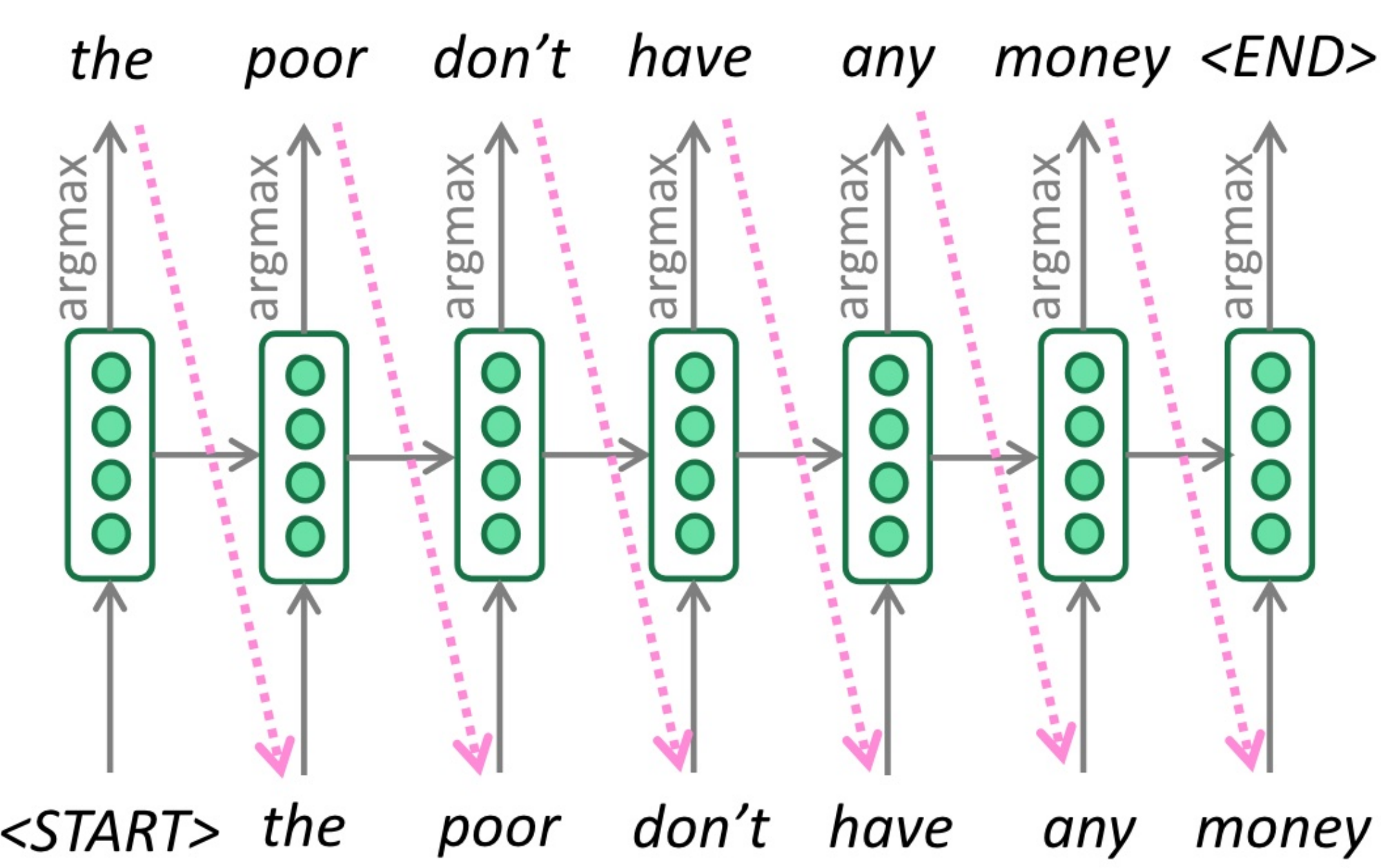
Machine Translation

Jordan Boyd-Graber

University of Maryland

Decoding

Adapted from material by Mohit Iyyer, Luke Zettlemoyer, Kalpesh Krishna, Karthik Narasimhan, Greg Durrett, Chris Manning, Dan Jurafsky



Argmax at every time step

Sampling Methods

Softmax distribution

$$p(w) = \frac{\exp\left\{\beta \cdot \vec{f}(w)\right\}}{\sum_{w'} \exp\left\{\beta \cdot \vec{f}(w')\right\}} \tag{1}$$

- top-k
- Nucleus / top-p
- Temperature

Sampling Methods

Softmax distribution

$$p(w) = \frac{\exp\left\{\beta \cdot \vec{f}(w)\right\}}{\sum_{w'} \exp\left\{\beta \cdot \vec{f}(w')\right\}} \tag{1}$$

- top-k: Only sample from k items with highest probability
- Nucleus / top-p
- Temperature

Sampling Methods

Softmax distribution

$$p(w) = \frac{\exp\left\{\beta \cdot \vec{f}(w)\right\}}{\sum_{w'} \exp\left\{\beta \cdot \vec{f}(w')\right\}} \tag{1}$$

- top-k: Only sample from k items with highest probability
- Nucleus / top-p: Only sample from highest items with at least p probability
- Temperature

