Polyphone disambiguation and accent prediction using pre-trained language models in Japanese TTS front-end

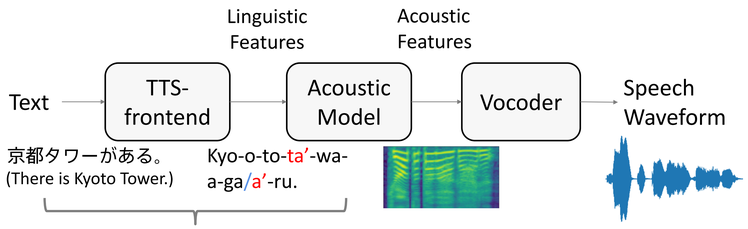
Rem Hida, Masaki Hamada, Chie Kamada, Emiru Tsunoo, Toshiyuki Sekiya, Toshiyuki Kumakura
Sony Group Corporation R&D Center, Remu.Hida@sony.com

Summary

- We propose a method for **polyphone disambiguation** and **accent prediction** in Japanese TTS front-end.
- Our proposed method combines **explicit features** extracted from morphological analysis and **implicit features** extracted from **pre-trained language models**.
- The combination of explicit and implicit features **improves both polyphone disambiguation and accent prediction performance**.
- The TTS system using our method **achieves a mean opinion score close to that of synthesized speech with ground truth pronunciation and accent**.
- The effectiveness of pre-trained language model type (BERT/Flair) depends on tasks.

TTS frontend

TTS frontend converts text to phonetic symbol sequences.



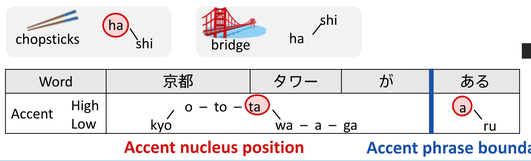
Highly language dependent
Japanese has mainly two characteristics.

Japanese Characteristics related to TTS

・Some Kanji have multiple candidate pronunciations, each corresponding to a different meaning.

このカレーは辛い。 ka-ra-i (spicy) / tsu-ra-i (hard) → **Polyphone disambiguation**

・Japanese is pitch (High/Low)-accent language.
Some words have the same pronunciation but different accents and meanings.



→ **Accent prediction**

Wrong pronunciation & accent lead wrong comprehension.
Japanese TTS system requires "**polyphone disambiguation(PD)**" and "**accent prediction(AP)**."

Motivation

Pronunciation and accent depend on context.

PD: このカレーはとても**辛い (ka-ra-i)**。体調が悪くてとても**辛い (tsu-ra-i)** 。
This curry is very **spicy**. I'm sick and it's very **hard**.

AP:

京都 (Kyoto)	タワー (tower)	上空 (above)
kyo'-o-to	ta'-wa-a	jo-o-ku-u
kyo-o-to-ta'-wa-a		jo-o-ku-u

Semantic relationship

However, existing methods only utilize local context.

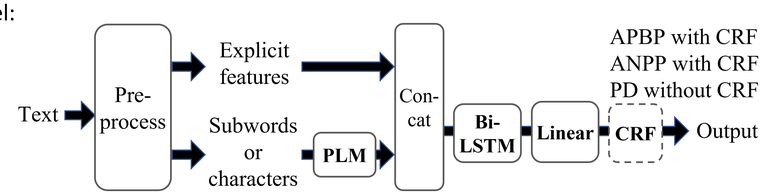
PD: KyTea[Neubig+,10] (pointwise prediction)

AP: TASET[Suzuki+,17] (linear-chain CRF)

How to take "longer/rich context" into account?
-> Using **Pre-trained Language Models**.

Japanese TTS-frontend with Pretrained Language Models

Model:



Features

Explicit(EF): features derived from morphological analysis

Word	京都	タワー	が	ある
POS	Noun	Noun	Particle	Verb
Original pronunciation	kyo-o-to	ta-wa-a	ga	a-ru
Accent nucleus position of each word	kyo — o – to	ta — wa – a	ga	a — ru
Other features	・・・	・・・	・・・	・・・

Explicit Features (EFs)

Implicit(PLM): features from Pretrained Language Models

BERT: subword based masked language model

Flair: character based bidirectional encoder

Explicit and implicit features are concatenated and input into BiLSTM.

Dataset for Experiments

・Polyphone disambiguation

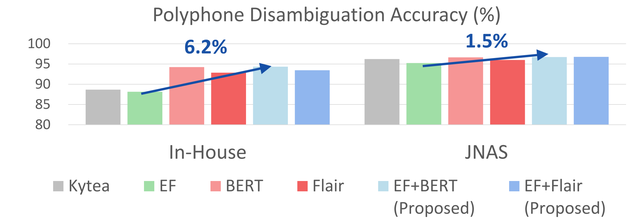
Focus on 92 frequently used polyphonic words

	#sentence	usage	Source
In-house	39,353 (24,117 / 5,156 / 10,080)	Train/dev/test	Wikipedia/TV captions/ novels/CSJ/JSUT
Public (JNAS)	5,642	test	JNAS

・Accent prediction

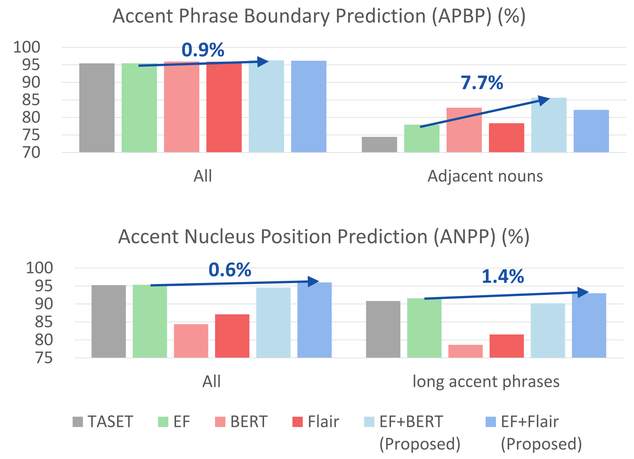
	#sentence	usage	Source
In-house	9,497 (7,768 / 864 / 865)	Train/dev/test	TV captions

Results on polyphone disambiguation



Bi-LSTM w/ EF+BERT improves by 6.2/1.5% from only EF

Results on accent prediction



Bi-LSTM w/ EF+PLM improves by 0.9/7.7% on APBP, 0.6/1.4% on ANPP from only EF.
BERT for APBP, Flair for ANPP

TTS quality subjective evaluation

TTS settings

Acoustic model: Tacotron2 w/ Global Style Token [Shen+,18. Wang+,18]

Vocoder: Parallel WaveGAN [Yamamoto+, 20]

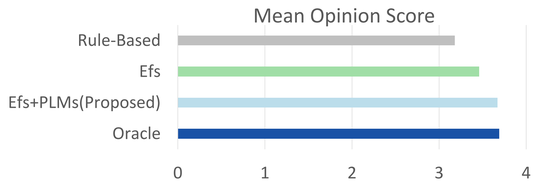
Training data: Sub-corpus of JSUT for both models

Evaluator

30 native Japanese Speakers

Evaluation text

25 utterance samples from in-house data



Proposed method achieved almost the same speech quality as Oracle.