How to deal with a vocabulary?

Outline

- Computing *softmax* for a large vocabulary is slow!
  - Hierarchical softmax
- Even a large vocabulary has *OOV words*:
  - Copy mechanism
  - Sub-word modeling
    - Word-character hybrid models
    - Byte-pair encoding

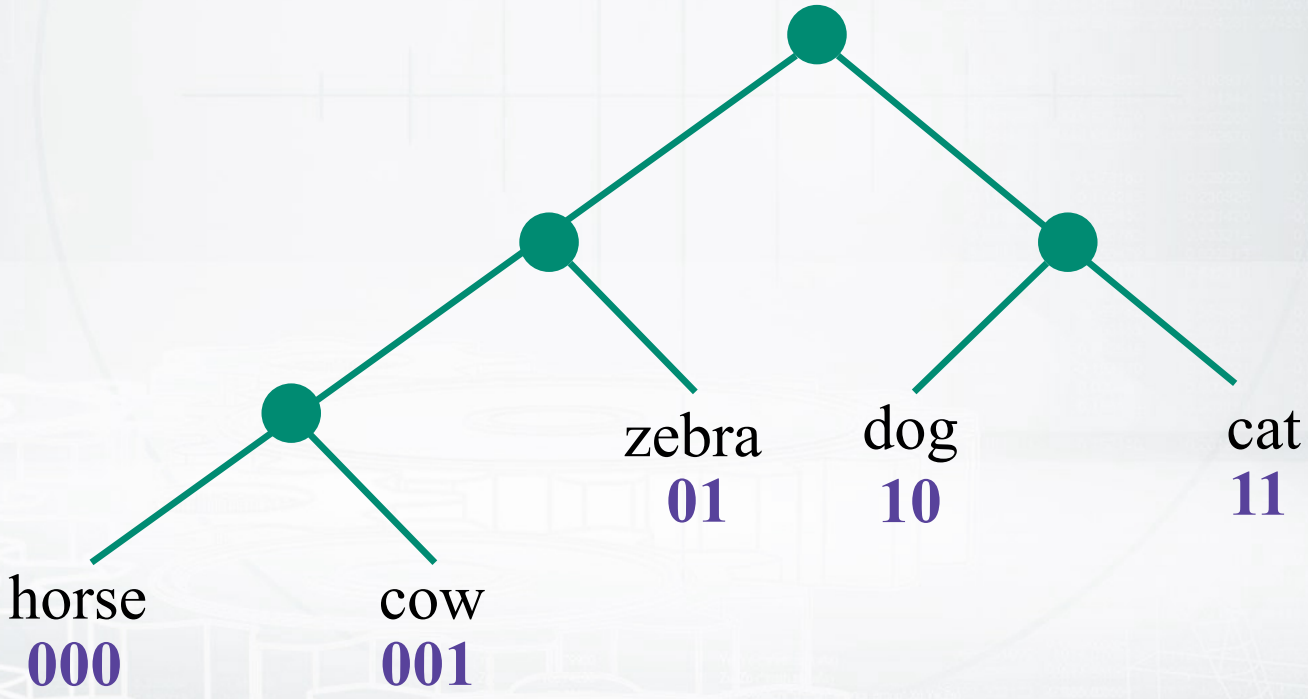
Outline

- Computing *softmax* for a large vocabulary is slow!
  - **Hierarchical softmax**
- Even a large vocabulary has *OOV words*:
  - Copy mechanism
  - Sub-word modeling
    - Word-character hybrid models
    - Byte-pair encoding

Hierarchical softmax

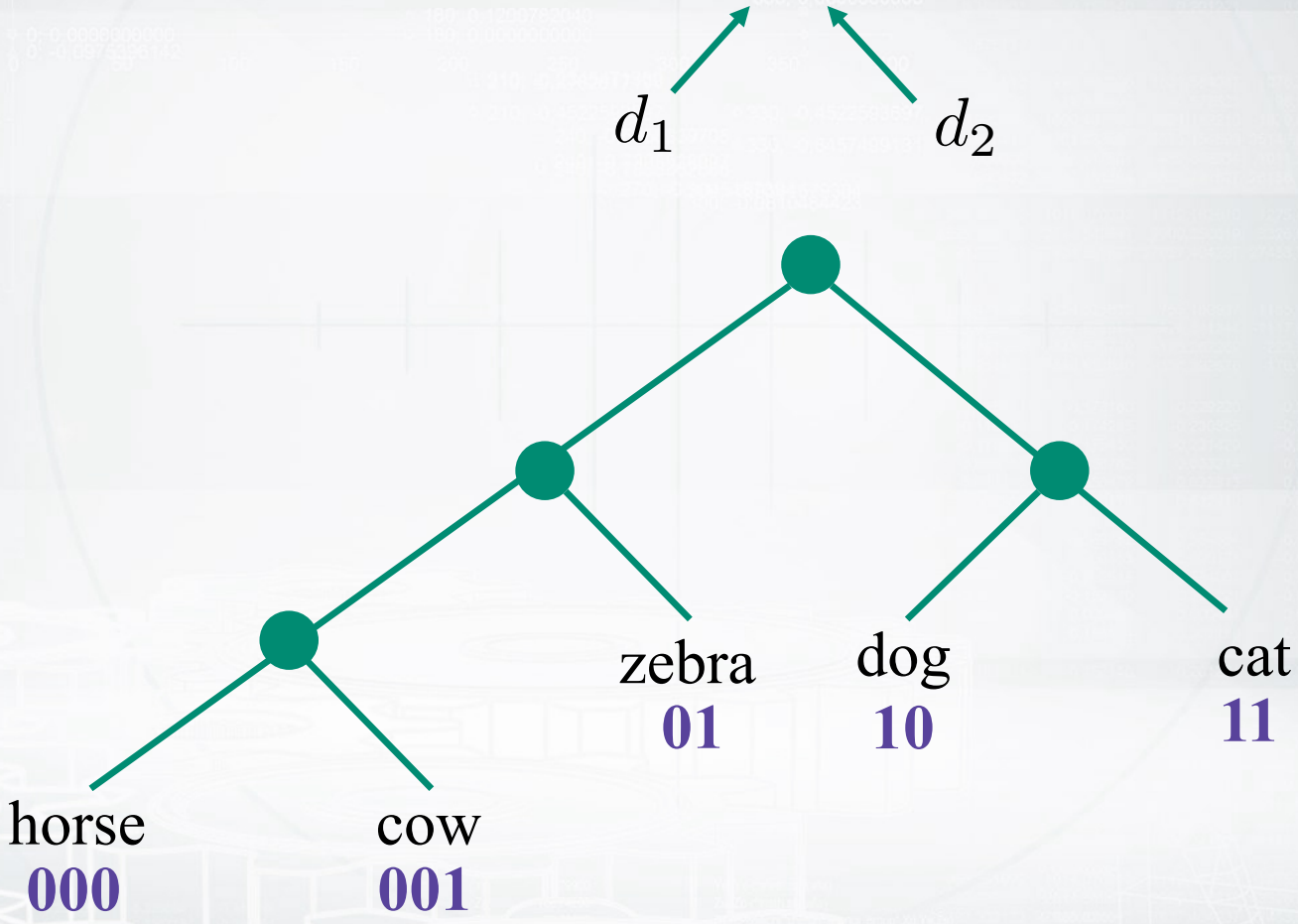
Each word is uniquely represented by a binary code:

