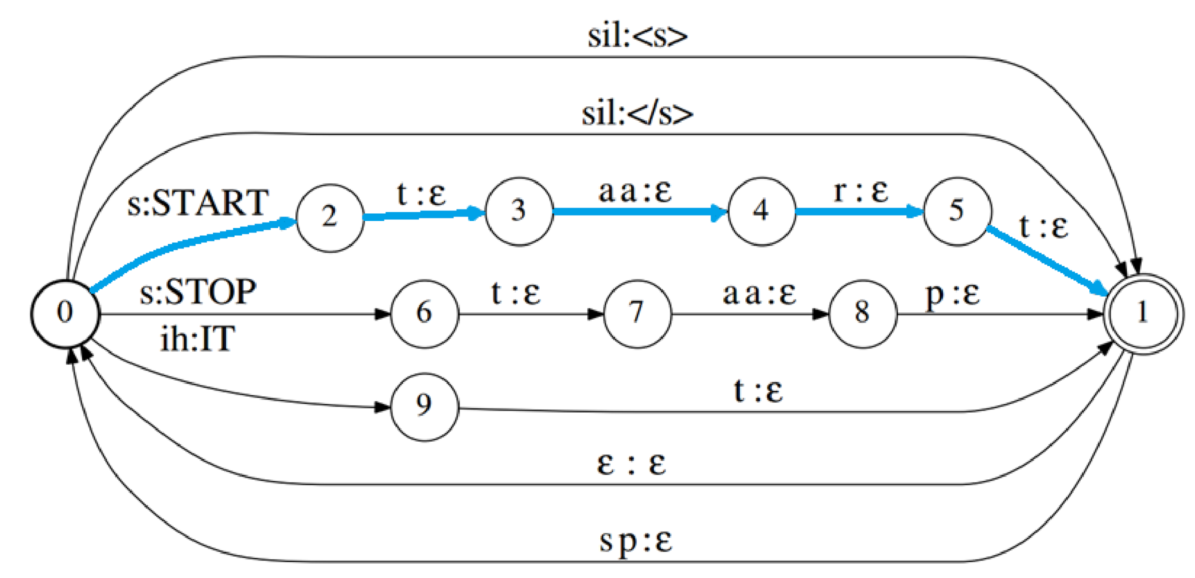


Linguistic Search Optimization for Deep Learning Based Speech Recognition

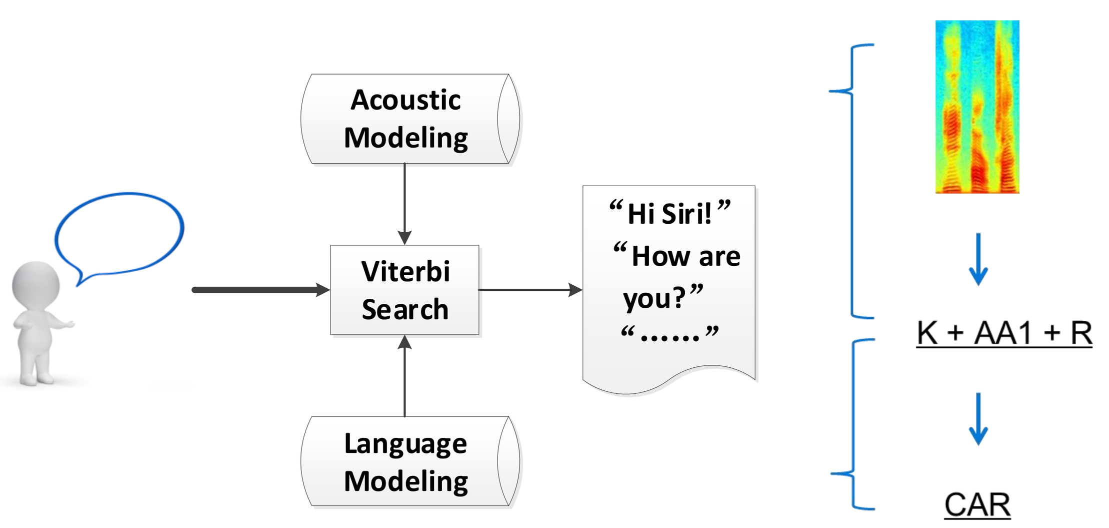
Zehuai Chen

chenzhehuai@sjtu.edu.cn

SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China

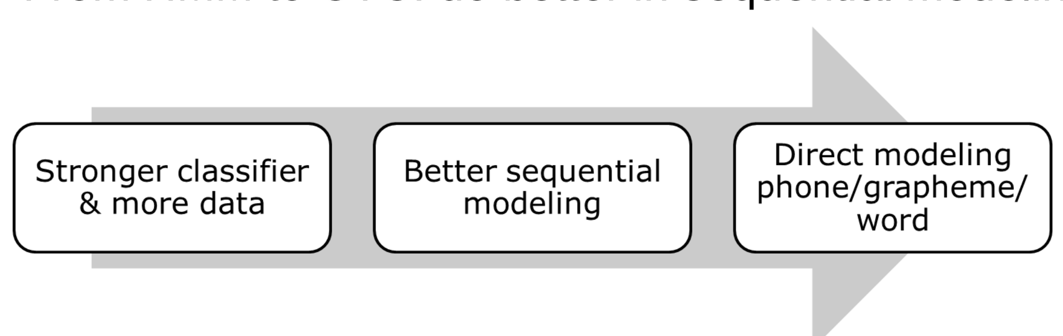
Overview

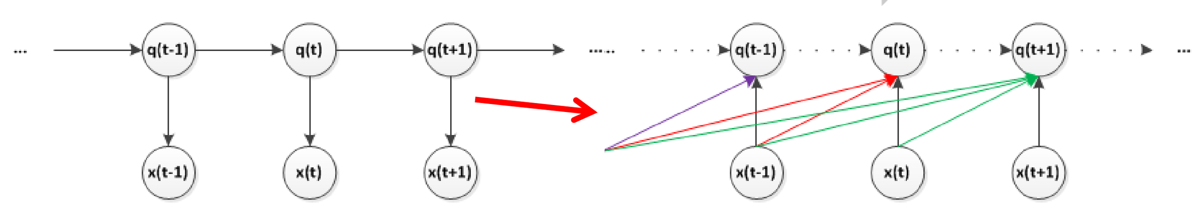
- **Problem: Linguistic search takes over 50% computation in Automatic Speech Recognition (ASR).**
- **Approach:**
  - **Reduce the Search Complexity by End-to-end Modeling**
  - **Accelerate the Search Speed using Parallel Computing**
- **Experiments & Discussion: 5 times and 50 times speedup respectively; able to combine**



