

Research Article

Full Diacritization of the Arabic Text to Improve Screen Readers for the Visually Impaired

Batool Abuali  and **Mohamad-Bassam Kurdy** 

Master in Web Sciences, Syrian Virtual University, Damascus, Syria

Correspondence should be addressed to Batool Abuali; batoulabuali@gmail.com

Received 9 November 2021; Revised 14 January 2022; Accepted 17 May 2022; Published 16 July 2022

Academic Editor: Alessandra Agostini

Copyright © 2022 Batool Abuali and Mohamad-Bassam Kurdy. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper aims to find the relationship between the full diacritization of the Arabic text and the quality of the speech synthesized in screen readers and presents a new methodology to develop screen readers for the visually impaired, focusing on preprocessing and diacritization of the text before converting it to audio. First, the actual need for our proposal was measured by conducting a MOS (Mean Opinion Score) questionnaire to evaluate the quality of the speech synthesized before and after full diacritization in the NVDA (<https://www.nvda-ar.org/>) screen reader. Then, an e-reader was built by integrating two models: the first one is for automatic Arabic diacritization (depending on Shakkala), and the second is a TTS model (depending on Tacotron). The quality of our proposed system was measured in terms of (1) pronunciation and (2) intelligibility, in which our system outperformed the commercial screen readers, NVDA and IBSAR (<https://www.sakhr.com>), as it recorded 60.67%, 17.67%, and 21.67% as correct, incorrect, and partially correct, respectively, for the isolated word test, and 84% correct results for the homograph test, and 78.50% and 93% correct results, respectively, for the DRT and DMRT tests.

1. Introduction

Text-to-Speech (TTS) systems are used to convert plain text included in digital documents into an audible format. It is a very essential technology in many fields and is used in multiple applications, especially those that help the visually impaired.

Even though Arabic is one of the most widely used languages worldwide, Arabic TTS systems are still in their infancy in comparison to other languages like English.

When we talk about Arabic natural language processing, the absence of diacritics poses a major dilemma for a group of software, especially speech synthesizers and screen readers, which help the visually impaired to overcome the barrier of using technology.

Arabic diacritics are divided into basic diacritics which play a grammatical role and give different meanings to words, for example, (عَلَم) could mean flag (عَلَمٌ), or knowledge (عِلْمٌ), and case endings or syntactic diacritics which determine words' place in the context so it would be

understood, for example, school (مدرسة) could be a subject (مدرسة), or an object (مدرسة), depending on the diacritization of the last letter.

Full diacritization improves the process of automatic reading of texts written in Arabic in screen reader applications. The basic Arabic alphabet contains 28 letters and 8 diacritics encoded in the standard hexadecimal notation under 0600–06FF as shown in Figure 1 [1].

TTS systems consist of two modules: (1) the NLP module and (2) the Digital Signal Processor module. In this research, we focus on the first part of a TTS system, suggest a change in the way that the input text is usually processed, and propose the possibility to improve screen readers used by the visually impaired by adding a full diacritization stage in the process of building an Arabic TTS system.

We measured the actual need for our proposal by conducting a 5-scale MOS questionnaire [2] (see Appendix), to evaluate the quality of speech synthesized before and after full diacritization in the NVDA screen reader in 7 categories. We selected our dataset to be balanced in the sense that it

contains a paragraph which is easy to pronounce, another paragraph which is slightly hard to pronounce, and a third one which is hard to pronounce. 152 native Arabic speakers aged 18–50 years participated in the survey.

The results shown in Figure 2 reflected a remarkable improvement, especially in the pronunciation part, where we recorded an increase of about 1 out of 5 overall degrees. This encouraged us to continue our work and be one of the first studies that focused on the importance and necessity of full diacritization of Arabic text in TTS systems, hoping to make the synthesized speech smoother and closer to the natural sound.

2. Literature Reviews

2.1. Automatic Arabic Diacritization. Regarding the Arabic language, the past approaches used in automatic diacritization are roughly classified as (1) rule-based, (2) statistical, and (3) hybrid [3]. We reviewed the approaches in the recently published literature on the diacritization problem as follows:

2.1.1. Rule-Based Approaches. These approaches used Arabic’s well-formed rules through methods like morphological analysis, syntactic analysis, and morph-phonological modules [3–5].

2.1.2. Statistical Approaches. The used methods include using HMM (Hidden Markov Models) [3, 6], N-grams models [3, 7, 8], Dynamic Programming methods [8, 9], classical Machine Learning models like MaxEnt (Maximum Entropy) classifier [10], and DL (Deep Learning) methods [11–13].

2.1.3. Hybrid Approaches. These approaches mix rule-based methods and statistical ones, and this includes a combination of linguistic knowledge (well-formed rules and dictionary retrievals with morphological analysis) and other techniques, like N-grams, HMM, DL models, and Machine Learning methods [3, 11, 14–17].