- 0 means “go left”, 1 means “go right”



Hierarchical softmax

E.g. for **zebra** the code is $d = (0, 1)$



Scaling softmax

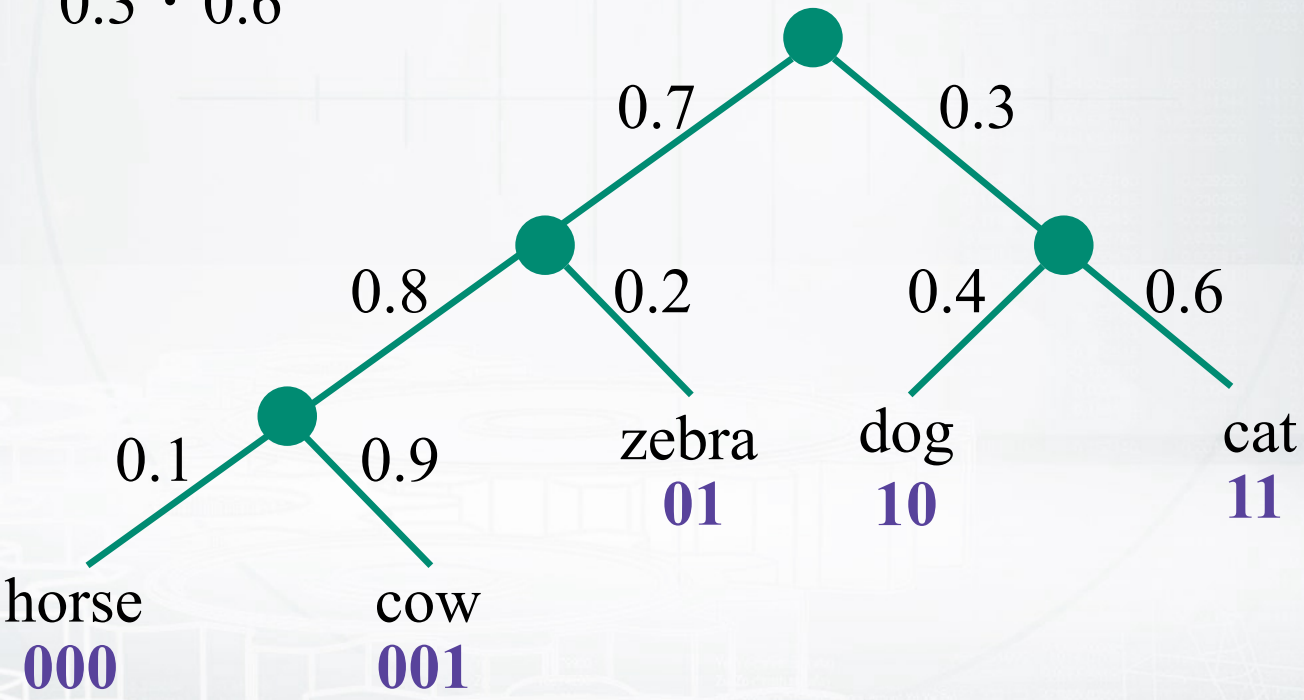
Express the probability of a word (zebra) as a product of probabilities of the binary decisions along the path (d_1, d_2).

$$p(w_n = w | w_1^{n-1}) = \prod_i p(d_i | w_1^{n-1})$$

Do you believe that it sums to 1?

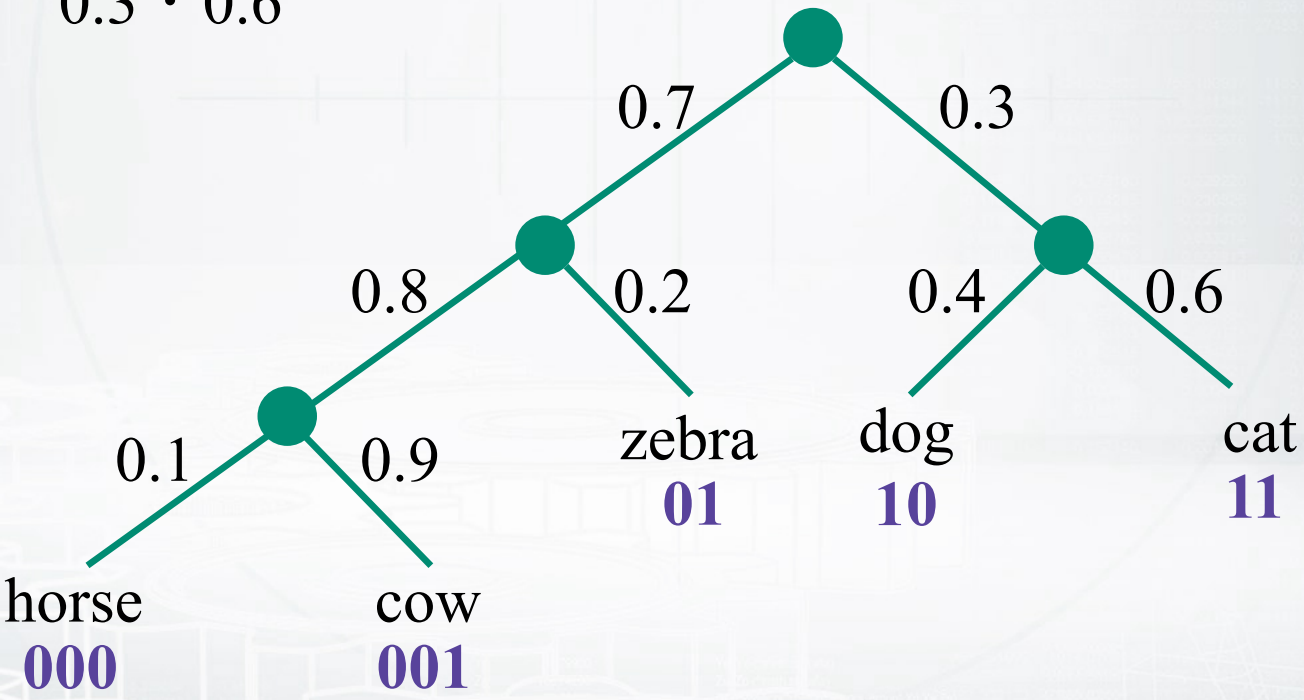
Hierarchical softmax

$$
\begin{aligned}
& 0.7 \cdot 0.8 \cdot 0.1 \\
& 0.7 \cdot 0.8 \cdot 0.9 \\
+ \quad & 0.7 \cdot 0.2 \\
& 0.3 \cdot 0.4 \\
& 0.3 \cdot 0.6
\end{aligned}
$$



