the following architectures and variations for the experiments:

Architectures	LSTM	Dense (ReLu)	Dense (Softmax)
1	64 + 0.1 Dropout	32	26
2	64 + 0.1 Dropout	32	26
3	64 + 0.1 Dropout	32	26
4	64 + 0.1 Dropout	32	26
5	200 + 0.3 Dropout	100	26
6	300 LSTM + Dropout(0,3) + 200 LSTM + Dropout (0,3)	150 + Dropout (0,2)	26
7	Bidirectional(300 LSTM) + Dropout(0,3) +Bidirectional (200 LSTM) + Dropout (0,3)	150 + Dropout (0,2)	26

Architectures	Max Sequence Length	Max N Words	Epoch	batch size
1	max	50000	15	2000
2	max	50000	20	32

3	max	50000	15	300
4	10	20000	15	300
5	20	20000	15	300
6	10	20000	15	300
7	10	20000	15	300

We show the mecris from MADAR-DIR-Scorer in the same fashion as the previous section for the relevant cases (similar cases with almost same accuracy are omitted).

Architectures	4	5	6	7
MACRO AVERAGE PRECISION SCORE	44,95%	47,94%	47,70%	47,56%
MACRO AVERAGE RECALL SCORE	44,38%	46,79%	47,15%	47,06%
MACRO AVERAGE F1 SCORE	44,21%	46,89%	47,06%	47,03%
OVERALL ACCURACY	44,38%	46,79%	47,15%	47,06%

	Architecture 4			Architecture 5			Architecture 6		
Labels	PRECISION SCORE	RECALL SCORE	F1 SCORE	PRECISION SCORE	RECALL SCORE	F1 SCORE	PRECISION SCORE	RECALL SCORE	F1 SCORE
ALE	44,50%	44,50%	44,50 %	60,87%	42,00%	49,70 %	55.28 %	44.50 %	49.31 %
ALG	63,22%	55,00%	58,82 %	55,05%	60,00%	57,42 %	49.80 %	62.50 %	55.43 %
ALX	37,86%	39,00%	38,42 %	32,32%	48,00%	38,63 %	34.34 %	45.50 %	39.14 %
AMM	27,39%	31,50%	29,30 %	37,78%	34,00%	35,79 %	38.41 %	31.50 %	34.62 %
ASW	48,84%	31,50%	38,30 %	40,10%	38,50%	39,29 %	46.41 %	48.50 %	47.43 %

BAG	44,91%	48,50%	46,63 %	59,44%	42,50%	49,56 %	48.10 %	50.50 %	49.27 %
BAS	42,05%	55,50%	47,84 %	47,96%	53,00%	50,36 %	54.80 %	48.50 %	51.46 %
BEI	40,96%	51,00%	45,43 %	63,16%	48,00%	54,55 %	66.42 %	44.50 %	53.29 %
BEN	48,40%	45,50%	46,91 %	53,05%	56,50%	54,72 %	43.18 %	47.50 %	45.24 %
CAI	33,17%	34,00%	33,58 %	42,50%	34,00%	37,78 %	36.45 %	37.00 %	36.72 %
DAM	45,38%	29,50%	35,76 %	41,61%	33,50%	37,12 %	42.51 %	35.50 %	38.69 %
DOH	41,71%	39,00%	40,31 %	36,32%	42,50%	39,17 %	48.99 %	36.50 %	41.83 %
FES	51,95%	60,00%	55,68 %	52,97%	62,50%	57,34 %	49.62 %	65.50 %	56.47 %
JED	38,78%	38,00%	38,38 %	35,18%	44,50%	39,29 %	42.08 %	42.50 %	42.29 %
JER	36,95%	37,50%	37,22 %	34,18%	40,50%	37,07 %	37.90 %	41.50 %	39.62 %
KHA	42,86%	58,50%	49,47 %	58,33%	56,00%	57,14 %	50.68 %	56.00 %	53.21 %
MOS	56,81%	60,50%	58,60 %	72,29%	60,00%	65,57 %	69.31 %	65.50 %	67.35 %
MSA	47,39%	54,50%	50,70 %	52,94%	63,00%	57,53 %	58.76 %	57.00 %	57.87 %
MUS	37,28%	31,50%	34,15 %	41,28%	35,50%	38,17 %	35.75 %	37.00 %	36.36 %
RAB	60,61%	40,00%	48,19 %	58,11%	43,00%	49,43 %	53.90 %	38.00 %	44.57 %
RIY	32,23%	39,00%	35,29 %	41,49%	39,00%	40,21 %	38.14 %	41.00 %	39.52 %
SAL	39,04%	28,50%	32,95 %	34,82%	39,00%	36,79 %	39.13 %	36.00 %	37.50 %
SAN	46,44%	55,50%	50,57 %	59,54%	51,50%	55,23 %	51.92 %	54.00 %	52.94 %