However, the accessible tools for Arabic text diacritization are still limited as most of the systems from the literature are not available for free use.

2.1.4. Comparison. A previous study compared the three past approaches from different points of view, and the results showed that hybrid approaches record higher accuracy in comparison with the other two [3]. Another review of the existing diacritization systems was provided and showed that Shakkala, which is a DL approach, is quite better than the traditional rule-based approaches and other existing systems and tools [11].

2.2. TTS. We reviewed the existing systems and methods for the Arabic speech synthesizing problem as follows:

	U+062x	U+063x	U+064x	U+065x
0		ذ		ـ
1	ء	ر	ف	ـ
2	آ	ز	ق	ـ
3	أ	س	ك	
4	ؤ	ش	ل	
5	!	ص	م	
6	ئ	ض	ن	
7	ا	ط	ه	
8	ب	ظ	و	
9	ة	ع	ى	
A	ن	غ	ي	
B	ث		ـ	
C	ج		ـ	
D	ح		ـ	
E	خ		ـ	
F	د		ـ	

FIGURE 1: Basic Arabic (Unicode block): 28 letters and 8 diacritics.

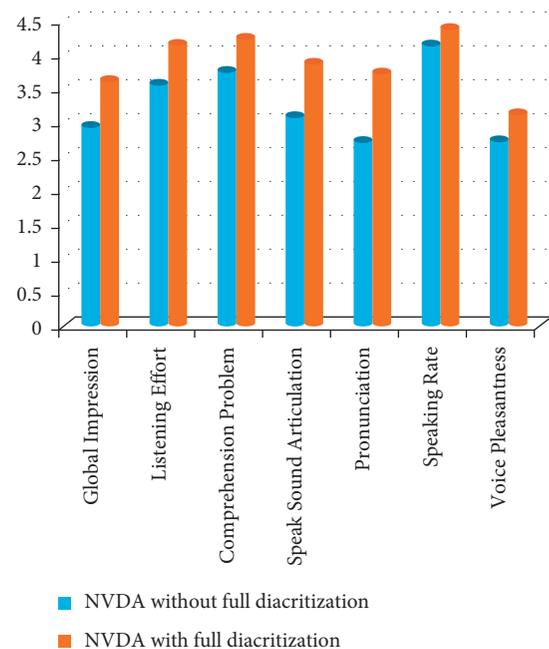


FIGURE 2: MOS questionnaire results comparing the quality of the output in NVDA before and after full diacritization.

2.2.1. Screen Readers. In the current time, there are multiple commercially available Arabic TTS systems, like Sakhr TTS (2012), Acapela (2017), Natural Soft (2017), CSTR Festival (2017), and MBROLA (2017). But the visually impaired cannot use these systems because they are not available for free. NVDA (2012) is a free software [18], which makes it one of the most widely used screen readers by the visually impaired in the Arab world.

A TTS system for diacritic Arabic texts was designed based on unit selection using a bigrams model [19]. An Arabic TTS support was developed to be included and integrated into the eSpeak system [20].

Methods	Advantages	Disadvantages
HMM	Flexible with changing voice characteristics and the system is robust	The acoustic features are oversmoothed, making the generated speech sounds muffled
RBM	Can better describe the distribution of high-dimensional spectral envelopes to alleviate the over-smooth problem	Suffer from the fragmentation problem of training data
DBN	Cannot suffer from the training data fragmentation problem and reduce the over-smoothing problem	The quality of generated speech will be degraded
DMDN	Can solve the single modality problem	Can only leverage limited contexts and each frame is mapped independently
DBLSTM	Can fully leverage contextual information	Still needs a vocoder to synthesize waveform
WaveNet	Can produce high-quality speech waveforms	Too slow and the errors from the front-end will affect the synthesis effect
Tacotron	Fully end-to-end speech synthesis model and can produce high-quality speech waveforms	Quite costly to train the model
CNN	Fast to train the model	The speech quality might be degraded

FIGURE 3: The advantages and disadvantages of the different methods in speech synthesis [21].

Arabic TTS model	MOS
Concatenative with HMM	3.89
Tacotron 1	4.01
Tacotron 2	4.38

FIGURE 4: Results of comparing two DL models (Tacotron 1 and 2) and a concatenative system [22].

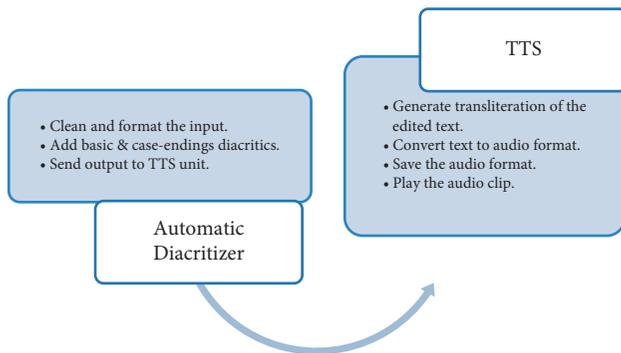


FIGURE 5: Units of the proposed system.

