Dalarm

달달한 콤퓨터 달콤팀

강승군 김세희 남병욱 위광진

목차

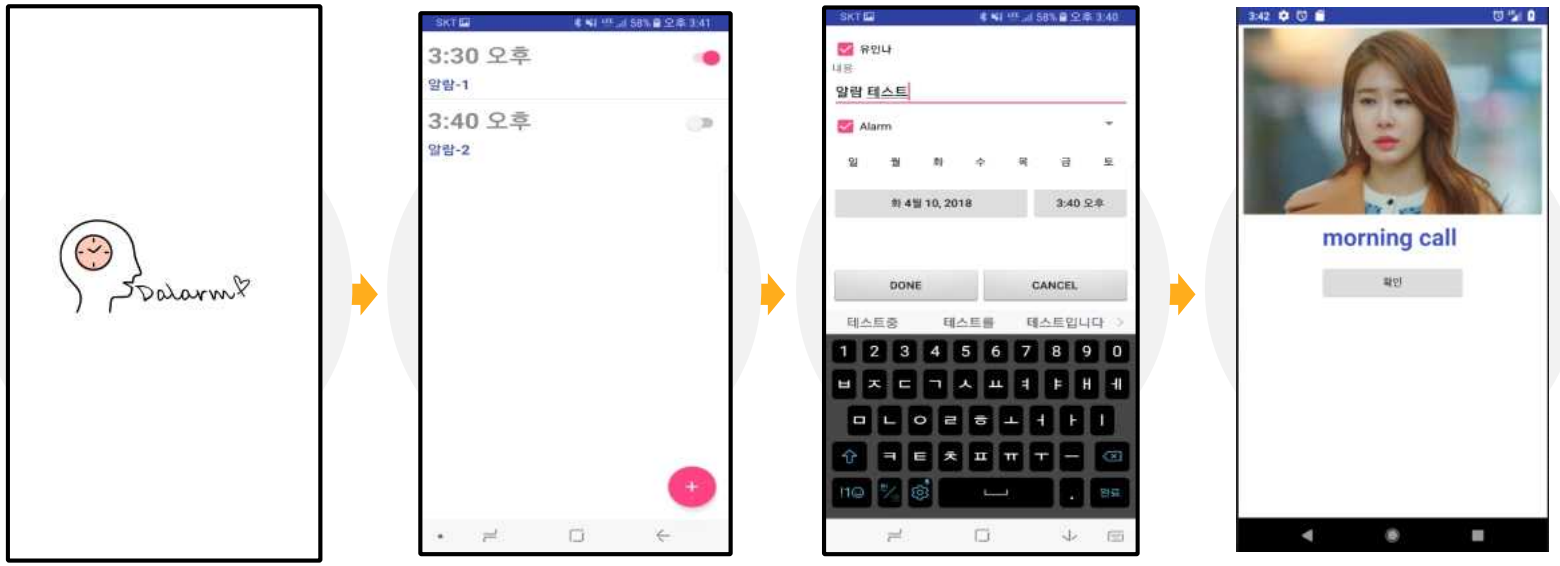
알람

유명인의 목소리를 이용

1. 사용자가 입력한 메세지를
   유명인의 목소리로 듣는 알람 어플리케이션
2. TTS(Text-to-Speech)를 위해서는
   유명인 목소리를 이용한 딥러닝 과정이 필요

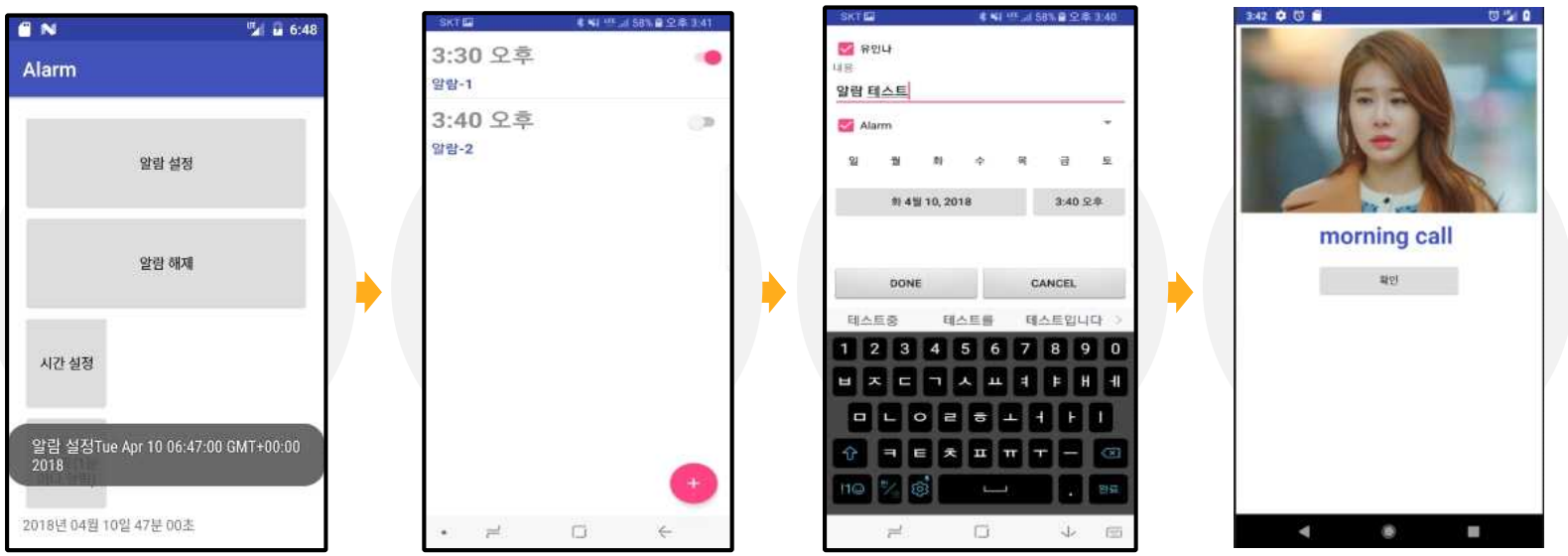
Dalarm 어플리케이션 UI by 김세희

- 어플리케이션 초기 logo splash 페이지 구현
- 알람 메인/추가/실행 페이지 제작
- 그 외 기능 페이지 제작
- 새로운 logo 제작



