NATIQ: AN END-TO-END TEXT-TO-SPEECH SYSTEM FOR ARABIC

Ahmed Abdelali[1], Nadir Durrani[1], Cenk Demiroglu[2],
Fahim Dalvi[1], Hamdy Mubarak[1], Kareem Darwish[1]

[1]Qatar Computing Research Institute - Hamad Bin Khalifa University, Doha, Qatar
[2]Özyeğin University, Istanbul, Türkiye

ABSTRACT

NatiQ is end-to-end text-to-speech system for Arabic. Our speech synthesizer uses an encoder-decoder architecture with attention. We used both tacotron-based models (tacotron-1 and tacotron-2) and the faster transformer model for generating mel-spectrograms from characters. We concatenated Tacotron1 with the WaveRNN vocoder, Tacotron2 with the WaveGlow vocoder and ESPnet transformer with the parallel wavegan vocoder to synthesize waveforms from the spectrograms. We used in-house speech data for two voices: 1) neutral male "Hamza"- narrating general content and news, and 2) expressive female "Amina"- narrating children story books to train our models. Our best systems achieve an average Mean Opinion Score (MOS) of 4.21 and 4.40 for Amina and Hamza respectively.The objective evaluation of the systems using word and character error rate (WER and CER) as well as the response time measured by real-time factor favored the end-to-end architecture ESPnet.NatiQ demo is available online at `https://tts.qcri.org`.

1. INTRODUCTION

Text to speech (TTS) is among the technologies that enables many solutions across different sectors. In the current pandemic time, education system is challenged with the new norm of distance and remote education. Teachers are not able to provide needed attention and support for every student; more precisely for lower elementary schools where students are very dependent on the teacher's guidance to follow the instructions. TTS can elevate some of this burden by allowing the young children to hear the content and have it read to them in a very fluent and pleasing voice. Advances in Neural technology allow achieving more natural voice compared to previous technologies [1].

We present **NatiQ**, an end-to-end speech system for Arabic. The system is composed of two independent modules: i) the web application and ii) the speech synthesizer. The web application uses *React Javascript* framework to handle dynamic User Interface and *MangoDB* to handle session related information. The system is built upon modern web technologies, allowing it to run cross-browsers and platforms. Figure 1 presents a screenshot of the interface.

Our best synthesizer is based on ESPnet Transformer TTS [2] architecture that takes input characters in an encoder-decoder framework to output mel-spectograms. The intermediate form is then converted into wav form using the Generative Adversarial Networks vocoder WaveGAN [3, 4]. We explored additional architecture including Tacotron1 [5] and 2 [6] and for vocoders WaveRNN [7] and WaveGlow [8] to synthesize waveforms from the decoded mel-spectograms.

We built two in-house speech corpora *Amina* – a female speaker with expressive narration and *Hamza* – a male speaker with neutral narration. The former is targeted towards education and the latter is more suitable to broadcast media.

Given that Arabic is typically written with no short vowels, this required to include additional processing to the text before exploiting it in the training. In addition to the short vowels restoration, diacritization, the pre-processing steps involves segmentation, transcript matching, voice normalization and silence reduction. We will further describe the pipeline and the architecture in detail. The resulting systems were evaluated using both objective and subjective approaches employing automatic metrics such as CER and WER; and using MOS. Lastly, the systems were assessed with Real-time Factor to evaluated decoding speed of each model.

2. SYSTEM ARCHITECTURE

Our NatiQ system is a web-based demonstration that is composed of two main components:

2.1. Web Application

The web application has two major components; the frontend and the backend. The frontend is created using the React Javascript framework to handle the dynamic User Interface (UI) changes such updates in generation. The backend is built using NodeJS and MongoDB to handle sessions, data associated with these sessions, communication with models, request inference and authentication. The frontend presents the user