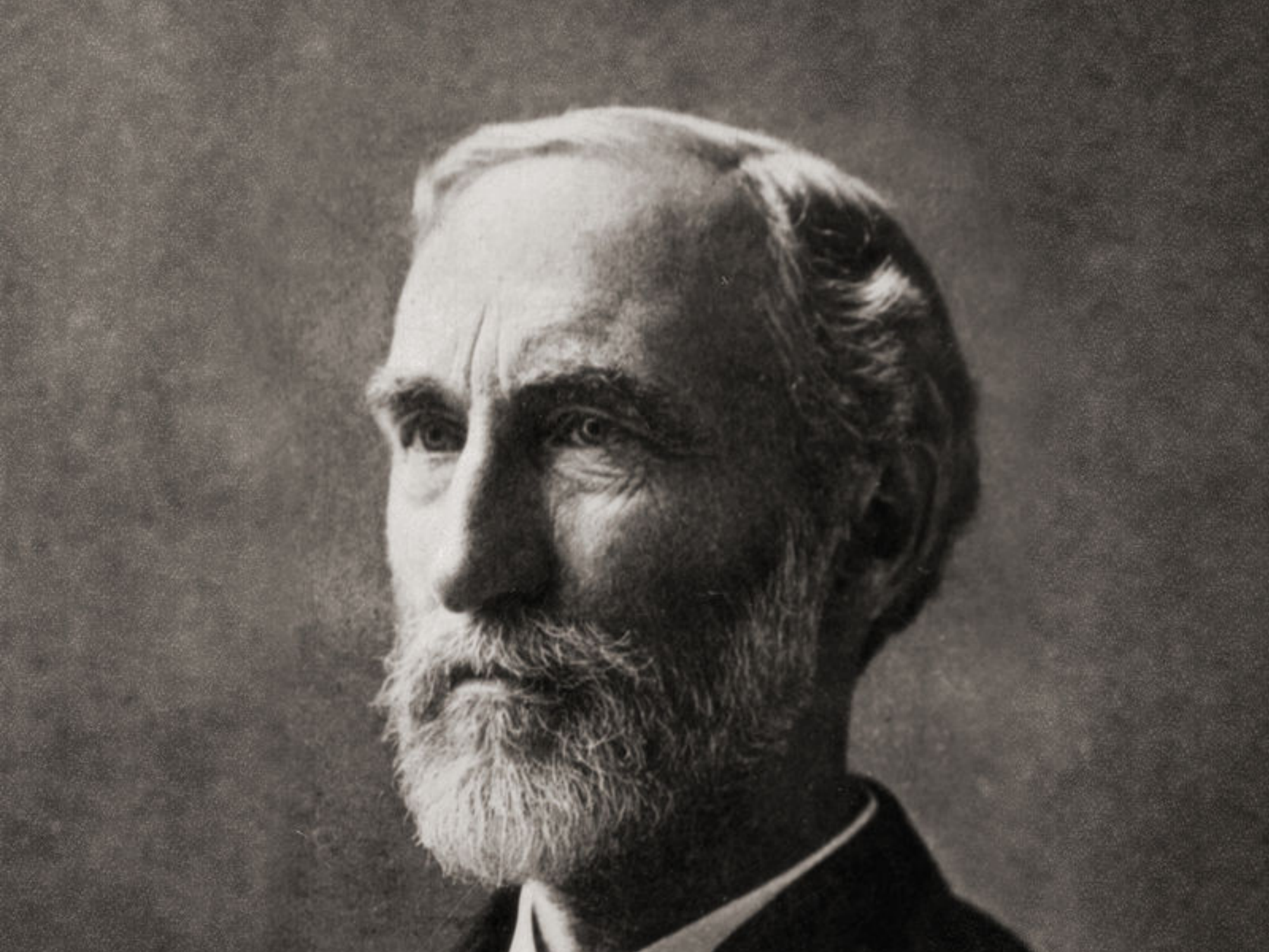
Sampling Methods

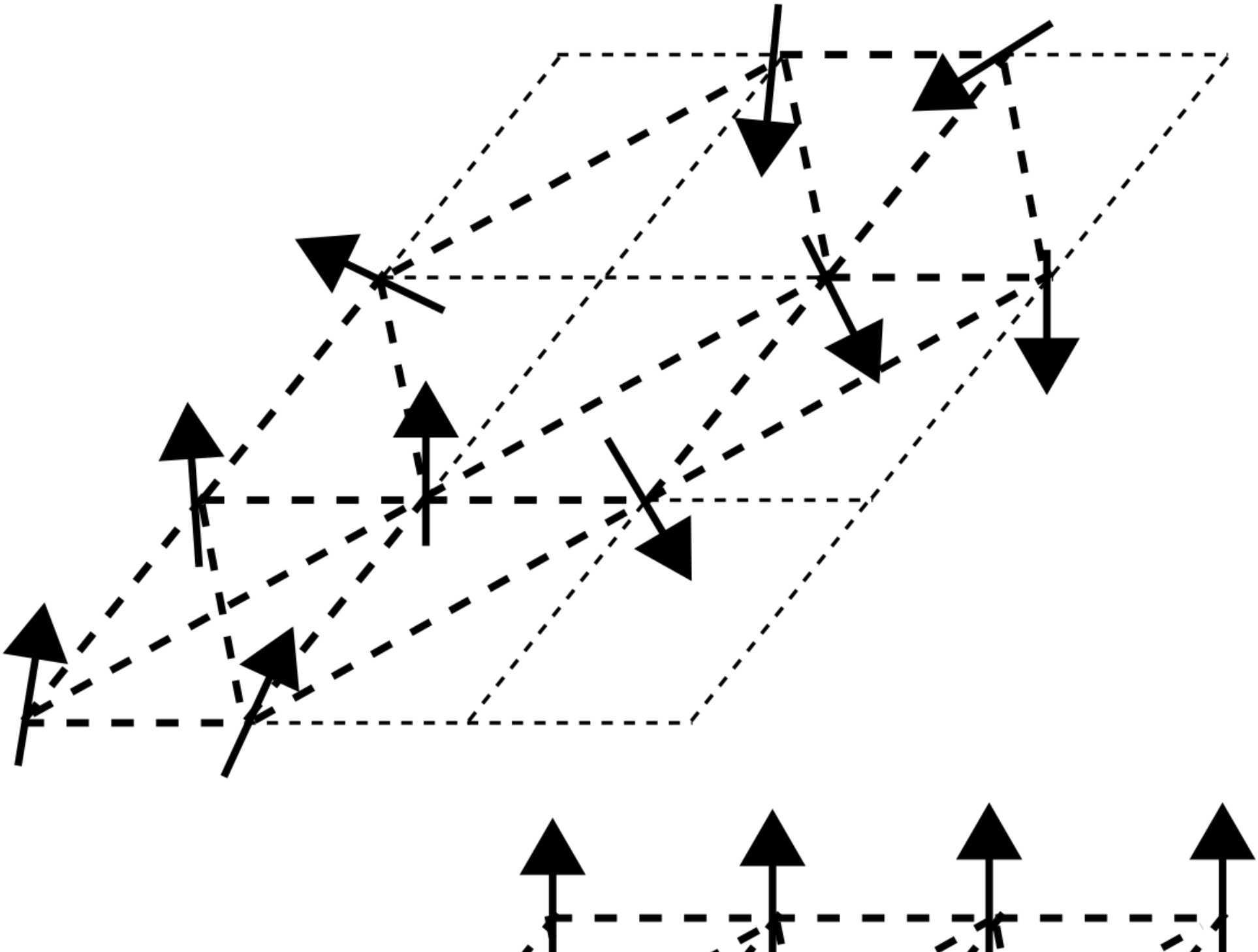
Softmax distribution

$$p(w) = \frac{\exp\left\{\beta \cdot \vec{f}(w)\right\}}{\sum_{w'} \exp\left\{\beta \cdot \vec{f}(w')\right\}} \tag{1}$$

- top-k: Only sample from k items with highest probability
- Nucleus / top-p: Only sample from highest items with at least p probability
- Temperature

$$p(w) = \frac{\exp\left\{\frac{\beta \cdot \vec{f}(w)}{T}\right\}}{\sum_{w'} \exp\left\{\frac{\beta \cdot \vec{f}(w')}{T}\right\}} \tag{2}$$





Sampling Methods

Softmax distribution

$$p(w) = \frac{\exp\left\{\beta \cdot \vec{f}(w)\right\}}{\sum_{w'} \exp\left\{\beta \cdot \vec{f}(w')\right\}} \tag{3}$$

- top-k
- Nucleus / top-p
- Temperature

Sampling Methods

Softmax distribution

$$p(w) = \frac{\exp\left\{\beta \cdot \vec{f}(w)\right\}}{\sum_{w'} \exp\left\{\beta \cdot \vec{f}(w')\right\}} \tag{3}$$

- top-k: Only sample from k items with highest probability
- Nucleus / top-p
- Temperature

Sampling Methods

Softmax distribution

$$p(w) = \frac{\exp\left\{\beta \cdot \vec{f}(w)\right\}}{\sum_{w'} \exp\left\{\beta \cdot \vec{f}(w')\right\}} \tag{3}$$

- top-k: Only sample from k items with highest probability
- Nucleus / top-p: Only sample from highest items with at least p probability
- Temperature

