Deep Learning Model for Identifying the Arabic Language Learners based on Gated Recurrent Unit Network

Seifeddine Mechti[1]
MIRACL Laboratory
Sfax University,
Sfax,Tunisia

Roobaea Alroobaea[2]
College of
Computers & Information
Technology
Taif University, Taif,
Saudi Arabia

Moez Krichen[3]
Faculty of CSIT,
AlBaha University,
AlBaha, Saudi Arabia
ReDCAD Laboratory,
Sfax University,
Sfax, Tunisia

Saeed Rubaiee[4], Anas Ahmed[5]
Department of
Industrial & Systems
Engineering,
College of Engineering,
University of Jeddah,
Jeddah, Saudi Arabia

***Abstract*—This paper focuses on identifying the Arabic Language learners. The main contribution of the proposed method is to use a deep learning model based on the Gated Recurrent Unit Network (GRUN). The proposed model explores a multitude of stylistic features such as the syntax, the lexical and the n-grams ones. To the best of our awareness, the obtained results outperform those obtained by the best existing systems. Our accuracy is the best comparing with the pioneers (45% vs 41%), considering the limited data and the unavailability of accurate tools dedicated to the Arabic language.**

***Keywords*—Arabic; Native Language Identification (NLI); deep learning; Gated Recurrent Unit Network (GRUN)**

I. Introduction

Technological progress and the unprecedented sharing of resources on the Internet has generated a huge number of documents on the web and especially on social networks. These documents and / or publications belong to different author profiles. Unfortunately, many Internet users do not reveal their real identity and give false information regarding their age, sex, nationality, level of education, mother tongue, etc. For this, several works have been interested in identifying the source of information.

In fact, in the commercial sector there is a need to know the age, gender, origin, and other details in order to offer potential buyers' products that are suitable to their profiles. Also, the products should be offered to them in perfect harmony with their preferences and moods. Furthermore, the origins of clients from their texts and their languages should be known. In this same framework, our work aims to detect the mother tongue of users.

Another application of mother tongue detection is the educational field. Indeed, for the learners of a given language one needs to know the level of mastery of the language in order to classify them into different learning groups corresponding to different levels of education. For example, for learners of the Arabic language, three levels of learning can be used, which are non-native learners, medium learners and native learners.

This article are interested in the detection of the mother tongue of the authors for learners of the Arabic language. This task is part of computational linguistics. We have based on the series of experimentation on the Gated Recurrent Unit (GRU) model. Our model contributes to overcoming the limitations of RNN.

This paper is divided into five sections. Section II is given a short overview about related works. Section III discusses our Baseline approach of ANLI. Section IV presents our new deep learning approach based on GRU. Finally, concluding remarks are detailed and upcoming outlines of research are provided in Section VI.

II. Related Work

Nowadays, other languages apart from English language [1] paying attention to researchers in order to evaluate the applicability of Natural Language Interaction (NLI) methods to other languages [2].

To the best of our knowledge, Malmasi and Drass [3] and Lan and Hayato [4] focused on the Chinese language. The former research proposed a system that introduced the first expansion of Natural Language Interaction applied to non-English data using a set of features such as "n-grams", "part-of-speech tags", "context-free grammar production rules" and "function words". The system found that the adoption of integrated features surpassed the employ of single features with 70.61% precision.

In [4], the authors resort to "skip-grams" as to solve "Natural Language Interaction" problem using lexical attributes built on JCLC ("Jinan Chinese Learner Corpus"). As the dimension of the "skip-gram" function increases tremendously, they decide to take as informative features "n-grams" with 10 occurrences. A simple example is proposed in Fig. 1.

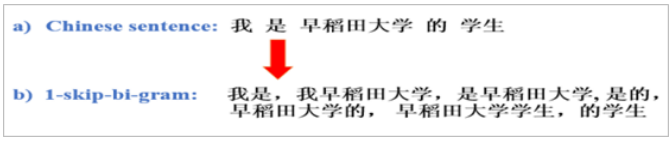

Fig. 1. An Example of a Chineese Sentence.

Unlike most of the Natural Language Interaction researches which used TF ("term frequency") or TF-IDF ("term frequency–inverse document frequency"). The big advantage of this analysis is that careful consideration is given to assigning each function the appropriate weight. They followed the "BM25 term-weighing" process [5]. Using hierarchical "linear SVM" classifiers, their proposed method achieved a higher score with 75% per cent accuracy.