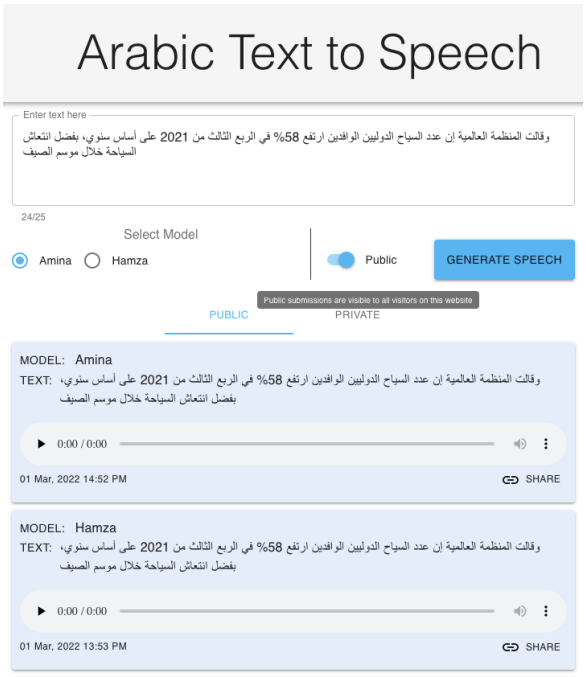

Fig. 1. NatiQ system in action

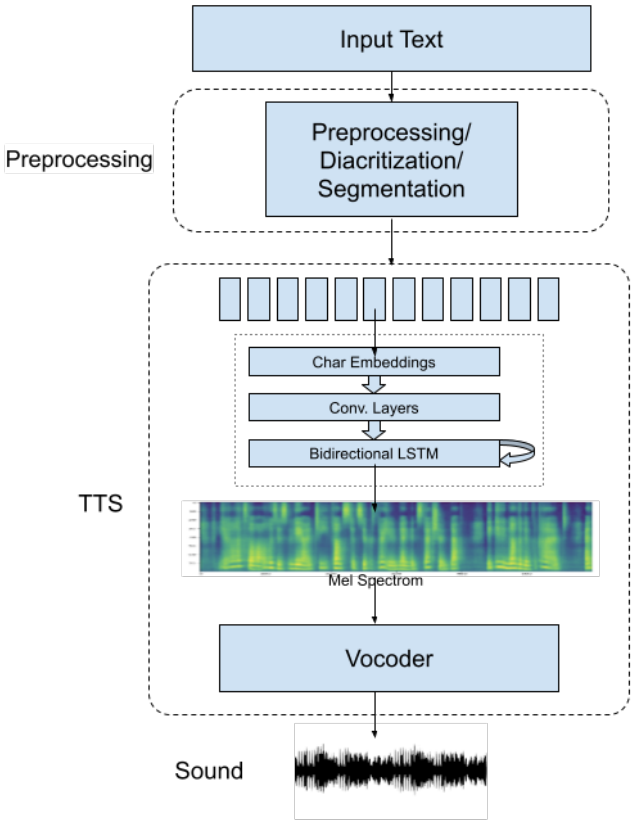

Fig. 2. NatiQ Architecture.

with an input text box and choice of speakers to choose from. Figure 1 shows a screenshot for the frontend. The responses from the backend will be presented to the user in a wave form that the user can listen to or download.

2.2. Speech Synthesis

Now we will describe the overall architecture of our synthesis model. Figure 2 shows the system architecture. The preprocessing module involves converting the numbers, abbreviations and dates into their vocalized form using linguistic and custom rules. Next the text is vowelized using Farasa [9], which diacritize and restore short vowels using the syntactic structure of the sentence.

The synthesizer is an encoder-decoder model cascaded with a vocoder to generate the wave-forms. The former converts the preprocessed text into a mel-spectrum. The latter convert the melspectogram representation into a wave form. Below we describe different components of our model:

2.2.1. Data

We acquired high quality speech data recorded at a sampling rate of 44kHz from two speakers. A female speaker *Amina* was recorded reading selected passages mainly from children books in Modern Standard Arabic. The data contains 3964 segments and 50,714 words in total. The style for this recording is expressive. The second data *Hamza* was recorded by a male speaker and in neutral style. This data contains 6005 segments and 80,409 words in total. Figure 3 shows the segments length distribution for each of the speakers. For both of the speakers, the average length of the segments is around 7 seconds or around 12 words per segment.