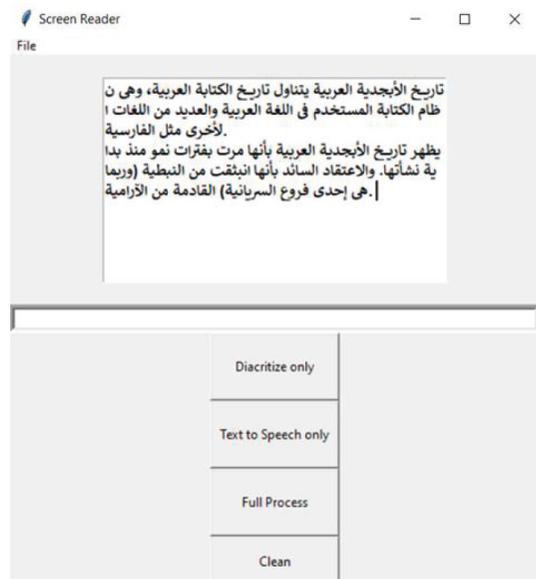


FIGURE 6: Example of an input.

2.2.2. Comparison of Speech Synthesis Methods. A review of the different speech synthesis methods, including HMM, RBM (Restrictive Boltzmann machine), DBN (Deep belief network), DMDN (Deep mixture density network), DBLSTM (Deep bidirectional long short term memory), WaveNet, Tacotron, and CNN (Convolutional neural network) was presented, discussing the advantages and disadvantages of each method, as shown in Figure 3, and drawing the conclusion that DL-based models achieve a higher quality of the synthesized speech than the traditional methods [21].

Also, Arabic speech synthesis using deep learning architectures was explored in another study where the main two models utilized in an end-to-end Arabic TTS system were compared to a concatenative TTS system as shown in Figure 4 [22].

2.2.3. Evaluation of Screen Readers. TTS systems can be measured against multiple criteria like pronunciation,

intelligibility, and comprehensibility. We reviewed the recent studies on the evaluation of Arabic TTS systems. In their study [23], the authors evaluate six Arabic TTS systems using four intelligibility tests: (1) Diagnostic Rhyme, (2) Modified Rhyme, (3) Phonetically Confusable Sentences, and the fourth one is related to the prediction of the diacritics of the input text and was proposed by the authors themselves. Another two tests were performed: (5) Arabic Text with All Sounds and (6) Best/Worst Pleasant Voice, which were proposed by the authors to determine the voice pleasantness. The authors also conducted an objective evaluation using two types of measures: (1) signal-to-noise variation and (2) linear predictive measure. In [18], the authors evaluate two of the most popular screen readers used in the Arab community using two pronunciation tests: (1) the isolated word test and (2) homographs, and two intelligibility tests: (1) DRT and (2) DMRT.

In building our proposed system in this study, we used Shakkala as it achieved the best results in full diacritization in our opinion and it is available as an open source, and we used Tacotron, as it is one of the best options available for

