Investigating Linguistic Features for Arabic NLI

Yasmeen Bassas
Indiana University, Yanbu Industrial College
ybassas@iu.edu

Sandra Kübler
Indiana University
skuebler@iu.edu

Abstract

Native Language Identification (NLI) is concerned with predicting the native language of an author writing in a second language. We investigate NLI for Arabic, with a focus on the types of linguistic information given that Arabic is morphologically rich. We use the Arabic Learner Corpus (ALC) for training and testing along with a linear SVM. We explore lexical, morpho-syntactic, and syntactic features. Results show that the best single type of information is character n-grams ranging from 2 to 6. Using this model, we achieve an accuracy of 61.84%, thus outperforming previous results (Ionescu, 2015) by 11.74% even though we use an additional 2 L1s. However, when using prefix and suffix sequences, we reach an accuracy of 53.95%, showing that an approximation of unlexicalized features still reaches solid results.

1 Introduction

Native Language Identification (NLI) task is concerned with predicting the native language of texts written by learners of a second language (L2). NLI relies on the assumption that speakers of the same native language display certain linguistic patterns in their L2 texts which can be used as traces in NLI to predict their L1. Work on NLI has exploited various types of these linguistic features such as function words, character n-grams, POS n-grams, syntactic structure, and spelling mistakes (Koppel et al., 2005; Wong and Dras, 2009).

Previous work (e.g., Malmasi and Dras (2014a); Malmasi et al. (2015)) has argued for unlexicalized features as opposed to lexicalized features because they are less biased to the prompt and domain of the data. Lexicalized features consist of word n-grams while unlexicalized features are content-independent and non-lexical (such as POS tags or function words) and are thus less dependent on text vocabulary. Lexicalized features are considered less desirable because they cause topic bias since they depend on the topic of the text (Malmasi and Dras, 2014a) and may thus not be useful for texts on different topics (e.g., different prompts in TOEFL).

In our current work, we investigate this argument using the Arabic Learner Corpus (ALC), a rather small collection of texts written by learners of Arabic. The contribution of this paper is to examine which types of features will be more accurate and informative for Arabic NLI: lexicalized, unlexicalized, morpho-syntactic, syntactic features, or their combinations. We also provide a comparison between our system and three other systems that have used the same corpus. To our knowledge, this is the first attempt in Arabic NLI that examines the performance of lexicalized features and compare them to the unlexicalized ones using ALC data. We also propose a new feature set, consisting of prefix-suffix sequences, which is close to an unlexicalized feature set and can be useful in situations with domain drift.

The rest of the paper is organized as follows: section 2 describes related work, section 3 describes the data, section 4 explains the methodology, section 5 shows the results and a discussion, and section 6 presents our conclusions and future work.

2 Related Work

In text analysis tasks such as NLI, various types of linguistic features have been exploited. Certain linguistic features such as words, lemmas, tokens, and characters n-grams have been shown to be effective (e.g., Wong and Dras (2009); Gebre et al. (2013); Markov et al. (2017)). However, these types of linguistic features can be problematic since they are content/domain dependent. By using such features, topic bias can occur, especially when prompts or topics are not equally distributed across texts. This will cause a classifier to indirectly learn to recognize topics.

Therefore, other research in NLI (e.g., Malmasi

Proceedings of The Second Arabic Natural Language Processing Conference, pages 183–192

and Dras (2014b); Wong and Dras (2011); Malmasi et al. (2015)) reported the importance and usefulness of using features that are content independent to avoid these issues. Malmasi and Dras (2014a), who introduced the first application of NLI using Arabic learner data, employed content independent features such as function words, POS n-grams, and context-free grammar production rules. They used an SVM classifier and the second version of the Arabic Learner corpus (ALC) (Alfaifi et al., 2014). More specifically, they used a subset by choosing the top seven L1s represented in the corpus: Chinese, English, French, Fulani, Malay, Urdu, and Yoruba. Their results show that POS n-grams were the most useful feature (37.60%) as a single feature type. However, the highest accuracy (41% against the baseline of 23%) was achieved when all features were combined. These results show that those features were effective in discriminating L1 groups.