Dalarm 어플리케이션 기능 by 남병욱

- 기본적인 알람 기능 구현
- 알람 추가 기능 구현
- 토글 버튼을 사용한 알람 비/활성화 기능 구현
- 기타 필요 기능 구현(알람 삭제, 수정, 텍스트 보내기)

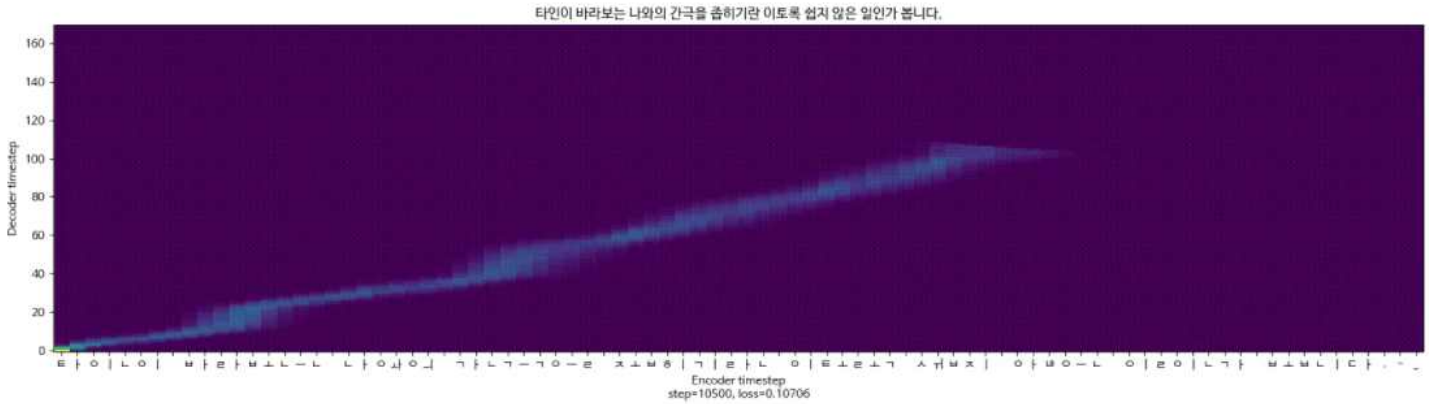


DB, 서버 구축 및 연동 by 강승군

- DB 설계 및 어플 연동
- Apache 서버 구축 및 서버와 어플 연동
- 앱에서 웹서버에 있는 음성파일에 접근하여 스트리밍하도록 구현
- python 서버에서 네이버TTS를 이용하여 음성파일을 생성하도록 구현
- 기타(초기 logo 제작, 영상편집, 피피티 제작)

딥러닝 개발 환경 구축 및 테스트 by위광진

- Tacotron을 사용하기 위해 필요한 개발 환경을 구축
- 손석희 dataset을 공개한 개발자의 github를 가져와서 음성합성 test를 수행
- tacotron 소스 파일 분석



딥러닝 공부 및 Tacotron 논문 스터디 by 모두

- 딥러닝 - CS231n 강의 듣기
- 각종 Tacotron 논문 스터디

TACOTRON: TOWARDS END-TO-END SPEECH SYNTHESIS

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, Rif A. Saurous*

Google, Inc.
{yxwang,rjryan,rif}@google.com

ABSTRACT

A text-to-speech synthesis system typically consists of multiple stages, such as a text analysis frontend, an acoustic model and an audio synthesis module. Building these components often requires extensive domain expertise and may contain brittle design choices. In this paper, we present Tacotron, an end-to-end generative text-to-speech model that synthesizes speech directly from characters. Given <text, audio> pairs, the model can be trained completely from scratch with random initialization. We present several key techniques to make the sequence-to-sequence framework perform well for this challenging task. Tacotron achieves a 3.82 subjective 5-scale mean opinion score on US English, outperforming a production parametric system in terms of naturalness. In addition, since Tacotron generates speech at the frame level, it's substantially faster than sample-level autoregressive methods.

1 INTRODUCTION

Modern text-to-speech (TTS) pipelines are complex (Taylor, 2009). For example, it is common for statistical parametric TTS to have a text frontend extracting various linguistic features, a duration model, an acoustic feature prediction model and a complex signal-processing-based vocoder (Zen et al., 2009; Agiomyrgiannakis, 2015). These components are based on extensive domain expertise and are laborious to design. They are also trained independently, so errors from each component may compound. The complexity of modern TTS designs thus leads to substantial engineering efforts when building a new system.

There are thus many advantages of an integrated end-to-end TTS system that can be trained on <text, audio> pairs with minimal human annotation. First, such a system alleviates the need for laborious feature engineering, which may involve heuristics and brittle design choices. Second, it more easily allows for rich conditioning on various attributes, such as speaker or language, or high-level features like sentiment. This is because conditioning can occur at the very beginning of the model rather than only on certain components. Similarly, adaptation to new data might also be easier. Finally, a single model is likely to be more robust than a multi-stage model where each component's errors can compound. These advantages imply that an end-to-end model could allow us to train on huge amounts of rich, expressive yet often noisy data found in the real world.

TTS is a large-scale inverse problem: a highly compressed source (text) is "decompressed" into audio. Since the same text can correspond to different pronunciations or speaking styles, this is a particularly difficult learning task for an end-to-end model: it must cope with large variations at the signal level for a given input. Moreover, unlike end-to-end speech recognition (Chan et al., 2016)

*These authors really like tacos.
†These authors would prefer sushi.