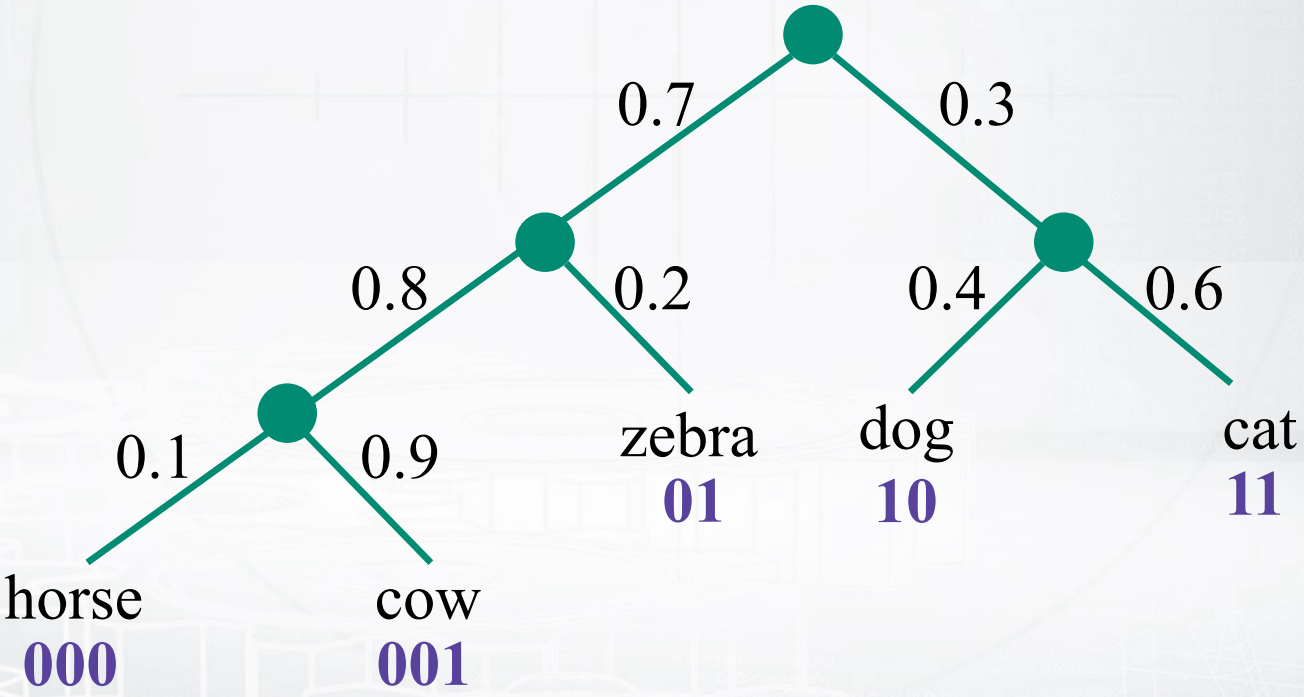
Hierarchical softmax

$$
\begin{aligned}
& 0.7 \cdot 0.8 \cdot 0.1 \\
& 0.7 \cdot 0.8 \cdot 0.9 \\
+ \quad & 0.7 \cdot 0.2 \\
& 0.3 \cdot 0.4 \\
& 0.3 \cdot 0.6
\end{aligned}
$$



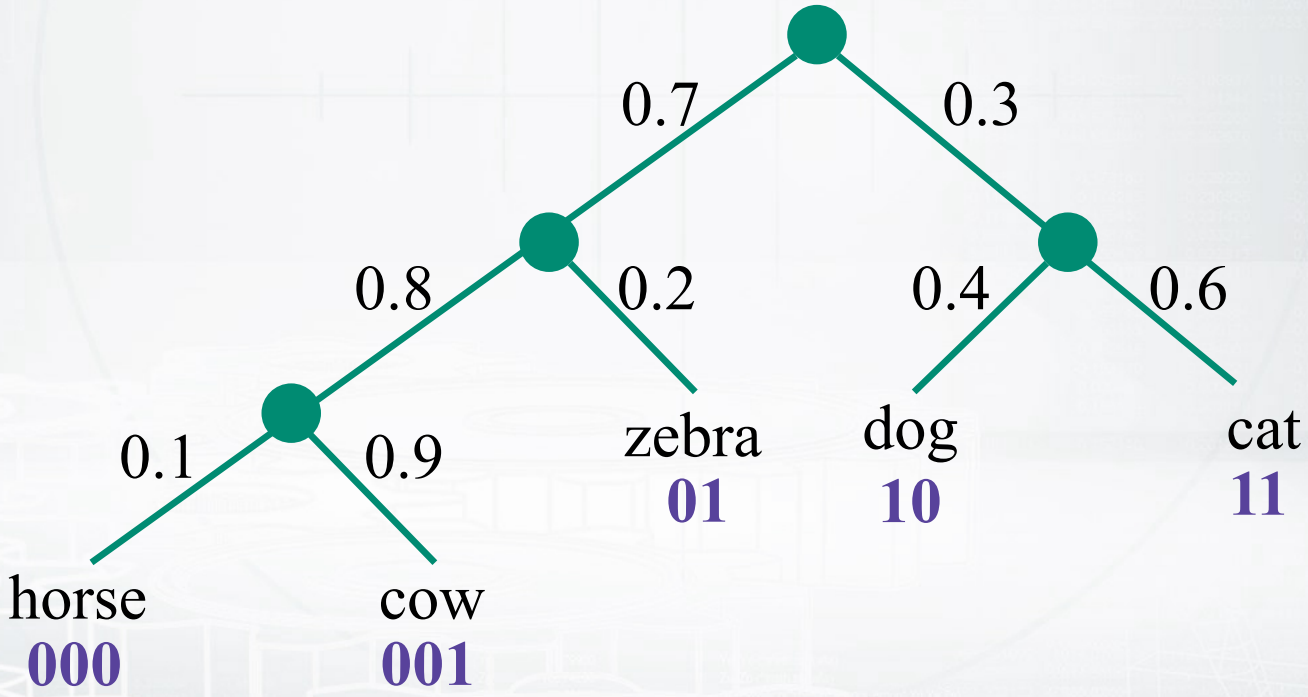
Hierarchical softmax

$$
\begin{aligned}
& 0.7 \cdot 0.8 \\
+\ & 0.7 \cdot 0.2 \\
+\ & 0.3 \cdot 0.4 \\
+\ & 0.3 \cdot 0.6
\end{aligned}
$$