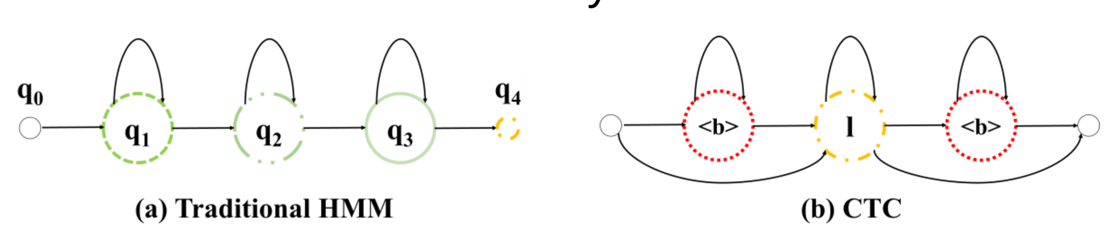
From HMM to CTC model

- From HMM to CTC: do better in sequential modeling

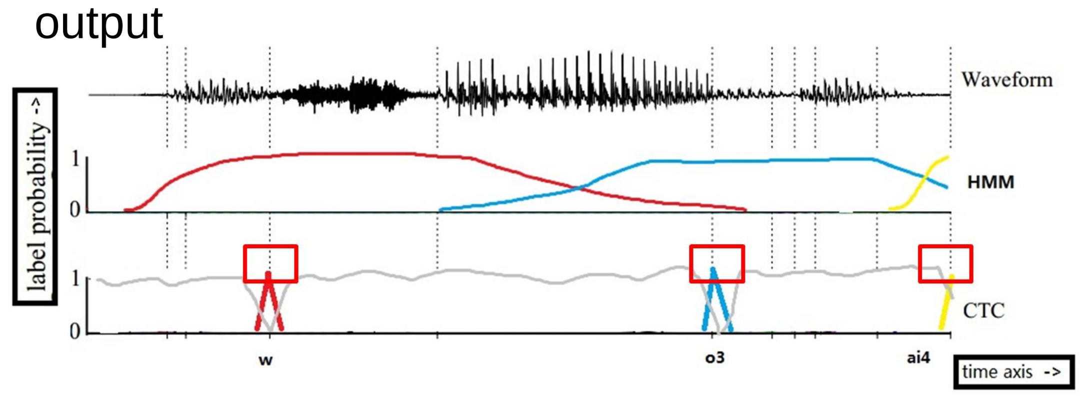




- CTC model: learn the many-to-one function



- peaky distribution and concentrated information output



Frame Sync. To Phone Sync.

- **Frame synchronous Viterbi beam search in CTC**

$$\mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w}}\{P(\mathbf{w})p(\mathbf{x}|\mathbf{w})\} = \operatorname*{argmax}_{\mathbf{w}}\{P(\mathbf{w})p(\mathbf{x}|\mathbf{l_w})\}$$

$$= \operatorname*{argmax}_{\mathbf{w}}\left\{P(\mathbf{w})\max_{\mathbf{l_w}}\frac{P(\mathbf{l_w}|\mathbf{x})}{P(\mathbf{l_w})}\right\}$$

$$\cong \operatorname*{argmax}_{\mathbf{w}}\left\{P(\mathbf{w})\max_{\pi:\pi\in L',\mathcal{B}(\pi_{1:T})=\mathbf{l_w}}\frac{1}{P(\mathbf{l_w})}\prod_{t=1}^{\boxed{T}} y^t_{\pi_t}\right\}$$

$\pi_{1:T} = (\pi_1,\dots,\pi_T)$ is the frame-wise decoding path
$\mathbf{l_w}$ is phone sequence corresponding to $\mathbf{w}$ in dictionary
$l \in L$ and L is the phone set
$\pi \in L'$ and $L' = L \cup \{\text{blank}\}$

- **Frame synchronous to phone synchronous decoding**

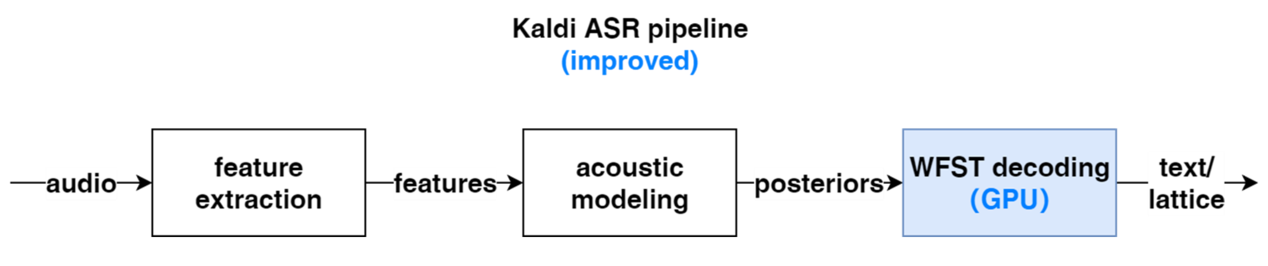
$$\mathbf{w}^* \cong \operatorname*{argmax}_{\mathbf{w}}\left\{P(\mathbf{w})\max_{\pi:\pi\in L',\mathcal{B}(\pi_{1:T})=\mathbf{l_w}}\frac{1}{P(\mathbf{l_w})}\left\{\prod_{t\notin U} y^t_{\pi_t}\cdot\cancel{\prod_{t\in U} y^t_{\text{blank}}}\right\}\right\}$$