Sampling Methods

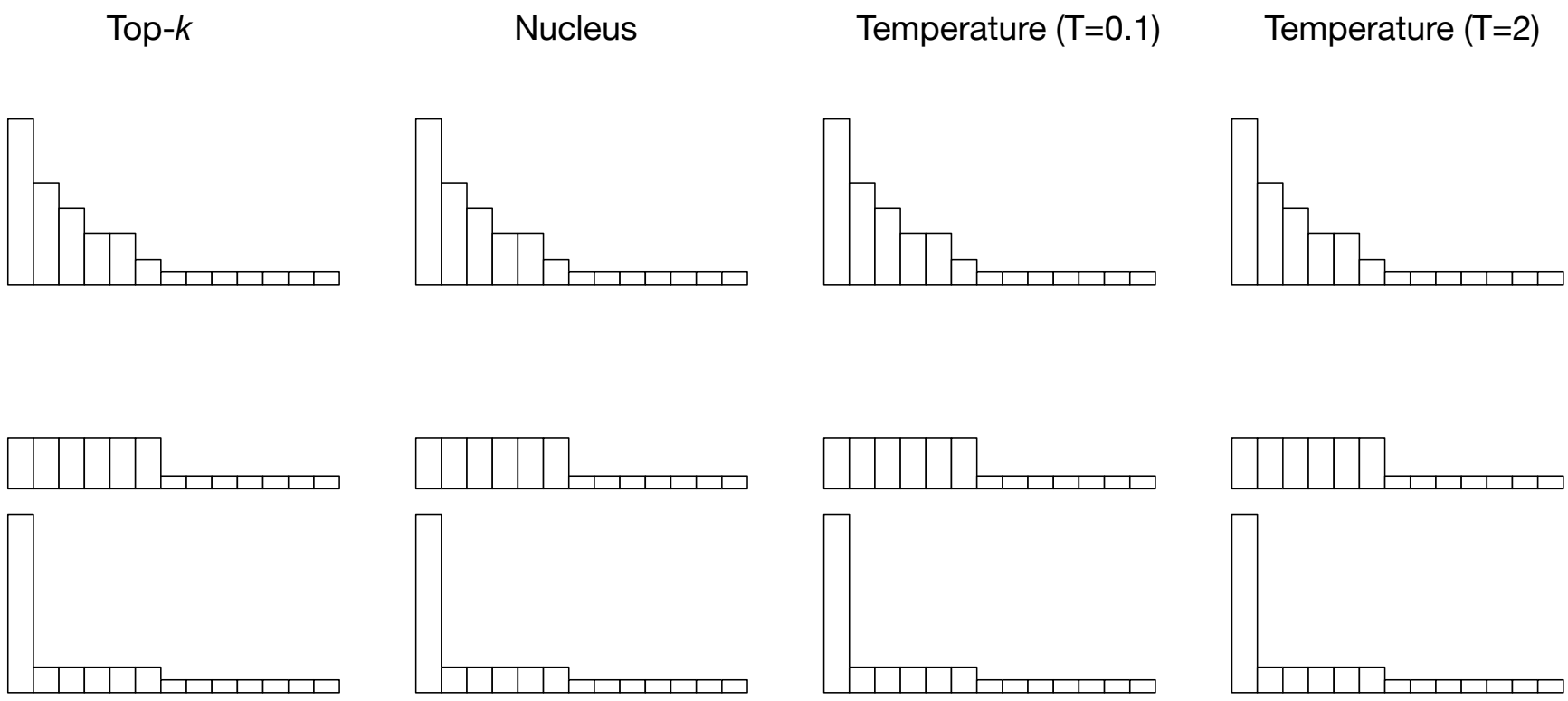
Softmax distribution

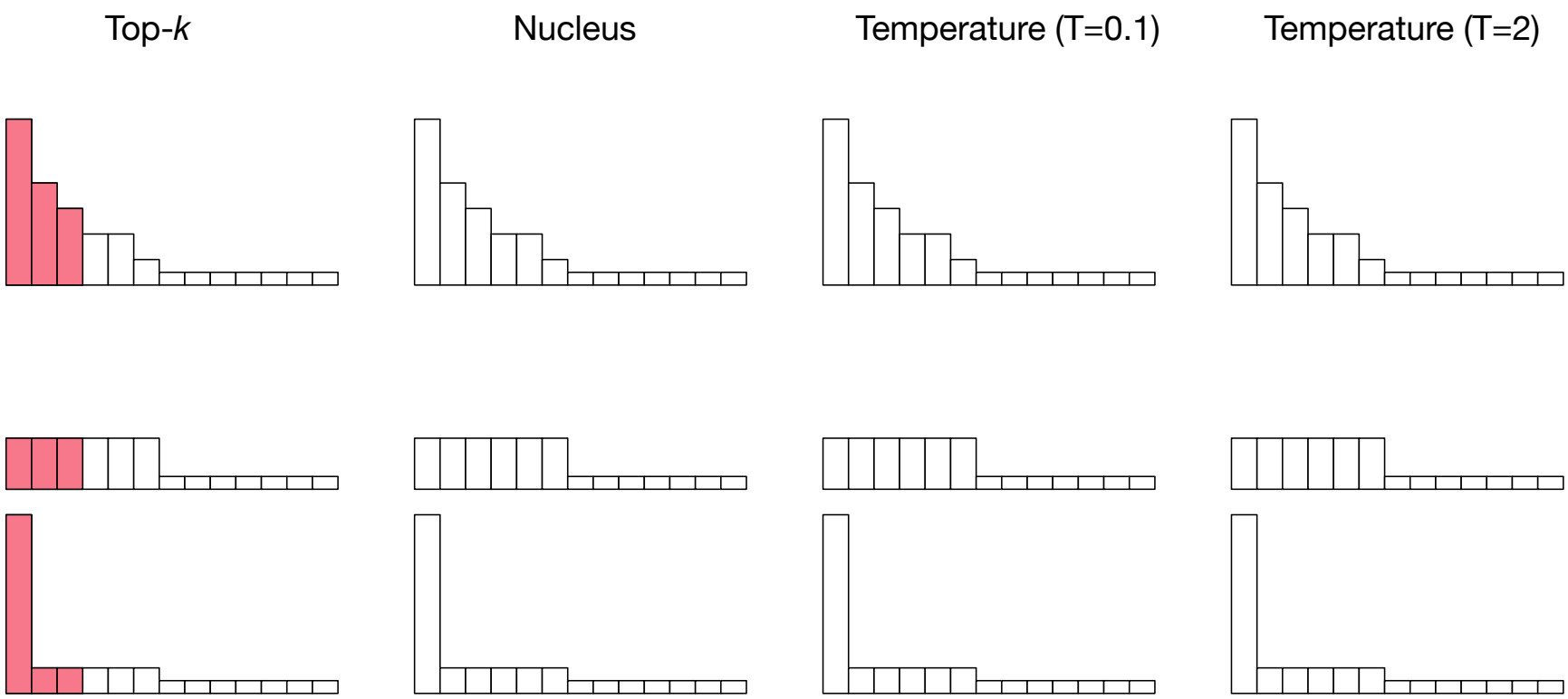
$$p(w) = \frac{\exp\left\{\beta \cdot \vec{f}(w)\right\}}{\sum_{w'} \exp\left\{\beta \cdot \vec{f}(w')\right\}} \tag{3}$$

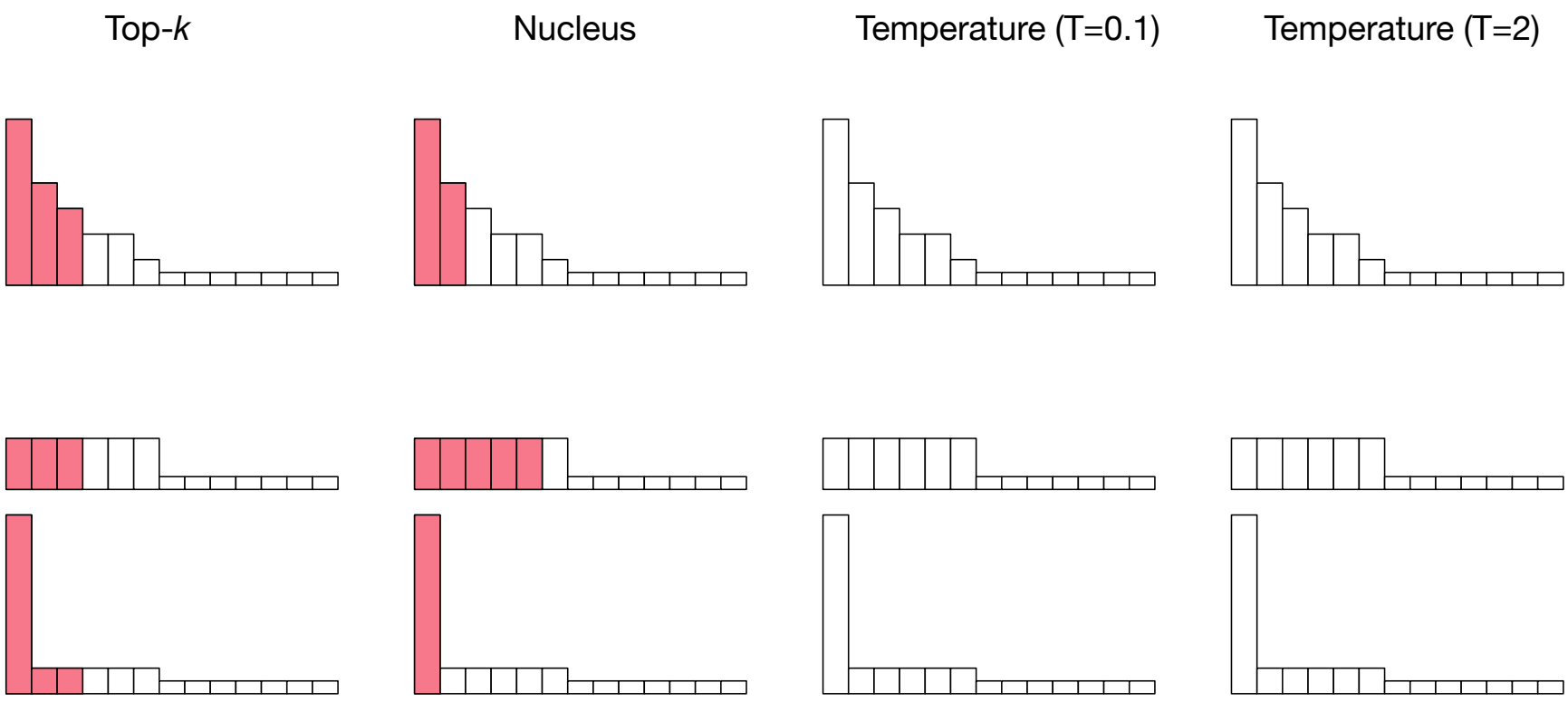
- top-k: Only sample from k items with highest probability
- Nucleus / top-p: Only sample from highest items with at least p probability
- Temperature

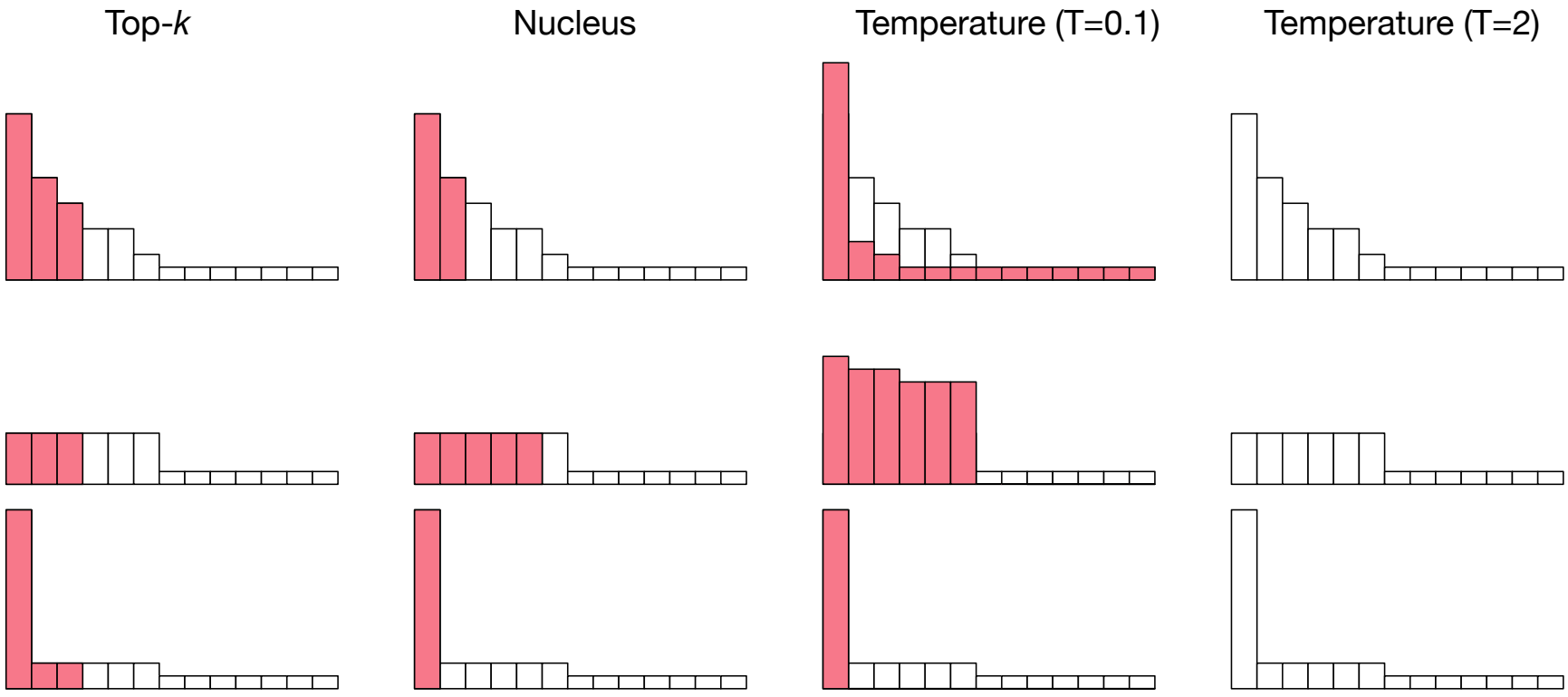
$$p(w) = \frac{\exp\left\{\frac{\beta \cdot \vec{f}(w)}{T}\right\}}{\sum_{w'} \exp\left\{\frac{\beta \cdot \vec{f}(w')}{T}\right\}} \tag{4}$$