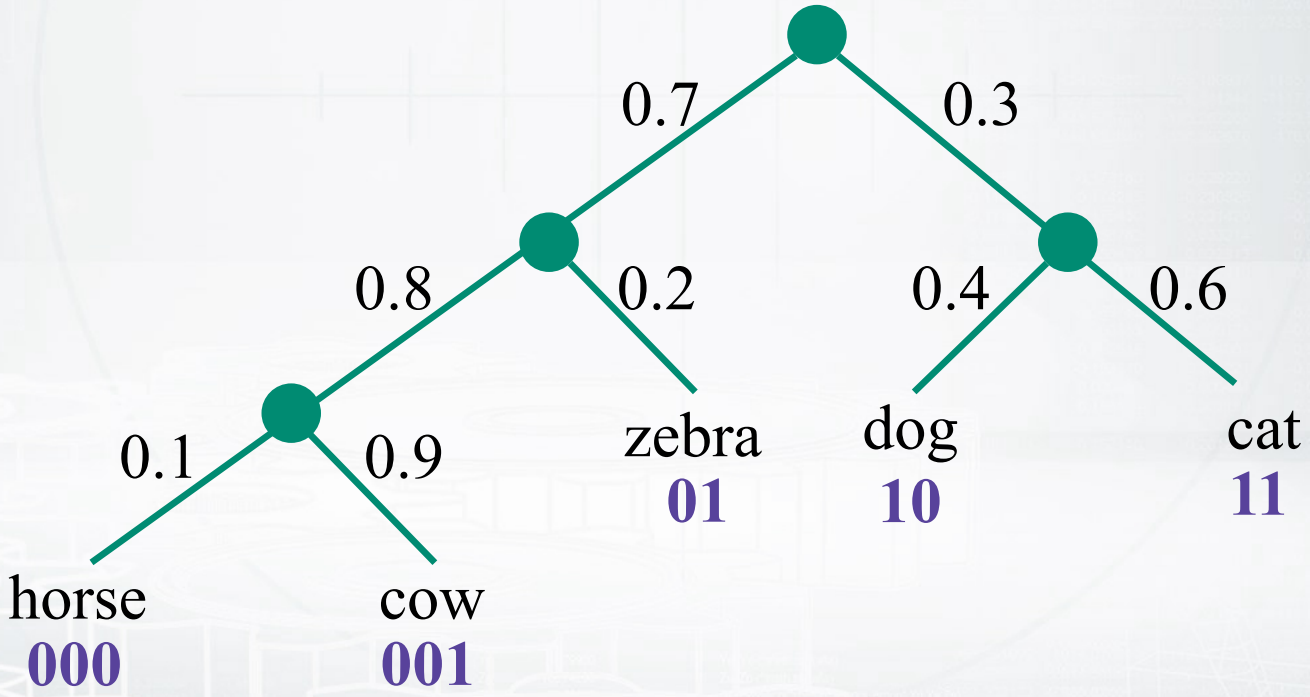
Hierarchical softmax

$$
\begin{aligned}
& 0.7 \cdot 0.8 \\
& 0.7 \cdot 0.2 \\
+ \quad & 0.3 \cdot 0.4 \\
& 0.3 \cdot 0.6
\end{aligned}
$$



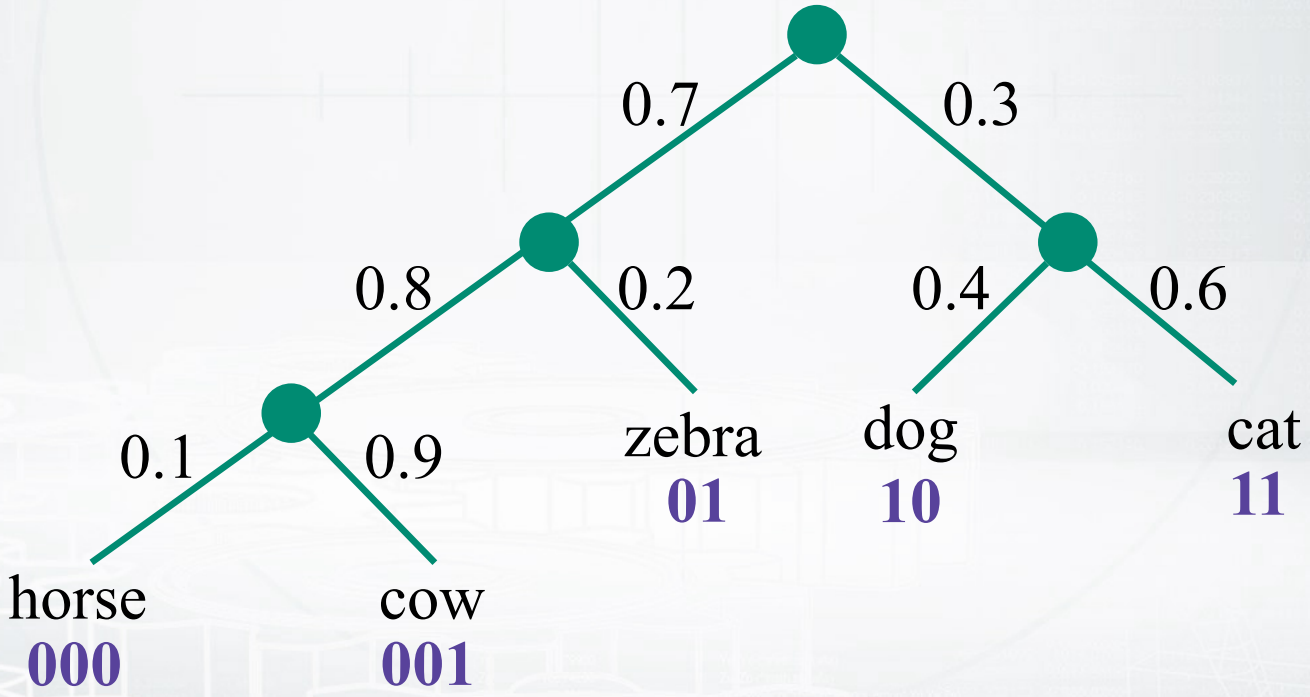
Hierarchical softmax

$$
\begin{aligned}
& 0.7 \\
+\quad & 0.3 \cdot 0.4 \\
+\quad & 0.3 \cdot 0.6
\end{aligned}
$$