$$= \operatorname*{argmax}_{\mathbf{w}}\left\{P(\mathbf{w})\max_{\pi':\pi'\in L,\mathcal{B}(\pi'_{1:J})=\mathbf{l_w}}\frac{1}{P(\mathbf{l_w})}\prod_{j=1}^{\boxed{J}} y^{t_j}_{\pi'_j}\right\}$$

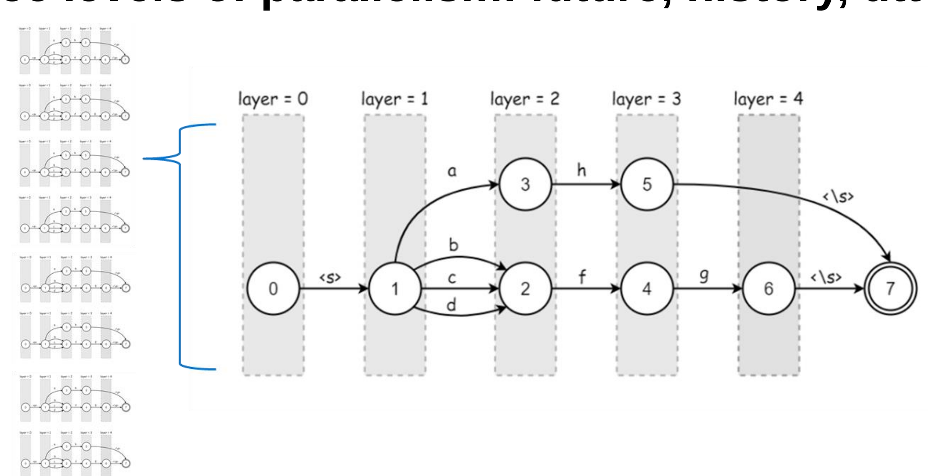
$U = \{u : y^u_{\text{blank}} \simeq 1\}$ is the set of common *blank* time indexes