SFX	55,85%	52,50%	54,12 %	42,14%	59,00%	49,17 %	52.70 %	58.50 %	55.45 %
TRI	53,22%	45,50%	49,06 %	50,00%	51,50%	50,74 %	46.61 %	55.00 %	50.46 %
TUN	50,79%	48,00%	49,36 %	43,82%	39,00%	41,27 %	48.94 %	46.00 %	47.42 %

	Architecture 7		
Labels	PRECISION SCORE	RECALL SCORE	F1 SCORE
ALE	54,30%	50,50%	52,33%
ALG	62,86%	55,00%	58,67%
ALX	41,14%	36,00%	38,40%
AMM	36,31%	32,50%	34,30%
ASW	43,75%	49,00%	46,23%
BAG	47,78%	48,50%	48,14%
BAS	50,00%	49,50%	49,75%
BEI	45,24%	47,50%	46,34%
BEN	55,69%	46,50%	50,68%
CAI	42,38%	32,00%	36,47%
DAM	39,51%	32,00%	35,36%
DOH	47,59%	39,50%	43,17%
FES	52,82%	51,50%	52,15%
JED	44,68%	42,00%	43,30%
JER	32,90%	50,50%	39,84%
KHA	57,92%	53,00%	55,35%
MOS	58,60%	63,00%	60,72%
MSA	54,08%	63,00%	58,20%
MUS	33,07%	41,50%	36,81%
RAB	49,78%	55,50%	52,48%
RIY	38,02%	46,00%	41,63%

SAL	37,99%	34,00%	35,88%
SAN	61,68%	51,50%	56,13%
SFX	47,84%	55,50%	51,39%
TRI	50,70%	54,00%	52,30%
TUN	50,00%	44,00%	46,81%

From previous tables we can infer that architectures with multiple LSTM layers and greater number of neurons like architecture 6 or adding Bidirecctional LSTM like in architecture 7 do not have substantial increase in performance in comparison with simple architectures consisting on one LSTM layer and fewer neurons such architecture 4 or 5 (just 2% increase in accuracy).

We now show statistics as usual:

Dialects	**Average**	**std**
ALE	6,0%	49,24%
ALG	3,2%	58,45%
ALX	4,2%	38,86%
AMM	3,4%	33,21%
ASW	5,7%	41,72%
BAG	4,7%	48,44%
BAS	3,7%	49,55%
BEI	6,5%	49,13%
BEN	4,2%	50,88%
CAI	3,9%	36,21%
DAM	4,9%	36,64%
DOH	3,2%	41,03%
FES	4,0%	55,21%
JED	3,3%	40,46%
JER	5,1%	38,52%
KHA	5,2%	54,29%
MOS	4,7%	61,79%
MSA	5,3%	55,70%

MUS	3,4%	36,58%
RAB	6,7%	50,79%
RIY	3,9%	39,21%
SAL	3,4%	35,44%
SAN	4,7%	54,23%
SFX	5,0%	51,95%
TRI	2,5%	50,78%
TUN	4,1%	45,89%

We can infer that in this case all dialects have low sensitivity to model architecture in that the standard deviation reported for different architectures and all dialects is low. Same as in Random Forest case and much lower than Multilayer Perceptron.

Overall we do not see an increase in performance for LSTM architectures in comparison with Multilayer Perceptron Architectures. And on specific dialects we see MOS performing the best on average with 61.79% and AMM, CAI and DAM perfoming the worst.

Case 2:

For this case we apply the same procedure as in case 1 except that we use "full_uni_cbow_300_twitter.mdl". Because the difference is negligible with case 1 we just report the MADAR-DIR-Scores Metrics.

Architectures	4	5	6	7
MACRO AVERAGE PRECISION SCORE	44,38%	47,94%	48,13%	47,56%
MACRO AVERAGE RECALL SCORE	43,90%	46,79%	47,73%	47,06%
MACRO AVERAGE F1 SCORE	43,79%	46,89%	47,47%	47,03%
OVERALL ACCURACY	43,90%	46,79%	47,73%	47,06%

Case 3:

Same as before but we use 'full_grams_cbow_100_twitter.mdl' for embeddings with 100 dimensional vectors this time.

