Effect of choice of probability distribution, randomness, and search methods for alignment modeling in sequence-to-sequence text-to-speech synthesis using hard alignment

[1,2] Yusuke Yasuda, [1] Xin Wang, [1] Junichi Yamagishi

[1] National Institute of Informatics, Japan

[2] SOKENDAI, Japan

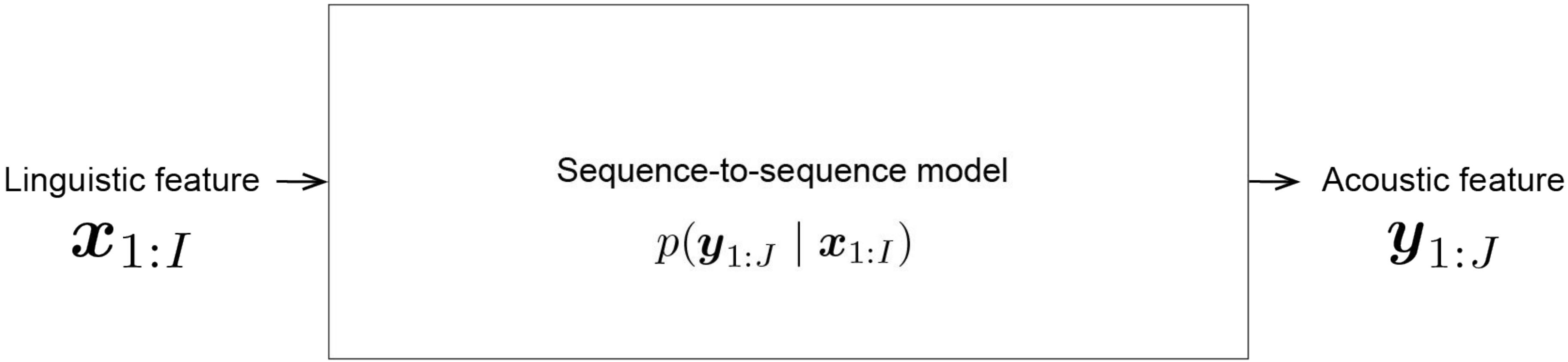
ICASSP 2020

Contents

- Introduction to SSNT-TTS
- Investigation of alignment prediction methods
  - Randomness
  - Search methods
  - Probability distributions
- Experiments
- Results
  - Randomness
  - Search methods
  - Probability distributions
- Conclusion

