+

CHULA ΣNGINEERING COMPUTER
Foundation toward Innovation

NLP

Decoding

2110572: Natural Language Processing Systems

Peerapon Vateekul & Ekapol Chuangsuwanich
Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University

Credit:

- Kasidis Kanwatchara
- Can Udomcharoenchaikit & Nattachai Tretasayuth

Outline

- Part1) Introduction
- Part2) Greedy decoding
- Part3) Beam search
- Part4) Random sampling

+

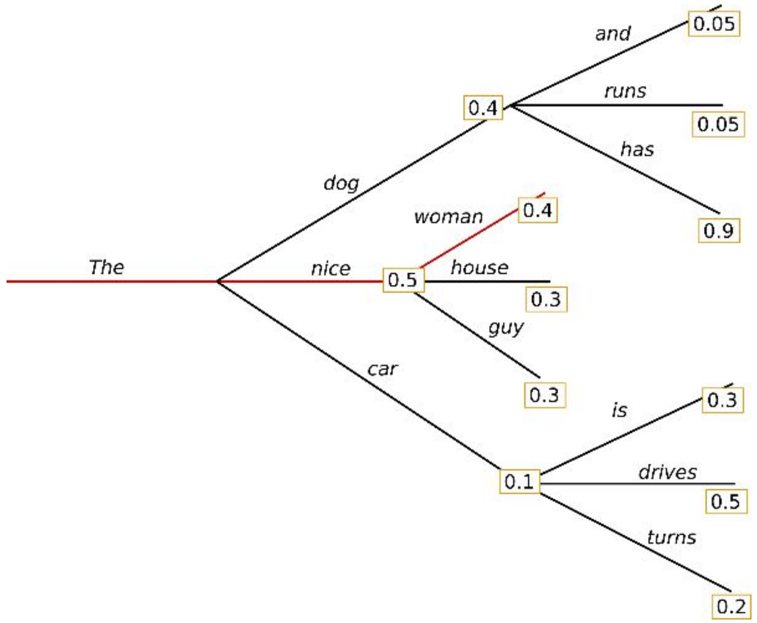
Part1) Introduction

Introduction

In sequence generation tasks, the task of selecting what the model outputs as a prediction is called **decoding**.

There are 3 methods for decoding:

1. Greedy decoding
2. Beam search
3. Random sampling



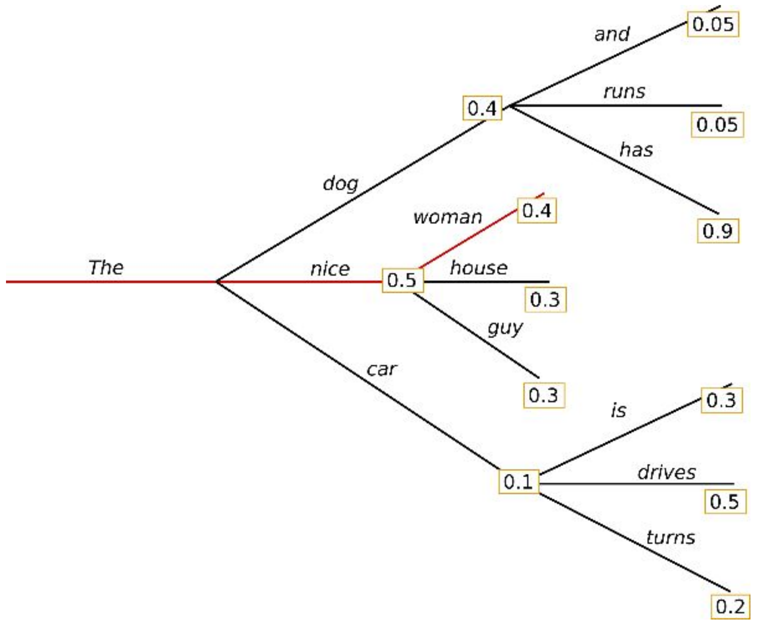
+

Part2) Greedy decoding

Greedy decoding

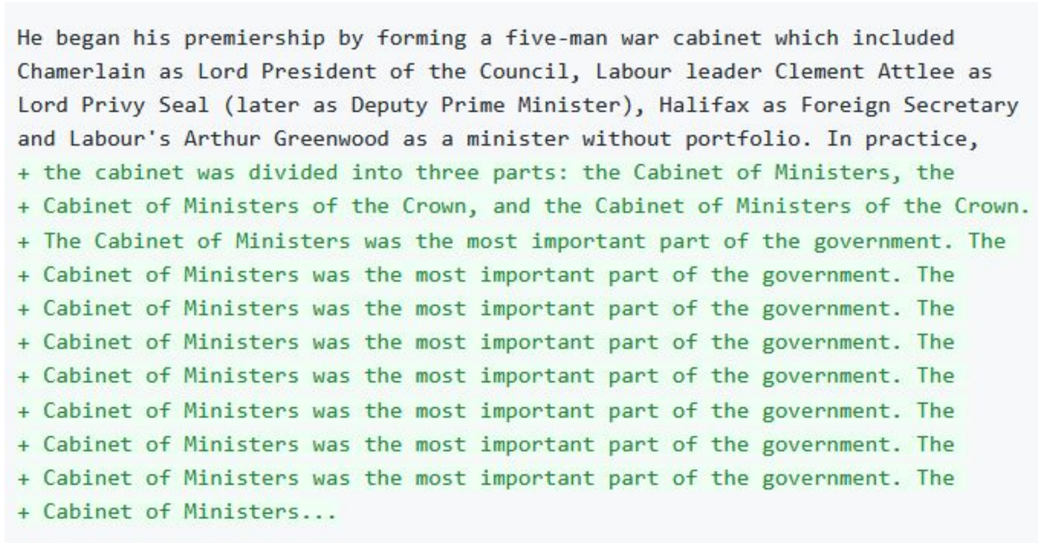
Greedy decoding simply selects the token with **the highest probability** as the next token.

As shown in the picture, after "the", the continuation with the highest probability is the word "nice" therefore it is selected as the next token. This is done until it reaches the model's max sequence length or upon encountering an end-of-sentence token.



Greedy decoding

Greedy decoding is fast and simple, however, the generated text is usually sub-optimal. Sometimes, the model can even repeat itself.



```
He began his premiership by forming a five-man war cabinet which included
Chamerlain as Lord President of the Council, Labour leader Clement Attlee as
Lord Privy Seal (later as Deputy Prime Minister), Halifax as Foreign Secretary
and Labour's Arthur Greenwood as a minister without portfolio. In practice,
+ the cabinet was divided into three parts: the Cabinet of Ministers, the
+ Cabinet of Ministers of the Crown, and the Cabinet of Ministers of the Crown.
+ The Cabinet of Ministers was the most important part of the government. The
+ Cabinet of Ministers was the most important part of the government. The
+ Cabinet of Ministers was the most important part of the government. The
+ Cabinet of Ministers was the most important part of the government. The
+ Cabinet of Ministers was the most important part of the government. The
+ Cabinet of Ministers was the most important part of the government. The
+ Cabinet of Ministers was the most important part of the government. The
+ Cabinet of Ministers was the most important part of the government. The
+ Cabinet of Ministers...
```

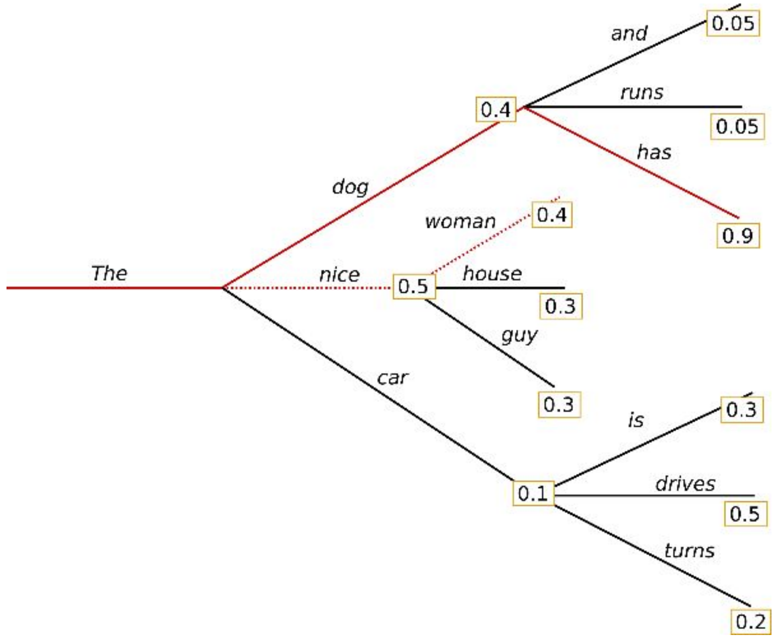
+

Part3) Beam search

Beam Search

Beam search reduces the risk of **missing hidden high probability word sequences** by keeping the most likely *num_beams* of hypotheses at each time step and eventually choosing the hypothesis that has the overall highest probability.