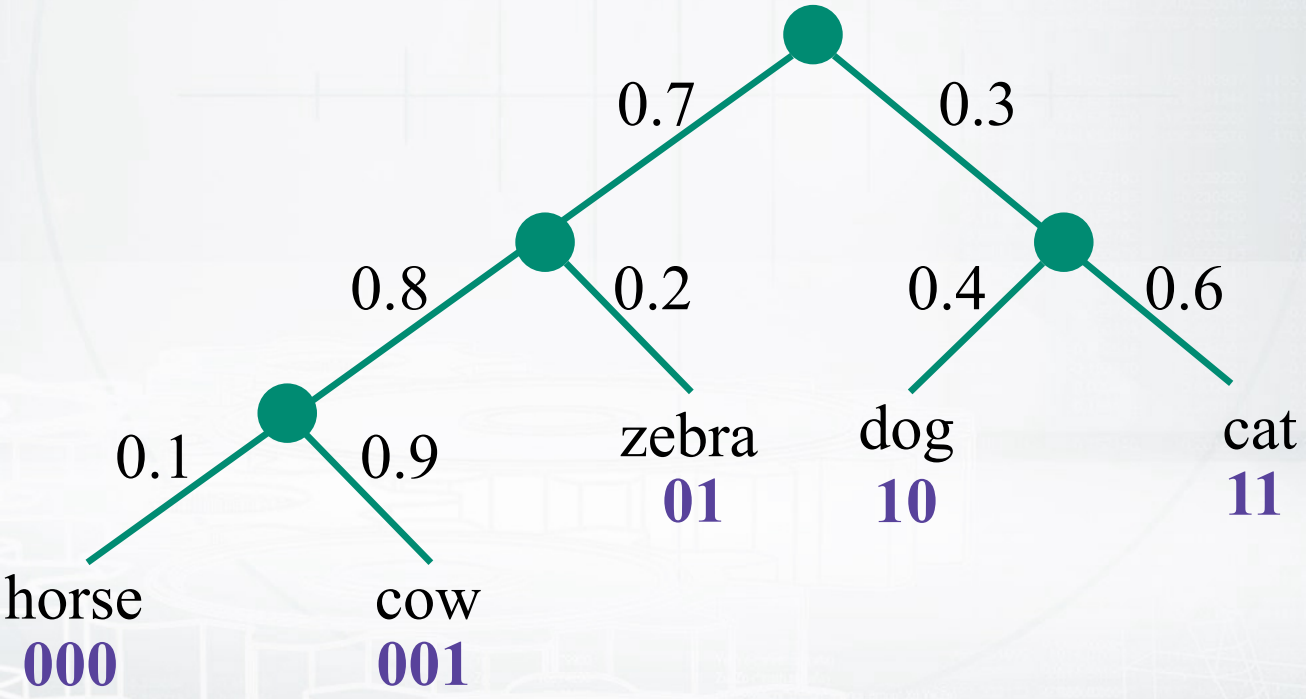
Hierarchical softmax

$$
\begin{aligned}
& 0.7 \\
+\ & 0.3 \cdot 0.4 \\
+\ & 0.3 \cdot 0.6
\end{aligned}
$$



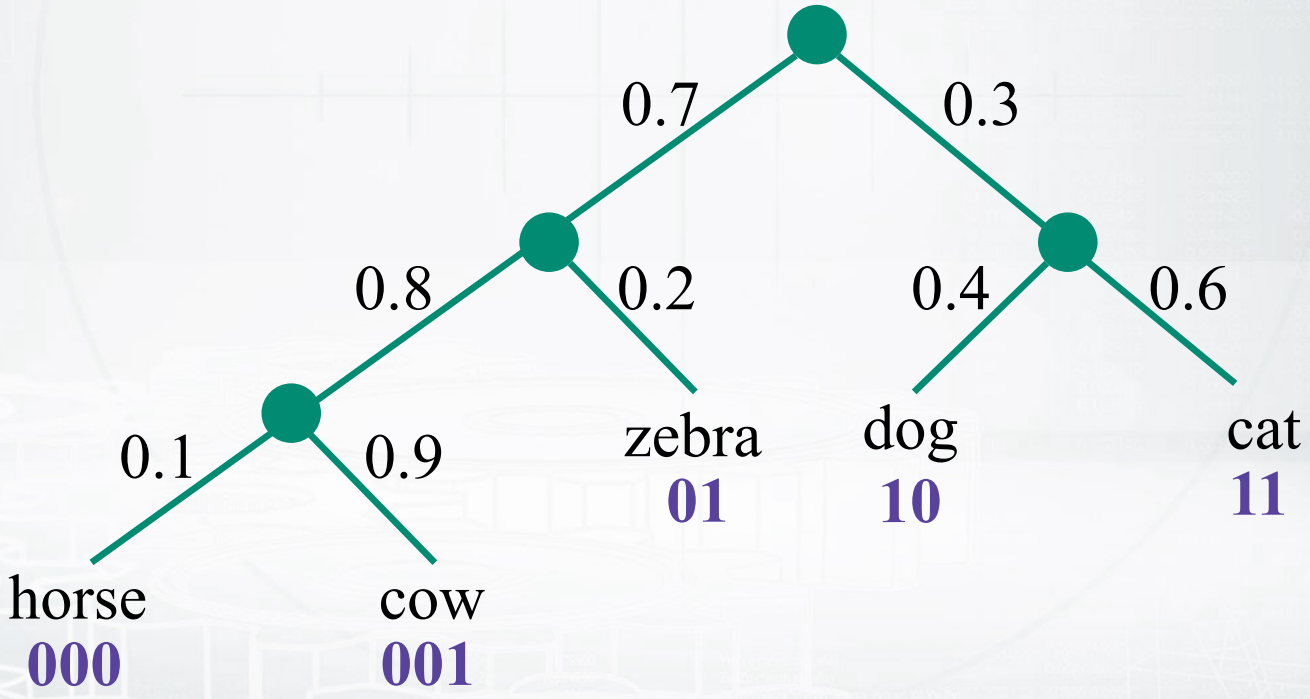
Hierarchical softmax

0.7
0.3
+



Hierarchical softmax

0.7
0.3
+

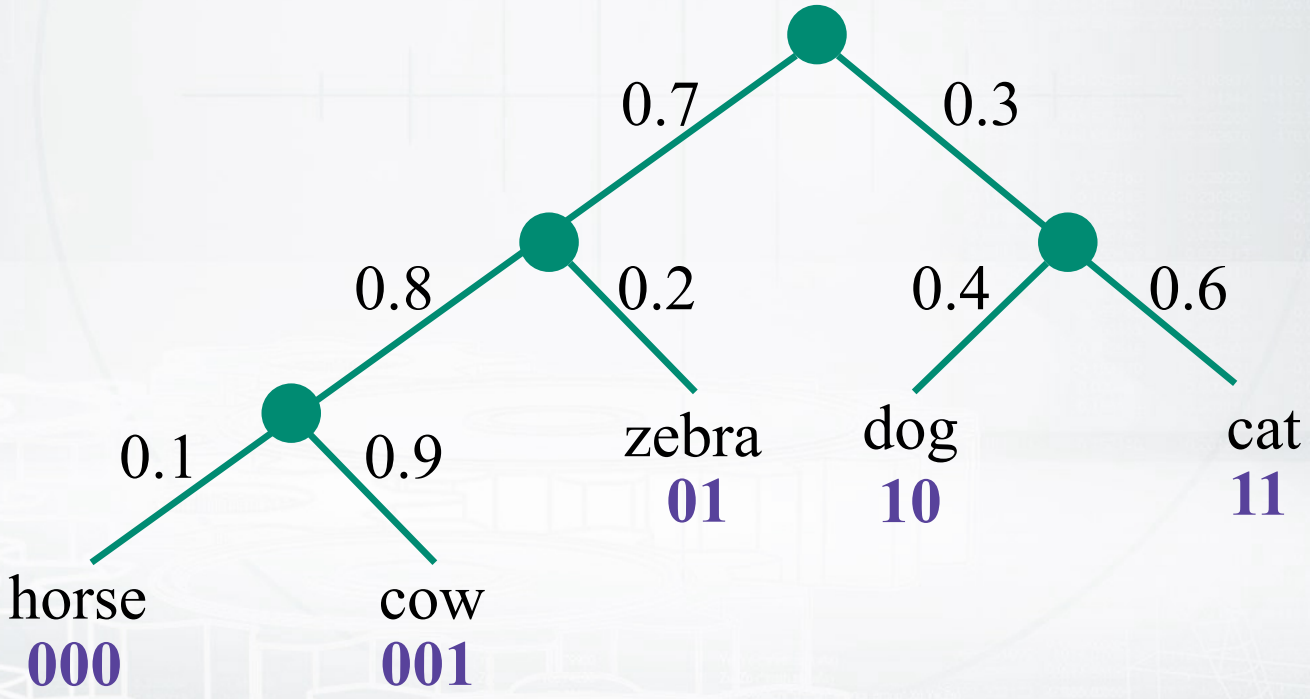


Hierarchical softmax

1.0

+

Congratulations!