Similarly, Mechti et al. (2020) introduced a deep learning approach using a Gated Recurrent Unit Network (GRU) to identify the L1s of Arabic learners. They utilized the same subset (top 7 L1s) of Arabic Learner Corpus (ALC) and the same feature sets used by Malmasi and Dras (2014a). They employed standard deviation for feature selection. Their classification accuracy reached 45%, which outperformed the system by Malmasi and Dras (2014a). Ionescu (2015) proposed a string kernel system based on Local Rank Distance metric (LRD), which was applied on the character level. The author used the same subset of Arabic Learner Corpus used by Malmasi and Dras (2014a) in order to compare results. Since LRD works on character level, Ionescu (2015) did not perform any pre-processing except for normalization (removing any tabs, spaces, and new lines). A kernel based on LRD was implemented by using the Kernel Ridge Regression (KRR) classifier, using a combination of 3-5 character n-grams as features. Comparing the results of this study with the results by Malmasi and Dras (2014a) showed that this system achieved an accuracy of 50.1%, which was 10% higher than the results by Malmasi and Dras (2014a) (41%).

As described above, some NLI studies argued for unlexicalized or syntactic features since they are content independent. Almost all work done on Arabic NLI focused only on syntactic or morpho-syntactic features while ignoring lexical features. For this reason, we investigate a wide range of lexicalized and unlexicalized features, to determine the differences in performance when the L2 is morphologically rich.

Native Language	No. Texts
Chinese	76
Urdu	64
Malay	46
French	44
Fulani	36
English	35
Yoruba	28
Somali	26
Tagalog	25
Total	380

Table 1: 9 L1s with number of texts taken from ALC website.

3 Data

Data set We use the second version of Arabic Learner Corpus (ALC) by Alfaifi (2015). The corpus is small in terms of the number of learner texts, but it contains 66 L1s. We chose the top 9 L1s, for which we have 25 texts or more. The corpus consists of 380 documents of written essays and spoken interviews and uses two prompts: narrative (a vacation trip) and discussion (my study interest). 365 (95.04%) of these texts are from the written data and 15 (4.96%) are from spoken data. Table 1 shows the 9 L1s that are used in this study and the number of texts. We merged Tagalog and Filipino into the same language.

Preprocessing For preprocessing, we cleaned the transcribed spoken data for consistency with the written data. Punctuation, such as double dash (marking incomplete sentences), is removed; other punctuation, such as period, comma, and question marks, is preserved. Filled pauses and hesitations, marked by م and ي, are removed. Furthermore, partial words transcribed with a single dash are also deleted. Additionally, mispronounced words transcribed using a preceding plus sign, are removed.

The original corpus is anonymized, all personal information such as names, email addresses were replaced by an Arabic phrase. We replaced this phrase by the English term 'UNKNOWN' to indicate the anonymization without overly interfering with the n-grams. In addition, any non-Arabic words are manually removed during preprocessing as long as removing them will not affect the structure of the sentence.

Linguistic annotation We tokenized and POS-tagged the data using MADAMIRA (Pasha et al., 2014). We explore three types of MADAMIRA tokenization schemes: 1) ATB, which segments all clitics, except the definite article, and normalizes alefs/yaa. 2) D1, which only segments question and conjunction proclitics as well as normalizes alefs/yaa. 3) D2, which is similar to D1 but also segments particle clitics.

We examine two POS tagsets: 1) the PATB tagset (Maamouri et al., 2004), which makes use of around 21 POS tags, and 2) the CATiB tagset (Habash and Roth, 2009), which uses a very coarse-grained POS tagset of 6 POS tags.

We parsed the data for dependencies using CamelParser (Shahrour et al., 2016) with the configuration used by Habash et al. (2022). The parser is used with the default setting, it is trained on PATB.

4 Methodology

Machine learner We use linear Support Vector Machines (linear SVM), in the implementation of Scikit-Learn (Pedregosa et al., 2011). We select one-versus-rest multi-class classification, and we also use Mutual Information (MI) for feature selection. Feature weights consisting of TF-IDF weights are employed; other parameter tuning is performed using a 10-fold cross validation setting.

Results are reported under stratified 10-fold cross validation.