Additionally, other languages were considered such as the Finnish and the Norwegian languages in [6], [7], the potuguese [8] and the indien[9]. These works aimed to identify if the NLI methods earlier used in level two English can be effective to other languages. Their findings provided encouraging signs that the NLI strategies are applicable to other languages.

A. Arabic Native Language Identification

Arabic is generally viewed as a language that is vital and of strategic use. However, the work [10] by Malmasi and Dras is the first experience which deals with Arabic. Their objective was to examine the utility of syntactic characteristics, primarily "CFG development laws", "Arabic function words", and "n-grams part-of-speech". They used a controlled approach to classification of multiple classes. As a result, these studies appeared to be effectively usable for "Arabic NLI".

In addition, it is noteworthy that merging features resulted in a fair precision of approximately 41%. That was attributed, first, to the reason that Arabic's morphological and syntactic diversity varies substantially from English and, on the other side, to the compact size of the dataset that is used in learning process.

B. NLI Shared Task

The growing interest in the NLI field reflected by a number of papers that have been published motivated research groups to organize shared Tasks [11] (to our knowledge this the first and the only shared task). The key goal of the mission was to further homogenise the group and support the field advance by creating a favorable framework for direct comparison of the systems.

In this task, 29 teams from different countries participated and 24 teams were elected to write papers describing their systems. These 24 teams competed across three different subtasks. The same test set of data was used for each task. Only the training data changed from a task to another. The teams developed systems trained on Data compiled from the TOEFIL11 corpus only, from External corpora and from both, respectively in the closed-task, open1-task and open2-task.

The teams were free to choose the convenient learner methods and features. Based on the report of [11], it is observed that "word", "character" and "POS n-gram" features were the most common features (see Table I). Unsurprisingly, "Support Vector Machines" was the most used among other machine learning algorithms.

C. Gated Recurrent Unit (GRU)

This technique is similar to "Long Short-Term Memory" (LSTM) [12]. It was proposed by [13]. GRU was developed to solve the problem of short-term memory. It has two gates

TABLE I. COMMON FEATURES ADOPTED IN THE SHARED TASK.

Feature	Type	# of teams
Character N-Grams	N between 1 and 9	16
POS N-Grams	N between 1 and 5	15
Word N-grams	N between 1 and 5	18
Function N-grams		2
Syntactic Features	TSG, Dependencies, Adaptor Grammars, Productions	6

which are "update gate" and "reset gate" to control the memory flow as shown in Fig. 2. It uses the memory to store the value for a certain amount of time and at a critical point dragging that value out and utilizing it with the present state to update at a future date. To sum up, it has less tensor operations than LSTM. To some extent, it is a slight quicker to train than LSTM [14], [15].

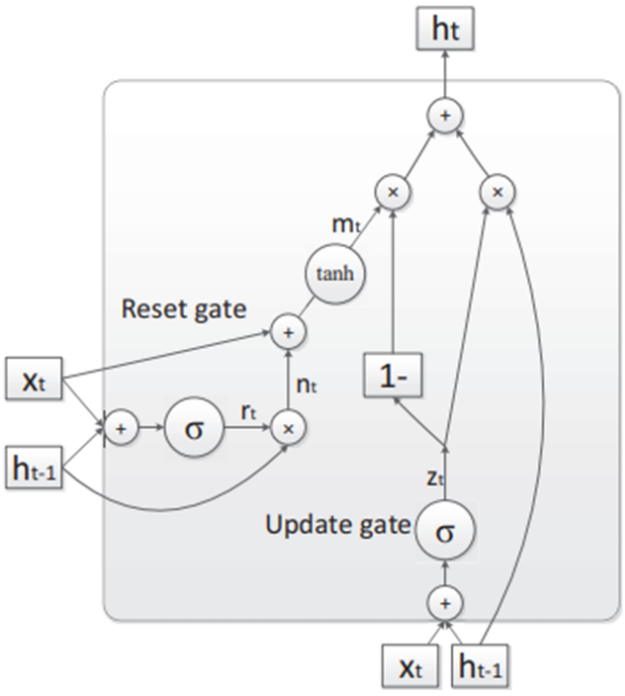

Fig. 2. Update Gate and Reset Gate in GRU [14].

