Rap Generation with GPT-2

introduction to natural language generation

Alex Honchar | Data Science UA | 2020

About Alex Honchar

Entrepreneur

creating and selling AI solutions worldwide

Practitioner

7 years building data-driven products

Educator

universities, workshops and blogging

Plan for today

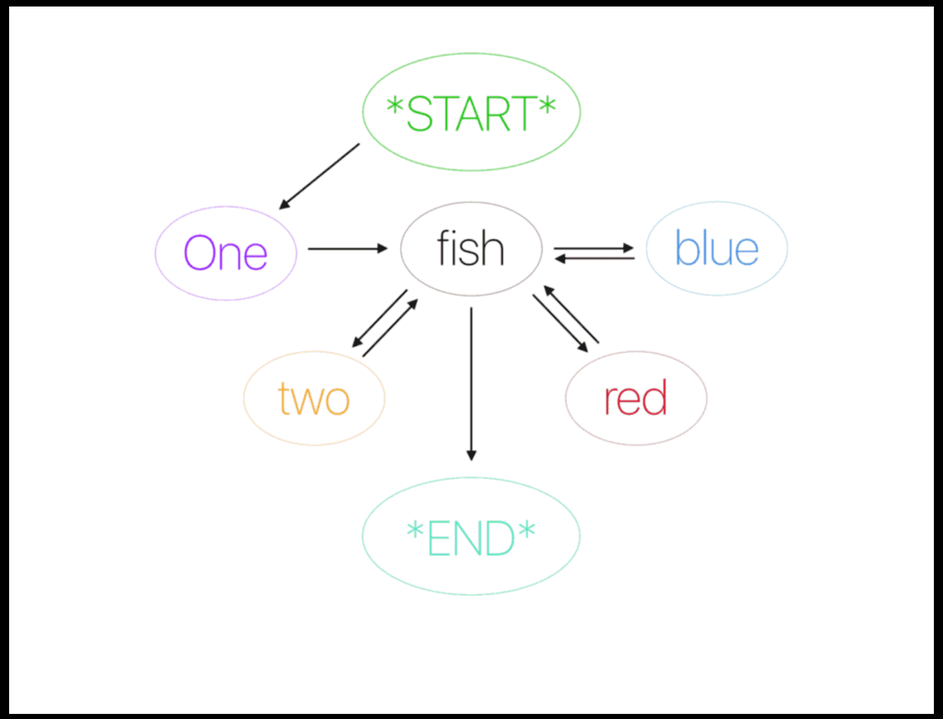
it’s OG Loc, baby!

1. Fundamentals of text generation
2. Transformer architecture
3. Intro to Hugging Face
4. English rap generation
5. Russian rap generation
6. Bonus: jokes generation