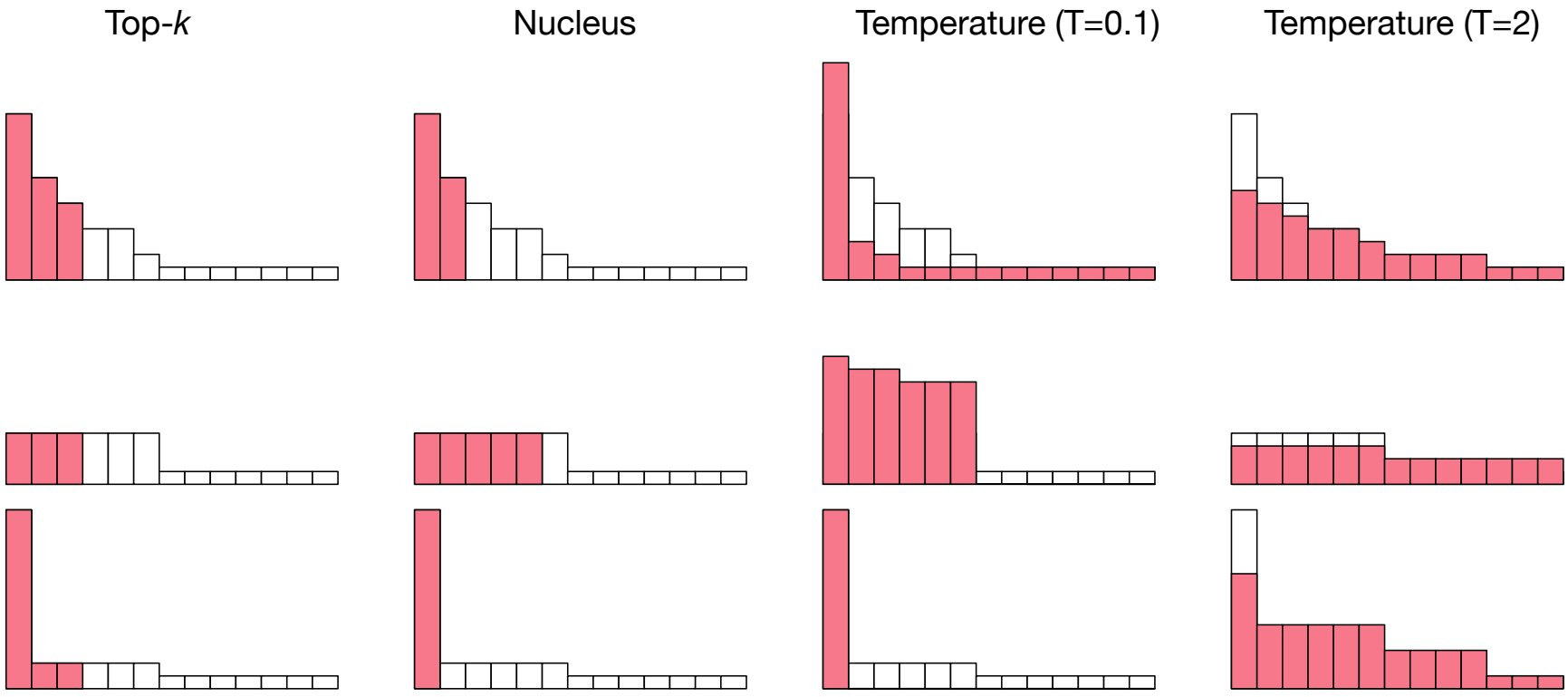
Top-*k* Nucleus Temperature (T=0.1) Temperature (T=2)

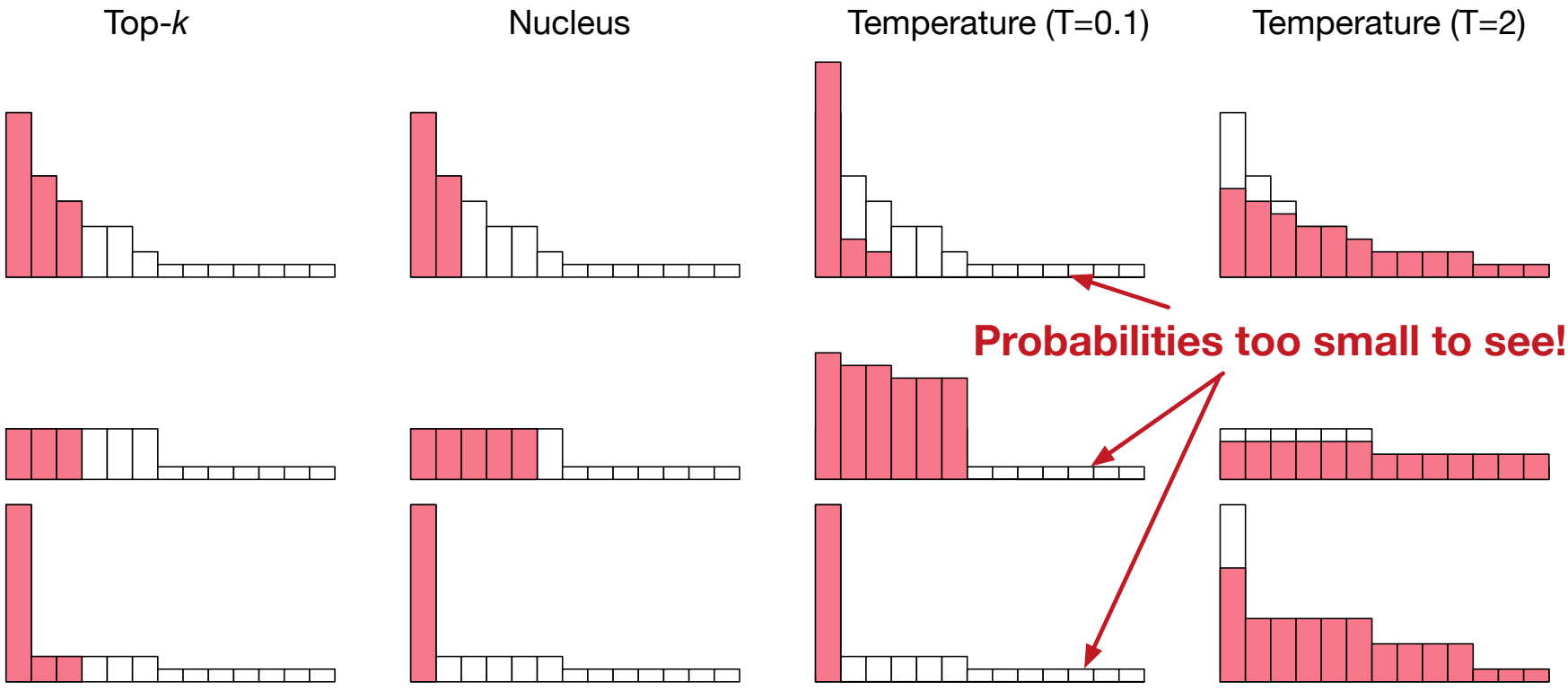

Top-*k*
Nucleus
Temperature (T=0.1)
Temperature (T=2)


Top-*k*
Nucleus
Temperature (T=0.1)
Temperature (T=2)


Top-*k*
Nucleus
Temperature (T=0.1)
Temperature (T=2)