III. METHODOLOGY

This paper aims to predict Arabic learners' native language as encouraged by the work of [16]. To get the best classification model, the feature-selection step was used as it was not given great attention in most of the aforementioned studies. The best classification model refers to improve the performance and reduce the features.

Three stages were adopted in the proposed methods here to get the best classification model. The first stage is pre-processing. The text that will be used in the next stage was well-prepared. The second stage is feature-extraction. The set of features that seem to be useful will be extracted for level one learners' background discrimination. The final stage is a classification algorithm that will be applied to build the classification model. Noticeably, the last two stages are achieved by a sub-stage of feature selection.

A. Text Pre-Processing

This stage aims to prepare for the deep Learning (DL) stage. Indeed, the texts are written by non-Arab individuals from all over the world, who studies in the Kingdom Saudi Arabia the Arabic language. Analysis of the texts has shown that there are several inconsistencies and many errors in the corpus. Words, characters and URLs appear in the texts as shown in the example given in Fig. 3. However, notes are inserted in the texts during the transcription.

Note	Meaning
كلمة غير موجودة (indefinite word)	Indicates that the considered word does not exist in the Standard Arabic language.
معلومة شخصية محذوفة (Personal information Deleted)	Indicates that some personal data concerning the text's author was deleted (e.g. learner's name, contacts, etc.).

Fig. 3. Notes and Corresponding Meanings.

In this case, deleting these annotations can change the structure of the sentences. This explains why these notes were treated case by case by replacing them with suitable words to keep a good syntactic structure of the sentence.

In the extreme case where we do not find an appropriate word, so it was deleted. Once the text pre-processing phase is achieved (i.e. the corpus data are transformed into usable data), the texts are ready for the next phase where features will be extracted from them (as shown in Fig. 4).

B. Extraction and Selection of Features

Three syntactic feature categories were discovered: POS n-grmas, function words and production rules. Thus, three collections of features were generated for each text. For every individual feature, frequency (TF) was calculated.

a) Function words: Namely empty words, In this study, 411 common Arabic function words were adopted and classified into 17 types. Fig. 5 shows examples of the Arabic function words listed by types.

b) POS n-grams: These features highpoint the words' linguistic class. The tagger was applied to assign the grammatical category for each word.

c) Production Rules: This terminology define both the syntactic class of the "words" and "sentences" structures. Fig. 6 shows some production rules extracted from the corresponding parse tree of a given sentence.

The first production rule "S $\longrightarrow$ VP | VB NP" indicates that the sentence (S) is constituted by a verb phrase (VP) or by verb phrase (VP) followed by noun phrase (NP). The second rule indicate that Verb phrase (VP) is constituted by verb followed by prepositional phrase introduced by a subordinating (SBAR) conjunction and so on. The Arabic syntactic tag set is slightly different from the tagset used for English given the major differences between the two syntactic systems. The full list of syntactic tags used in this study is detailed [17].

Since the rules extracted are often errored and not acceptable by Arabic syntax, we think of how to decide if a rule is valid or not. The solution is to compare it with an existed list of rules that we know that it is correct in advance.

Thus, we use a base of rules extracted from the Penn Arabic Treebank (ATB) by [18] which is a collection of text gathered from the Lebanese newspaper An-Nahar. While ATB texts are written by Journalists specialized in editing, we assume that they respect the Arabic syntax rules and so on we accept only rule appear in that base and throw away the rest.

For the step of feature selection, the idea was to use standard deviation to select features that contribute most to the classification. We calculated the standard deviation for each feature and sorted them in ascending order as described in the Algorithm shown in Fig. 7. xij is the weight of feature fi in document docj. The idea of the Algorithm is to use standard deviation to select features that contribute most to the classification.

We then muted features which have the lesser standard deviations to pick only the most important features. This was achieved since a lower standard deviation implies that the values of the function are placed in close proximity to the mean, that is not appropriate for class discrimination. Then we were training our model utilizing the latest features sub-set. We repeated the method until we omitted no features without compromising precision. We trained the ultimate model later, with the features picked. The process is described in the following the informal algorithm shown in Fig. 8.

The algorithm starts with the full feature set and, for each step, the "p" worst features (in terms of Standard deviation) are excluded from the set. The number of removed features p is determined dynamically at the beginning of the algorithm (P¡M where M is the size of the feature set). Then, the new feature set is evaluated by applying a given classification algorithm in order to make a comparison of the performance of the new set with respect to the precedent set. The process is run repeatedly making sure that no loss in prediction performance occurred (the stop criterion is not verified).