We only report the relevant cases to make a point. Comclusion is that 300 dimension embedding is better in terms of performance. We just use the following architectures:

Architectures	**LSTM**	**Dense (ReLu)**	**Dense (Softmax)**
1	64 + 0.1 Dropout	32	26
4	64 + 0.1 Dropout	32	26
6	300 LSTM + Dropout(0,3) + 200 LSTM + Dropout (0,3)	150 + Dropout (0,2)	26

Architectures	**Max Sequence Length**	**Max N Words**	**Epoch**	**batch size**
1	max	50000	15	2000
4	10	20000	15	100
6	10	20000	15	100

We obtain the following training, validation and test accuracies (remember in all cases we have used 80% of MADAR_Corpus_26_Train for Training and 20% for Validation, and full MADAR_Corpus_26_dev for test).

Architectures	**Train Acc**	**Val Acc**	**Test Acc**
1	0,7272	0,3444	0,389
4	0,5457	0,3399	0,37961
6	0,8033	0,3916	0,4794

We see an overall decrease in accuracy for all cases in comparison with 300 dimensional embeddings.

CASE 6 DIALECTS:

Now we want to show a specific case of the 6 dialects case (MADAR_Corpus_6) and see how the evaluation metrics are much better for this problem than the 26 dialect case. It makes sense because the multilabel classification is much harder with more than 4 times the number of classes. We have decided to focus on the 26 case as is more challenging and we therefore only show the performance for the 6 dialect case for the following specific architecture:

Architectures	**LSTM**	**Dense (ReLu)**	**Dense (Softmax)**	**Train Acc**	**Val Acc**	**Test Acc**
6	300 LSTM + Dropout(0,3) + 200 LSTM + Dropout (0,3)	150 + Dropout (0,2)	26	0,9617	0,8222	0,83716

Architectures	6
MACRO AVERAGE PRECISION SCORE	83,84%
MACRO AVERAGE RECALL SCORE	83,72%
MACRO AVERAGE F1 SCORE	83,74%
OVERALL ACCURACY	83,72%

	Architecture 6		
Dialects	PRECISION	RECALL	F1 SCORE
BEI	79.31 %	82.40 %	82.40 %
CAI	82.07 %	81.00 %	81.00 %
DOH	78.76 %	81.60 %	81.60 %
MSA	85.77 %	89.20 %	89.20 %
RAB	88.24 %	86.30 %	86.30 %
TUN	88.91 %	81.80 %	81.80 %

We can see how in the 26 dialect case we could not reach more than 55% at most accuacy for all casses with LSTM, whereas in this case with the LSTM architecture 6 we reach a 84% test accuracy which is quite good. We do not report the rest of the cases for 6 dialects but all cases have a huge improvement in evaluation metrics when passing from 26 Dialect problem to 6 dialect problem.

3 BERT:

For this section we apply BERT with a pretrained model from transformers: “asafaya/bert-base-arabic” which is a model from github araBERT specifically trained on arabic language.

We have encountered many issues in terms of memory errors both for CPU and GPU processing and base don research on the internet this ussually happens with BERT as it is a huge consumption model. For that reason we needed to cut the number of training instances, which is not a big deal as the model is already pretrained but, on the other hand we had to cut the number of instances labels to match the training instances and not all 26 dialects from the MADAR_Corpus_26 appeared on training and validation. This caused some of the dialects to have zero accuracy and also to reduce overall accuracy.

We apply the following combination of hyperparameters for the BERT architectures:

N of instances is the number of training instances we feed to the pretrained model for fine tunning. In all these cases the validation and test sets are a portion of the MADAR_Corpus_dev and the training instances are picked from MADAR_Corpus_Train. Training and Validation set sizes are proportional to the original MADAR_Corpus Train and dev dcouments.

Architectures	Documents	N of instances	Max_length_padding	batch_size	epochs	Adam Learning
1	MADAR 26	1000	20	20	10	1,00E-06
2	MADAR 6	2000	20	100	10	1,00E-07
3	MADAR 6	5000	20	100	3	1,00E-07
4	MADAR 26	8000	20	100	3	1,00E-07

Max_length_padding is the maximum length of each sentence to be feed to the BERT model.