Figure 1: *Model architecture. The model takes characters as input and outputs the corresponding raw spectrogram, which is then fed to the Griffin-Lim reconstruction algorithm to synthesize speech.*

or machine translation (Wu et al., 2016), TTS outputs are continuous, and output sequences are usually much longer than those of the input. These attributes cause prediction errors to accumulate quickly. In this paper, we propose Tacotron, an end-to-end generative TTS model based on the sequence-to-sequence (seq2seq) (Sutskever et al., 2014) with attention paradigm (Bahdanau et al., 2014). Our model takes characters as input and outputs raw spectrogram, using several techniques to improve the capability of a vanilla seq2seq model. Given <text, audio> pairs, Tacotron can be trained completely from scratch with random initialization. It does not require phoneme-level alignment, so it can easily scale to using large amounts of acoustic data with transcripts. With a simple waveform synthesis technique, Tacotron produces a 3.82 mean opinion score (MOS) on an US English eval set, outperforming a production parametric system in terms of naturalness†.

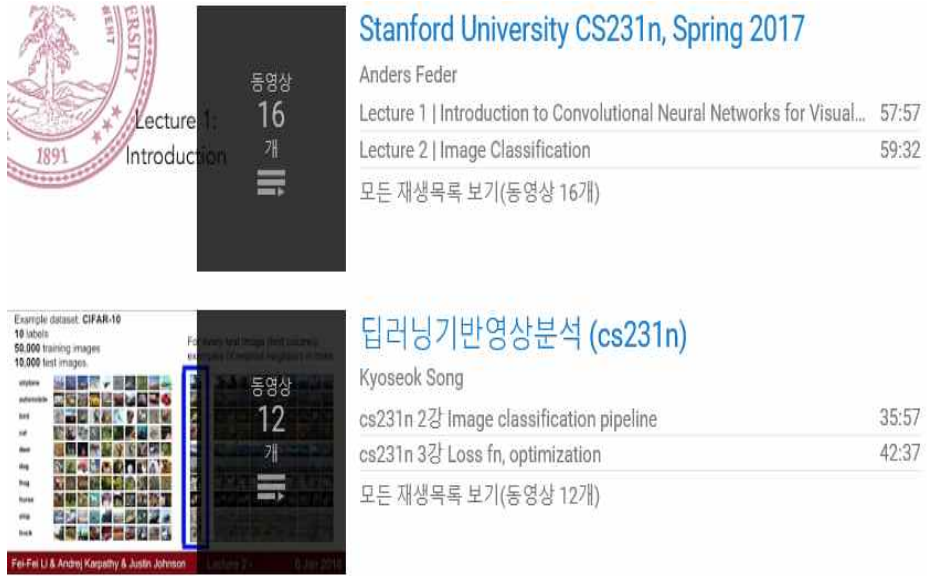
2 RELATED WORK

WaveNet (van den Oord et al., 2016) is a powerful generative model of audio. It works well for TTS, but is slow due to its sample-level autoregressive nature. It also requires conditioning on linguistic features from an existing TTS frontend, and thus is not end-to-end: it only replaces the vocoder and acoustic model. Another recently-developed neural model is DeepVoice (Arik et al., 2017), which replaces every component in a typical TTS pipeline by a corresponding neural network. However, each component is independently trained, and it's nontrivial to change the system to train in an end-to-end fashion.

To our knowledge, Wang et al. (2016) is the earliest work touching end-to-end TTS using seq2seq with attention. However, it requires a pre-trained hidden Markov model (HMM) aligner to help the seq2seq model learn the alignment. It's hard to tell how much alignment is learned by the seq2seq per se. Second, a few tricks are used to get the model trained, which the authors note hurts prosody. Third, it predicts vocoder parameters hence needs a vocoder. Furthermore, the model is trained on phoneme inputs and the experimental results seem to be somewhat limited.

Char2Wav (Sotelo et al., 2017) is an independently-developed end-to-end model that can be trained

Tacotron 논문 스터디



cs321n 강의 듣기

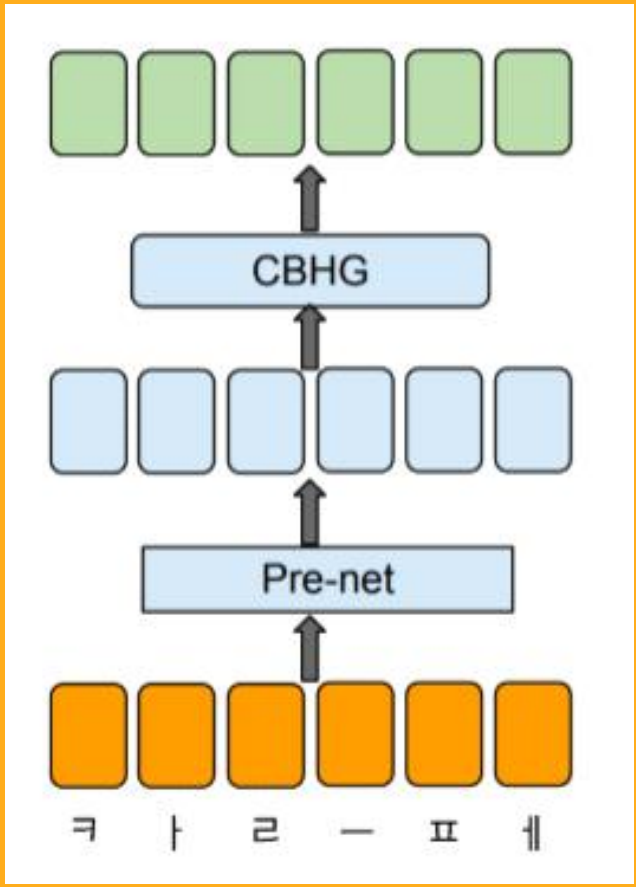
3. 딥러닝 학습

사용자가 입력한 메세지를 유명인의 목소리로 들려주는 TTS(Text-to-Speech)를 위해서는 유명인의 목소리를 딥러닝 시켜주는 과정이 필요합니다. 따라서 목소리와 그에 대응되는 글자를 대량으로 입력시킨 후 우리가 사용할 알고리즘 '타코트론'을 이용하여 딥러닝 학습을 진행합니다. 이 학습이 본 달람 프로젝트의 핵심이자 근본적인 챌린지입니다.