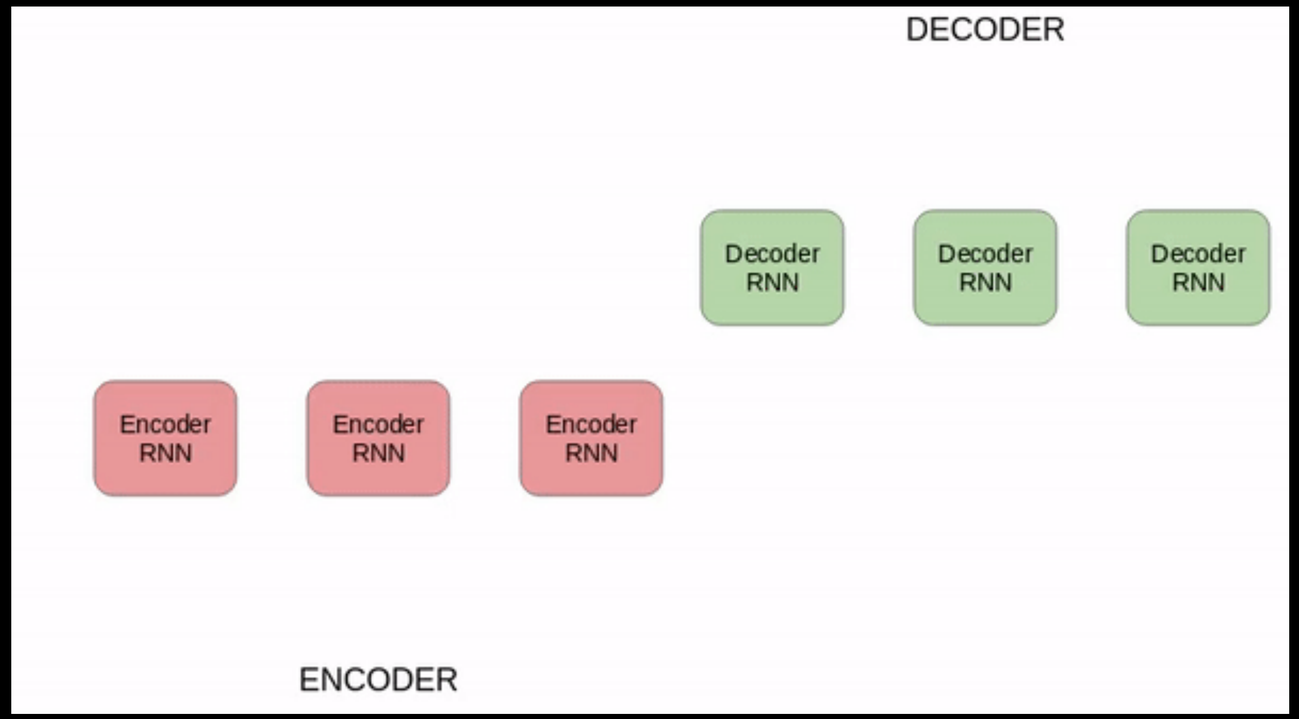
we are going to work with rap generation, take your kids off the screens