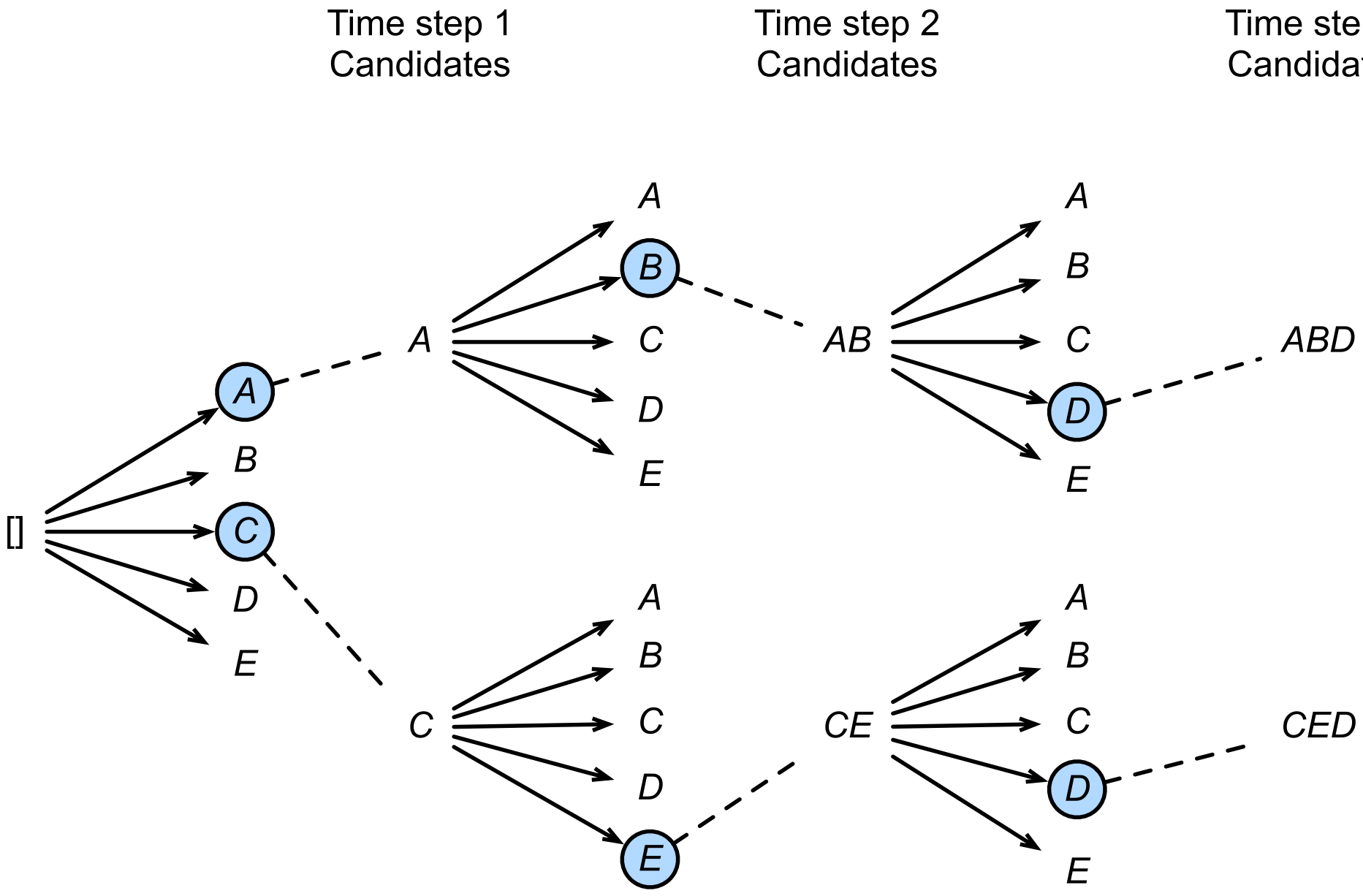

Top-*k*
Nucleus
Temperature (T=0.1)
Temperature (T=2)


Top-*k*
Nucleus
Temperature (T=0.1)
Temperature (T=2)


Top-k
Nucleus
Temperature (T=0.1)
Temperature (T=2)
Probabilities too small to see!

What do you do with samples?

- Getting out of being stuck in a garden path
- Getting diverse outputs
- Combining multiple models together
- Rescoring by a non-probability metric



From Zhang et al. (Dive into Deep Learning)


<s>
</s>
</s>
</s>
</s>

<s> →

→ </s>

→ </s> 

→ </s>

→ </s>

<s>

</s>

</s>

</s>

</s>

<s>

</s>

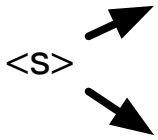
</s>

</s>

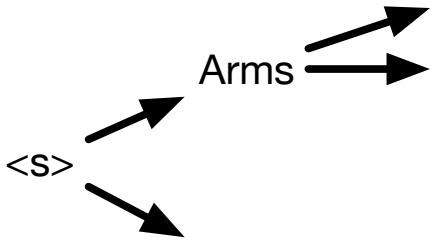
</s>

<s>

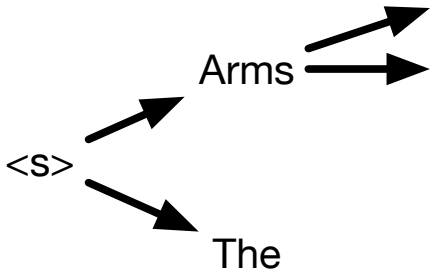
Beam Search Decoding for: Die Arme haben kein Geld



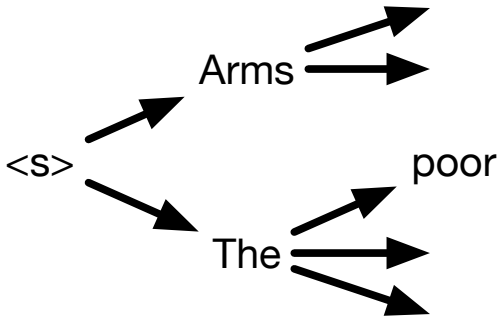
Beam Search Decoding for: Die Arme haben kein Geld



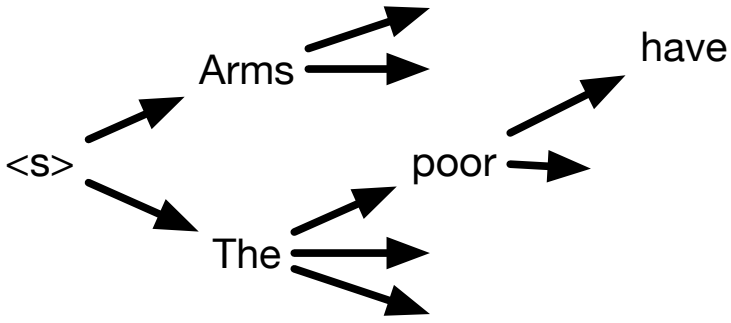
Beam Search Decoding for: Die Arme haben kein Geld



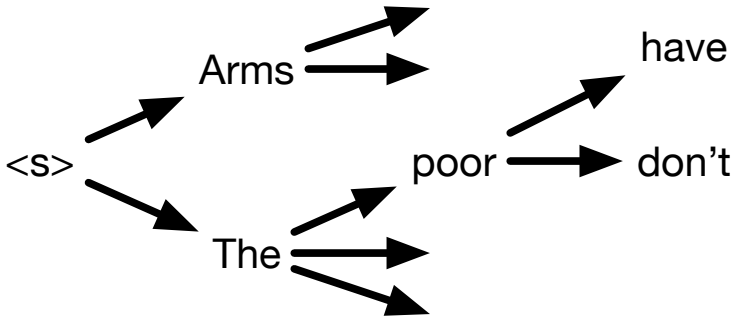
Beam Search Decoding for: Die Arme haben kein Geld



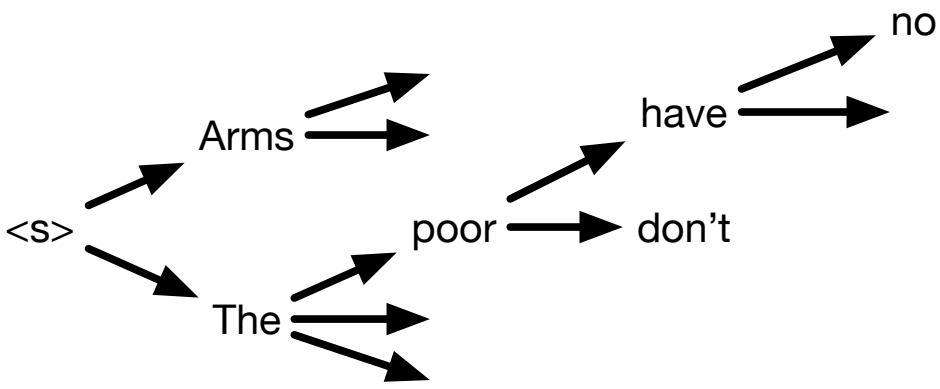
Beam Search Decoding for: Die Arme haben kein Geld



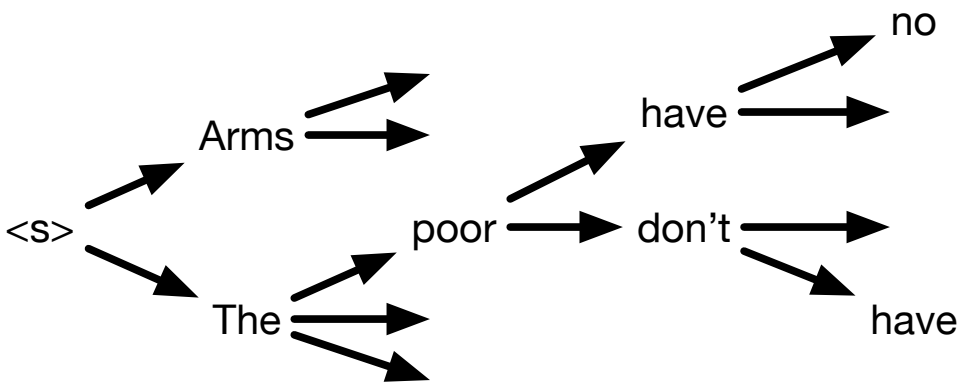
Beam Search Decoding for: Die Arme haben kein Geld