음성 합성 Process

Encoder → Decoder → Vocoder

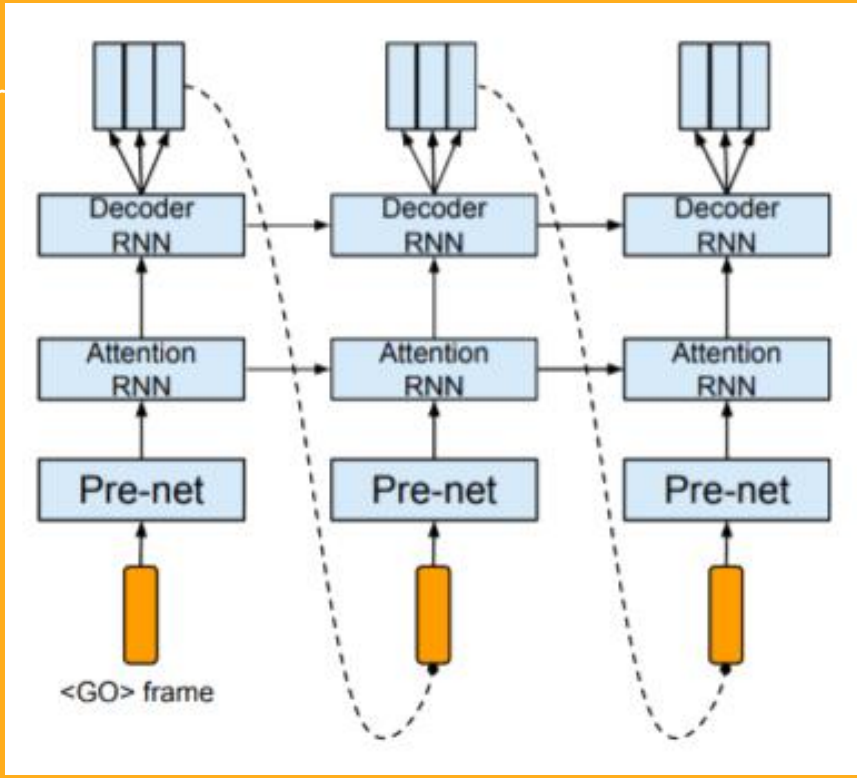
ENCODER

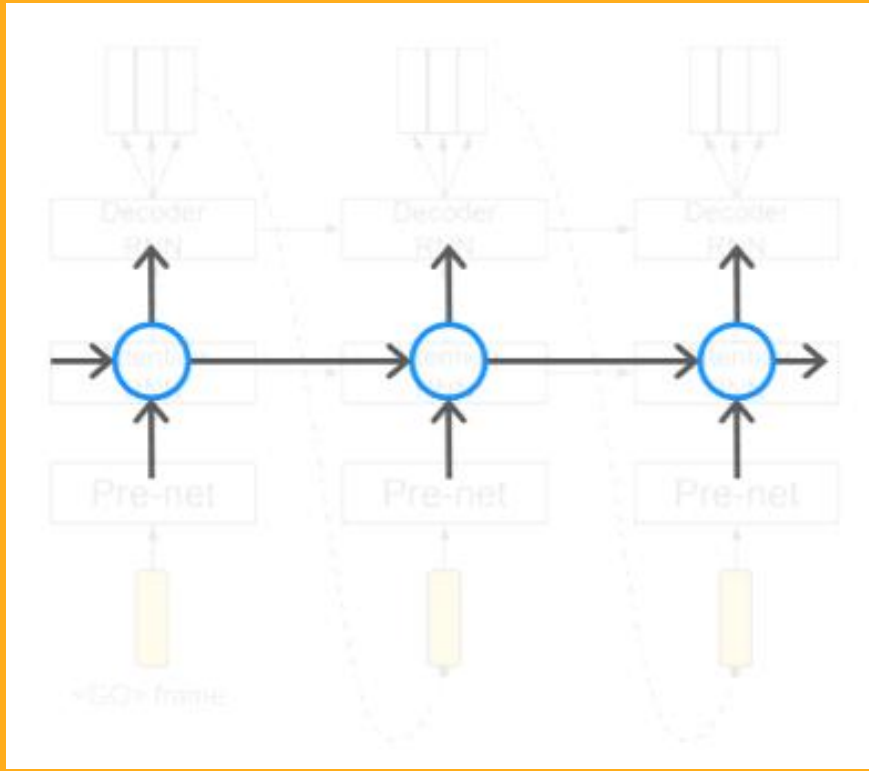


텍스트를 텍스트 정보를 잘 나타내는 숫자로 변환해주는 역할

DECODER