Hierarchical softmax

Model binary decisions along the path in the tree:

$$p(w_n = w | w_1^{n-1}) = \prod_i p(d_i | w_1^{n-1})$$

How to construct a tree (balanced vs. semantic):

- Based on some pre-built ontology
- Based on semantic clustering from data
- Huffman tree
- Random

Outline

- Computing *softmax* for a large vocabulary is slow!
  - Hierarchical softmax
- Even a large vocabulary has *OOV words*:
  - **Copy mechanism**
  - Sub-word modeling
    - Word-character hybrid models
    - Byte-pair encoding

Copy mechanism

- Scaling *softmax* is insufficient!
- What do we do with OOV words?
  - Names, numbers, rare words…

	ecotax			*Pont-de-Buis*
The	UNK	portico	in	UNK

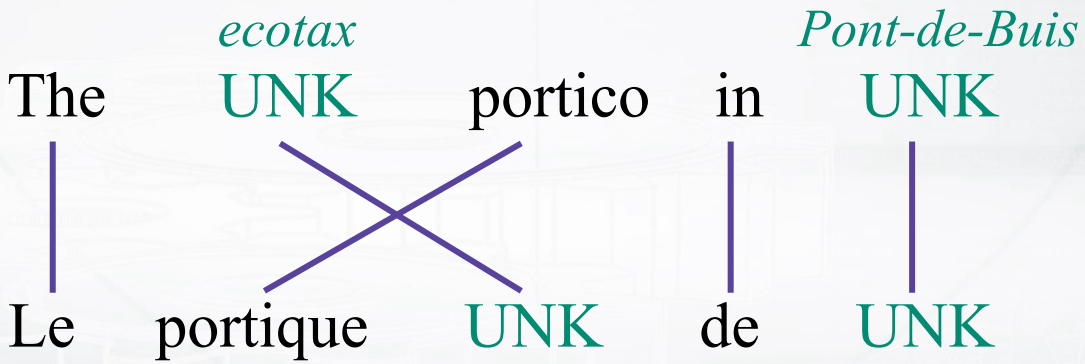
Copy mechanism

- Scaling *softmax* is insufficient!
- What do we do with OOV words?
  - Names, numbers, rare words…

	ecotax			*Pont-de-Buis*
The	UNK	portico	in	UNK
Le	portique	UNK	de	UNK

Copy mechanism

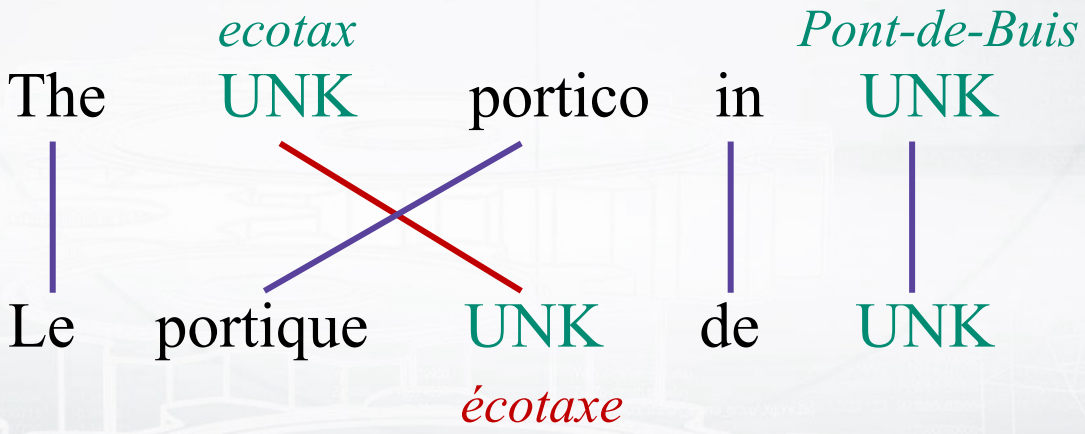
- Scaling *softmax* is insufficient!
- What do we do with OOV words?
  - Names, numbers, rare words…



Copy mechanism

- Scaling *softmax* is insufficient!
- What do we do with OOV words?
  - Names, numbers, rare words…

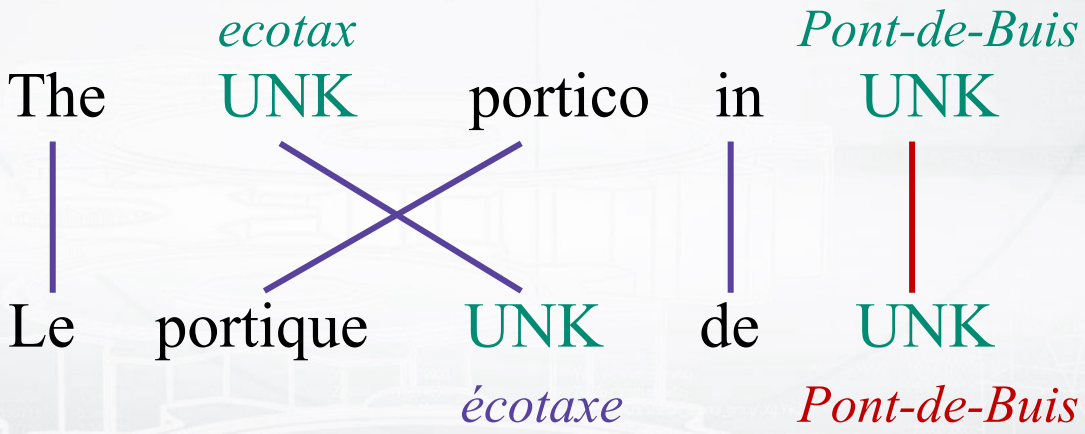
Look-up in a dictionary



Copy mechanism

- Scaling *softmax* is insufficient!
- What do we do with OOV words?
  - Names, numbers, rare words…

Look-up in a dictionary **Copy name**



Copy mechanism

Algorithm:

- Provide word alignments in train time
- Learn relative positions for UNK tokens with NMT
- Post-process the translation:
  - Copy the source word
  - Look up in a dictionary

Simple, but super useful technique!