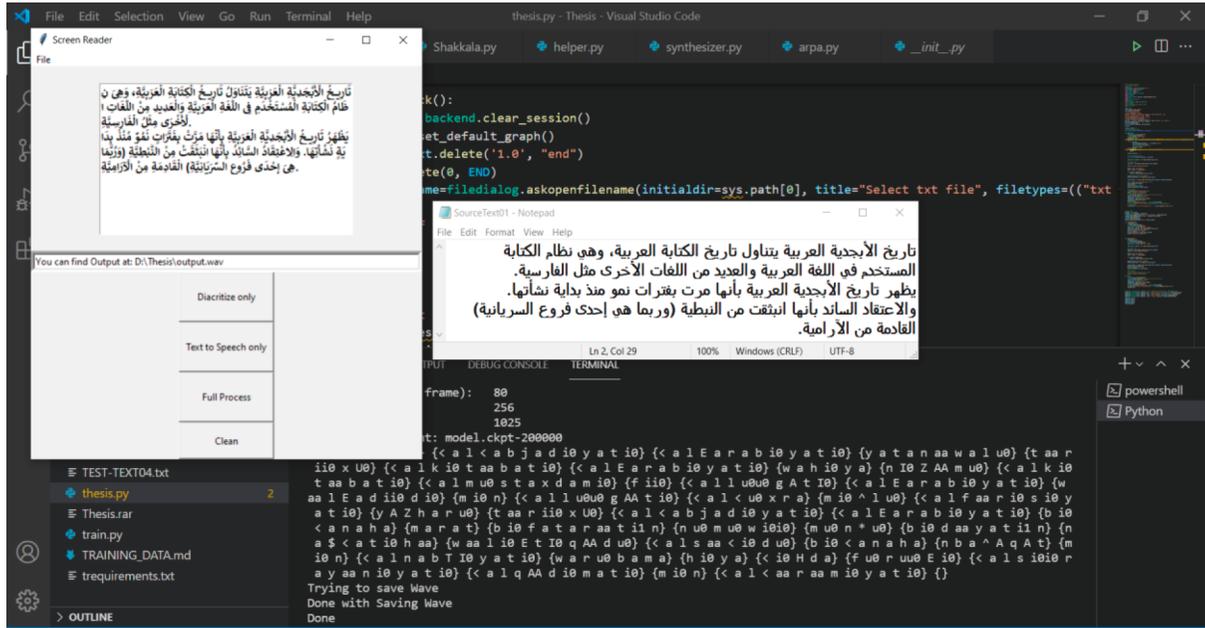


FIGURE 7: Output of the proposed system: after the full diacritization, the system shows the transliteration of diacritized speech, generates and plays the audio file.

speech synthesis, and here we would like to make a point, that Google TTS is not accessible to us as Syrians so we could not use it in the proposed system or the evaluation. For evaluation, we used the same methods in [18], as our focus is on improving the pronunciation and differentiation of words so that the output is clearer and more natural for the Arabic listener.

It is important to emphasize that the purpose of our study is not to compete with other Arabic TTS systems, but to point out the importance of the full diacritization, and suggest the possibility of improving existing e-readers or building new ones that handle the initial text more efficiently.

3. Methodology

TTS systems consist of two modules: (1) the NLP module and (2) the Digital Signal Processor module. As our study focuses on the impact of full diacritization, as shown in Figure 5, our proposed system consists of two main units:

- (i) Automatic diacritizer unit.
- (ii) Text-to-speech unit.

3.1. Work Stages

First: add input; an input text is presented to the proposed system by either writing it directly in the GUI textbox shown in Figure 6, or by adding a whole file via the button named Full Process.

Second: clean input; the input text is processed and previous diacritics (if any) are removed.

Third: full diacritization; the processed text is fully diacritized (basic and case-ending diacritics).

TABLE 1: Libraries used.

Package	Version
Bleach	1.5.0
Click	7.1.2
enum34	1.1.10
h5py	2.10.0
Keras	2.1.3
Markdown	3.3.3
Nltk	3.5
Numpy	1.15
Protobuf	3.14
PyYAML	5.3.1
SciPy	1.5.4
TensorFlow	1.4.0
Six	1.15
Tensorflow-tensorboard	0.4.0
Werkzeug	1.0.1
Falcon	1.2.0
Librosa	0.5.1
Matplotlib	2.0.2
Tqdm	4.11.2
Unidecode	0.4.20
Arabic-pronounce	0.2.5

Fourth: TTS; the fully diacritized text is converted to audio.

Fifth: save and play; a wave file is built and played, as shown in Figure 7. Check the file titled “output.wav” in the Supplementary Material and listen to the speech synthesized from the same input example.

3.2. *More Details on the Proposed Model.* To build our proposed model, we used two pretrained models, the first one is Shakkala model [24], and the second is an open-

TABLE 2: Word list for isolated word test.

No.	Word
1	بَخَسَ (bakhs)
2	أَغْنَى (aghnaa)
3	أَنْذَرَ (andhar)
4	بَدَرَ (badr)
5	بَدَّخَ (badhkh)
6	فَوَّزَ (fwz)
7	حَدَّوَهَ (hadhwah)
8	حَضَّيَضَ (hadid)
9	كَافَأَتْ (kafat)
10	مَأْخُذَ (makhudh)
11	أَضْدَادَ (addad)
12	أَشْخَصَ (ashkhas)
13	إِثْمَ (iithm)
14	بَخَشَّشَ (bikhashish)
15	عَجَزَ (ejz)
16	فَحَمَ (fahm)
17	عَطَّشَانَ (eatshan)
18	دَغَّلَ (ladagh)
19	حَطَّطَ (lahazt)
20	مَأْدَبَةَ (maduba)
21	مَعْسُولَ (maeswl)
22	مَدَّحَ (madah)
23	مَرْيَمَ (marym)
24	مَشْكَاةَ (mishkat)
25	وَقْفَ (waqf)
26	قَطْنَ (qutn)
27	رَاسِيَّ (rasi)
28	بَعُونَسَ (sabeun)
29	يَصِطْفَى (yastafi)
30	قَهْمَزَرَ (rzquhm)

source implementation of Tacotron [25]. We integrated the two models and made some improvements.

A virtual environment was created using Visual Studio Code (<https://code.visualstudio.com/>) and the Python language to integrate both models. Several packages were tested for stability on those that make the two models work without errors, as in Table 1.

Also, some modifications were made to improve our proposed system, like adding a function to delete the pre-existing diacritics in the first unit of the system and editing the way that Tacotron pronounces specific parts of the Arabic words by modifying the Arabic-pronounce package.