$J = T - |U|$ is the number of output phone labels

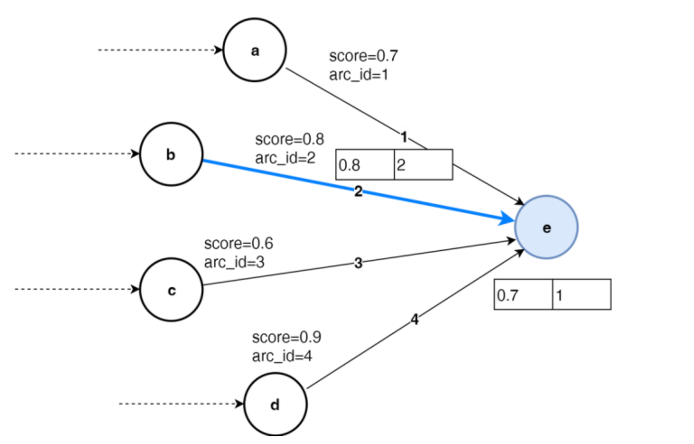
Parallel Viterbi Decoding



- **Three levels of parallelism: future, history, utterance**

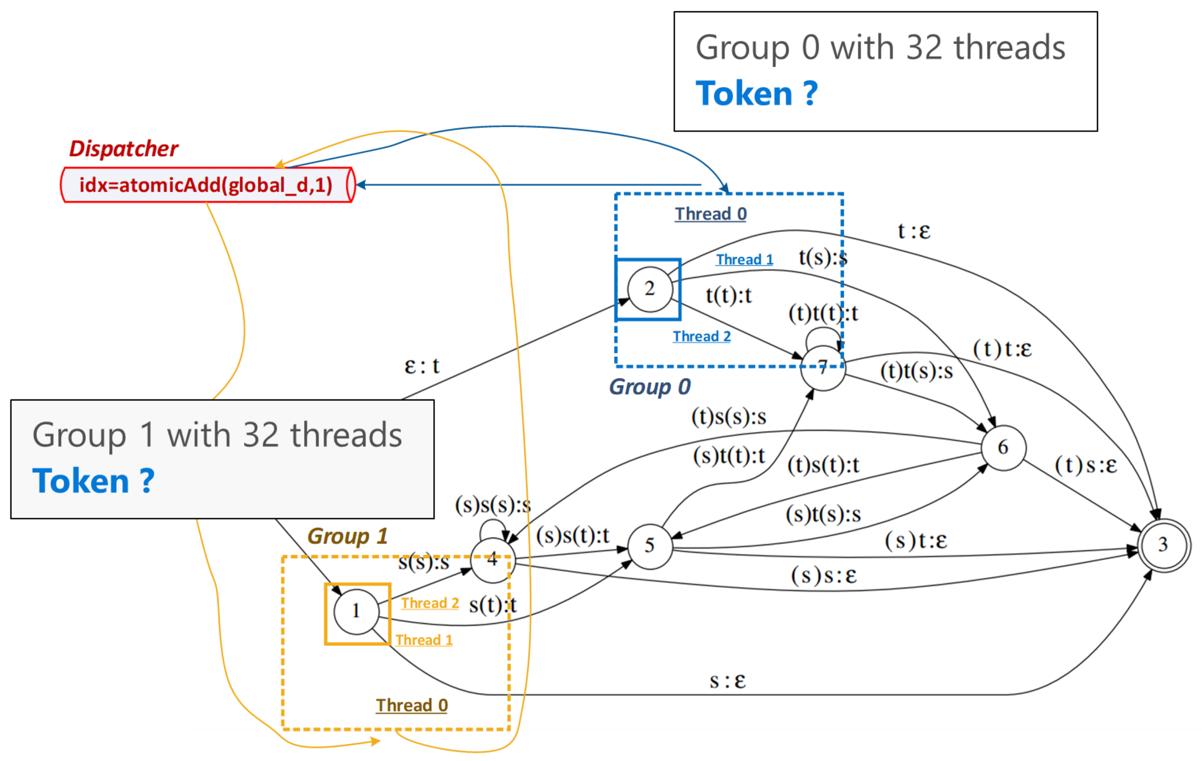


- **Atomic Token Recombination**



```
int64* atomicMax(*address, val) {
  original = *address;
  *address = max(val, *address);
  return original;
}
```

- **Dynamic Load Balancing**



- **Lattice Processing**
  - Linkedlist → vector
  - Atomic operations e.g. memory allocation
  - Parallel lattice pruning

Experiments

- **Experimental Setup**
- Switchboard 300 hours corpus, Cross Entropy & LF-MMI acoustic models (AM)
- 30k-vocabulary, several tri-gram language models (LM)
- Baseline: Kaldi 1-best decoder, Kaldi lattice decoder
- GPU Optimization: Fast memcpy; merge GPU kernels by adding grid sync.; etc. (rel. 20% speedup)
  - https://github.com/chenzhehuai/kaldi/tree/gpu-decoder

subset	performance FSD↦PSD WER	Δ(%)	search speed-up FSD↦PSD SRTF	Δ(%)	FSD↦PSD #AT	Δ(%)
swb	18.7	+0.5	0.075	-71	2221	-77
callhm	33.3	+0.0	0.073	-70	2211	-77