Towards open vocabulary

Still problems:

- Transliteration: Christopher ↦ Kryštof
- Multi-word alignment: Solar system ↦ Sonnensystem
- Rich morphology: nejneobhospodařovávatelnějšímu
- Informal spelling: goooooood morning !!!!!

Outline

- Computing *softmax* for a large vocabulary is slow!
  - Hierarchical softmax
- Even a large vocabulary has *OOV words*:
  - Copy mechanism
  - **Sub-word modeling**
    - Word-character hybrid models
    - Byte-pair encoding

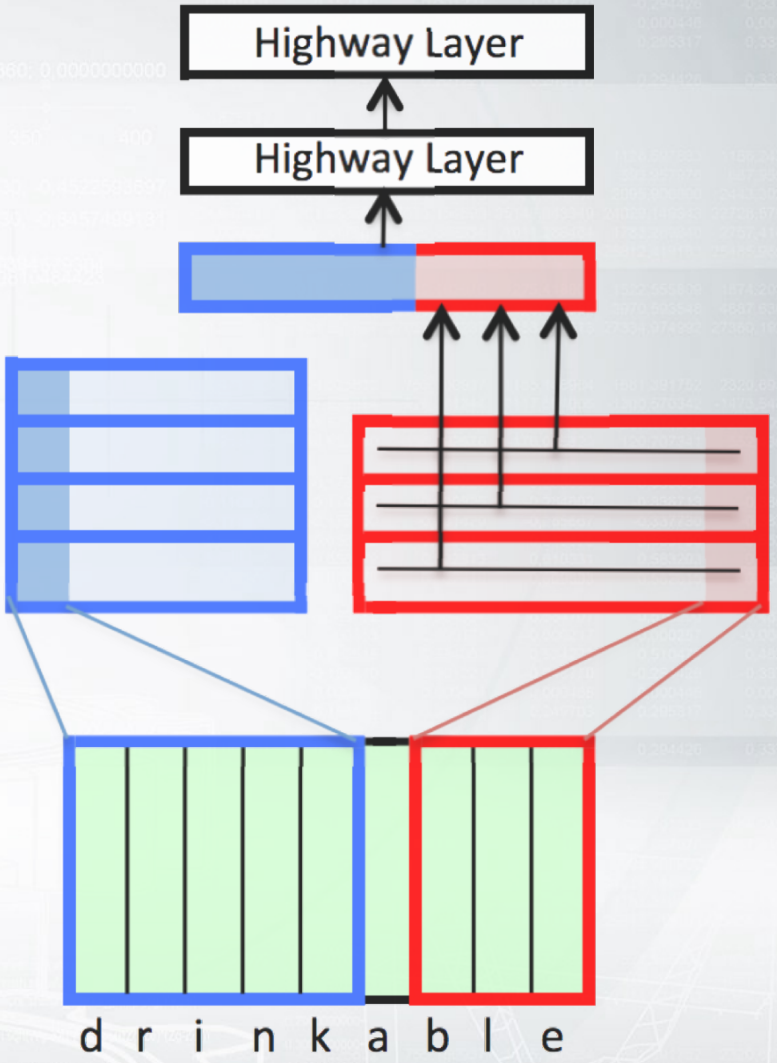
Outline

- Computing *softmax* for a large vocabulary is slow!
  - Hierarchical softmax
- Even a large vocabulary has *OOV words*:
  - Copy mechanism
  - Sub-word modeling
    - **Word-character hybrid models**
    - Byte-pair encoding

Character-based models

Character-based encoder is good for source languages with rich morphology!

- Bi-LSTMs to build word embeddings from characters
- CNNs on characters



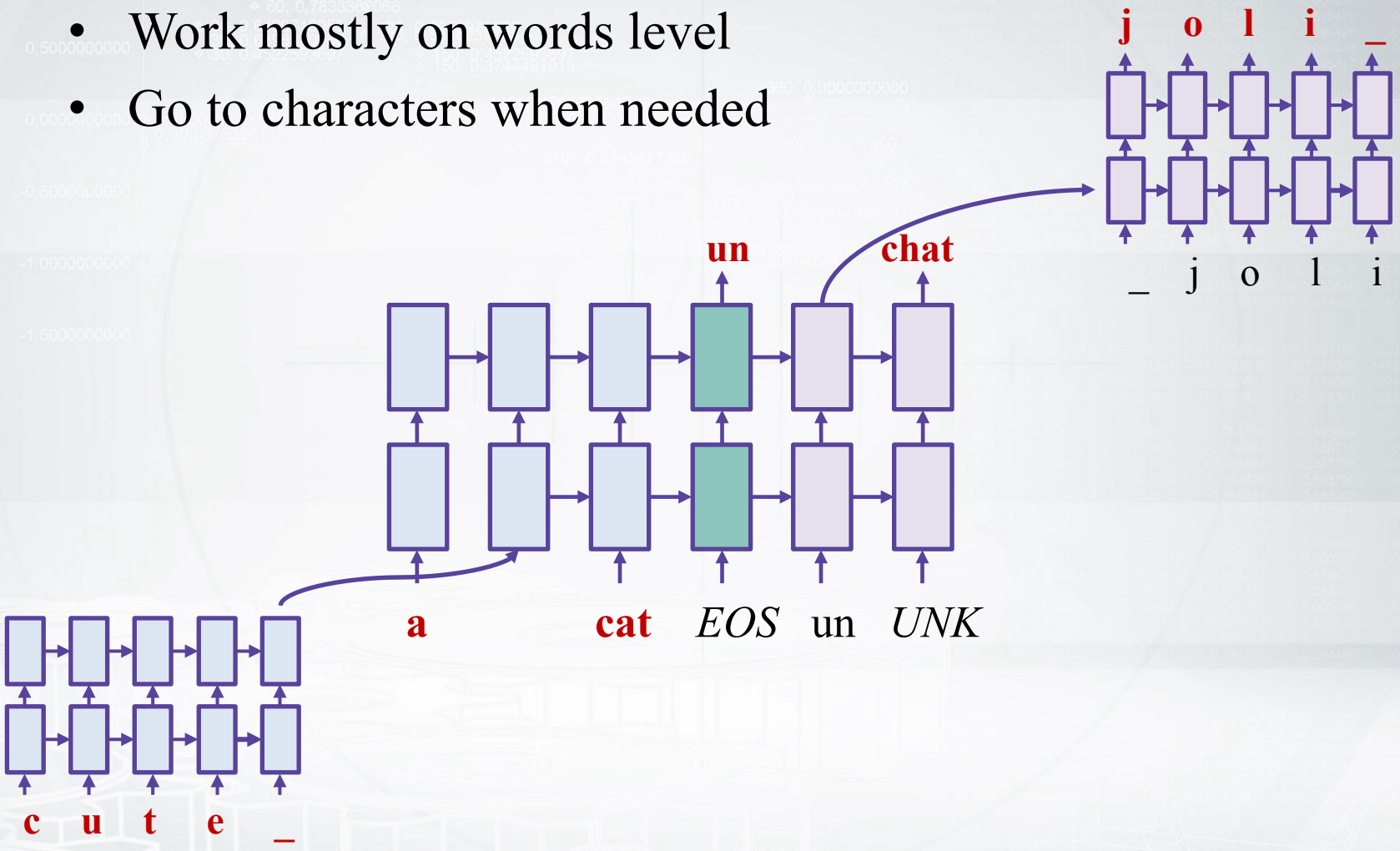
Ling, et. al. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. EMNLP 2015.