Linguistic Features We explore a wide range of linguistic features: lexical features, morpho-syntactic features, syntactic features, and combinations of these. We investigate six different lexicalized feature sets:

- word n-grams (1-5).
- prefix-stem-suffix n-grams (1-5; Madamira ATB, D1, and D2 schemes)
- character n-grams (2-8)
- a combination of character and word n-grams
- a combination of character and prefix-stem-suffix n-grams
- dependency triples (word, head, dependency relation)

For unlexicalized, morpho-syntactic and syntactic features, we investigate the following feature sets:

- function words[1]
- prefix-suffix n-grams (1-5; Madamira ATB, D1, and D2)
- a combination of function words and prefix-suffix n-grams
- POS n-grams (1-5; CATiB and PATB tagsets)
- dependency triples (POS of word, POS of head, dependency label)
- a combination of POS n-grams and function words
- a combination of POS and prefix-suffix n-grams.
- a combination of dependency triples and prefix-suffix n-grams

For the combination of lexicalized, unlexicalized, morpho-syntactic, and dependency-based features, we combine the best performing models of each feature set:

- POS and prefix-stem-suffix n-grams
- POS and character n-grams
- POS and word n-grams
- function words and prefix-stem-suffix n-grams
- function words and character n-grams.
- function words and word $n-$grams.
- function words and lexicalized dependency triples
- prefix-suffix n-grams and lexicalized dependency triples

5 Results and Discussion

5.1 Lexicalized Features Experiments

We first have a look at the experiments using lexicalized features. While such features may not be

[1]We use a function word list consisting of 1 353 words based on the lists by Salloum and Habash (2012) and Alrefaie et al. (2016)).

Type	n-gram	Accuracy
Word n-grams	1-2	51.58
	1-3	51.58
Character n-grams	2-3	58.95
	2-4	59.74
	2-5	59.47
	2-6	**61.84**
	All	59.74
Word & character n-grams	1; 2-4	58.42
	1; all	58.95
	1-3; 2-6	60.00
	1-3; 2-7	60.00
	1-3; all	59.74
Prefix-stem-suffix, D1 scheme	1-3	55.26
Prefix-stem-suffix, D2 scheme	1-3	54.21
Prefix-stem-suffix, ATB scheme	1-3	54.74
Prefix-stem-suffix D1 scheme & character n-grams	1-3; 2-4	56.32
	1-3; all	57.11
Lexicalized dependency triples		18.68

Table 2: Results of the lexicalized experiments.

optimal in practical settings because of topic shift and imbalance, we need to understand what the upper bound is and to what degree systems suffer when they do not have access to lexical information. Ideally, we are interested in finding a solution that will allow us to reach a solid performance without being too dependent on characteristics of the data.

Table 2 shows a selection of results of the lexicalized experiments. Since feature selection (using Mutual Information) gives better results in the majority of the cases, we do not report the results of the experiments using the full feature sets.

A general strategy for NLI is to use word n-grams. Since Arabic is a morphologically rich language, this strategy may not be optimal in our case, especially given the small corpus. The results show that we reach an accuracy of 51.58% when using uni- and bigrams, and remains the same when we add trigrams. This corroborates our assumption that word features are not a good fit for a morphologically rich language.

One method of mitigating variation in word forms is to use character n-grams. Using these features, we obtain the best results, 61.84% using 2-6-grams. This is in line with the findings of Kulmizev et al. (2017) and Ionescu et al. (2014) for English NLI. However, when we use all character n-grams, the results are lower at 59.74%, even after feature selection. This shows how sensitive this type of features is. The best result is based on about 30 000 features, out of 731 000 features for all character n-grams. Combining word and character n-grams is not successful, the highest accuracy of 60.00% is reached when we use word 1-3-grams and character 2-6-grams.

Since we assume that the morphological richness causes problems, we also investigate whether we can mitigate these problems by tokenizing the texts. Since this splits words into smaller units, our hope is that this will result in higher frequencies of features. We use MADAMIRA's D1, D2, and ATB tokenization schemes and split words into prefix, stem, and suffix sequences. However, the best results reach an accuracy of 55.26%. We note that the D1 scheme is a better fit than D2 or ATB for our task. In other words, the minimal tokenization, which only segments question and conjunction clitics, provides the strongest signal.