Beam search is relatively computationally expensive since it basically generates multiple sequences but it always find an output sequence that is more probable than greedy decoding.



Beam Search

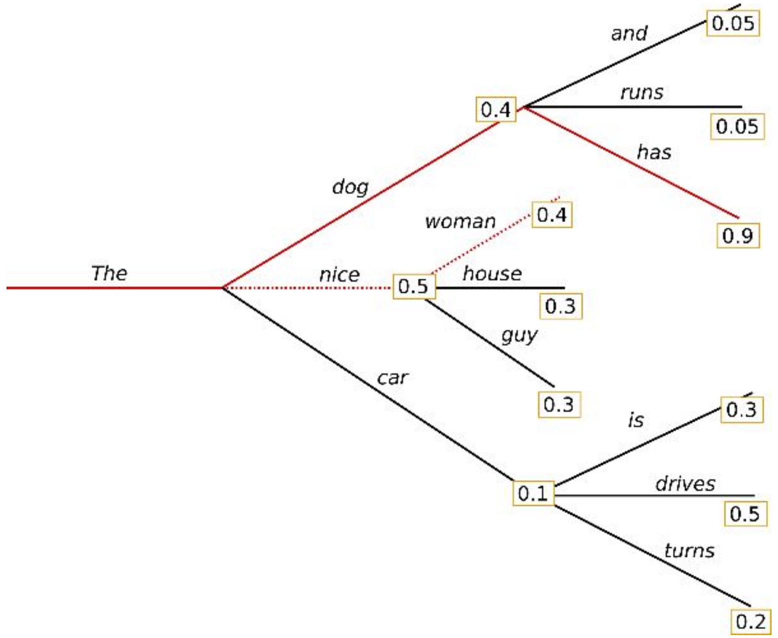
From the example, we consider 2 “beams”, i.e., we only keep the top 2 most probable sequence while we go through the generation process.

At time step 1, the beam search algorithm keeps tab on 2 most probable continuations: (“The”, “nice”) and (“The”, “dog”).

At time step 2, it continues to find the next word for each beam:

(“The”, “dog”, “has”) = 0.4*0.9 = **0.36**
(“The”, “nice”, “woman”) = 0.5*0.4 = 0.2

Suppose this is the end of the generation, the output of the algorithm will be “The dog has”.



Beam Search

As can be seen, the five beam hypotheses are only marginally different to each other

```
0: I enjoy walking with my cute dog, but I'm not sure if I'll ever be able to walk with him again.
I've been thinking about this for a while now, and I think it's time for me to

1: I enjoy walking with my cute dog, but I'm not sure if I'll ever be able to walk with her again.
I've been thinking about this for a while now, and I think it's time for me to

2: I enjoy walking with my cute dog, but I'm not sure if I'll ever be able to walk with him again.
I've been thinking about this for a while now, and I think it's a good idea to

3: I enjoy walking with my cute dog, but I'm not sure if I'll ever be able to walk with him again.
I've been thinking about this for a while now, and I think it's time to take a

4: I enjoy walking with my cute dog, but I'm not sure if I'll ever be able to walk with him again.
I've been thinking about this for a while now, and I think it's a good idea.
```

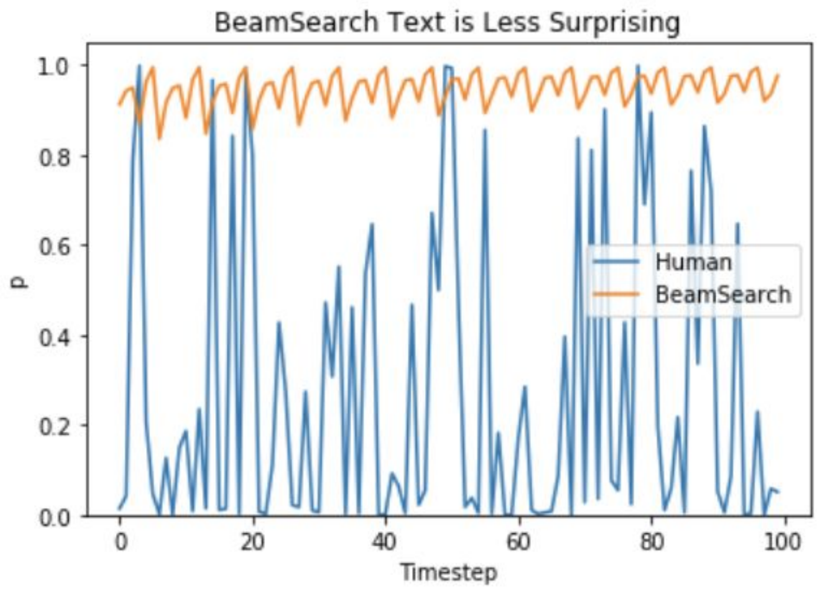
https://huggingface.co/blog/how-to-generate

Beam Search

Beam search can work very well in tasks where the length of the desired generation is predictable as in **MT** or **summarization**.