C. Classification Model

Once the attributes are defined and selected, We train the final model by executing a learning algorithm. The output would be a classification model that is able to predict the native language for the response to new data.

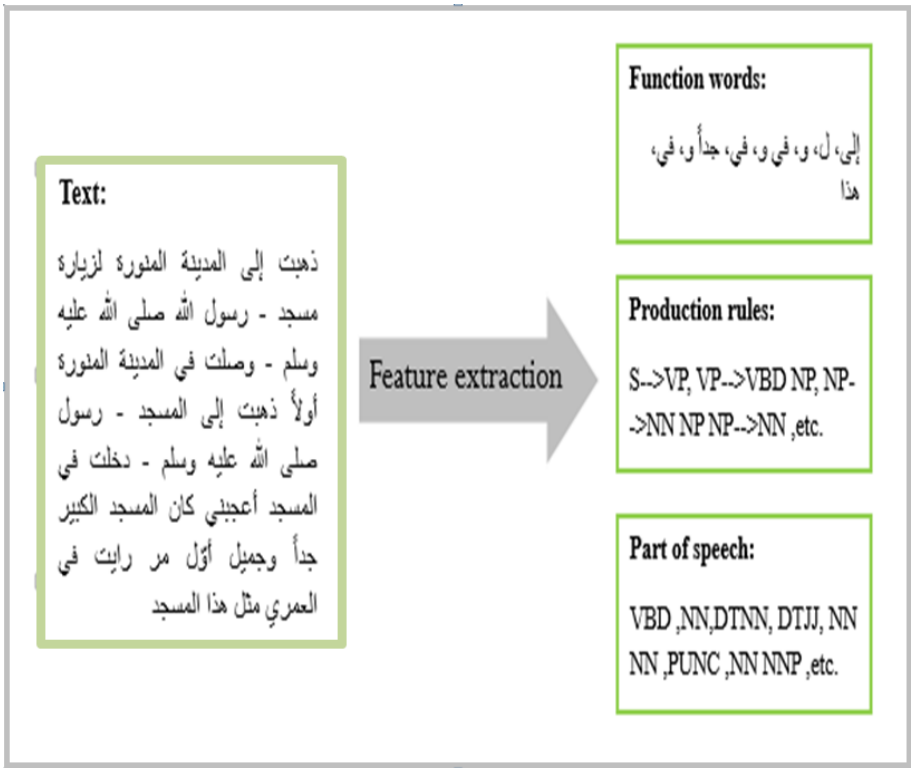

Fig. 4. Feature Sets Extraction.

Type	Examples
Linking words	• furthermore = علاوة على • despite = بالرغم • whereas = حيث أن • etc.
Conjunctions	• or = أو • but/ rather = بل • and = و • etc.
Prepositions	• from = من • to = إلى • in = في • on = على • etc.

Fig. 5. Function Words

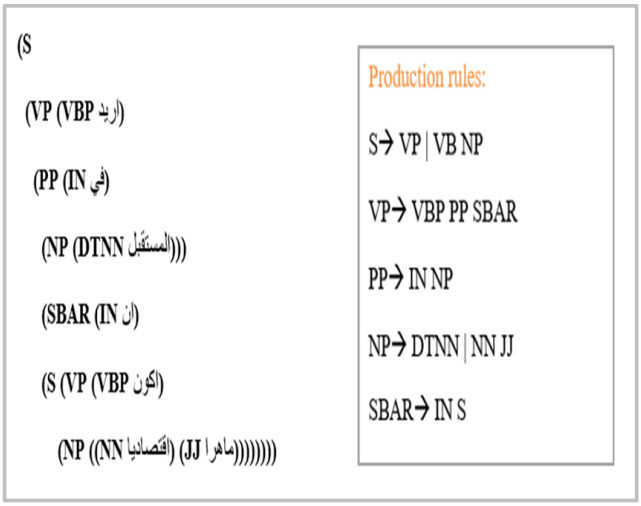

Fig. 6. Constituent Parse Tree and Grammar Production Rules.

IV. Experiments

Several series of experiments were carried out on the test corpus. These experiences have been validated and evaluated by the following techniques.