- **3 times speedup from end-to-end modeling**

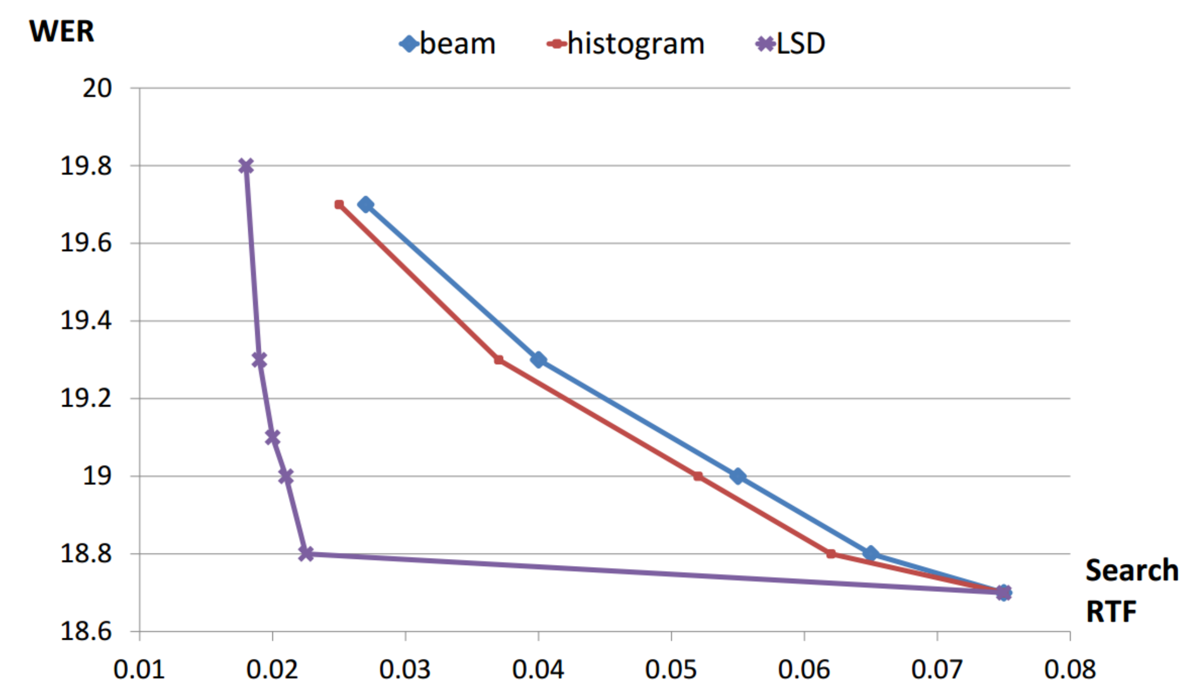
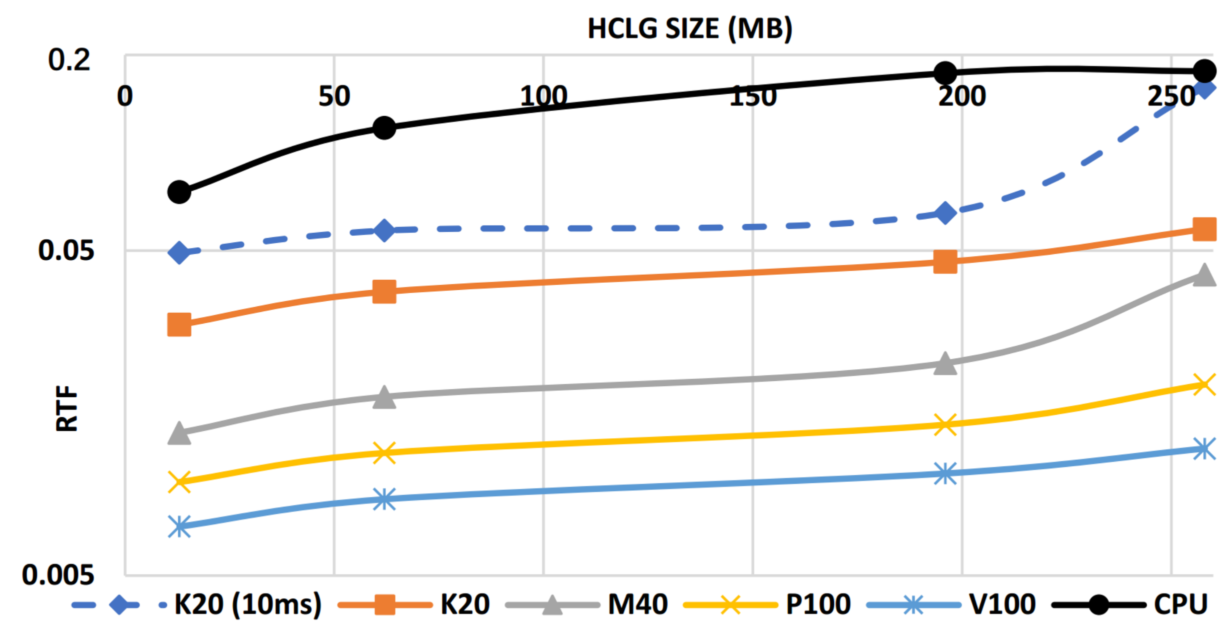