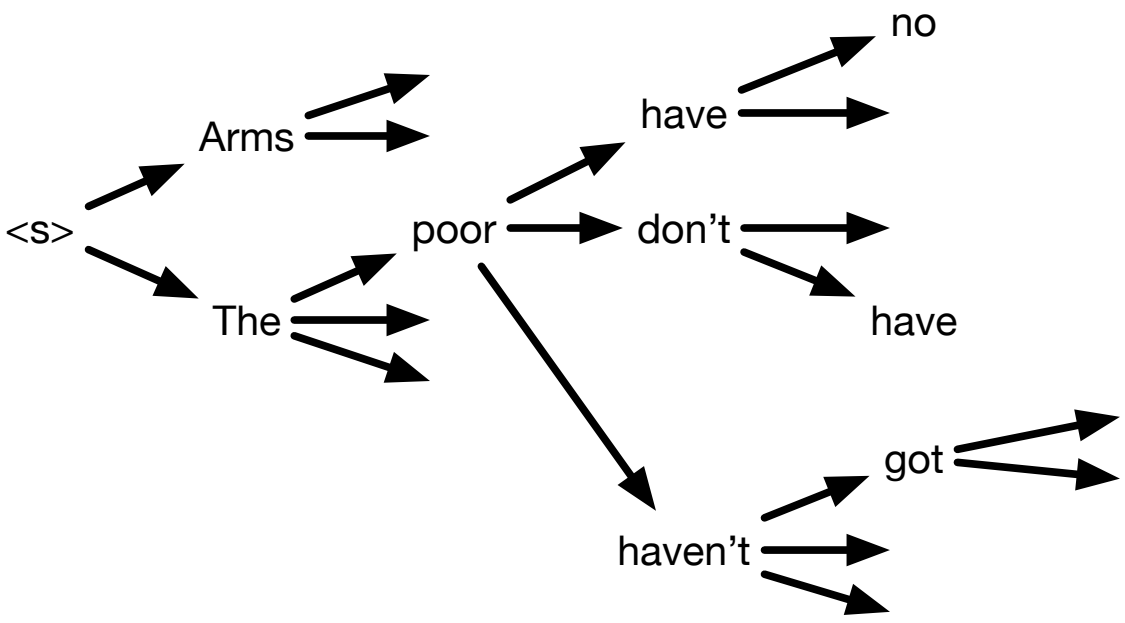
Beam Search Decoding for: Die Arme haben kein Geld



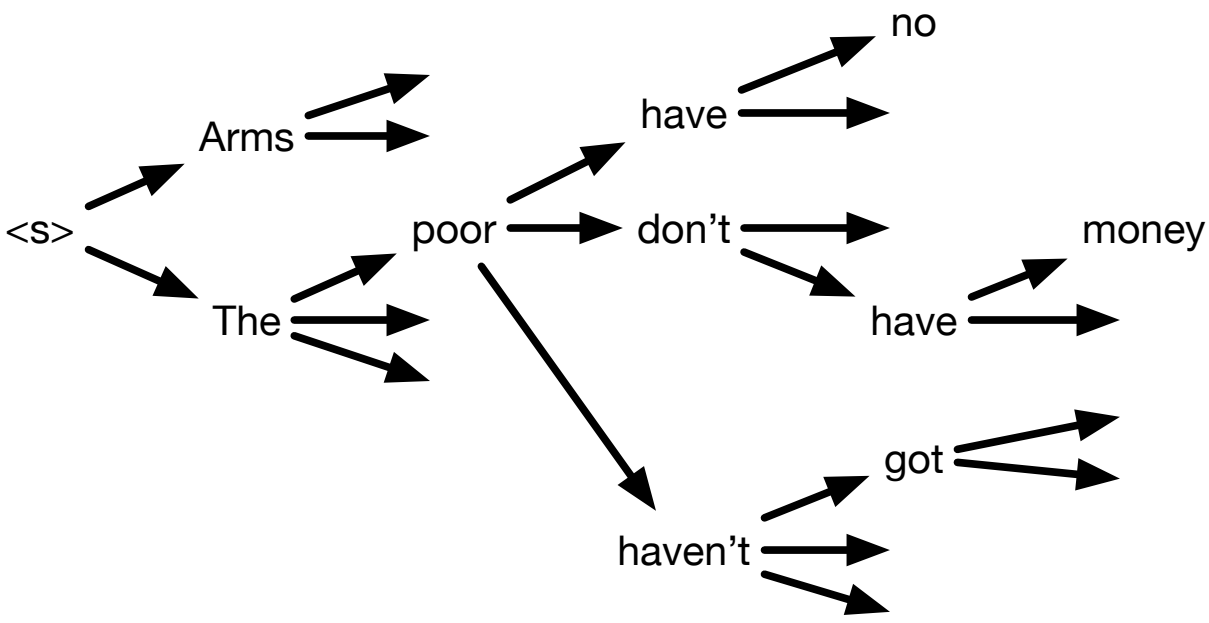
Beam Search Decoding for: Die Arme haben kein Geld



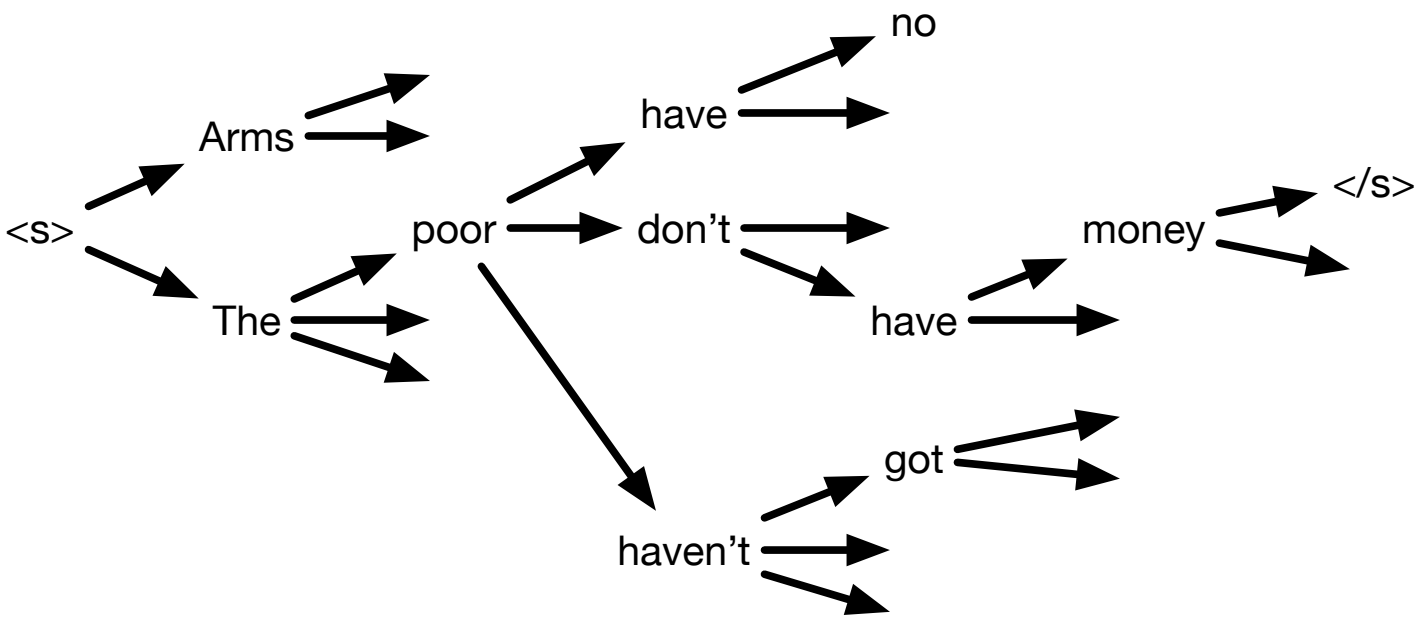
Beam Search Decoding for: Die Arme haben kein Geld



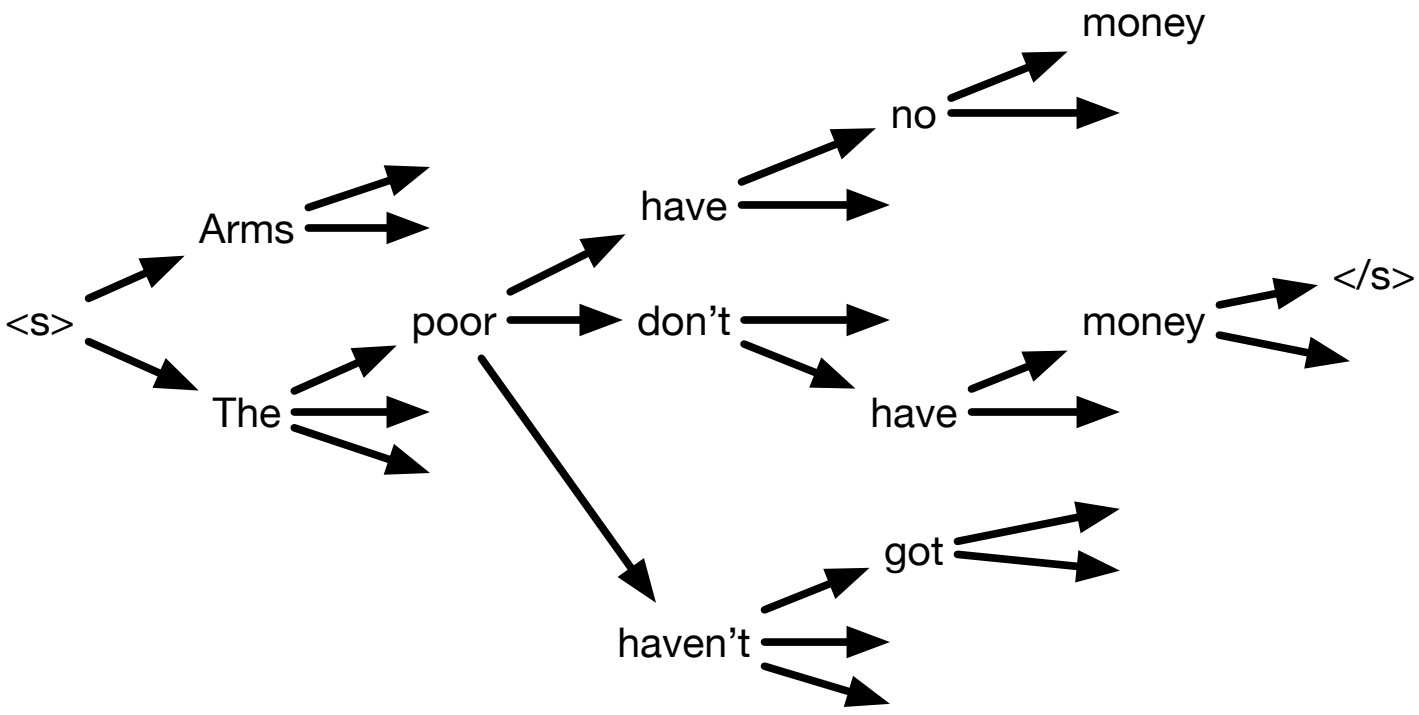
Beam Search Decoding for: Die Arme haben kein Geld