One of the most popular cross-validation techniques is K-fold. It consists in dividing the data into k subsets; one subset serves as validation data and the others act as training data. The validation process is then repeated k times. This technique becomes the de facto standard for communicating the results of the NLI; therefore, we reported our experimental results under cross validation K, with k = 10.

Because our training data set is imbalanced, to test the classification model, the adoption of different performance indicators can be a useful approach to tackle this issue. Thus, we have been using three variables frequently adopted in data mining assessment to estimate the efficiency of our strategy: recall, accuracy and precision.

A. Data Description

Our model was trained to [19], the section of the second edition of the "Arabic learner corpus" (ALC). The above

Algorithm: Calculation of the Standard deviation of the features
Input: feature set, term-document matrix weight
Output: set of features sorted by standard deviation

Begin
(1) Calculating the mean for each feature
(2) Calculating the Standard deviation for each feature
(3) Sorting the feature's sets of values in ascending order
(4) Return the Sorted set
End

Fig. 7. Calculation of Standard Deviation.

Algorithm: Feature selection Using Standard Deviation
Input: sorted feature set, a given classification algorithm (classifier) and desired number of features to exclude in each step (p)
Output: subset of most confident features

Begin
(1) Apply classifier using the full set
(2) Update feature set by removing the p first features
(3) Evaluate the new set by applying classifier using the new subset
(4) If (stop criterion not verified) return to (2)
(5) Return the new set
End

Fig. 8. Feature Selection.

includes texts produced by Native and non-native persons speaking 66 distinct Native language. In this trial we have included the seven top L1s with text length of 166 words in terms of text numbers. Table II shows the distribution of L1 broken down by word number and text number.

ALC texts are available in two computerized formats, TXT and XML formats. Those texts are annotated by author's native language within other metadata such as age, gender, etc. Each text has a title and content. The title specifies the topic of the text. Fig. 9 shows an example of XML text used in this study.

B. Architecture Layers

The different layers of the GRU model are:

- Input Layer: In this layer, Each unit directly transfers its allocated value to the Embedding layer.

TABLE II. L1 DISTRIBUTION BY NUMBER OF TEXTS AND WORDS.

Native Language	# of Texts	# of Words
Chin	76	∼ 11000
Urd	64	∼ 12300
Mal	46	∼ 6700
Fren	44	∼ 6000
Ful	36	∼ 5800
Eng	35	∼ 5800
Yor	28	∼ 5800
TOTAL	329	∼ 52200

- Embedding layer: To initialize the GRU's embedding layer, we used a bag of words that were strained via a shallow neural network. This bag defines words for determining the resemblance between words by a vector. In reality, the similitude search is based on "word2vec" techniques. In reality word2vec is a two-way combination.
- BI-GRU Layer: There are two gates in the GRU cell: an "upgrade gate", and a "reset gate". It diminishes the 3 gates that are specified in LSTM.
- Activation Layer: For the hidden layers, the most recent deep learning networks used rectified linear units (ReLUs). Many frameworks, such as "TF Learn" and "Tensor Flow" and allow the use of ReLUs on hidden layers simpler.
- Drop Out Layer: Since the size of our model is fairly large and we have a bent to implement dropout to regulate the network size and to adjust the number of hidden choices among the recurrent layers to prevent overfitting drawback.
- Dense Layer: Sigmoid was adopted as an activation function to complete the flow of information within the two gates created by the Bi-GRU sheet.