It is important to emphasize that we did not write a single Python code from scratch, but most of the Python codes used in this study are from previously mentioned studies, which have been cited.

4. Evaluation

Based on the evaluation methods used in a previous comparative study [18], we compared the output of our proposed system with the previous results of both IBSAR and NVDA programs in two tests: (1) pronunciation and (2) intelligibility.

4.1. Pronunciation Test. Two different pronunciation tests were conducted. The first one is called the “isolated word

TABLE 3: Word list for homograph test.

No.	Word
1	عَصَرَ (Asara)
2	عَصَرَ (Usira)
3	مَلَّسَ (Sallama)
4	مَلَّسَ (Sollima)
5	جَمَعَ (Jamaa)
6	جَمَعَ (Jammaa)
7	بَعَرَ (Rabbaa)
8	بَعَرَ (Robbiaa)
9	قَطَرَ (Qatar)
10	قَطَرَ (Qutr)
11	كَتَبَ (Kataba)
12	كَتَبَ (Kotiba)
13	ذَهَبَ (Zahab)
14	ذَهَبَ (Zahaba)
15	عَبَّرَ (Aabra)
16	عَبَّرَ (Ebra)
17	وَجَدَ (Wagada)
18	وَجَدَ (Wogeda)
19	عَيْنَ (Aaen)
20	عَيْنَ (Oean)

test,” where we pronounce each word without being in any surrounding context, and the second is the homograph test, where we present each homograph in a single sentence context.

In Arabic, homographs are words composed of the exact same letters but have different pronunciations and meanings, e.g. كَتَبَ (Kataba, means wrote) and كَتَبَ (Kotiba, means written).

For the isolated word test, we selected a list consisting of 30 words from a database of Arabic phonemes [26].

We selected the dataset to contain 10 words that are easy to pronounce, 10 words that are slightly hard to pronounce, and 10 words that are hard to pronounce. The words (written in Arabic and their transliteration in English) are shown in Table 2.

Similarly, we selected a list of 10 homographs and embedded each one of them in a one-sentence context. The homographs (written in Arabic and their transliteration in English) are shown in Table 3.

Ten native Arabic speakers who are university students with vision impairment participated in the experiments and were provided with instructions as follows:

In the isolated word test, you will hear one word at a time. Then, you will read a word written in braille. Please define the word as:

- (i) Correct: if the spoken word matches the one written in braille, write number 1.
- (ii) Incorrect: if the spoken word is not the same as that written in braille, write number 2.
- (iii) Partially correct: if you are not sure if the spoken word is the same as that presented in braille, write number 3.

Similarly, the homograph test was conducted where each homograph was embedded in a one-sentence context.

TABLE 4: Word lists for experiment 2 (DRT and DMRT).

	DRT	DMRT
1	(hanan) حنان	(wassaf) وصال
	(janan) جانان	(wassal) وصال
2	(hibal) حبال	(kassar) كسار
	(hibal) هبال	(kassab) كساب
3	(minshar) منشار	(yaltaqit) يلقط
	(mustashar) مستشار	(yaltahim) يلقح
4	(sarab) بارص	(enkitam) كتمن
	(sarab) بارس	(enkitab) كتبن
5	(qasim) قاسم	(basil) باسل
	(kasim) كاسم	(basim) باسم
6	(nil) نيل	(tahjuz) تجج
	(nil) ميل	(tahjub) تججب
7	(tin) تين	(Jamie) جامع
	(din) دين	(jamih) جامح
8	(ghab) باغ	(haris) حريص
	(khab) خاب	(hariq) حريق
9	(sadd) سدس	(rayid) ريدار
	(shadd) شدد	(rayie) ريدار
10	(nah) حان	(sanaqul) قولنس
	(lah) حال	(sanaqum) قومنس
11	(tab) باط	(sharif) شريف
	(tab) تاب	(sharik) شريك
12	(ramat) متر	(sayid) يداص
	(namat) متن	(sayib) يصاص
13	(khar) خار	(tawabie) وابعط
	(har) حار	(tawabiq) وابعق
14	(zala) ظل	(eajayib) عجائب
	(dhala) ذل	(eajayiz) عجائز
15	(nawe) وعن	(ghamid) مضاع
	(rawe) وعر	(ghalib) بلاغ
16	(ghab) باغ	(qari) قارئ
	(eab) عاب	(qarib) قارب
17	(sawq) وقس	(karl) كارل
	(shawq) شوق	(kard) كارد
18	(tilal) تلال	(mamun) مأمون
	(silal) لالس	(mamur) مأمور
19	(zwm) ومز	(natasalal) تسللن
	(thum) ثوم	(natasalaq) تسلقن
20	(rasibat) ببتسار	(waqie) واقع
	(hasibat) حسبت	(waqif) واقف

4.2. *Intelligibility Test.* We determine intelligibility by whether the human user can understand the output of the TTS system or not. To achieve that, we used the most common tests: DRT (Diagnostic Rhyme Test) and DMRT (Diagnostic Medial Consonant Test).