But this is NOT the case for open-ended generation where the desired output length can vary greatly, e.g. dialog and story generation.

As argued in Ari Holtzman et al. (2019), high quality human language does not follow a distribution of high probability next words. In other words, as humans, we want generated text to surprise us and not to be boring/predictable



So let's stop being boring and introduce some randomness 🤪.

https://huggingface.co/blog/how-to-generate

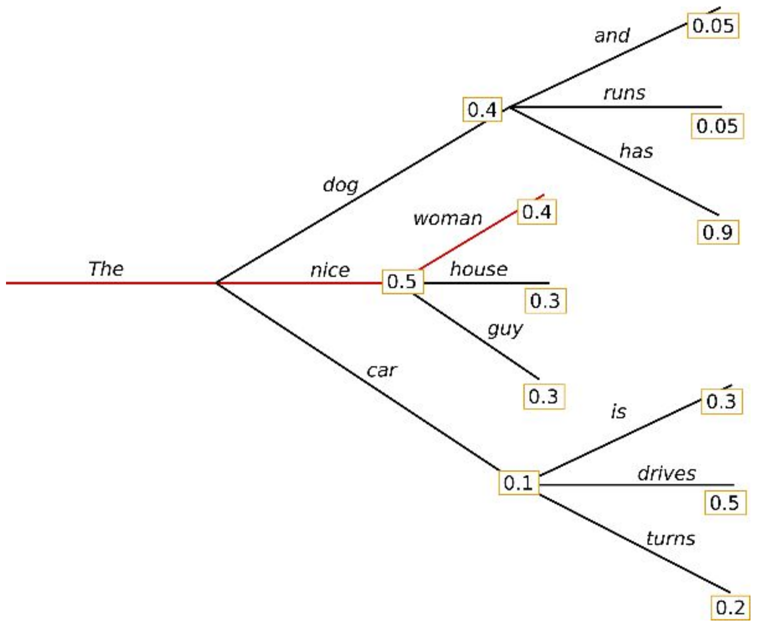
+

Part4) Random sampling

Random sampling

Random sampling chooses the next token **based on the probabilities**.

Using random sampling, the probability of selecting the token "nice", "dog", and "car" as the continuation is 50%, 40%, and 10%, respectively. The decoding also proceeds until reaching max sequence length or encountering the end-of-sentence token.

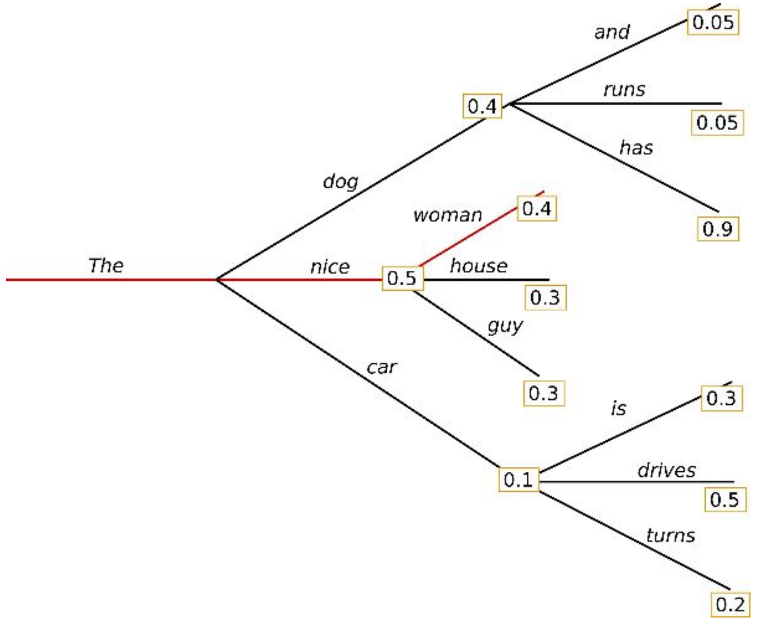


Random sampling

By the laws of probability, you are bound to eventually generating something gibberish by *selecting multiple low probability tokens* in a row.