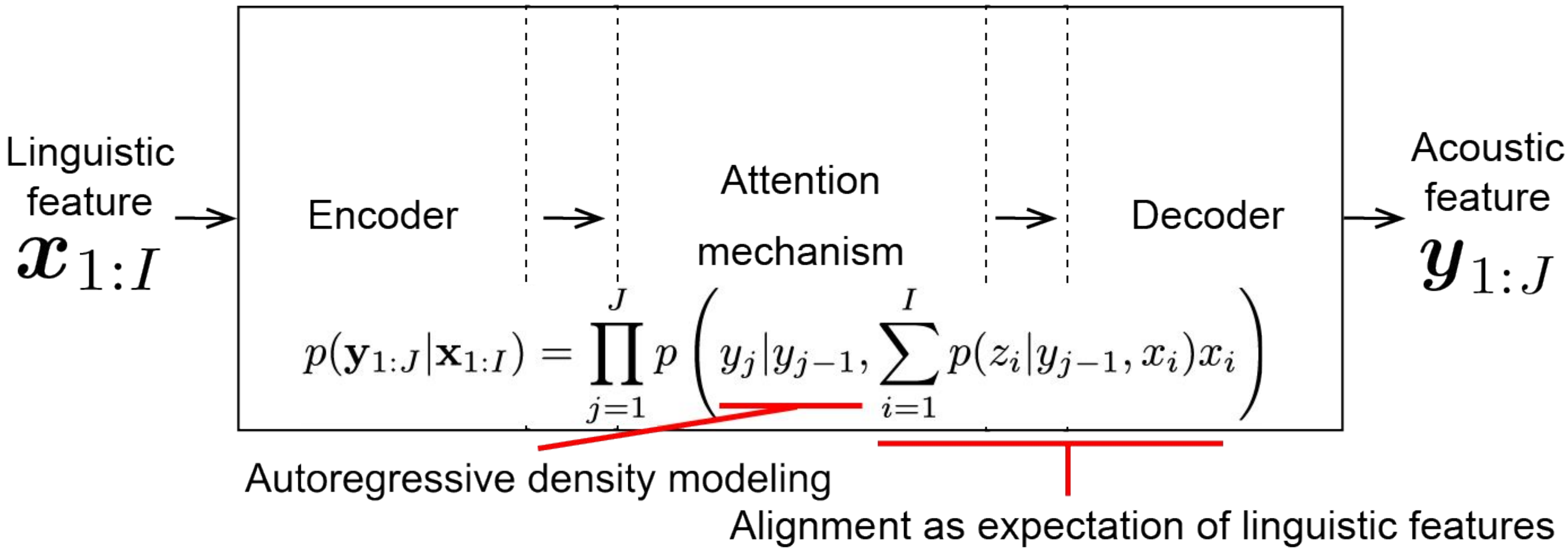
Introduction to SSNT-TTS

Sequence-to-sequence text-to-speech synthesis

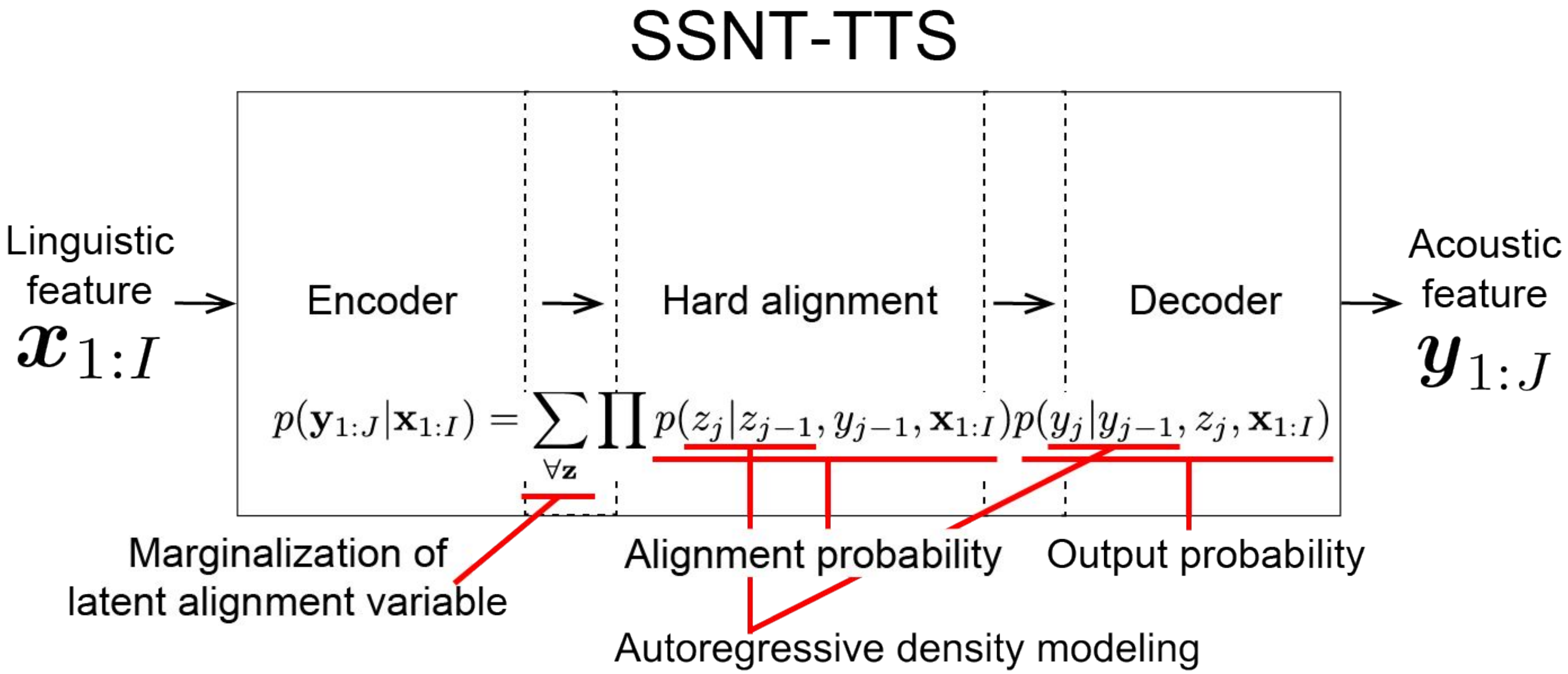


Tacotron vs SSNT-TTS: Tacotron

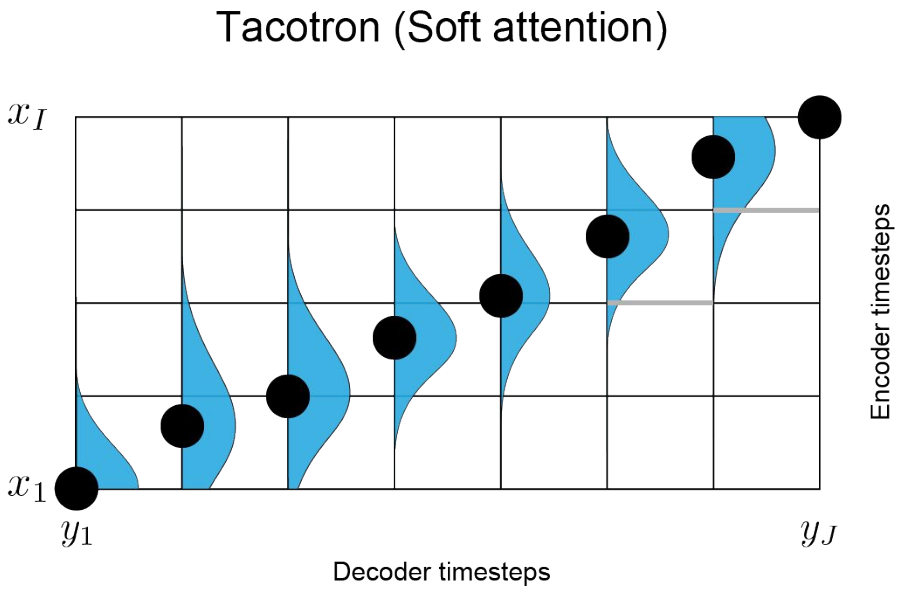
Tacotron

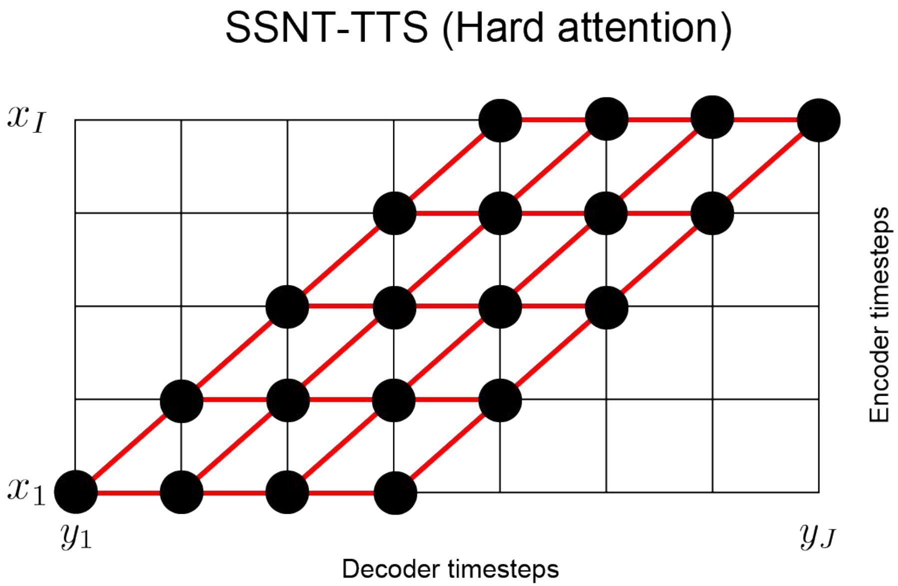


Tacotron vs SSNT-TTS: SSNT-TTS



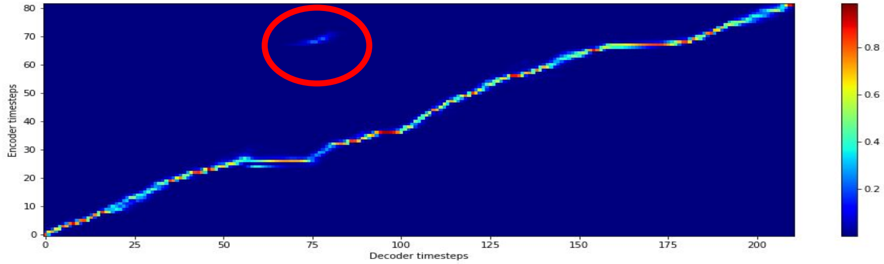
Tacotron vs SSNT-TTS: Alignment methods



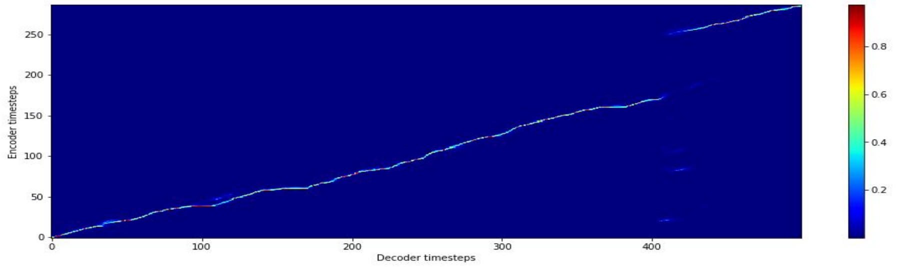


Tacotron vs SSNT-TTS: problems of soft attention

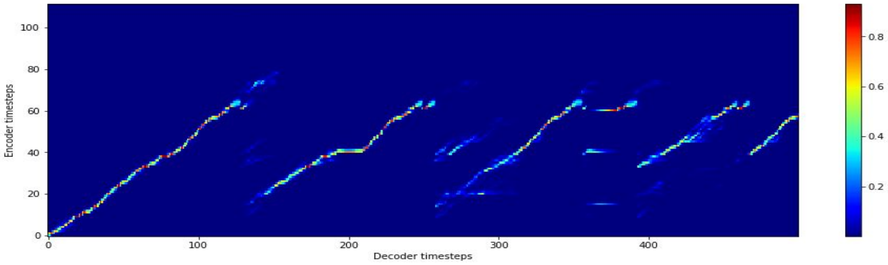
Mode split



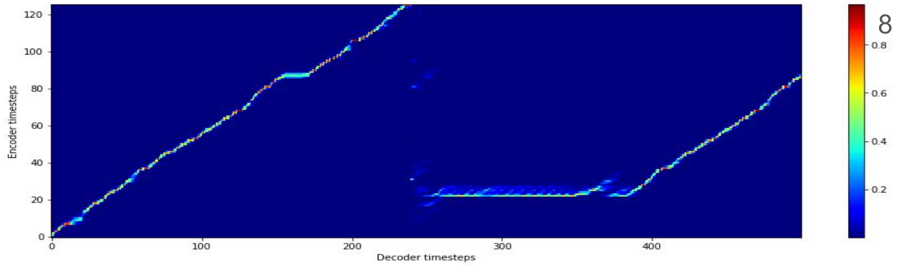
Skip



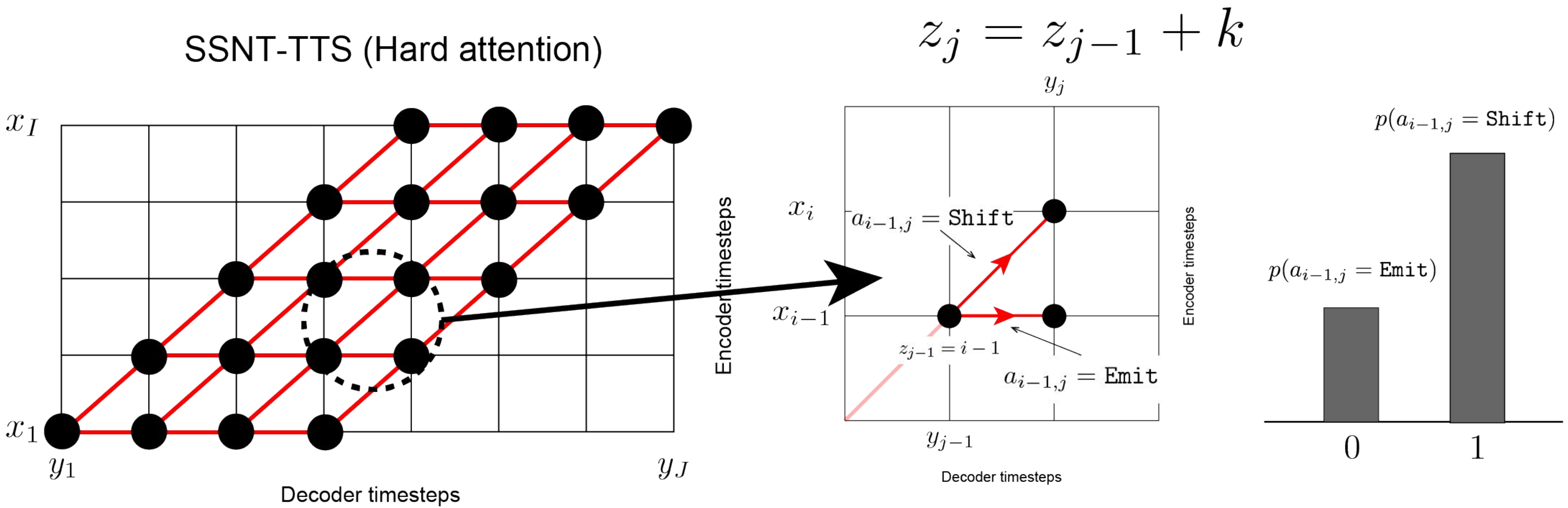
Repeat



Late termination



SSNT-TTS: monotonic alignment structure

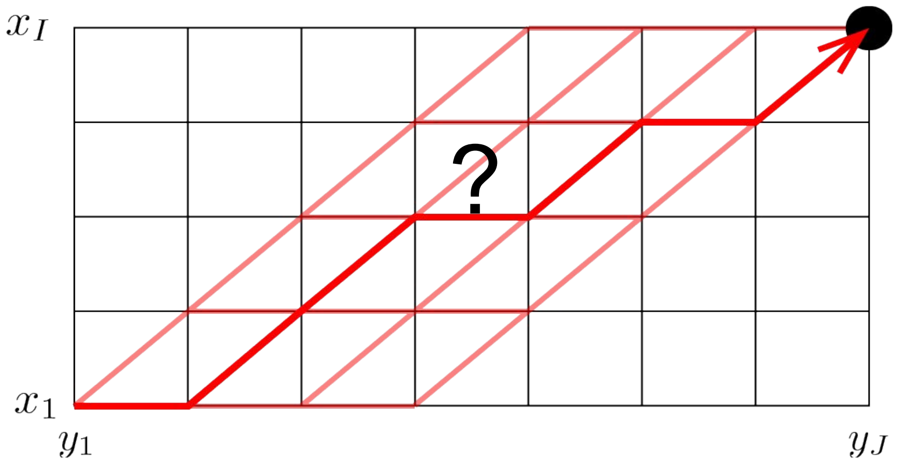


Yasuda et al., Initial investigation of encoder-decoder end-to-end TTS framework using marginalization of monotonic hard alignments. SSW10, 2019.
Yu et al., Online Segment to Segment Neural Transduction. EMNLP 2016.

Topic: Investigation of alignment prediction methods

How can we find the most optimal alignment during inference?

1. Randomness ← Nondeterministic nature of speech
2. Search methods ← Autoregressive decoding
3. Probability distributions ← Suitable distribution for random sampling