Ten native Arabic speakers who are university students with vision impairment participated in the experiments, and were provided with instructions as follows:

Each time you hear a word, you will read two words written in braille. Please choose the word you heard.

In both DRT and DMRT tests, we selected 20 rhyming word pairs which differ in their initial consonant, e.g. تلال (tilal means hills) and لالس (silal means baskets) in DRT, and in the intervocalic consonant, e.g. حريق (hariq means fire) and حريص (haris means careful) in DMRT. The words (written in Arabic and their transliteration in English) are shown in Table 4.

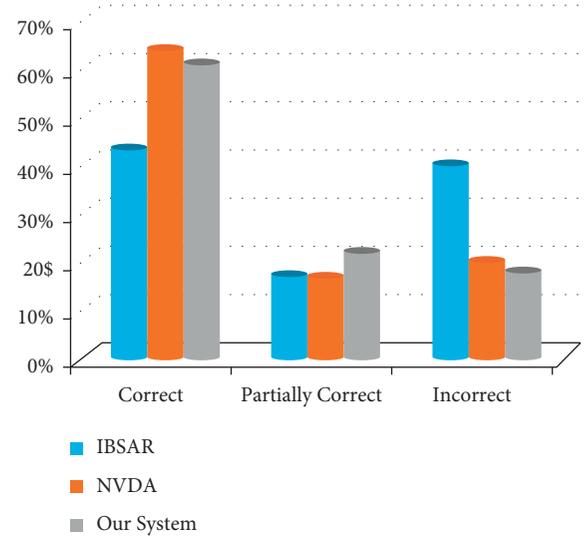


FIGURE 8: Isolated word test results.

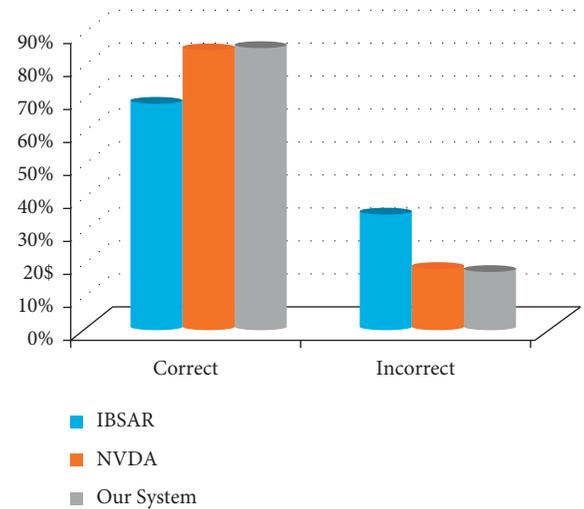


FIGURE 9: Homograph test results.

5. Results and Discussion

5.1. *Punctuation Test.* Answers were listed as correct, incorrect, or partially correct for each word. The results revealed that for the first test (isolated word test): 60.67% of the answers were correct, 17.67% were incorrect, and 21.67% were partially correct, as shown in Figure 8, approximating the performance of NVDA and outperforming the performance of the IBSAR program, as stated in the previous study which was mentioned above.

Similarly, for the homograph test, the results revealed that 84% of the answers were correct, and 16% were incorrect, as shown in Figure 9, approximating the performance of NVDA and outperforming the performance of IBSAR.

5.2. *Intelligibility Test.* Answers were listed as correct or incorrect for each test. The results of DRT revealed that 79%

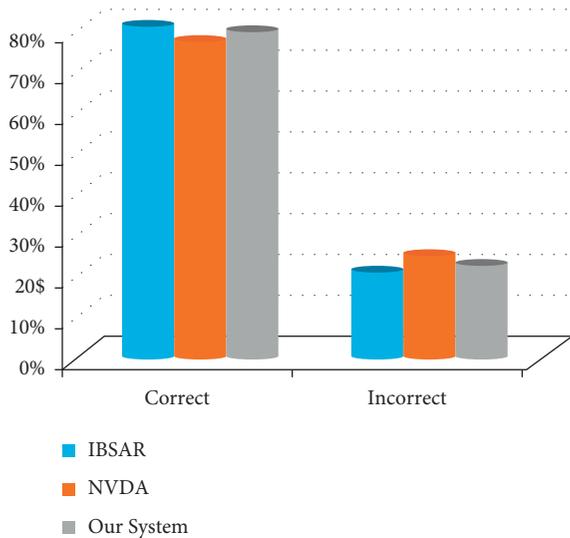


FIGURE 10: DRT test results.