From the tokenization experiments we learn that while these features are more successful than words, they do not perform as well as character n-grams. Combining these two feature types improves over the prefix, stem, and suffix sequences, but does not reach the results of using only character n-grams.

A final type of features consists of dependency triples, i.e., we extract for each word the triple containing the word, its head, and the dependency

Type	n-gram	Accuracy
POS: CATiB	1	19.74
	1-2	16.32
	1-3	17.89
POS: PATB	1	16.58
	1-2	18.68
	1-3	17.89
Function words		32.11
Unlexicalized dependency triples		12.63
Prefix-suffix n-grams: D1 scheme	1-3	53.16
	1-4	53.16
	All	53.16
Prefix-suffix n-grams: D2 scheme	1-3	53.16
	1-4	**53.95**
	All	51.05
Prefix-suffix n-grams: ATB scheme	1-3	51.05
	1-4	50.53
	All	50.79
POS: CATiB & prefix-suffix n-grams: D1	1; 1-3	48.68
	1; 1-4	46.84
POS: CATiB & prefix-suffix n-grams: D2	1; 1-3	47.63
	1: 1-4	45.00
POS: PATB & prefix-suffix n-grams: D1	1-2; 1-3	43.16
POS: CATiB & function words	1	31.58
	1-3	26.58
POS: PATB & function words	1-2	27.11
	1-3	26.58
Prefix-suffix n-grams: D1 & function words	1-3	48.95
	1-4	47.89
Prefix-suffix n-grams: D2 & function words	1-3	48.16
	1-4	49.47
Unlexicalized dependency triples & prefix-suffix n-grams: D2	1-4	42.89

Table 3: Results of the unlexicalized experiments.

label. These features have been shown to be successful for English (Bykh et al., 2013; Cimino and Dell'Orletta, 2017), but in our case, they only reach an accuracy of 18.68%. We attribute this to the small size of the corpus.

5.2 Unlexicalized Experiments using Morpho-Syntactic and Syntactic Features

Our second set of experiments concerns unlexicalized features. Researchers often resort to POS and function word features (Malmasi and Dras, 2014a; Malmasi et al., 2015), in order to avoid domain shift or bias. However, it is unclear to what degree this strategy will work for Arabic, since the tagsets are small and do not provide morphological information. We did not experiment with adding morphological information since this would have resulted in severe data sparsity.

We use different types of features that are less corpus dependent: unlexicalized, morpho-syntactic, syntactic, and their combinations. Table 3 shows the best results of those models.

We first investigate using POS tag n-grams, using either the PATB or the CATiB POS tagset. The results range between about 16% to about 20%. The highest results are reached using CATiB POS unigrams. Whenever POS features are used separately or with other features, the CATiB tagset shows better performance compared to the PATB tagset. This may indicate that using minimal information, i.e., the small tagset of 6 POS tags in unigrams, provides the best basis for documenting

Type	n-gram	Accuracy
Word n-grams	1-2 / 1-3	51.58
Word n-grams & POS: CATiB	1; 1-2	48.42
	1; 1-3	45.79
Word n-grams & function words	1+2	51.58
	1-3	52.89
Character n-grams	2-6	**61.84**
Character n-grams & POS: CATiB	1; 2-6	57.63
	1; 2-7	57.37
Character n-grams &function words	2-6	58.68
	2-7	60.00
Prefix-stem-suffix n-grams: D1	1-3	55.26
Prefix-stem-suffix n-grams: D1 & POS: CATiB	1; 1-3	52.37
Prefix-stem-suffix n-grams: D1 & POS: PATB	1-2; 1-3	41.05
Prefix-stem-suffix n-grams: D1 & function words	1-3	53.42
Lexicalized dependency triples		18.68
Lexicalized dependency triples & function words		32.63
Lexicalized dependency triples & prefix-suffix n-grams: D1	1-3	44.21
	1-4	43.42
Lexicalized dependency triples & prefix-suffix n-grams: D2	1-3	42.63

Table 4: Results of combining lexicalized and unlexicalized features.