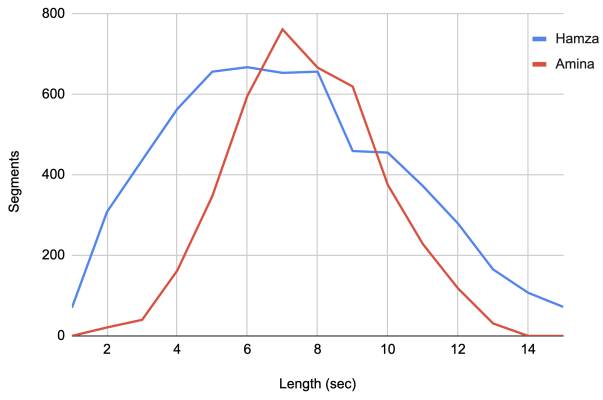

Fig. 3. Distribution of segments lengths for each speaker

2.2.2. Preprocessing

Data preprocessing steps involve: i) diacritization, ii) speech transcript matching, iii) segmentation, and iv) vowel normalization and silence reduction.

Diacritization Arabic has two types of vowels; namely long vowels, which are explicitly written in the text, and short vowels (aka diacritics) which are typically omitted in modern writings as native speakers can infer them based on contextual information. In order to read Arabic words properly, readers need to restore the missing diacritics and this is important for machines to pronounce the text correctly.

We diacritized the text using `Farasa` [9]. Although Farasa gives an accuracy above 94% the automatic diacritized data was, neverthless, reviewed by a language expert to ensure the accuracy of the annotations. This is important as some cases (for example named entities and foreign words) are often even challenging for a native speaker let alone for the automatic system. It's worth mentioning that we built a text normalization layer to convert digits, abbreviations, and special symbols to words to be fully diacritized by Farasa. Due to Arabic complexity and ambiguity, this conversion was not trivial in many cases.

Speech Transcript Matching Although native speakers don't require short vowels to correctly pronounce a word, in some rare cases they may make mistake of pronouncing a word with a wrong vowel. Rather than correcting the speaker which might require going back to the studio and re-record the segment again, we opted to change the transcript in such cases to reflect what was spoken. This will save both time and efforts required from the speaker and the recording studio.

Segmentation Due to the limitation of neural architectures to handle long audio samples [6], the data is sampled into frames of 10 seconds in average. The segmentation has to consider the sentence boundaries and not to break nor the context or the prosody. In general cases, long silences between segments is a good indicator but exception were found when related context or supplemental material that is still considered a part of the sentence still comes after a long pause.

Text Normalization This includes spelling out numbers, fractions, abbreviations and titles into their textual format such as "16.43" to "ستة عشر وأربعة وثلاثين جزء من المئة" (stp Ecr wOrbEp wvlAvyn jzC mn AlmQp)[1] or "وقال أ. د. ماجد" (wqAl O. d. mAjd) to "وقال الأستاذ الدكتور ماجد" (wqAl AlOstAV Aldktwr mAjd).

2.2.3. Models

We trained three models based on Tacotron-1 [5], Tacotron-2 [6] and Transformer TTS [2] recipes. The choice of these models was driven mainly by two main goals: Real-time decoding and high-quality voice.

[1]Using Safe Buckwalter Arabic encoding

Model Tacotron1 builds on top of RNN sequence-to-sequence architecture. It includes an encoder, an attention-based decoder, and a post-processing module. The former takes text as characters and generates a mel-spectrogram. The post-processing module then generates waveform from the mel-spectogram. Tacotron1 uses a CBHG-based encoder which consists of a bank of 1-D convolutional filters, followed by highway networks and a bidirectional gated recurrent unit (GRU). The decoder is a content-based tanh attention decoder that generates an 80-band mel-scale spectrogram as the target. Finally we use **WaveRNN** [7] on top to generate waveforms from the generated mel-spectograms. WaveRNN is a single layered RNN network that generates raw audio samples.