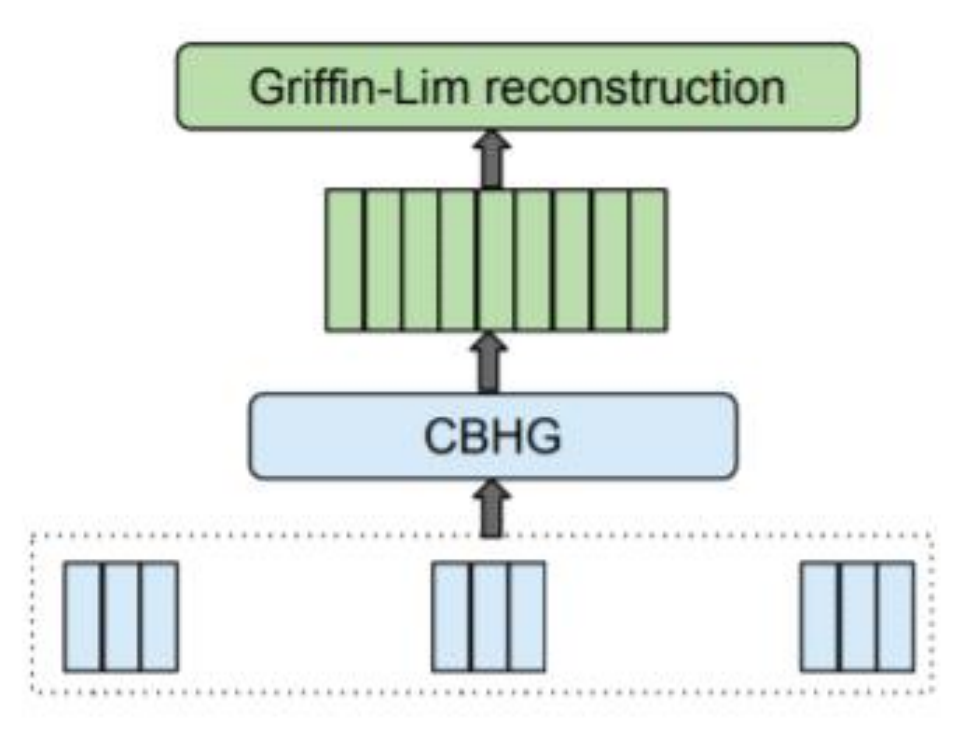
Encoder output





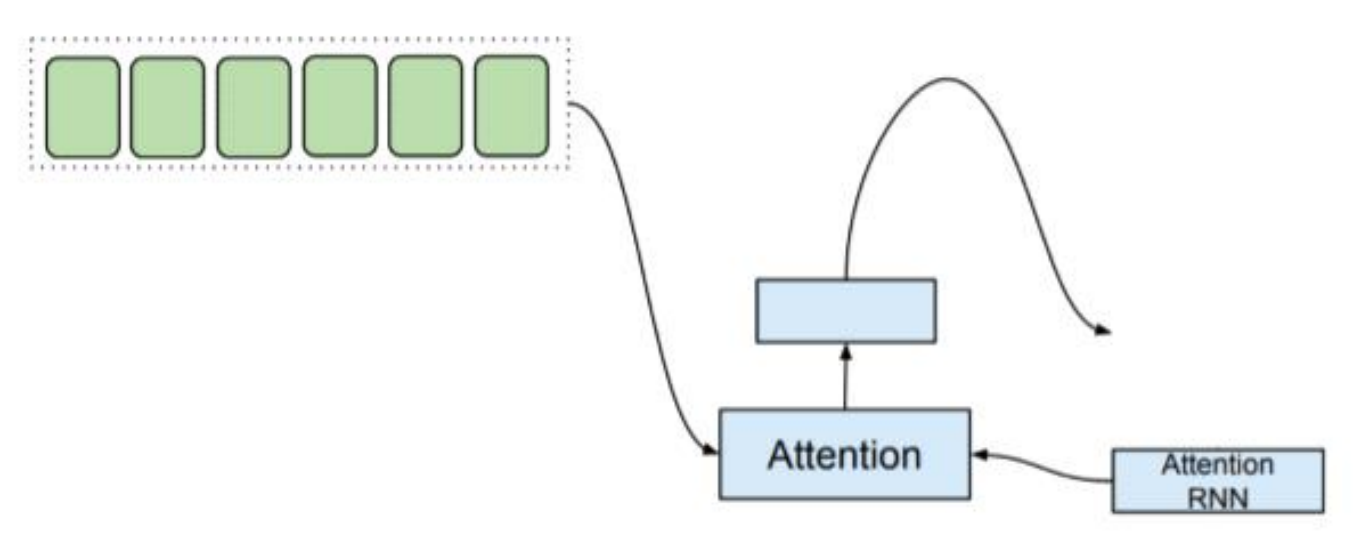
크게 보면 그냥 RNN

VOCODER



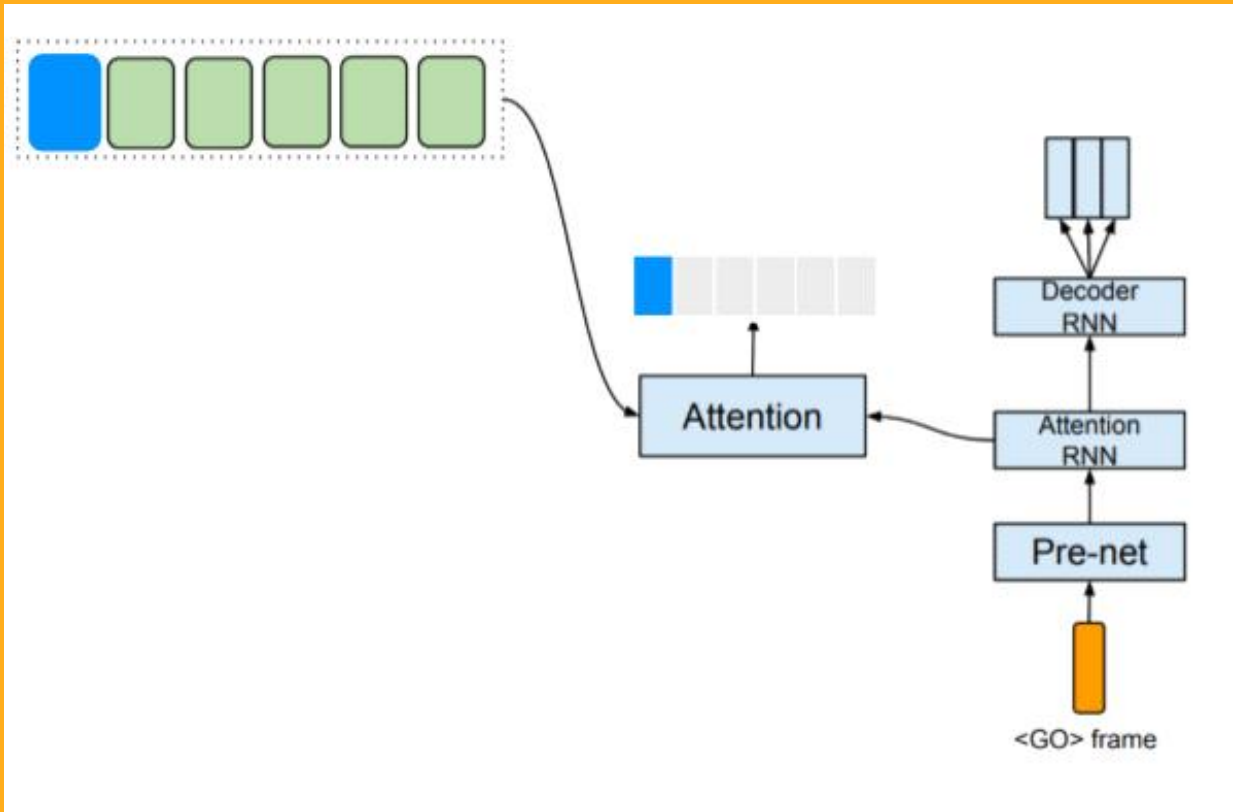
스펙트로그램을 음성으로 변환하는 부분이다.