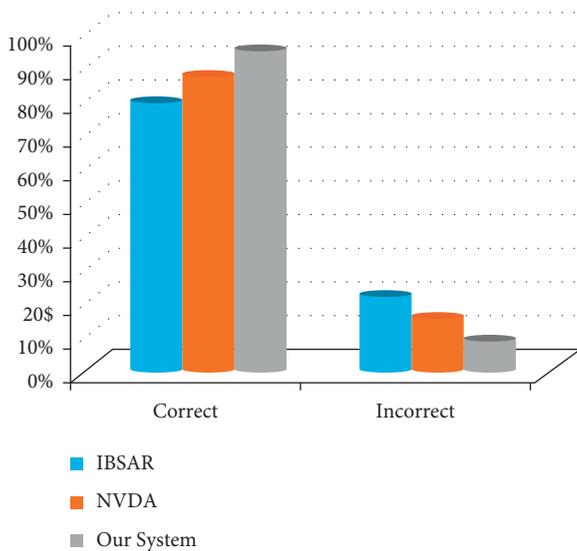


FIGURE 11: DMRT test results.

of our system answers were correct, and 21% were incorrect, as shown in Figure 10, outperforming the performance of NVDA and approximating IBSAR results, as stated in the previous study that was mentioned above.

Similarly, for the DMRT test, the answers were also listed as correct or incorrect for each test. The results revealed that 93% of our system’s answers were correct, and 7% were incorrect, as shown in Figure 11, clearly outperforming both of the other programs.

6. Conclusion

In this work, a new approach has been proposed to improve Arabic screen readers by focusing on the full diacritization of the text. Our proposed model was built by integrating two models: the first one is for automatic Arabic diacritization (depending on Shakkala), and the second is a TTS model (depending on Tacotron).

We evaluated our proposed system in terms of (1) pronunciation and (2) intelligibility, and compared it to the commercial screen readers: NVDA and IBSAR.

The results showed that the overall quality of our proposed system is better than the other two.

In the future, we will work to improve our proposed system by:

- (i) Training the system (both models) on a larger set of data.
- (ii) Adding rules related to reading known numbers, dates, and names.
- (iii) Working on the prosody factor that affects the quality, naturalness, and intelligibility of synthesized speech.
- (iv) Building an interactive website that can contribute to evaluating the system

Appendix

MOS questionnaire

Q1: Global impression

Please rate the sound quality of the voice you heard.

- Bad
- Poor
- Fair
- Good
- Excellent

Q2: Listening effort

Please rate the degree of effort you had to make to understand the message.

- Message not understood with any feasible effort
- Major effort required
- Effort required
- Slight effort required
- No Effort Required

Q3: Comprehension problems

Were single words hard to understand?

- Every word
- Many
- Some
- Few
- None

Q4: Speech sound articulation

Were the speech sounds clearly distinguishable?

- Not at all
- Not very clear
- Fairly clear
- Clearly enough

Very Clear

Q5: *Pronunciation*

Did you notice any anomalies in the naturalness of sentence pronunciation?

- Yes, very
- Yes, annoying
- Yes, slightly annoying
- Yes, but not annoying
- No

Q6: *Speaking rate*

Was the speed of delivery of the message appropriate?

- No, too fast
- No, too slow
- Yes, but faster than preferred
- Yes, but slower than preferred
- Yes

Q7: *Voice Pleasantness*

Was the voice you heard pleasant to listen to?

- Very Unpleasant
- Unpleasant
- Fair
- Pleasant
- Very Pleasant

Data Availability

Most Python codes used in this study are from previously reported studies, which have been cited. Any further data or content used in the experiments are available from author Batool Abuali upon request via: batoulabuali@gmail.com and batool_110906@svuonline.org.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Batool Abuali designed the research plan, reviewed the previous studies, wrote the literature review, collected the data, organized and ran the experiments, presented, analyzed, and interpreted the results, and prepared and wrote the draft manuscript. Prof. Mohamad-Bassam Kurdy supervised the study and aided in revising the manuscript.

Acknowledgments

The authors extend their appreciation to the visually impaired students and the management team at Light Initiative for helping run the experiments.

Supplementary Materials

Audio file titled “output.wav” is an example of the speech synthesized by our proposed system. (*Supplementary Materials*)