C. Results using the GRU

For the GRU, we disseminate a batch size of 1000. We use an unfold dimension of 20-time steps. We apply dropout, with a 0.8 probability for the item. To clip enormous gradients that may otherwise cause drop minima, we tend to apply a gradient cap of 5. For the training, we apply 10 iterations. We run 5 algorithms: Adam, RMSprop, Adagrad, Adadelta, and SGD. Our model is trained best based on Adam optimizer with a 0.001 learning rate. For the evaluation, we based accuracy, precision and recall. The confusion matrix shown in Fig. 10

```
<doc ID="S024_T1_M_Pre_NNAS_W_C">
  <header>
    <learner_profile>
      <age>25</age>
      <gender>ذكر</gender>
      <nationality>صيني</nationality>
      <mothertongue>الصينية</mothertongue>
      <nativeness>ناطق بغير العربية</nativeness>
      <No_languages_spoken>3</No_languages_spoken>
      ...
    </learner_profile>
    <text_profile>
      <genre>سردي</genre>
      <where>في الصف</where>
      <year>1434</year>
      ...
    </text_profile>
  </header>
  <text>
    <title>الرحلة إلى دمام.</title>
  <text_body>
    كان لنا فرصة السّفر إلى دمام ، وذلك بدأ بتسجيل أسماء الراغبين للسفر، عرفنا تاريخ السفر أثناء التسجيل، واستعددنا
    بعض الأشياء له قبل تاريخه، وجاء وقت السفر فجأة بدون إدراك! لأن الدراسة كانت متوتّرة ومليئة بتنظيم الأوقات.
    لذا، لم نشعر ببطء مرور الأيام، بل بلعكس، شعرنا بسريعتها! على كل حال. السفر لا شك أنّه سارّ وممتع، وكنّا
    مسرورين بركوب الحافلة الجامعيّة إلى المقصد، وهو دمام . من المعروف أن دمام مدينة جميلة سياحية تستحقّ الزيارة
    لها . لذا ، كنا نشيطين داخل الحافلة، تعجبنا بللمناظر الطبيعية الجميلة، غنّينا غناءً ذا ميزات محلية ، ونزلنا من
    الحافلة بقرب البحر، سبحنا مباشرا فى البحر بسبب درجه حرارة الجو لعالية . وتجوّلنا فى شوارع تلك المنطقة .
    وتنزهنا فى استراحة ما هناك . وما شاء الله لهذه الرحلة !
  </text_body>
  </text>
</doc>
```

Fig. 9. XML Files Containing Metadata and Text Content.

displays the number of samples that were classified correctly and falsely.

V. Results and Interpretations

We run two sets of experiments in order to evaluate the performance of our suggested method. The first sets of experiments aim to evaluate the performance of learning algorithms (classifiers) and consequently we choose the most efficient one to be used in the next set of experiments. The second sets of experiment is dedicated to evaluate the contribution of our features set in different configurations: individually, together, with and without passing by the selection process.

We performed multiple 10-fold cross-validation experiments to test our features both separately and in combination. Table III summarizes the full classification accuracies of the different set of features both with and without using our proposed feature selection step. Malmasi and Drass [10]

Confusion matrix		Predicted class	
		Class A	Class b
True class	Class A	True Positives (TP)	False Negatives (FN)
	Class b	False Positives (FP)	True Negatives (TN)

Fig. 10. Confusion Matrix.

developed the first and the only NLI method addressed Arabic language. They rapport that the best accuracy is obtained using combination of syntactic features similar of that used in the current study. Our results outperform those reported by them, around 5% up in accuracy.

TABLE III. NUMBER OF FEATURES AND ACCURACY.

Features	Without feature selection		With feature selection	
	# of features	Accuracy	# of features	Accuracy
Production rules	1124	30.5 %	106	36,5 %
Function Words	17	31.0 %	11	31,0 %
POS unigrams	33	30.0 %	16	34.9 %
POS bigrams	594	35.4 %	145	38.0 %
POS trigrams	580	29.0 %	347	29.0 %
Combined	2348	41.9 %	278	45.0 %

Based on the experimental results described in the Table III, we found that removing lowest deviation features in term of standard enhanced the prediction capability of our solution. Indeed, applying our selection algorithm enable our system to obtain a gain in accuracy ranging from 3.9% (case of Production rules) to 5% (case of combined features), as well as a gain in terms of memory space: we managed to reduce the size of feature vector 10 times less than the size of the initial vector from 2348 to 278.

Fig. 11 shows the last four iterations of our feature selection algorithm (iteration #205 to iteration #208) with p=10 (i.e. the ten lowest values in term of standard deviation are removed each iteration). We can see that we reach a higher accuracy of 45% at the iteration #207 with set contains 287 features. After the 207th iteration the classification performance is dramatically decreased even when we force the algorithm to continue running (iteration #209 and iteration #210).