1. Randomness

— How to predict transition probabilities —
Deterministic prediction vs sampling from Bernoulli distribution

Randomness: Sampling from Bernoulli distribution

Gumbel-Max trick (Yellott, 1977):

An implementation of sampling from Bernoulli distribution.

$$\begin{aligned}\mathbb{P}(a_{i,j} = \texttt{Emit}) &= \mathbb{P}(G_1 + \log \alpha_1 > G_2 + \log \alpha_2) \\ &= \mathbb{P}(L + \log \alpha_1 > \log \alpha_2),\end{aligned}$$

Add Gumbel noise to logits.

Difference of two Gumbel noises is Logistic noise.

$$a_{i,j} = \text{argmax}(L + \log \alpha_1, \log \alpha_2).$$

Obtain discrete sample by argmax operator.

Yellott, The relationship between luce's choice axiom, thurstone's theory of comparative judgment, and the double exponential distribution. Journal of Mathematical Psychology, 1977

Randomness: Relationship with greedy decode

Greedy decode

$$a_{i,j} = \operatorname{argmax}(\log \alpha_1, \log \alpha_2)$$

Sampling from Bernoulli distribution

$$a_{i,j} = \operatorname{argmax}(\underline{L +} \log \alpha_1, \log \alpha_2)$$

The difference between Greedy decode and sampling from Bernoulli distribution is the presence of Logistic noise.