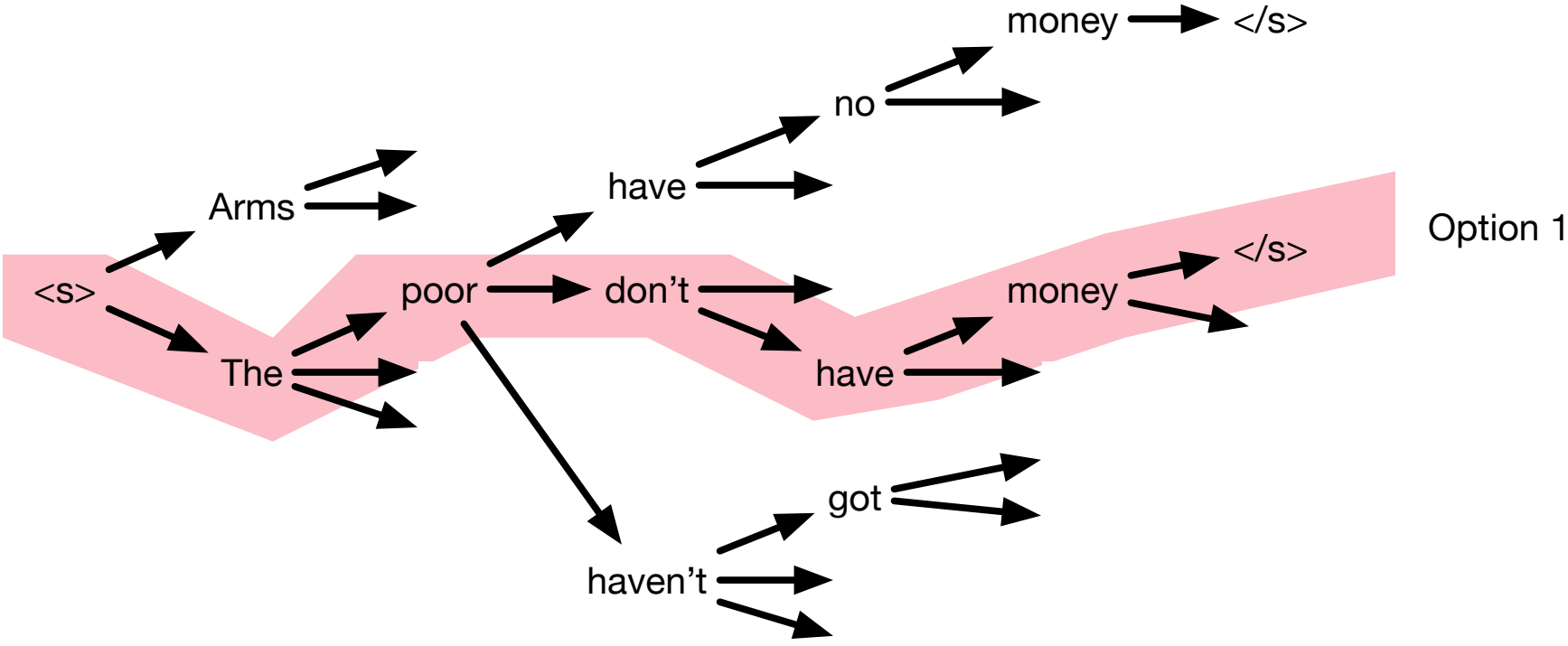
Beam Search Decoding for: Die Arme haben kein Geld



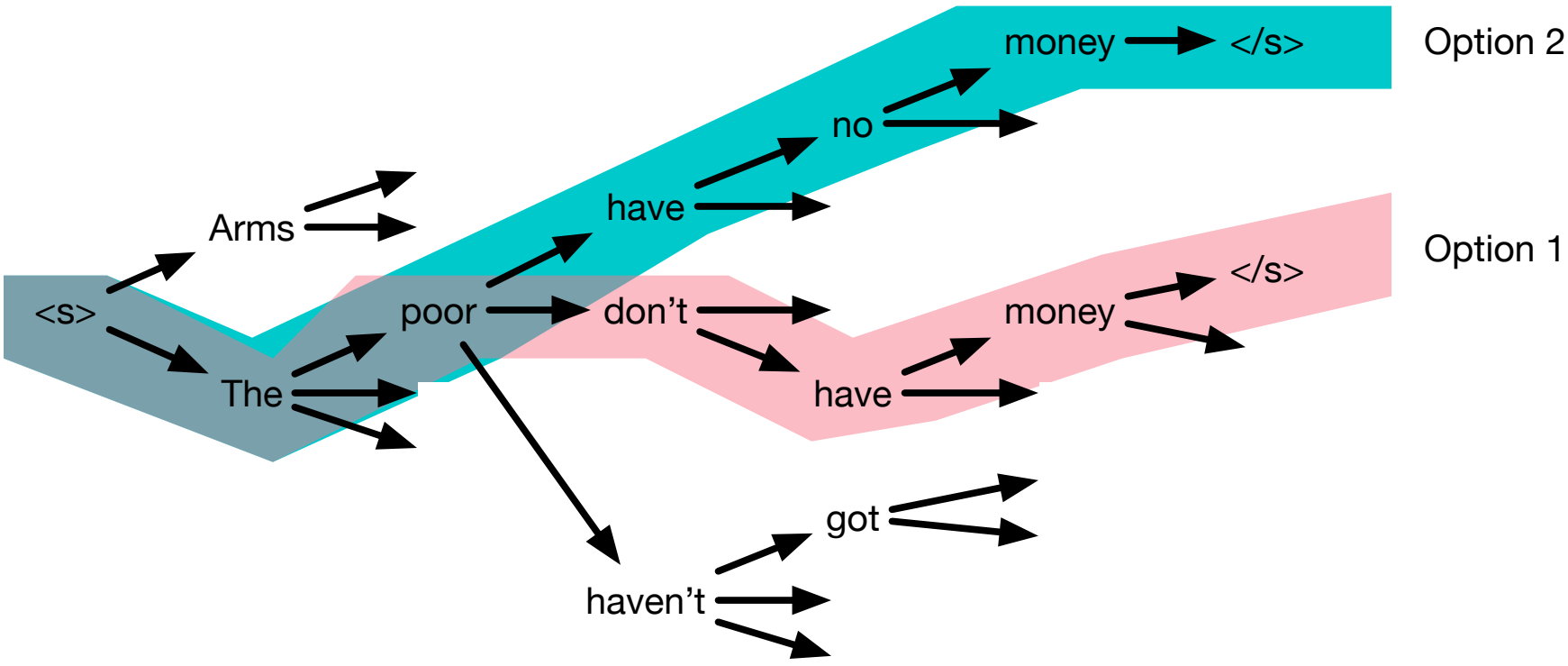
Beam Search Decoding for: Die Arme haben kein Geld



Beam Search Decoding for: Die Arme haben kein Geld



Beam Search Decoding for: Die Arme haben kein Geld



Beam Search Decoding for: Die Arme haben kein Geld

Using multiple sources

- Generate from multiple models
- Generate from multiple directions
- Generate from multiple data
- Generate from multiple temperatures

How to pick?

- Show to a user

How to pick?

- Show to a user
- Take highest probability

How to pick?

- Show to a user
- Take highest probability
- Rerank

Reinforcement Learning for Bandit Neural Machine Translation with Simulated Human Feedback

Khanh Nguyen[⊙♢] and **Hal Daumé III**[⊙♠♢♡] and **Jordan Boyd-Graber**[⊙♠♣♢]
University of Maryland: Computer Science,[⊙]Language Science,[♠]iSchool,[♣]UMIACS[♢]
Microsoft Research, New York[♡]
{kxnguyen,hal,jbg}@umiacs.umd.edu

Can Neural Machine Translation be Improved with User Feedback?