POS distributions. This goes against findings by Malmasi (2016) and Gyawali et al. (2013), who found that a more fine-grained POS tagset yields better accuracy for English NLI.

Next, we experiment with function words. This feature type reaches an accuracy of 32.11%, thus considerably higher than the POS results, but considerably lower than the lexicalized results. We also test unlexicalized dependency triples. It is not very surprising that those (unlexicalized dependency triples) perform worse than their lexicalized counterparts.

Our final feature set consists of prefix and suffix information. This is similar to the prefix-stem-suffix n-grams, but we argue that without using stem information, the features should be less susceptible to domain drift.

Our results show that this feature set yields the highest accuracy when using Madamira D2 tokenization (53.95%) and 1-4-grams. This setting is based on 4 000 features (selected using Mutual Information), as compared to 222 710 features for all n-grams. We note that the tokenization scheme makes little difference. Also note that while this setting does not reach the best lexicalized result (61.84% using character 2-6-grams), the results are higher than the word-based results, and very close to the prefix-stem-suffix results, thus showing that the stem information is not required in most cases.

When we look at combinations of the unlexicalized features, we see that none of the combinations reach the results of the prefix-suffix n-grams.

5.3 Experiments Combining Lexicalized, Unlexicalized, and Syntactic Features

Our third set of experiments focuses on the combination of the lexicalized and unlexicalized features, where we combine the best performing models of each of these categories in order to determine whether such a combination can provide additional information. Table 4 summarizes the best results of those combinations. For ease of comparison, we repeat the best lexicalized results per combination.

We obtain the best results (60.00%) in the combinations when we combine character 2-7-grams with function words. However, when we combine function words with word n-grams or with prefix, stem, and suffix n-grams, the accuracy decreases.

We also note that most combinations do not improve over their lexicalized individual models. The only exceptions are the word 1-3-grams combined with function words and the combination of lexicalized dependencies with either function words or with D1/D2 prefix-suffix 1-3-grams. The former (word 1-3-grams and function words) outperforms word n-grams by 1.31 points, the latter (lexical-

Type	n-gram	Accuracy
Character n-grams	2-6	**61.84**
Prefix-suffix n-grams: D2 scheme	1-4	53.95
Character n-grams &function words	2-7	60.00

Table 5: The best feature sets.

ized dependencies and the unlexicalized features) outperforms the dependency triples by more than 13 points. However, the combination of lexicalized dependencies and D1 prefix-suffix 1-3-grams shows a sizable decrease in performance compared to its unlexicalized prefix-suffix model (44.21% vs. 53.95%).

We assume that most of the lexical features suffer from data sparsity, given the small size of the corpus, while most of the unlexicalized features are too coarse to represent Arabic in sufficient detail to allow the model to recognize learner errors. It is important to know that the combination of such features does not provide any additional signal.

5.4 The Most Informative Features Per L1

Since the SVM allows us to determine which features are most predictive for a class, we had a look at the 5 most important features per L1, as indicated by the weights assigned by the SVM. Those features are shown in Table 6. The 5 most informative features include misspellings, Hamza position issues, and segmentation errors where the learner had merged two words. For example, for French and Yoruba L1s, three of the five top features include misspellings. For English L1, two of the five top features include Hamza position issue where Hamza is either missing or misplaced. For Malay L1, one of the five top features include a segmentation error. However, we also see that many of the highest ranked features are content features, i.e., correct usage. This shows that the system is sensitive to content words, even though the corpus is balanced wrt. prompts.

5.5 Comparison to Other Systems

In this section, we compare our system to the other systems that have used the same corpus for their experiments: The first system is the state of the art system obtained by Ionescu (2015), which uses a string kernel combined with Local Rank Distance metric (LRD). We also compare our system to the system by Mechti et al. (2020), who proposed using GRUs, and the system by Malmasi and Dras (2014a), who used an SVM with a range of unlexicalized features.

This comparison needs to be taken with a grain of salt since we use two additional languages: The three systems use the top 7 L1s of ALC data while we use the top 9 L1s of the same data. We chose larger number of languages to make the problem more realistic. All systems use 10-fold cross validation. In terms of feature sets, the system by Ionescu (2015) utilizes character n-grams ranging from 3 to 5. The system by Malmasi and Dras (2014a) uses function words, POS n-grams, and context-free grammar production rules. And the system by Mechti et al. (2020) employ three syntactic features: CFG production rules, function words (only 411 function words are used), and POS n-grams ranging from 1 to 3. We use our best performing feature set, character 2-6-grams, for the comparison.