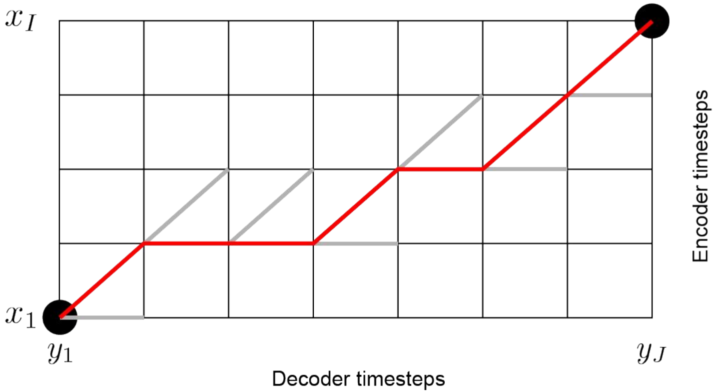
2. Search methods

— How to search the best path over trellis —

Greedy vs Beam search

Search: Greedy search

$$a_{i,j} = \text{argmax}(\log \alpha_1, \log \alpha_2) = \begin{cases} \texttt{Emit} & \text{if } \alpha_1 > \alpha_2 \\ \texttt{Shift} & \text{if } \alpha_1 < \alpha_2 \end{cases}$$



Greedy search takes a path with the highest probability at each time step.

Search: Beam search

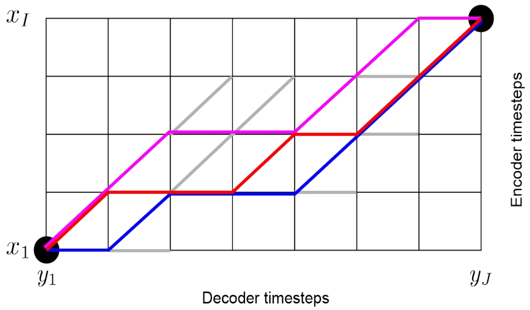
$$
\begin{aligned}
(a_j^{\text{beam1}}, \ldots, a_j^{\text{beamK}}) = \text{TopK}(&\log p(a_{j-1}^{\text{beam1}}) + \log \alpha_1^{\text{beam1}}, \\
&\log p(a_{j-1}^{\text{beam1}}) + \log \alpha_2^{\text{beam1}}, \\
&\ldots, \\
&\log p(a_{j-1}^{\text{beamK}}) + \log \alpha_1^{\text{beamK}}, \\
&\log p(a_{j-1}^{\text{beamK}}) + \log \alpha_2^{\text{beamK}})
\end{aligned}
$$

Keep top K alignment candidates at each time step.

$$
(a_1, \ldots, a_J) = \text{PathHistory}\left(\text{argmax}\left(p(a_J^{\text{beam1}}), \ldots, p(a_J^{\text{beamK}})\right)\right)
$$

Take the highest as a final alignment at the last time step.

K = 3



Greedy decode is a special case where K = 1.

1. Randomness & 2. Search: Stochastic search

$$a_{i,j} = \text{argmax}(\underline{L + \log}\, \alpha_1, \log \alpha_2)$$

Stochastic greedy search (sampling from Bernoulli distribution)

$$\begin{aligned}(a_j^{\text{beam1}}, \ldots, a_j^{\text{beamK}}) = \text{TopK}(\log p(a_{j-1}^{\text{beam1}}) \underline{+ L + \log}\, \alpha_1^{\text{beam1}}, \\ \log p(a_{j-1}^{\text{beam1}}) + \log \alpha_2^{\text{beam1}}, \\ \ldots, \\ \log p(a_{j-1}^{\text{beamK}}) \underline{+ L + \log}\, \alpha_1^{\text{beamK}}, \\ \log p(a_{j-1}^{\text{beamK}}) + \log \alpha_2^{\text{beamK}})\end{aligned}$$

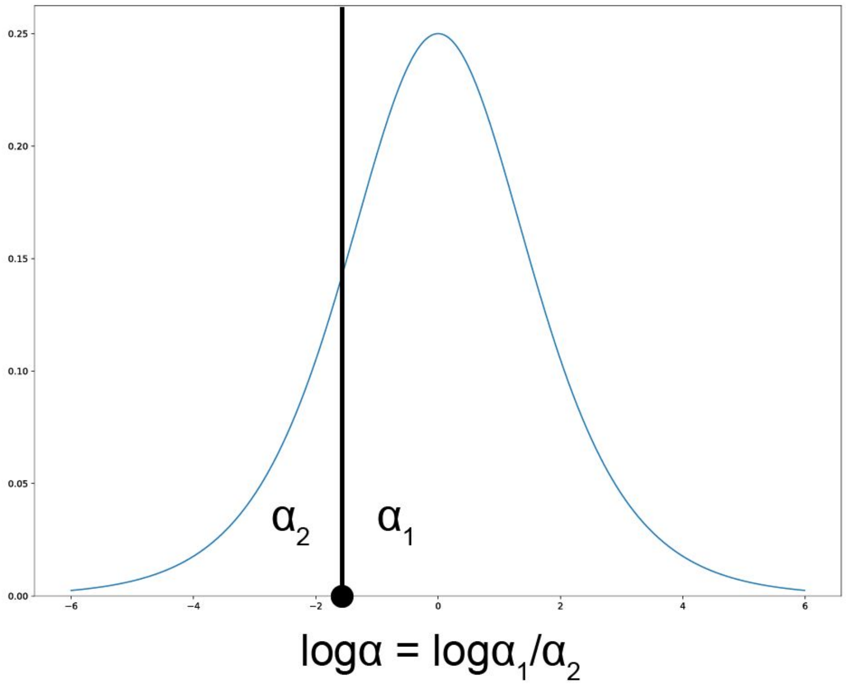
Stochastic beam search

3. Probability distributions

— What is the best probabilistic distribution for transition probabilities? —

Logistic vs binary Concrete distributions

Probability distributions: Logistic distribution

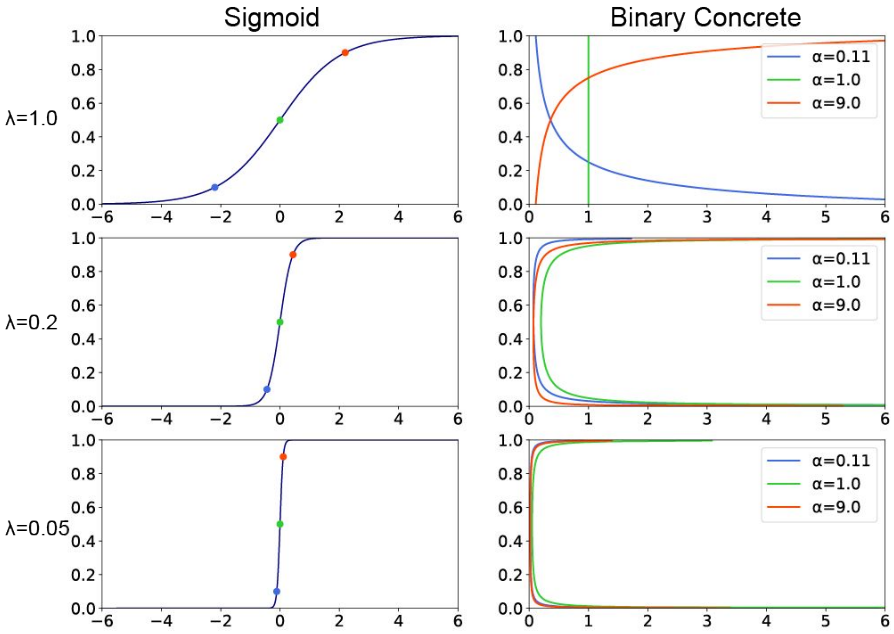


A sample from Bernoulli distribution can be drawn from Logistic distribution followed by argmax operator (Gumbel-max trick).