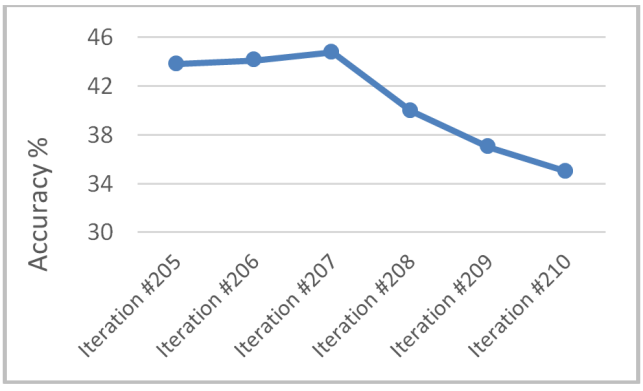

Fig. 11. Variation of Accuracies in the Last Iterations of Features Selection Procedure.

In the following, we will detail the result of individual and combined features after have been selected. For individual features, function words and production rules have demonstrated their capacity to distinguish L1 learners with and 31% accuracy for function words and 36.5% for production rules. Unsparingly, the POS n-grams consistently beaten the other syntactic features with those provided in past studies. The highest precision was obtained at 38%, with n=2. We notice that while 3-grams gave 29% of fair precision, it seems that coupling POS 3-grams with other attributes do not yield good results. The general result was quite underperformed. It may be attributed to the fact that, opposed to the other feature sets, these trigrams represent redundant information. And when we used features together, we ruled it out. We integrated 278 features divided as following: 145 bigrams, 16 unigrams, 106 production rules and 11 classes of function words. This set permitted to achieve a higher Arabic NLI classification result (45%), which prove the efficacy of the use of standard deviation.

TABLE IV. L1 DISTRIBUTION BY NUMBER OF TEXTS AND WORDS.

	Classified AS						
L1	Chin	Urd	Mal	Fren	Ful	Eng	Yor
Chin	56	14	3	3	-	-	-
Urd	13	40	6	5	-	-	-
Mal	9	11	21	2	2	1	-
Fren	8	16	6	13	1	-	-
Ful	3	12	7	7	7	-	-
Eng	10	12	3	3	-	7	-
Yor	8	10	3	4	-	-	4

The confusion matrix illustrated in Table IV presents the distribution of misclassified and correctly classified samples for the different native languages. A combination of production rules, POS and function words were adopted as classification features. The performance of the different native languages is slightly spaced. In fact, the experimental results reveal that it was possible to identify Asiatic Arabic learners better than European Arabic learners. For Chinese and Urdu, we obtained an precision rate of approximately 80% while this rate was 30% for English writers and 36% for French authors. In addition, we find out that most mispredicted samples are labelled as Chinese or Urdu sample. It is probably because Chinese and Urdu, compared to the other class, are over represented in term of samples number in training set, which is attributed to the idea of unbalanced training data and its impact in the effectiveness of the classification model.

Consequent to the tow above point, it was proven that Asian languages are effectively distinguished in the context of Arabic NLI. On the other side, the two closely related European often misclassified as Asian. African languages are the hardest to distinguish and represent the higher error rate. Especially for Yoruba, only one of seven texts is correctly classified. This may be because the deficiency of training data allocated for it.

VI. Conclusion

In this research work, we investigated the efficacy of language transfer to identify the first language of non-native Arabic speakers based on their text written in Arabic. In particular, we focused on the transfer related to the syntax. For this purpose, we presented a supervised method for Arabic NLI task based on syntactic features extracted automatically from text written by non-Arabic learners. Essentially, our method consisted of three steps where the input is a set of text and the output is a classification model able for predicting the class of unseen text: we started by pre-processing the text, in this step, we dealt with the inappropriate characters, words and marks by removing or replacing them depending on the case. Then texts passed to the next step where syntactic feature types were extracted. Therefore, the initial set transformed into space vector representation at final, the new text representation used as input for a deep learning algorithm that served to build the classification model. We found out that the features space is higher compared with number of simples. Indeed, it exceeded two thousands when we use all the features together. We assumed that many of them were redundant and non-informative. Based on this hypothesis we proposed an algorithm using a statistical metric (standard deviation) that enabled us to select the non-useful features. To accomplish the task we used the second version of ALC corpus. We included the seven top native languages: three Asian languages (Chinese, Urdu and Malay), two European (French and English) and two African (Fulani and Yoruba). In all we experimented using 329 texts of average 160 words per text.