Julia Kreutzer[1*] and **Shahram Khadivi**[3] and **Evgeny Matusov**[3] and **Stefan Riezler**[1,2]
[1]Computational Linguistics & [2]IWR, Heidelberg University, Germany
{kreutzer,riezler}@cl.uni-heidelberg.de
[3]eBay Inc., Aachen, Germany
{skhadivi,ematusov}@ebay.com

RankGen — Improving Text Generation with Large Ranking Models *(EMNLP 2022)*




Kalpesh Krishna Yapei Chang John Wieting Mohit Iyyer

UMassAmherst
Manning College of Information & Computer Sciences

Google AI

Decoding Method	GPT2-md PG19	GPT2-md wiki	GPT2-XL PG19	GPT2-XL wiki
Nucleus ($p = 0.9$)	73.0	74.6	74.4	75.0
Eta (Hewitt et al., 2022)	76.9	71.2	76.9	74.8
Contrastive methods				
search (Su et al., 2022)	5.3	21.2	54.0	43.2
decode (Li et al., 2022)	65.2	83.2	73.2	84.9
RANKGEN-all-XL (ours)				
rerank full ancestral	**79.0**	84.9	**79.0**	86.4
beam search nucleus	76.2	**88.9**	77.0	**89.4**

Automatic Song Translation for Tonal Languages

Fenfei Guo
University of Maryland
fguo1@umd.edu

Chen Zhang
Zhejiang University
zc99@zju.edu.cn

Zhirui Zhang
Tencent AI Lab
zrustc11@gmail.com

Qixin He
Purdue University
heqixin@purdue.edu

Kejun Zhang
Zhejiang University
zhangkejun@zju.edu.cn

Jun Xie
Alibaba DAMO Academy
qingjing.xj@alibaba-inc.com

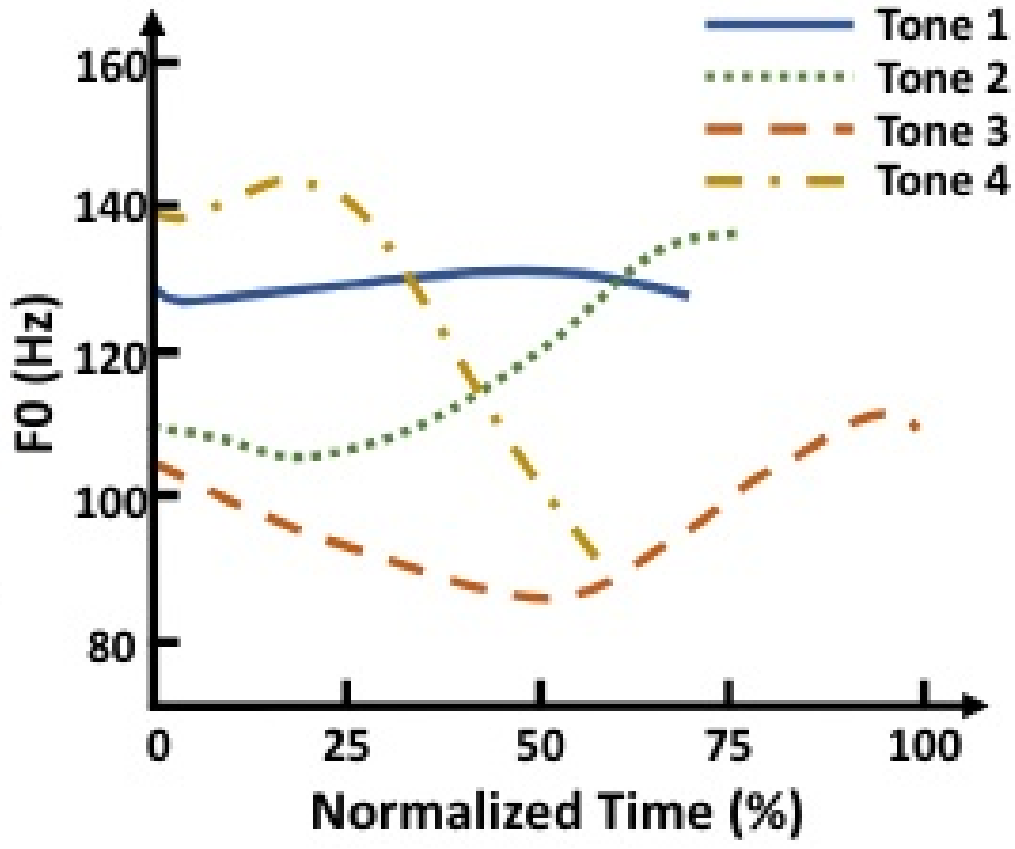
Jordan Boyd-Graber
CS, iSchool, UMIACS, LSC
University of Maryland
jbg@umiacs.umd.edu

Tone 1: 叔 shū (uncle)

Tone 2: 熟 shú (cooked, familiar)

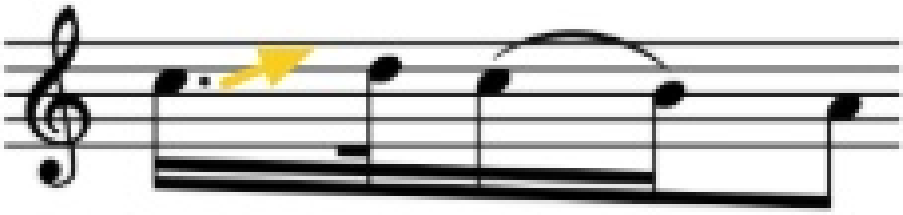
Tone 3: 鼠 shǔ (mouse, Muroidea)

Tone 4: 树 shù (tree)



Tones in Chinese (for “shu”, not “ma” like I said)

Original Lyrics (Inconsistent Tone)



sì zài yǎn qián

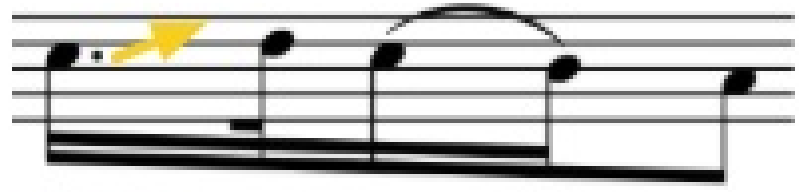
似 在 眼 前

appear where eye front

As if before my eyes

Inter-syllable pitch alignment score: 0.5

Misheard Lyrics (Consistent Tone)



sǐ zài yǎn qián

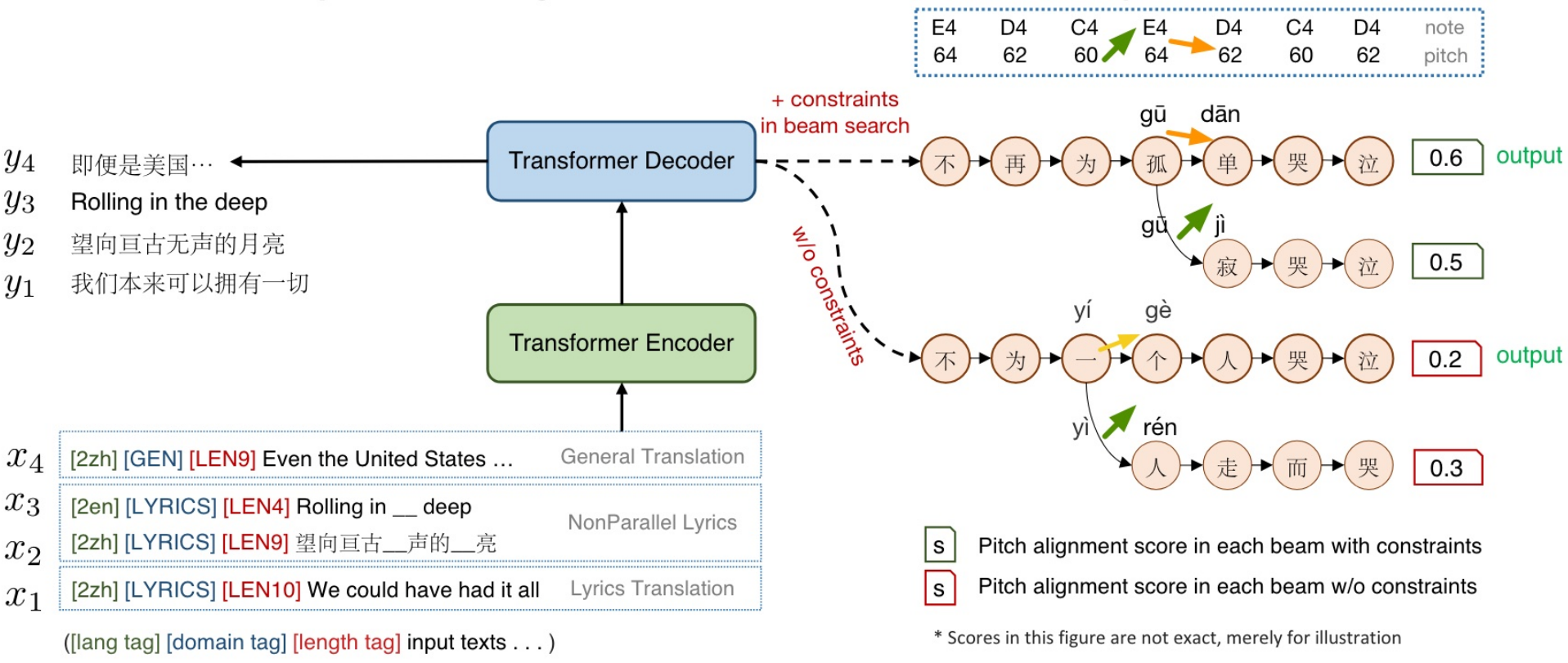
死 在 眼 前

death where eye front

Die before my eyes

Inter-syllable pitch alignment score: 0.75

Misheard lyrics when the tones are wrong



Decoding song translations with tones in decoder