ATTENTION

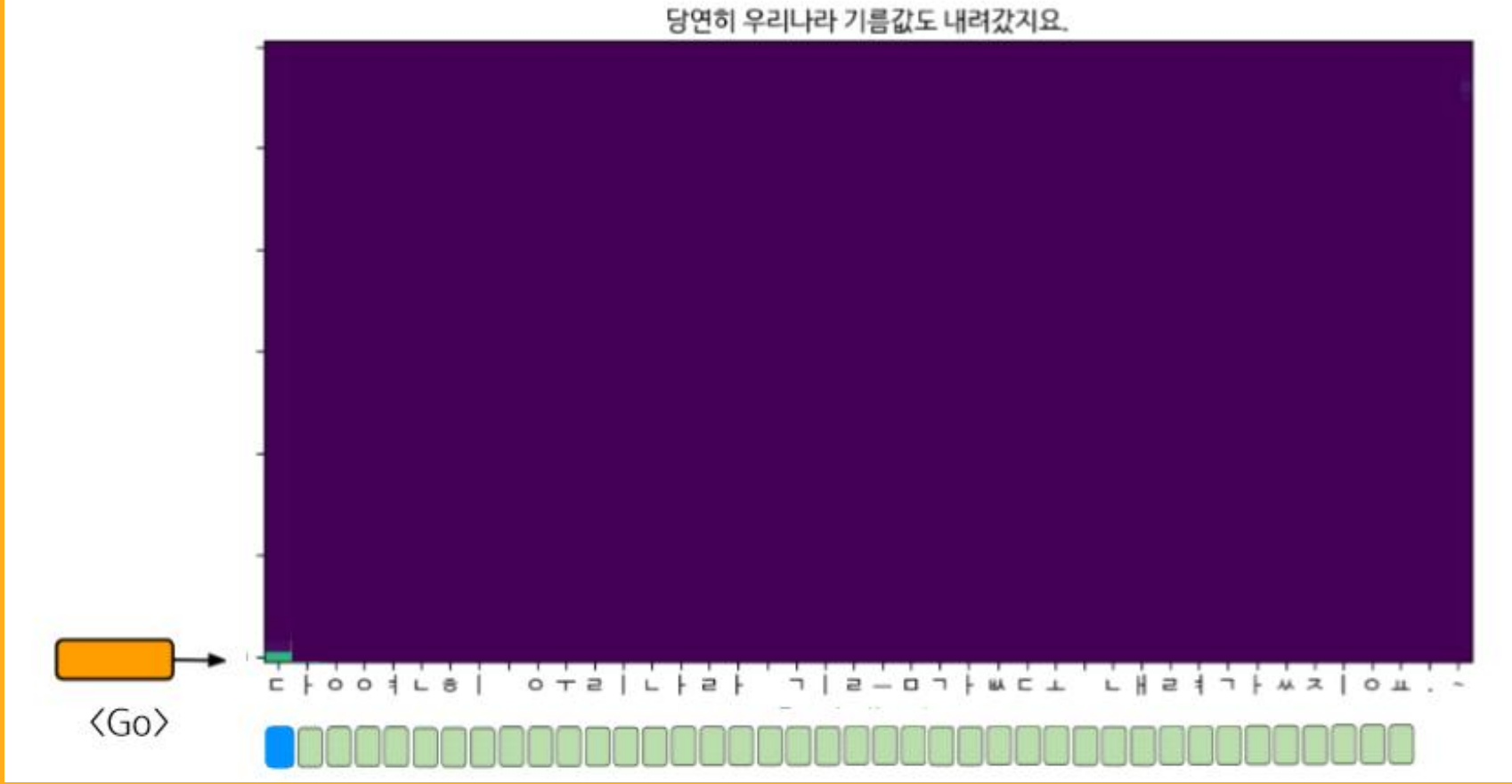


어디에 집중 할 것인가?