Table below we show MADAR_DIR_Scores for the 4 architectures and we specify on which dataset we run them on the Number of Dialects row (Either MADAR_Corpus_26 or 6)

N of Dialects (MADAR_Corpus)	26	6	6	26
Architectures	**1**	**2**	**3**	**4**
MACRO AVERAGE PRECISION SCORE	4,00%	7,76%	14,22%	2,67%
MACRO AVERAGE RECALL SCORE	20,00%	15,63%	15,22%	5,39%
MACRO AVERAGE F1 SCORE	6,67%	10,09%	9,86%	2,42%

OVERALL ACCURACY	20,00%	15,95%	14,80%	6,83%

We show the evaluation metrics per dialect for the 6 dialect architecture cases:

	2			3		
Dialects	PRECISION SCORE	RECALL SCORE	F1 SCORE	PRECISION SCORE	RECALL SCORE	F1 SCORE
BEI	0,0%	0,0%	0,0%	10,5%	2,7%	4,3%
CAI	0,0%	0,0%	0,0%	14,3%	66,7%	23,5%
DOH	16,8%	50,00%	25,19%	25,7%	9,28%	13,64%
MSA	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
RAB	18,5%	28,53%	22,45%	8,8%	6,10%	7,19%
TUN	11,2%	15,27%	12,93%	26,1%	6,59%	10,53%

Finally, we show the architecture 4 evaluation metrics per dialect for the 26 diealects. We can see that we do not have all of the 26 dialects on the table because the training and dev sets have been reduce to cope with the memory issues and therefore there are not cases on the training and validation for those dialects.

	Architecture 4		
Dialects	PRECISION SCORE	RECALL SCORE	F1 SCORE
ALE	6,59%	19,00%	9,78%
ALG	0,00%	0,00%	0,00%
ALX	0,00%	0,00%	0,00%
AMM	0,00%	0,00%	0,00%
BAS	0,00%	0,00%	0,00%
BEN	0,00%	0,00%	0,00%
CAI	0,00%	0,00%	0,00%

DAM	2,50%	0,50%	0,83%
FES	5,66%	3,00%	3,92%
JED	0,00%	0,00%	0,00%
MOS	7,81%	62,50%	13,89%
MUS	10,00%	0,50%	0,95%
RAB	9,43%	5,00%	6,54%
SAL	6,03%	9,50%	7,38%
TRI	2,75%	2,50%	2,62%

4 Conclusion:

These conclusions are applicable for the Multiclass Clasification problem for 26 dialects (MADAR_Corpus_26). Other conclusions are applicable for the 6 dialects problem which is much easier problem. We have decided to focus mainly on the 26 dialects case and report all the metrics for this specific case as it is the challenging one. However, evaluation metrics for the 6 dialect case are much higher on average and per dialect for every case evaluated in this report. So all evaluation metrics are 20 to 30% higher for all cases for the 6 dialect case.

For the 26 Dialect:

We can conclude that our models did not give very good results in terms of performance evaluation metrics if we compare to the same NLP task on other languages such as English, or if we compare to other NLP Tasks or Machine Learning applications such as Computer Vision. However, if we compare with the state-of-the-art performance on Arabic Dialect Classification we can see that we are not that bad comparing to L. Lulu and A. Elnagar (1) were they report a best test accuracy of 70% and we do get closet o 60% on some models overall and more than 70% on few dialects.

With this in mind, it is true that can be improved and we do not think is a problem of the modelling part of the assigment but of the preprocessing stage. Whereas we have implemented many of the standard proprocessing techniques, we think that if we apply further techniques such as steamming and lemmantization together with more elaborated techniques we can reach easily the state-of-the-art performance. And if more advanced

techniques are applied by native arabic speakers they would be able to improve that state-of-the-art performance.

On the other hand, the BERT section poor perfromance is not only due to the preprocessing stage that could be improve, but also on the fact that it is a very consuming method in terms of CPU and GPU memory and this had made for us imposible to feed the model with large amounts of data or use large extensions of BERT models.

Having said this though, we can also say that a 26 dialect multiclass classification problem is quite hard, because if we focus on the 6 dialect case, which is the one that most of the literature focus on, we can see accuracies from 70% to 90% for the test set which is in line with what we expect.

BIBLIOGRAPHY:

1) Leena Lulu, Ashraf Elnagar, Automatic Arabic Dialect Classification Using Deep Learning Models, Procedia Computer Science, Volume 142, 2018, Pages 262-269, ISSN 1877-0509,