We refer sampling from Bernoulli distribution as Logistic condition.

Probability distributions: binary Concrete distribution

$$\mathbb{P}(a_{i,j} = \texttt{Emit}) = \frac{1}{1 + \exp\left(-(\log\alpha + L)/\lambda\right)}$$



Concrete distribution: Continuous relaxation of discrete distribution (Maddison et al., 2017).

Parametrized with α and λ.

Sample can be drawn with sigmoid added Logistic noise.

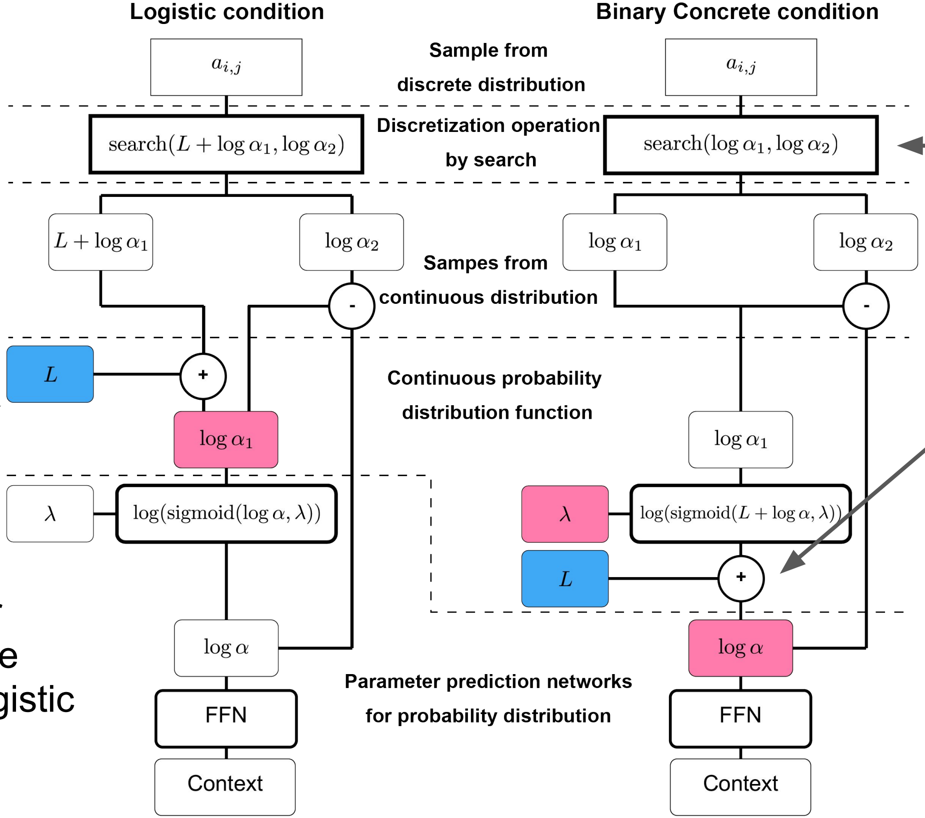
<u>Lower temperature λ encourages discretization.</u>

Maddison et al., The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. ICLR 2017

Randomness, Search, Probability distribution: all together

Logistic condition adds Logistic noise after sigmoid.

Deterministic or Stochastic mode with/without Logistic noise



Greedy search or Beam search

Binary Concrete condition adds Logistic noise before sigmoid.

Difference of the two probability distribution condition is the location to add Logistic noise.

Experiments & Results

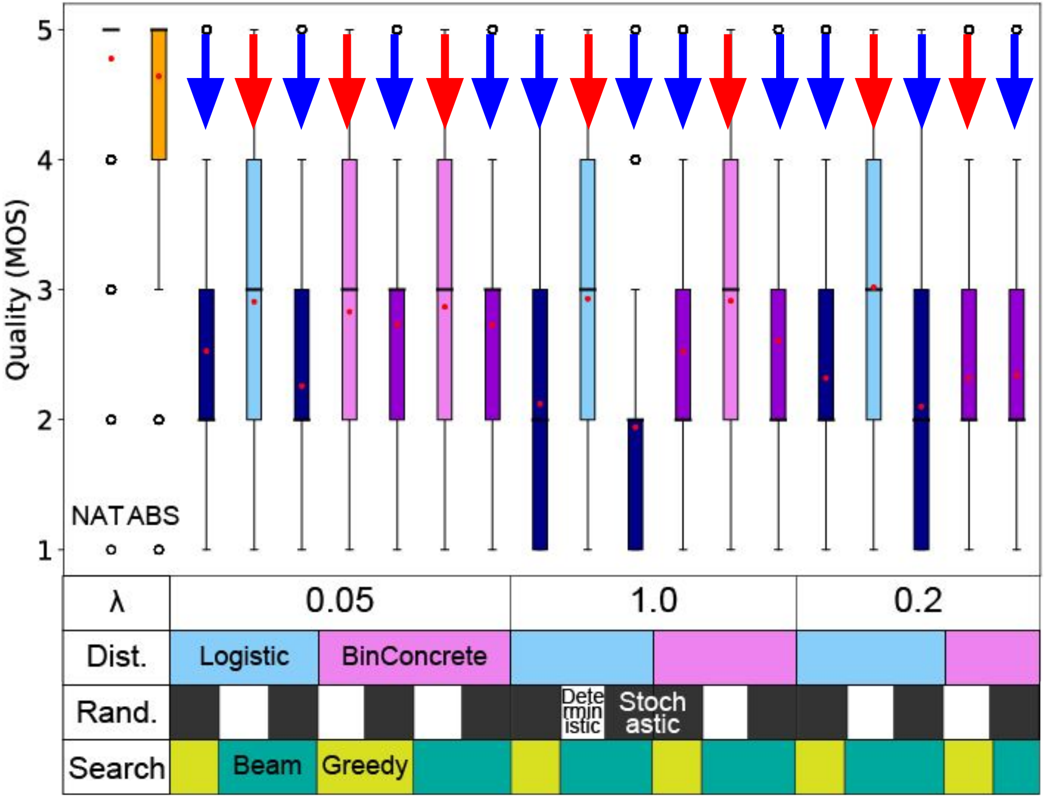
Experiments

- Corpus: ATR Ximera
  - Language: Japanese
  - Total duration: 46h
  - Total utterances: 28,259
- SSNT-TTS
  - Input: phoneme & accentual type
  - Output: mel spectrogram
- Vocoder: WaveNet
- Subjective evaluation
  - 5 grade MOS about naturalness
  - 193 native listeners
  - 28,800 evaluations