Model Tacotron2 follows the same recipe as Tacotron1 i.e. RNN-based sequence-to-sequence encoder-decoder architecture, it consists of a bi-directional LSTM-based encoder and a unidirectional LSTM-based decoder with location sensitive attention [10]. Additionally, the models employs different vocoder to generate waveforms. We used the **WaveGlow** [8], a flow-based network capable of generating high quality speech from melspectograms. WaveGlow is a generative model that generates audio by sampling from zero mean spherical Gaussian distribution. It uses 12 coupling layers and 12 invertible 1×1 convolutions.

Model ESPnet Transformer TTS Inspired by Neural Machine Translation, Transformer TTS [2] adapts multi-head self-attention mechanism and feed forward strategy to build an encoder-decoder model that would convert a sequence of inputs characters into an output sequence of acoustic features (log Mel-filter bank features), the model provide an adventage over the former models in the training speed as it uses a feed forward network compared to recurrent network based-models. Similarly to Tacotron1 and Tacotron2 models, Transformer TTS requires a vocoder to further convert the Mel features into wave form. We used Parallel WaveGAN [4] a non-autoregressive WaveNet that uses generative adversarial network to convert the Mel-filter bank sequences to a waveform.

3. EVALUATION

To evaluate the performance of each of the models, We built an evaluation test set composed of 100 sentences of varying lengths, collected from six domains including: Culture, Economy, Literature, Politics, Sports, and Technology. The sentences were collected between Jan 1st to Jan 20th, 2022. They include excerpts from current topics and news. We decoded each sentence using the three models and for each of the voices. This resulted in a pool of 600 audio files to evaluate. We carried automatic and manual (subjective) evaluations described below:

3.1. Automatic Evaluation

We used state-of-the-art Arabic ASR system [11] to decode the audio files generated by our TTS models. The ASR system gives state of the art performance on a number of standard data sets such as MGB-3 [12] and MGB-5 [13]. We then compare the generated transcripts against the input sentences for which TTS outputs are generated. As the ASR system generates unvowelized text, we strip short vowels from the reference original text to allow a fair comparison. We used standard evaluation metrics Word Error Rate (WER) and Charecter Error Rate (CER). Table 1 shows the results using the automatic approach. The system built using ESPnet2 gave the lowest WER and CER. Additionally, the neutral voice "Hamza" achieved a lower error rate when compared to the expressive "Amina". This highlights the challenges dealing with non-monotonic voices which are typically richer and has more features that the network needs to capture [14]. For Amina, Tacotron1 results are not worse than the leading ESPnet2 system; which potentially means that Tacotron1 is better at handling richer features. Tacotron2 suffers more from deletion, and substitution errors, this is the main cause for the CER/WER to be higher than other models.

3.2. Qualitative Evaluation

We recruited 14 individuals (7 females and 7 males) to carry the manual subjective evaluation. The participants were instructed to listen to the audio and give their opinion on the speech quality using a scale from 1 to 5; The five-category MOS scale [15]: 5 = excellent, 4 = good, 3 = fair, 2 = poor, 1 = bad. Each participants was presented with a set of 15 random samples from the pool. The overall results presented in Table 2 shows that the participants favored ESPNet:Hamza and Tacotron1:Amina. The results of ESPNet:Hamza are very comparable to the Tacotron1:Hamza. The results also shows that participants preferred the neutral voice over expressive one. Literature also reports that typically evaluators prefer neutral over expressive and expressivity is better perceived when the samples have a high quality [16]. The qualitative results are closely aligned with automatic evaluation, the differences in CER/WER between ESPNet:Amina and Tacotron1:Amina are less pronounced when compared to Hamza.

3.3. Speed

Lastly, another metric to evaluate the system, we used Real-time Factor (RTF): the ratio of the speech generation time to the utterance duration. Such measure is very crucial and essential in the deployment of any system, especially for real-time use. For a system to be considered real-time, RTF should be $<= 1$ [17]. Having a low RTF, will ensure that the system latency is reasonable and acceptable and indicate that the system can be used in real-time applications. Table 3 shows the average RTF for the three systems running on a 4 Cores Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz and 32Gb of RAM and powered by NVIDIA Tesla V100 SXM2 32Gb GPU. The end-to-end ESPnet2 system, is the clear winner with a an RTF equal to 0.09 which is 1.5 and 17 times faster than Tacotron2 and Tacotron1 respectively. None of the systems run real-time on CPU. Our fastest system ESPnet2 runs at a speed of 4.24xRT on CPU.