Table 2: *Speedup of the Proposed Method (beam=14).*

system	1-best RTF	Δ	+ lattice RTF	Δ
CPU	**0.16**	**1.0X**	**0.27**	**1.0X**
+ 8-sequence (1 socket)	-	-	0.15	1.8X
GPU	0.016	10X	0.080	3.3X
+ atomic operation	0.015	11X	0.077	3.5X
+ dyn. load balancing	**0.011**	**15X**	0.075	3.6X
+ lattice prune	-	-	**0.028**	**9.7X**
+ 8-sequence (MPS)	0.0035	46X	0.0080	34X

- **34 times speedup from parallel computing**

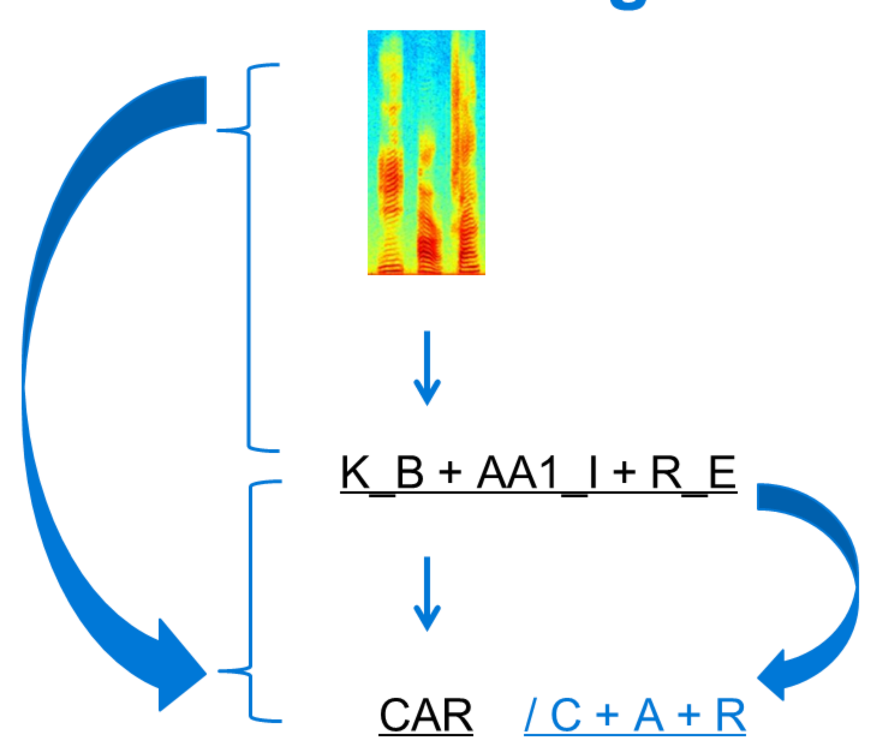


- **Varieties of GPU arch., WFST sizes and acoustic models.**

Conclusions

- **General speedup of linguistic search in speech recognition**

End-to-end Modeling



Parallel Computing

- **Future works:**
  - **Inspire more researches in GPU decoding**
  - **Combination of Both Techniques**