Conditions: 18 combinations

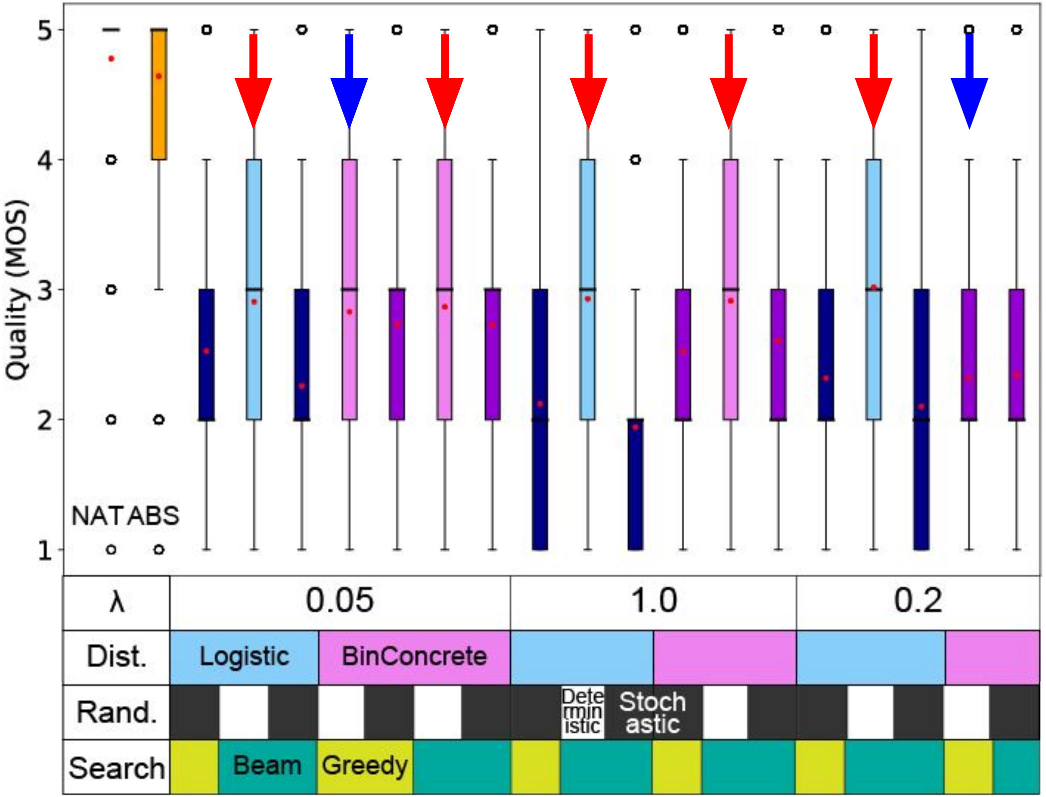
Distribution	Logistic/binary Concrete
Temperature λ	0.05/0.2/1.0
Randomness	Deterministic/Stochastic
Search	Greedy/Beam

Results: randomness



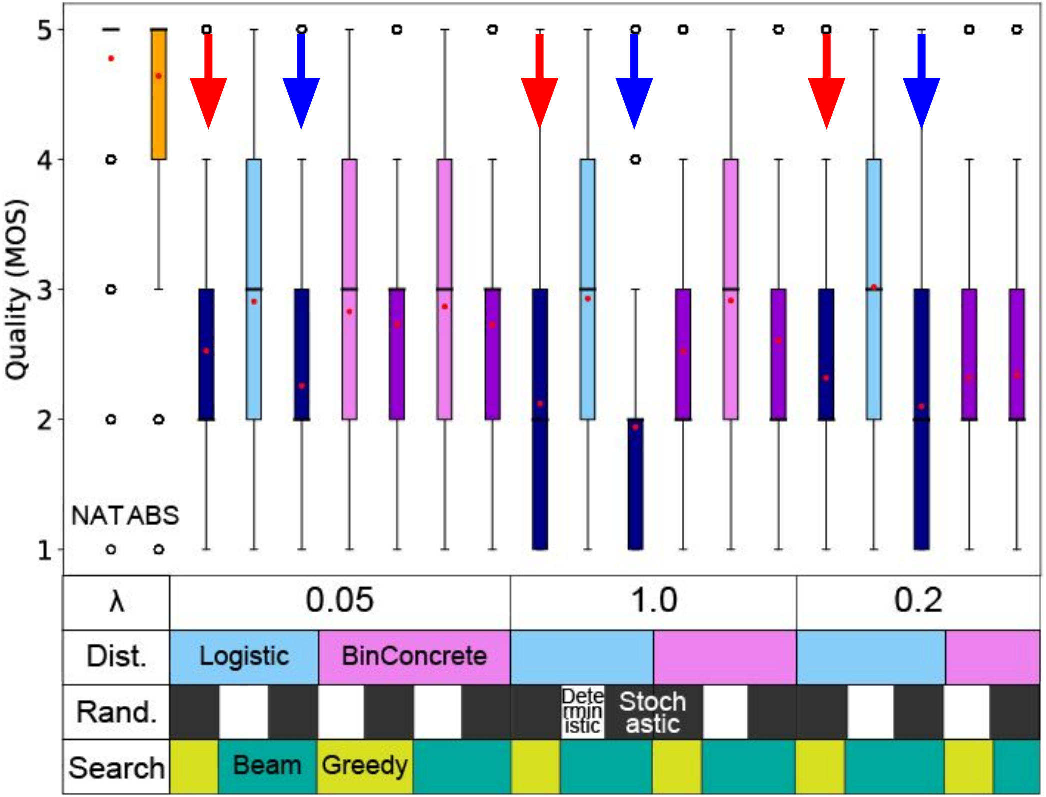
- Deterministic conditions outperformed stochastic conditions