References

- [1] G. A. Abandah, A. Graves, B. Al-Shagoor, A. Arabiyat, F. Jamour, and M. Al-Tae, “Automatic diacritization of Arabic text using recurrent neural networks,” *International Journal on Document Analysis and Recognition*, vol. 18, no. 2, pp. 183–197, 2015.
- [2] I. T. U. T. Recommendation, *Telephone Transmission Quality Subjective Opinion Tests. A Method for Subjective Performance Assessment of The Quality of Speech Voice Output Devices*, p. 85, International Telecommunication Union, Geneva, Switzerland, 1994.
- [3] A. M. Azmi and R. S. Almajed, “A survey of automatic Arabic diacritization techniques,” *Natural Language Engineering*, vol. 21, no. 3, pp. 477–495, 2015.
- [4] A. Fashwan and S. Alansary, “A rule based method for adding case ending diacritics for modern standard Arabic texts,” in *Proceedings of the 16th International Conference on Language Engineering*, Cairo, Egypt, December 2016.
- [5] S. Alansary, “Alserag: an automatic diacritization system for Arabic,” *Intelligent Natural Language Processing: Trends and Applications*, pp. 523–543, Springer, Cham, Switzerland, 2018.
- [6] M. S. Khorsheed, “Diacritizing Arabic text using a single hidden Markov model,” *IEEE Access*, vol. 6, pp. 36522–36529, 2018.
- [7] M. Alghamdi, Z. Muzaffar, and H. Alhakami, “Automatic restoration of Arabic diacritics: a simple, purely statistical approach,” *Arabian Journal for Science and Engineering*, vol. 35, no. 2, p. 125, 2010.
- [8] Y. Hifny, “Smoothing techniques for Arabic diacritics restoration,” in *Proceedings of the 12th International Conference on Natural Language Processing*, no. 1, Trivandrum, India, December 2012.
- [9] Y. Hifny, “Restoration of Arabic diacritics using dynamic programming,” in *Proceedings of the 2013 8th International Conference on Computer Engineering & Systems (ICCES)*, Cairo, Egypt, November 2013.
- [10] I. Zitouni, J. Sorensen, and R. Sarikaya, “Maximum entropy based restoration of Arabic diacritics,” in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, July 2006.
- [11] A. Fadel, I. Tuffaha, and M. Al-Ayyoub, “Arabic text diacritization using deep neural networks,” in *Proceedings of the 2019 2nd international conference on computer applications & information security (ICCAIS)*, Riyadh, Saudi Arabia, May 2019.
- [12] M. A. A. Rashwan, A. A. Al Sallab, H. M. Raafat, and A. Rafea, “Deep learning framework with confused sub-set resolution architecture for automatic Arabic diacritization,” *IEEE/ACM Transactions on Audio, Speech, And Language Processing*, vol. 23, no. 3, pp. 505–516, 2015.
- [13] Y. Belinkov and J. Glass, “Arabic diacritization with recurrent neural networks,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, September 2015.
- [14] A. Said, “A hybrid approach for Arabic diacritization,” in *Proceedings of the 18th International Conference on*

- Application of Natural Language to Information Systems*, Salford, UK, June 2013.
- [15] A. Fashwan and S. Alansary, "SHAKKIL: an automatic diacritization system for modern standard Arabic texts," in *Proceedings of the Third Arabic Natural Language Processing Workshop*, Valencia, Spain, April 2017.
 - [16] Y. Hifny, "Hybrid LSTM/MaxEnt networks for Arabic syntactic diacritics restoration," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1515–1519, 2018.
 - [17] H. Abbad and S. Xiong, "Multi-components system for automatic Arabic diacritization," *Advances in Information Retrieval*, vol. 12035, p. 341, 2020.
 - [18] N. K. Bakhsh, S. Alshomrani, and I. Khan, "A comparative study of Arabic text-to-speech synthesis systems," *International Journal of Information Engineering and Electronic Business*, vol. 6, no. 4, pp. 27–31, 2014.
 - [19] A. Amrouche, L. Falek, and H. Teffahi, "Design and implementation of a diacritic Arabic text-to-speech system," *The International Arab Journal of Information Technology*, vol. 14, no. 4, 2017.
 - [20] T. Zerrouki, M. M. A. Shquier, A. Balla, N. Bousbia, I. Sakraoui, and F. Boudardara, "Adapting espeak to Arabic language: converting Arabic text to speech language using espeak," *International Journal of Reasoning-based Intelligent Systems*, vol. 11, no. 1, p. 76, 2019.
 - [21] Y. Ning, S. He, Z. Wu, C. Xing, and L. J. Zhang, "A review of deep learning based speech synthesis," *Applied Sciences*, vol. 9, no. 19, p. 4050, 2019.
 - [22] A. H. Ali, "Arabic speech synthesis using deep neural networks," in *Proceedings of the 2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Sharjah, UAE, March 2021.
 - [23] I. Abu Doush, F. Alkhatib, and A. A. R. Bsoul, "What we have and what is needed, how to evaluate Arabic Speech Synthesizer?" *International Journal of Speech Technology*, vol. 19, no. 2, pp. 415–432, 2016.
 - [24] A. Barqawi and T. Zerrouki, *Shakkala, Arabic Text Vocalization*, <https://github.com/Barqawiz/Shakkala>, 2017.
 - [25] Y. Sharief, *End to End Arabic TTS System Based on Tacotron*.
 - [26] M. Alghamdi, "Database of Arabic sounds: words," in *Proceedings of the 15th National Computer Security Conference*, Baltimore, MA, USA, October 1992.