Table 7 shows the highest results of all systems. Results reveal that two systems that employ traditional machine learning approaches obtain better results compared to the GRU system, with our SVM-based system using character n-grams outperforming the string kernel system by more than 10 percent points. Our system yields the highest accuracy (61.84%) even though our system has to choose between more classes (9) compared to the other systems that have fewer classes (7).

6 Conclusion and Future Work

In our work, we explored different types of feature sets: lexical, morpho-syntactic, and syntactic features in order to determine the best feature type given that Arabic is a morphologically rich language and that the corpus is small in comparison to English NLI corpora. Our results indicate that lexical features are more informative and predictive, even for a morphologically rich language, compared to unlexicalized features. The best single feature type is character n-grams ranging from 2 to 6 (61.84%). Combining lexicalized and unlexicalized features does not result in an improvement of results. However, using prefix and suffix n-grams

Language	Feature	Type	Language	Feature	Type
Chinese	أصبحت 'I became'	C	Somali	أجانب 'foreigners'	C
	الأمثل 'the most ideal'	C		أستزيد 'I gain more'	C
	اسنتشعرنا 'we felt'	M		الإعادة 'the repetition'	C
	أغلى 'more expensive'	C		أقدمه 'I present it'	C
	الأماكن 'places'	C		أدحل 'I enter'	M
English	إثناء 'during'	H	Tagalog	الاسبانية 'Spanish'	C
	الاستماع 'listening'	H		أوجههم 'their faces'	C
	أرتحل 'I travel'	C		الأيام 'days'	C
	استيقظوا 'they woke up'	C		الاملابس 'clothes'	M
	أرشدني 'he guided me'	C		الأبعاء 'Wednesday'	M
French	اسنتشعرنا 'we felt'	M	Urdu	أزرع 'I instil'	C
	الأقربائي 'my relatives'	M		أغظم 'greater'	M
	استغربت 'I was surprised'	C		أبي 'my father'	C
	ءادم 'Adam'	H		أحققها 'I achieve it'	C
	أشبي 'I look like'	M		أجهزة 'devices'	C
Fulani	١٤٣٤ '1434'	C	Yoruba	ءلى 'to'	M
	الازهر 'Al-Azhar'	H		الإجرازات 'the procedures'	M
	أحرمت 'I wore Ihram'	C		أصلا 'originally'	C
	أصلو 'fundamentals'	M		آدات 'performing'	M
	أخي 'my brother'	C		75 بالمئة '75 percent'	C
Malay	استأجاره 'he rented it'	M			
	أنت 'you'	C			
	اعتنى 'he took care'	C			
	أقرأ 'I read'	C			
	إلاالله 'except God/Allah'	S			

Table 6: The five most informative features per L1. Feature types: C(ontent), H(amza error), M(isspelling), S(egmentation error).

shows promising results in case we need to address topic shift. While these results do not reach those of the lexicalized version using prefix-stem-suffix n-grams, they are close, thus showing that the stem does not provide much signal for NLI.

For future work, we are planning to investigate those linguistic features using a large scale data to determine to what degree our results are due to the small corpus size. We also need to broaden the spectrum and look at other L2s that are morphologically rich in order to see if similar regularities hold.

Acknowledgements

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

Systems	Model	Features	Accuracy
Ionescu (2015)	string kernel & LRD	3-5 char	50.10
Malmasi and Dras (2014a)	SVM	unlexicalized	41.00
Mechti et al. (2020)	GRU	syntactic features	45.00
Ours	SVM	2-6 char	61.84

Table 7: Comparison to three Arabic NLI systems.

References

Abdullah Alfaifi. 2015. *Building the Arabic Learner Corpus and a System for Arabic Error Annotation*. Ph.D. thesis, University of Leeds.

Abdullah Alfaifi, Eric Atwell, and Hedaya Ibraheem. 2014. 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*.