Results: search methods



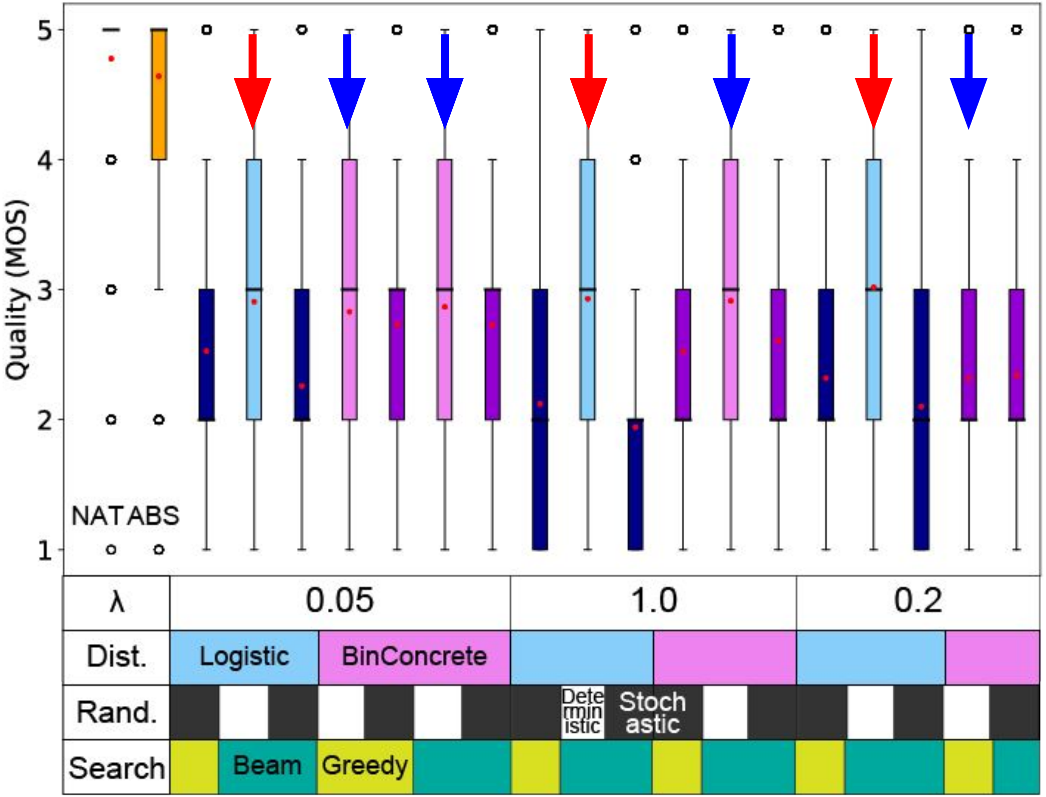
- Beam search performed better than greedy search under deterministic conditions

Results: search methods



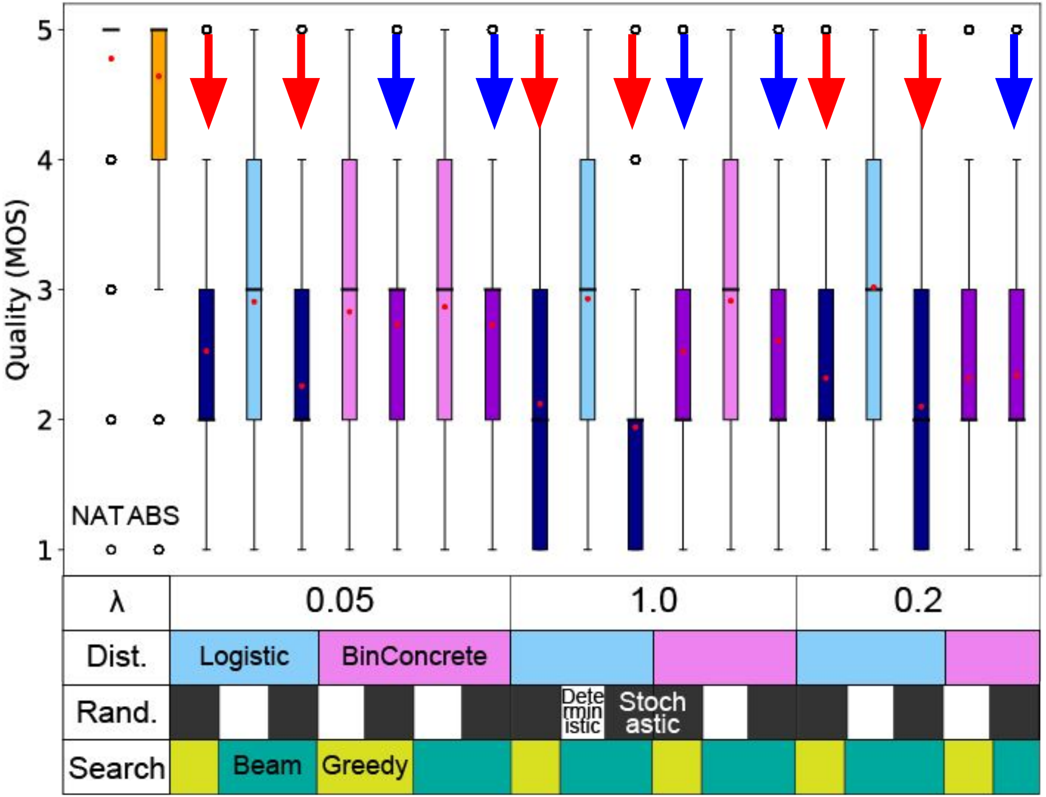
- Beam search performed worse than greedy search under stochastic and Logistic conditions

Results: probability distributions



- Performance of Logistic conditions is same as binary Concrete conditions under deterministic condition

Results: probability distributions



- Performance of Logistic conditions is much worse than binary Concrete conditions under stochastic condition
- The poor performance of Logistic condition is mitigated by lowering temperature parameter

Discussion & Summary

- The Logistic and binary Concrete conditions can estimate the alignment transition boundaries.
  - Both conditions had similar scores under deterministic search.
- The Logistic condition does not parametrize proper alignment transition distribution.
  - The Logistic condition performed very badly under stochastic search.
- The binary Concrete conditions can fill the gap between continuous and discrete distributions.
  - The binary Concrete conditions were relatively robust to stochastic search condition.

Conclusion

- Alignment prediction methods were investigated for SSNT-TTS
- The conditions for alignment prediction included:
  - Randomness
  - Search methods
  - Probability distributions
- Our experiment showed
  - Deterministic condition was favorable than stochastic condition
  - Beam search was helpful to improve naturalness
  - The binary Concrete distribution was relatively robust under stochastic search

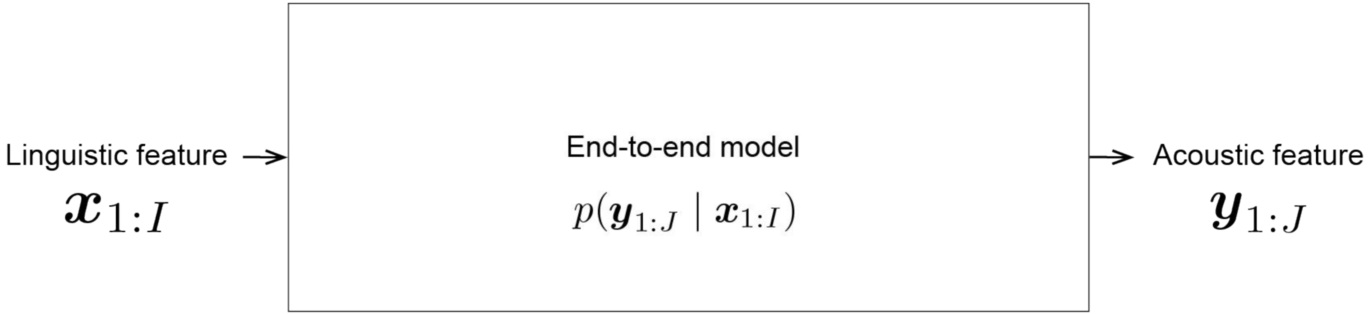
Audio samples: https://nii-yamagishilab.github.io/sample-ssnt-sampling-methods

SSNT-TTS (Yasuda et al., 2019 [1])

- Alignment structure is designed to be **monotonic**
- Alignment method is **hard attention**, instead of soft
- Alignment is a discrete **latent variable**
- Based on **SSNT** (Segment-to-Segment Neural Transduction) [2]
- Output distribution is continuous, instead of discrete

[1] Yasuda et al. SSW10, 2019.
[2] Yu et al., EMNLP, 2016.