It is worth pointing out that our results are promising, we outperform the state-of-art accuracy (45% vs 41%), given the issues that we faced in this study concerning the limited data and the unavailability of accurate tools dedicated to the Arabic language. Our methodology currently uses a static-learning model which adopts ALC as a dataset for training and testing. Therefore, we intend in future works to address this problem by developing new Arabic learner corpus which may be adopted to evaluate the generalisability of our method and more broadly to serve linguistic and computational research areas. Furthermore, the analysis of ALC texts showed that learners committed several errors of different types (orthography, morphology, syntax, semantics, etc.) when they express their ideas [20]. Exploitation of errors presents a perspective. Indeed, these errors reflect one of the main aspect of language transfer resulting from the difference between the learner's native language and that of Arabic.

References

[1] A. K. Hultgren, "English as the language for academic publication: on equity, disadvantage and 'non-nativeness' as a red herring," *Publications*, vol. 7, no. 2, p. 31, 2019. [Online]. Available: https://doi.org/10.3390/publications7020031

[2] C. S. C. Dalim, M. S. Sunar, A. Dey, and M. Billinghurst, "Using augmented reality with speech input for non-native children's language learning," *Int. J. Hum. Comput. Stud.*, vol. 134, pp. 44–64, 2020. [Online]. Available: https://doi.org/10.1016/j.ijhcs.2019.10.002

[3] S. Malmasi and M. Dras, "Chinese native language identification," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, 2014, pp. 95–99.

[4] W. Lan and Y. Hayato, "Robust chinese native language identification with skip-gram," in *DEIM Forum*, 2016.

[5] L. Wang, M. Tanaka, and H. Yamana, "What is your mother tongue?: Improving chinese native language identification by cleaning noisy data and adopting bm25," in *2016 IEEE International Conference on Big Data Analysis (ICBDA)*. IEEE, 2016, pp. 1–6.

[6] S. Malmasi and M. Dras, "Finnish native language identification," in *Proceedings of the Australasian Language Technology Association Workshop 2014*, 2014, pp. 139–144.

[7] S. Malmasi, M. Dras, and I. Temnikova, "Norwegian native language identification," in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, 2015, pp. 404–412.

[8] I. del Río, "Native language identification on L2 portuguese," in *Computational Processing of the Portuguese Language - 14th International Conference, PROPOR 2020, Evora, Portugal, March 2-4, 2020, Proceedings*, ser. Lecture Notes in Computer Science, P. Quaresma, R. Vieira, S. M. Aluísio, H. Moniz, F. Batista, and T. Gonçalves, Eds., vol. 12037. Springer, 2020, pp. 87–97. [Online]. Available: https://doi.org/10.1007/978-3-030-41505-1_9

[9] A. A. Chowdhury, V. S. Borkar, and G. K. Birajdar, "Indian language identification using time-frequency image textural descriptors and gwo-based feature selection," *J. Exp. Theor. Artif. Intell.*, vol. 32, no. 1, pp. 111–132, 2020. [Online]. Available: https://doi.org/10.1080/0952813X.2019.1631392

[10] S. Malmasi and M. Dras, "Arabic native language identification," in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, 2014, pp. 180–186.

[11] J. Tetreault, D. Blanchard, and A. Cahill, "A report on the first native language identification shared task," in *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, 2013, pp. 48–57.

[12] F. Adeeba and S. Hussain, "Native language identification in very short utterances using bidirectional long short-term memory network," *IEEE Access*, vol. 7, pp. 17 098–17 110, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2896453

[13] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[14] J. Kang, W.-Q. Zhang, and J. Liu, "Gated recurrent units based hybrid acoustic models for robust speech recognition," in *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2016, pp. 1–5.

[15] L. Jing, C. Gulcehre, J. Peurifoy, Y. Shen, M. Tegmark, M. Soljacic, and Y. Bengio, "Gated orthogonal recurrent units: On learning to forget," *Neural computation*, vol. 31, no. 4, pp. 765–783, 2019.

[16] S.-M. J. Wong and M. Dras, "Exploiting parse structures for native language identification," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1600–1610.

[17] A. Bies and M. Maamouri, "Penn arabic treebank guidelines," in *Proceedings of the Conference on Arabic language resources and tools*, 2003, pp. 466–467.

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

[19] A. Alfaifi, E. Atwell, and I. Hedaya, "Arabic learner corpus (alc) v2: a new written and spoken corpus of arabic learners," in *Proceedings of Learner Corpus Studies in Asia and the World 2014*, vol. 2. Kobe International Communication Center, 2014, pp. 77–89.

[20] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining text data*. Springer, 2012, pp. 163–222.