Kim, et. al. Character-Aware Neural Language Models. AAAI 2016.

Marta R. Costa-jussà and José A. R. Fonollosa. Character-based Neural Machine Translation. ACL 2016.

Hybrid models: the best of two worlds

- Work mostly on words level
- Go to characters when needed



Thang Luong and Chris Manning. Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models. ACL 2016.

Outline

- Computing *softmax* for a large vocabulary is slow!
  - Hierarchical softmax
- Even a large vocabulary has *OOV words*:
  - Copy mechanism
  - Sub-word modeling
    - Word-character hybrid models
    - **Byte-pair encoding**

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

She sells seashells by the seashore

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

S h e _ s e l l s _ s e a s h e l l s _ b y _ t h e _ s e a s h o r e _

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

S h e _ s e l l s _ s e a s h e l l s _ b y _ t h e _ s e a s h o r e _

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

S h e _ s e l l s _ s e a s h e l l s _ b y _ t h e _ s e a s h o r e _

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

S h e _ s e l l s _ s e a s h e l l s _ b y _ t h e _ s e a s h o r e _

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

S h e _ s e l l s _ s e a s h e l l s _ b y _ t h e _ s e a s h o r e _

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

Sh e _ se l l s _ se a sh e l l s _ b y _ t h e _ se a sh o r e _

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

Sh e _ se ll s _ se a sh e ll s _ b y _ t h e _ se a sh o r e _

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

Sh e _ se ll s _ se a sh e ll s _ b y _ t h e _ se a sh o r e _

Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

Sh e _ se ll s _ sea sh e ll s _ b y _ t h e _ sea sh o r e _

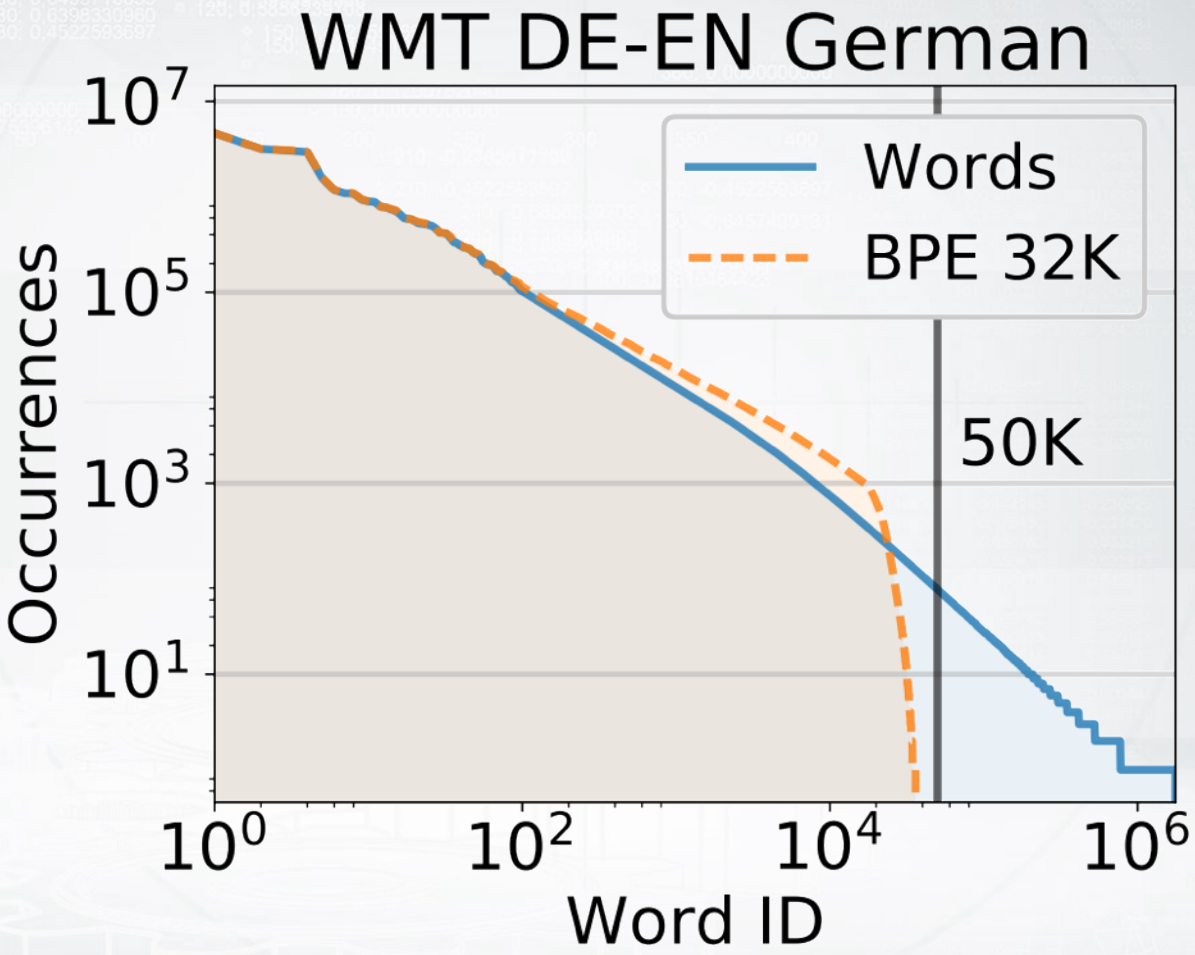
Byte-pair encoding

- Simple way to handle open vocabulary:
  - Start with characters
  - Iteratively replace the most frequent pair with one unit

Sh e _ se ll s _ sea sh e ll s _ b y _ t h e _ sea sh o r e _

- End whenever you reach the vocabulary size limit
- Stick to that vocabulary of sub-word units
- Apply the same algorithm to test sentences

Why is it so useful?



Denkowski, Neubig. Stronger Baselines for Trustable Results in Neural Machine Translation, 2017.

BLEU score comparison

	WMT			IWSLT	
	DE-EN	EN-FI	RO-EN	EN-FR	CS-EN
Words 50K	31.6	12.6	27.1	33.6	21.0
BPE 32K	**33.5**	**14.7**	**27.8**	34.5	22.6
BPE 16K	33.1	**14.7**	**27.8**	**34.8**	**23.0**

- Byte-pair encoding improves BLEU score
- It is a nice and simple way to handle the vocabulary
- Very common trick in modern NMT

Denkowski, Neubig. Stronger Baselines for Trustable Results in Neural Machine Translation, 2017.