SSNT-TTS: end-to-end TTS as a probabilistic model



SSNT-TTS: factorization for joint probability of alignment and output

$$p(\boldsymbol{y}_{1:J} \mid \boldsymbol{x}_{1:I}) = \sum_{\forall \mathbf{z}} \underline{p(\boldsymbol{y}_{1:J}, \mathbf{z} \mid \boldsymbol{x}_{1:I})}$$

Factorization for joint probability
of alignment and output

$$p(\boldsymbol{y}_{1:J}, \mathbf{z} \mid \boldsymbol{x}_{1:I}) \approx \prod_{j=1}^{J} \underbrace{p(z_j \mid z_{j-1}, \boldsymbol{y}_{1:j-1}, \boldsymbol{x}_{1:I})}_{\text{Alignment probability}} \underbrace{p(\boldsymbol{y}_j \mid \boldsymbol{y}_{1:j-1}, z_j, \boldsymbol{x}_{1:I})}_{\text{Output probability}}$$

SSNT-TTS: definition of alignment transition variables

$$\prod_{j=1}^{J} p(z_j \mid z_{j-1}, \boldsymbol{y}_{1:j-1}, \boldsymbol{x}_{1:I}) p(\boldsymbol{y}_j \mid \boldsymbol{y}_{1:j-1}, z_j, \boldsymbol{x}_{1:I})$$

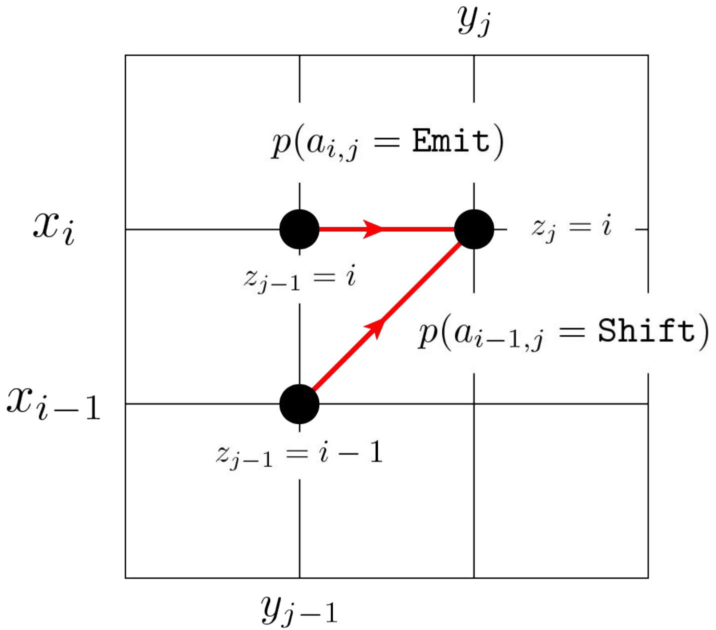
Alignment probability   Output probability

Binary alignment transition variable

$$a_{i,j} \in \{\texttt{Emit}, \texttt{Shift}\}$$

Probability when an alignment reaches input position *i* at timestep *j*

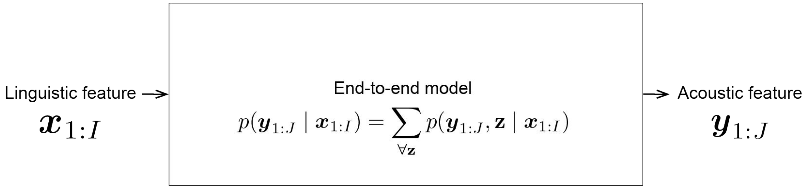
$$p(z_j = i \mid z_{j-1}, \boldsymbol{y}_{1:j-1}, \boldsymbol{x}_{1:I}) = \begin{cases} 0 & z_{j-1} > i \\ p(a_{i,j} = \texttt{Emit}) & z_{j-1} = i \\ p(a_{i-1,j} = \texttt{Shift}) & z_{j-1} = i-1 \\ 0 & z_{j-1} < i-1 \end{cases}$$



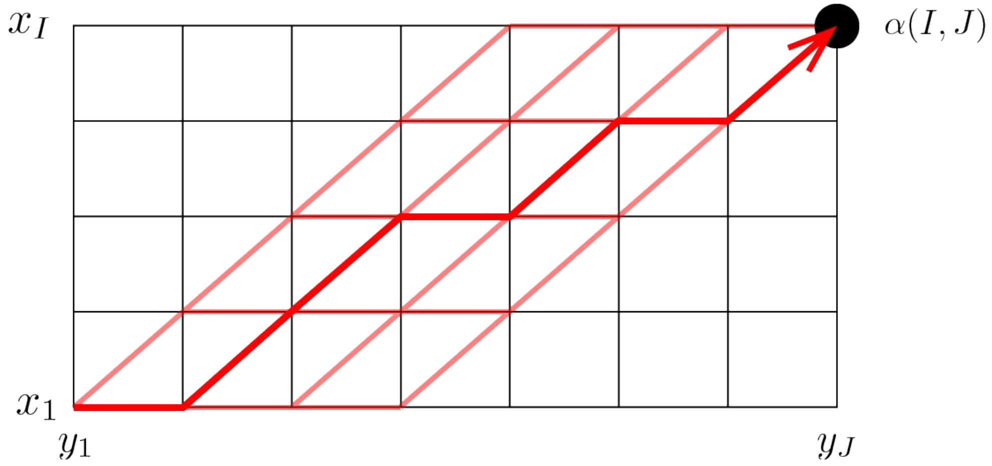
SSNT-TTS: Training with marginalization of alignments by forward probability

Maximize $p(\boldsymbol{y}_{1:J} \mid \boldsymbol{x}_{1:I})$

$$\mathcal{L}(\boldsymbol{\theta}) = -\log p(\boldsymbol{y}_{1:J} \mid \boldsymbol{x}_{1:I}; \boldsymbol{\theta})$$



$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= -\log p(\boldsymbol{y}_{1:J} \mid \boldsymbol{x}_{1:I}; \boldsymbol{\theta}) \\ &= -\sum_{\forall \mathbf{z}} p(\boldsymbol{y}_{1:J}, \mathbf{z} \mid \boldsymbol{x}_{1:I}) \\ &= -\log \alpha(I, J)\end{aligned}$$



SSNT-TTS: alignment prediction during inference

$$z_j = z_{j-1} + k$$

- Greedy decode $k = \text{argmax}\left(p(z_j = i \mid z_{j-1}, \boldsymbol{y}_{1:j-1}, \boldsymbol{x}_{1:I})\right)$

or

- Random sampling $k \sim \text{Bernoulli}\left(p(z_j = i \mid z_{j-1}, \boldsymbol{y}_{1:j-1}, \boldsymbol{x}_{1:I})\right)$