	Amina		Hamza	
	CER	WER	CER	WER
ESPnet2	**17.47**	**40.42**	**8.01**	**24.87**
Tacotron1	22.51	43.98	27.48	46.12
Tacotron2	40.76	64.80	82.38	93.62

Table 1. CER and WER evaluation results for the three systems.

	Amina	Hamza
ESPnet2	3.57	4.40
Tacotron1	4.21	4.38
Tacotron2	3.49	2.34

Table 2. MOS evaluation results for the three systems.

	RTF	
Model	GPU	CPU
ESPnet2	**0.09**	4.24
Tacotron1	1.66	-
Tacotron2	0.14	-

Table 3. Realtime Factor evaluation results.

4. CONCLUSION

We presented NatiQ Arabic text-to-speech system, a system based on end-to-end framework that combines Transformer encoder-decoder and WaveGAN vocoder. The system was evaluated using subjective metric, Mean Opinion Score and objective Speed, WER and CER. The system achieved a MOS of 4.35 and 4.72 for Amina and Hamza respectively. Such performance is very comparable to English systems [5, 6] . For the expressive speaker, the performance of the system still lags behind the neutral one. This is due to the complex and rich features encoded in expressive voice. We plan to explore different techniques that exploits the additional features in the voice such as [18] which aim to combine frames and style information as two objective functions to optimize while training the model.

5. REFERENCES

[1] Zvi Kons, Slava Shechtman, Alex Sorin, Carmel Rabinovitz, and Ron Hoory, "High quality, lightweight and adaptable tts using lpcnet," 2019.

[2] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019, AAAI'19/IAAI'19/EAAI'19, AAAI Press.

[3] Chris Donahue, Julian McAuley, and Miller Puckette, "Adversarial audio synthesis," 2019.

[4] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6199–6203, 2020.

[5] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous, "Tacotron: Towards end-to-end speech synthesis," 2017.

[6] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," 2018.

[7] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu, "Efficient neural audio synthesis," 2018.

[8] Ryan Prenger, Rafael Valle, and Bryan Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," 2018.

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

[10] Jing-Xuan Zhang, Zhen-Hua Ling, and Li-Rong Dai, "Forward attention in sequence- to-sequence acoustic modeling for speech synthesis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4789–4793.

[11] Amir Hussein, Shinji Watanabe, and Ahmed Ali, "Arabic speech recognition by end-to-end, modular systems and human," *Computer Speech & Language*, vol. 71, pp. 101272, 2022.

[12] Ahmed Ali, Stephan Vogel, and Steve Renals, "Speech recognition challenge in the wild: Arabic mgb-3," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 316–322.

[13] Ahmed Ali, Suwon Shon, Younes Samih, Hamdy Mubarak, Ahmed Abdelali, James Glass, Steve Renals, and Khalid Choukri, "The mgb-5 challenge: Recognition and dialect identification of dialectal arabic speech," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 1026–1033.

[14] Rafael Valle, Jason Li, Ryan Prenger, and Bryan Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," 2019.

[15] Rainer Guski, "Psychological methods for evaluating sound quality and assessing acoustic information," *Acta Acustica united with Acustica*, vol. 83, pp. 765–774, 09 1997.

[16] Marie Tahon, Gwénolé Lecorvé, Damien Lolive, and Raheel Qader, "Perception of expressivity in TTS: linguistics, phonetics or prosody?," in *Statistical Language and Speech Processing*, Le Mans, France, Oct. 2017, vol. 10583, pp. 262–274.

[17] Vineel Pratap, Qiantong Xu, Jacob Kahn, Gilad Avidov, Tatiana Likhomanenko, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert, "Scaling up online speech recognition using convnets," 2020.

[18] Rui Liu, Berrak Sisman, Guanglai Gao, and Haizhou Li, "Expressive tts training with frame and style reconstruction loss," 2020.