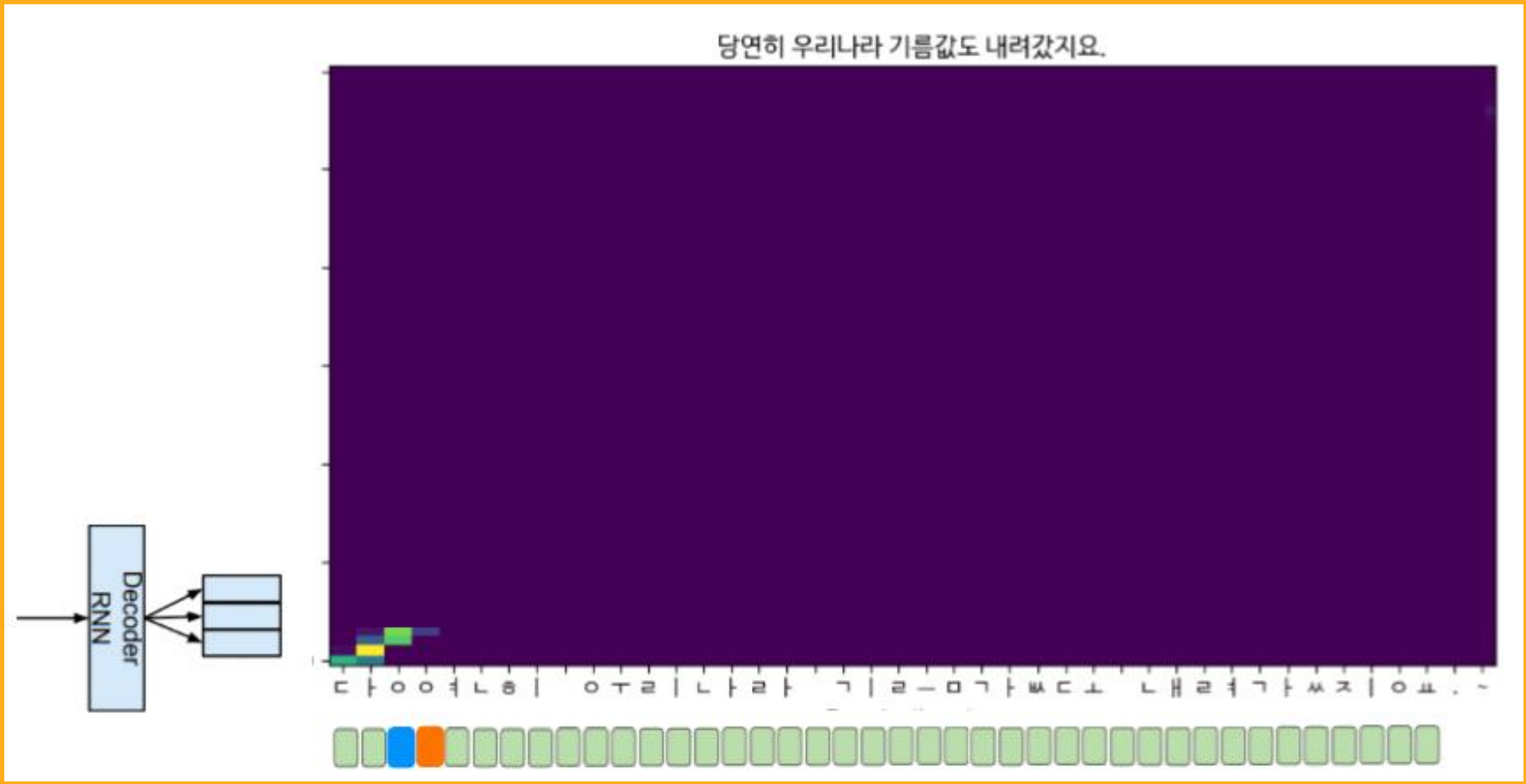
ATTENTION



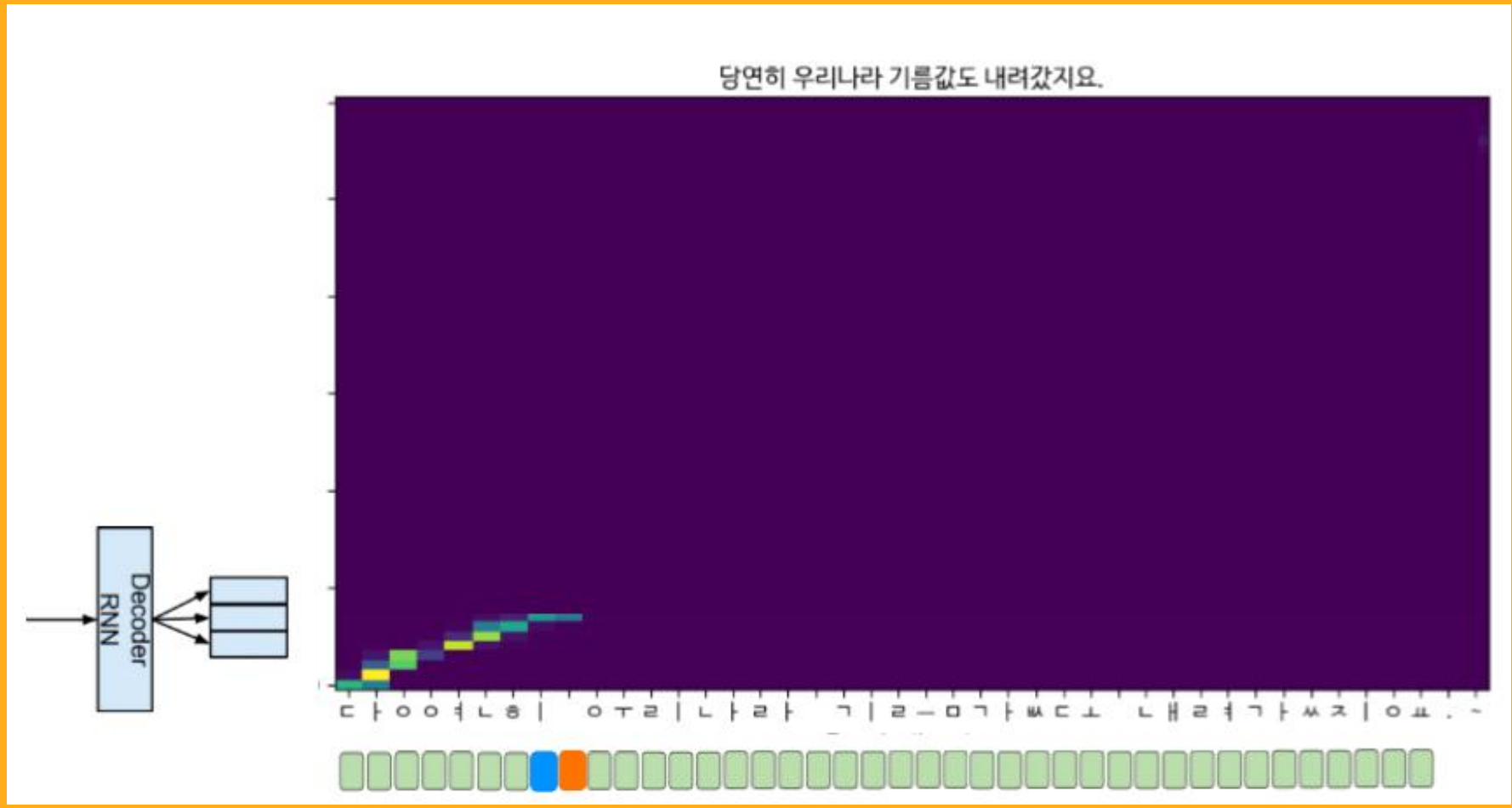
예 시



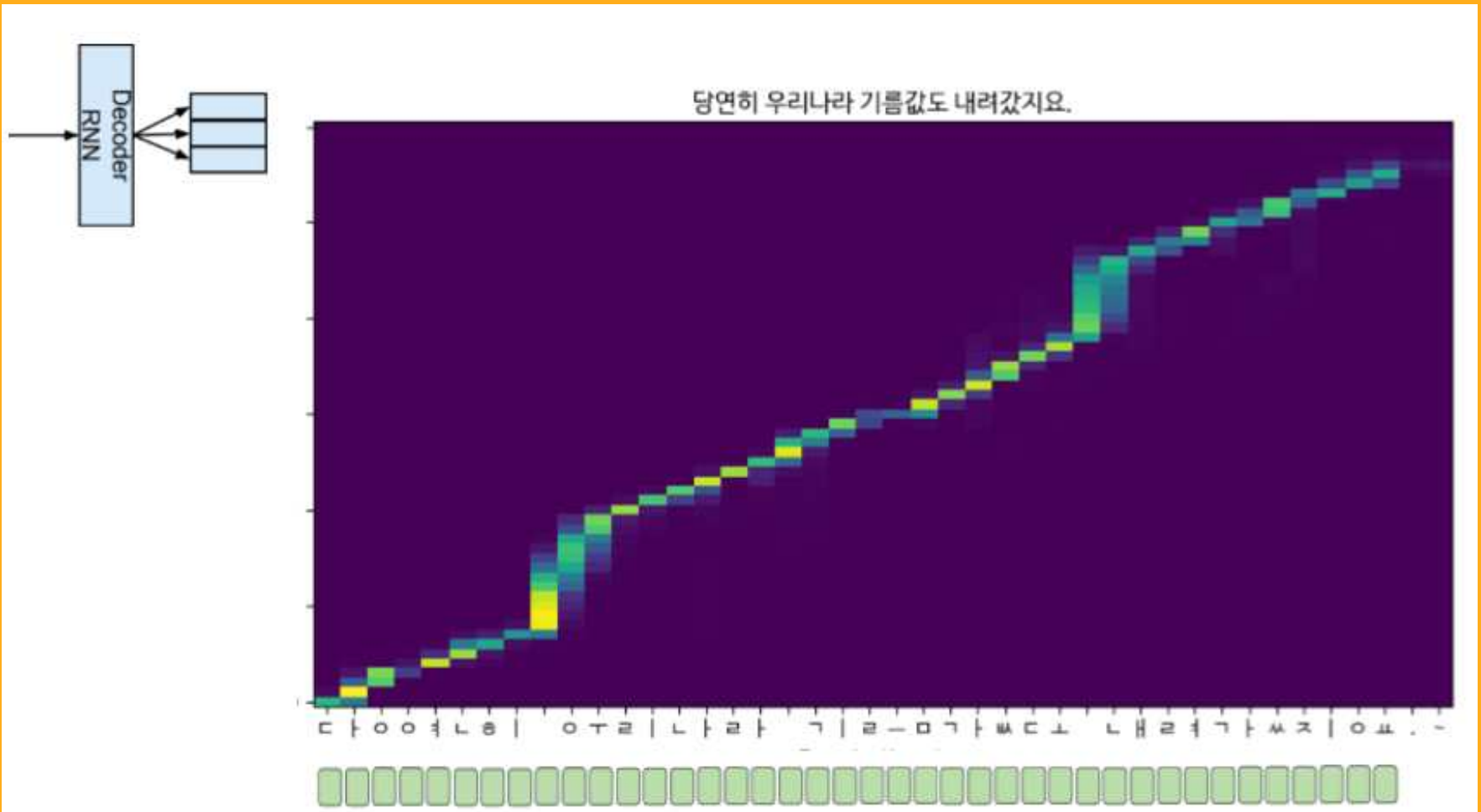
예 시



예 시



예 시




Made With
VivaVideo

Q. 보코더 등 하드웨어 구현 부분이 있는지?

Q. 목소리의 종류를 다양하게 하는데 무슨 어려움이 있는지?

Q. 본 업무를 진행할 때의 필요한 기술들에 대한 내용이 표현이 부족합니다. 아이디어를 실현하기 위한 기본 계획을 자세히 나타내길 바랍니다.

Q. 새로운 음성 합성을 위한 데이터 수집 및 딥러닝 학습 계획에 대해 보다 구체적인 계획이 필요합니다.

감사합니다