Mohamed Alrefaie, Noura Hussein, and Tarek Bazine. 2016. Arabic-stop-words. https://github.com/mohataher/arabic-stop-words/blob/master/LICENSE.

Serhiy Bykh, Sowmya Vajjala, Julia Krivanek, and Detmar Meurers. 2013. Combining shallow and linguistically motivated features in native language identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 197–206, Atlanta, Georgia. Association for Computational Linguistics.

Andrea Cimino and Felice Dell'Orletta. 2017. Stacked sentence-document classifier approach for improving native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 430–437, Copenhagen, Denmark. Association for Computational Linguistics.

Binyam Gebrekidan Gebre, Marcos Zampieri, Peter Wittenburg, and Tom Heskes. 2013. Improving native language identification with TF-IDF weighting. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 216–223, Atlanta, Georgia. Association for Computational Linguistics.

Binod Gyawali, Gabriela Ramirez, and Thamar Solorio. 2013. Native language identification: a simple n-gram based approach. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 224–231, Atlanta, Georgia. Association for Computational Linguistics.

Nizar Habash, Muhammed AbuOdeh, Dima Taji, Reem Faraj, Jamila El Gizuli, and Omar Kallas. 2022. Camel treebank: An open multi-genre Arabic dependency treebank. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2672–2681, Marseille, France. European Language Resources Association.

Nizar Habash and Ryan Roth. 2009. CATiB: The Columbia Arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, Suntec, Singapore. Association for Computational Linguistics.

Radu Tudor Ionescu. 2015. A fast algorithm for local rank distance: Application to arabic native language identification. In *Proceeings, Part II, of the 22nd International Conference on Neural Information Processing - Volume 9490*, ICONIP 2015, page 390–400, Berlin, Heidelberg. Springer-Verlag.

Radu Tudor Ionescu, Marius Popescu, and Aoife Cahill. 2014. Can characters reveal your native language? a language-independent approach to native language identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1363–1373, Doha, Qatar. Association for Computational Linguistics.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Determining an author's native language by mining a text for errors. In *Knowledge Discovery and Data Mining*.

Artur Kulmizev, Bo Blankers, Johannes Bjerva, Malvina Nissim, Gertjan van Noord, Barbara Plank, and Martijn Wieling. 2017. The power of character n-grams in native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 382–389, Copenhagen, Denmark. Association for Computational Linguistics.

Mohamed 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*.

Shervin Malmasi. 2016. *Native language identification: explorations and applications*. Ph.D. thesis, Macquarie University.

Shervin Malmasi and Mark Dras. 2014a. Arabic native language identification. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 180–186, Doha, Qatar. Association for Computational Linguistics.

Shervin Malmasi and Mark Dras. 2014b. Chinese native language identification. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2:*

Short Papers, pages 95–99, Gothenburg, Sweden. Association for Computational Linguistics.

Shervin Malmasi, Mark Dras, and Irina Temnikova. 2015. Norwegian native language identification. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 404–412, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Ilia Markov, Lingzhen Chen, Carlo Strapparava, and Grigori Sidorov. 2017. CIC-FBK approach to native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 374–381, Copenhagen, Denmark. Association for Computational Linguistics.

Seifeddine Mechti, Roobaea Alroobaea, Moez Krichen, Saeed Rubaiee, and Anas Ahmed. 2020. Deep learning model for identifying the arabic language learners based on gated recurrent unit network. *International Journal of Advanced Computer Science and Applications*, 11.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland. European Language Resources Association (ELRA).

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Wael Salloum and Nizar Habash. 2012. A modern standard arabic closed-class word list.

Anas Shahrour, Salam Khalifa, Dima Taji, and Nizar Habash. 2016. CamelParser: A system for Arabic syntactic analysis and morphological disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 228–232, Osaka, Japan. The COLING 2016 Organizing Committee.

Sze-Meng Jojo Wong and Mark Dras. 2009. Contrastive analysis and native language identification. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 53–61, Sydney, Australia.

Sze-Meng Jojo Wong and Mark Dras. 2011. Exploiting parse structures for native language identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1600–1610, Edinburgh, Scotland, UK. Association for Computational Linguistics.