I enjoy walking with my cute dog for the rest of the day, but this had me staying in an unusual room and not going on nights out with friends (which will always be wondered for a mere minute or so at this point).

To prevent this problem, top-k and top-p (nucleus sampling) are often used to improve the generation quality.



https://huggingface.co/blog/how-to-generate

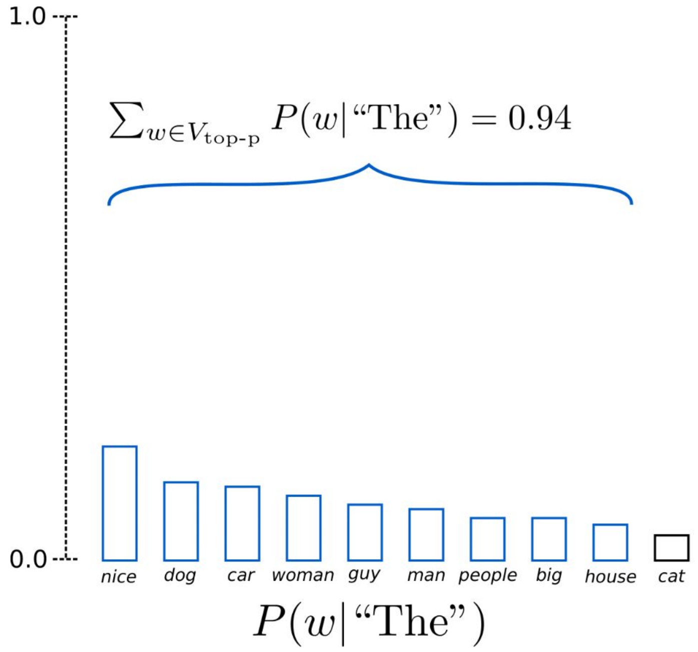
Random sampling

Top-k sampling simply limits the token selection to just top k (usually 20-40) words with the highest probabilities.

Top-p sampling or nucleus sampling dynamically limits the number of words by setting a probability threshold. Top-p sampling chooses from the smallest possible set of tokens whose cumulative probability exceeds the probability threshold.

For example, if we set p as 0.92, top-p sampling will select the *minimum* number of tokens whose cumulative probability is more than 92%

Note that both top-k and top-p can also be used together.



Softmax Temperature

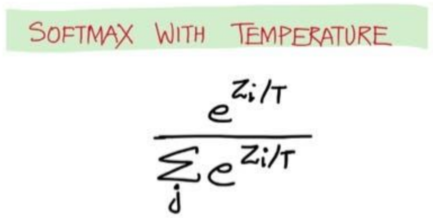
In model Generation, we could also control the *temperature* of the softmax.

This would make the probability distribution sharper or more leveled.

Low Temperature (<1 e.g. 0.2, 0.5)
Makes the model more **deterministic**

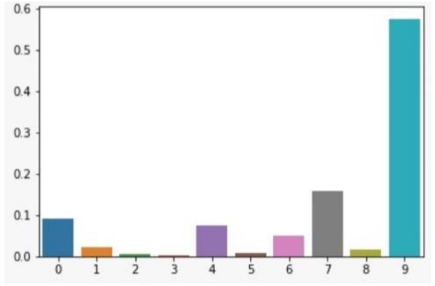
Temperature = 1
Default softmax behavior

High Temperature (>1 e.g. 1.5, 2)
Makes the model more **random** and **diverse**.

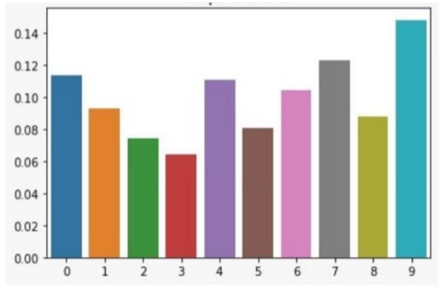


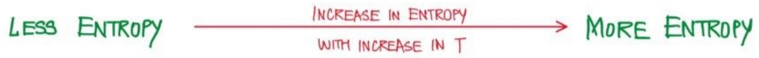
$$\frac{e^{z_i/T}}{\sum_j e^{z_j/T}}$$

T < 1



T > 1


0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
1
2
3
4
5
6
7
8
9


LESS ENTROPY
INCREASE IN ENTROPY
WITH INCREASE IN T
MORE ENTROPY