Text generation in a nutshell

The era before transformers



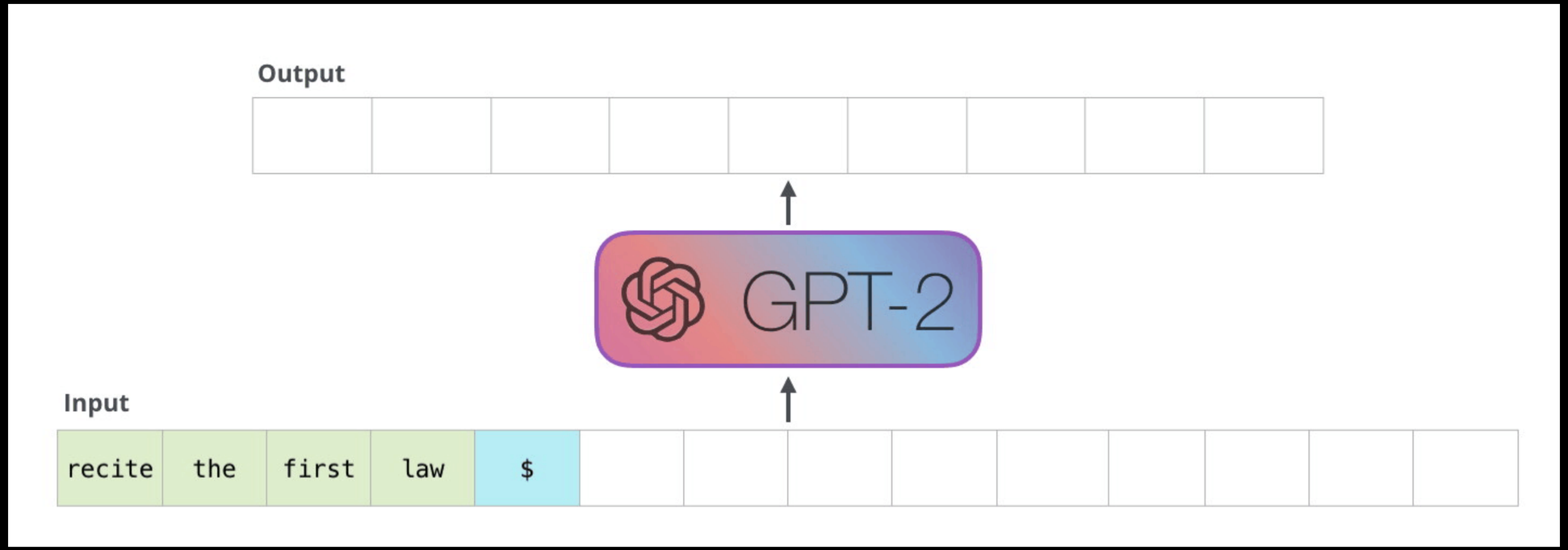
Markov Chains



Sequence2sequence

Transformers in NLP

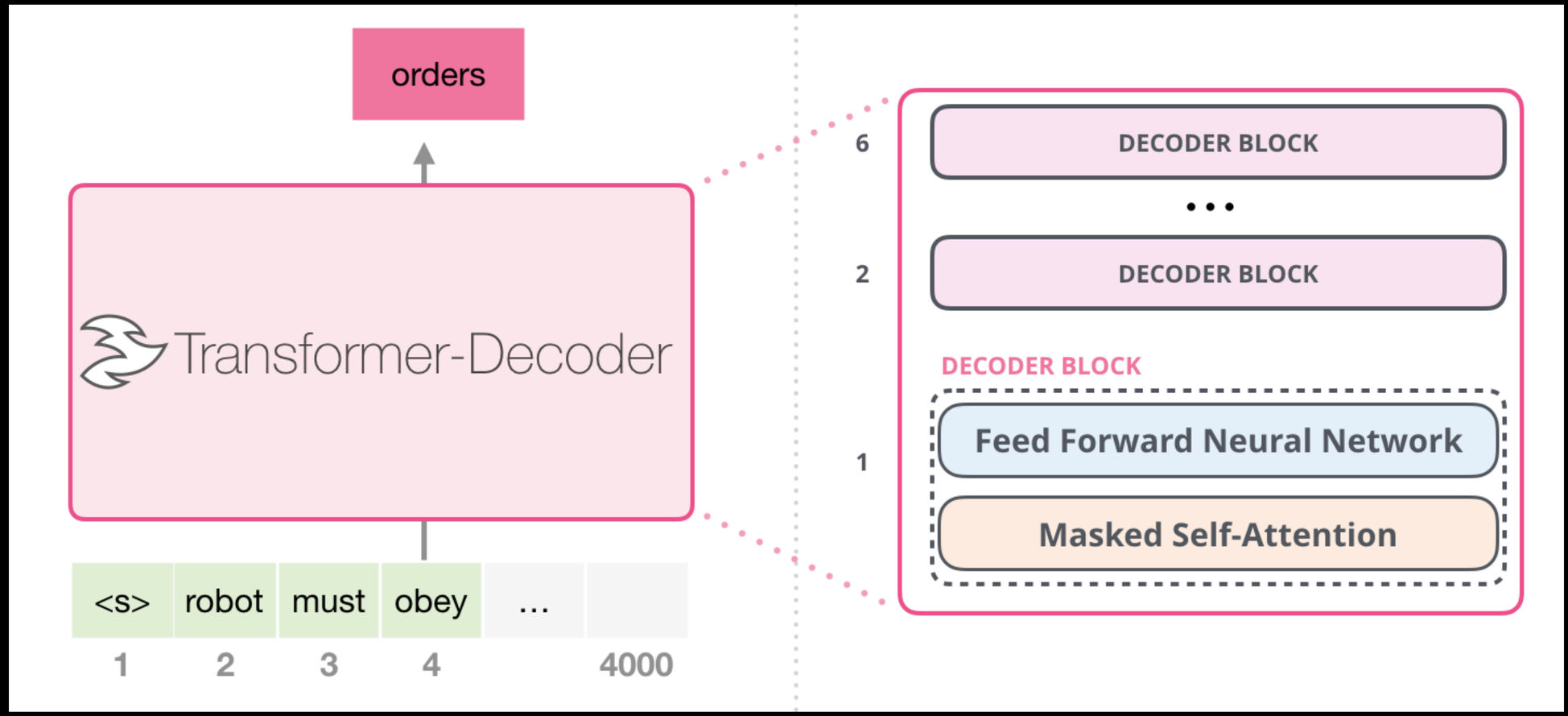
attention is all you need!



GPT-2 autoregressive model

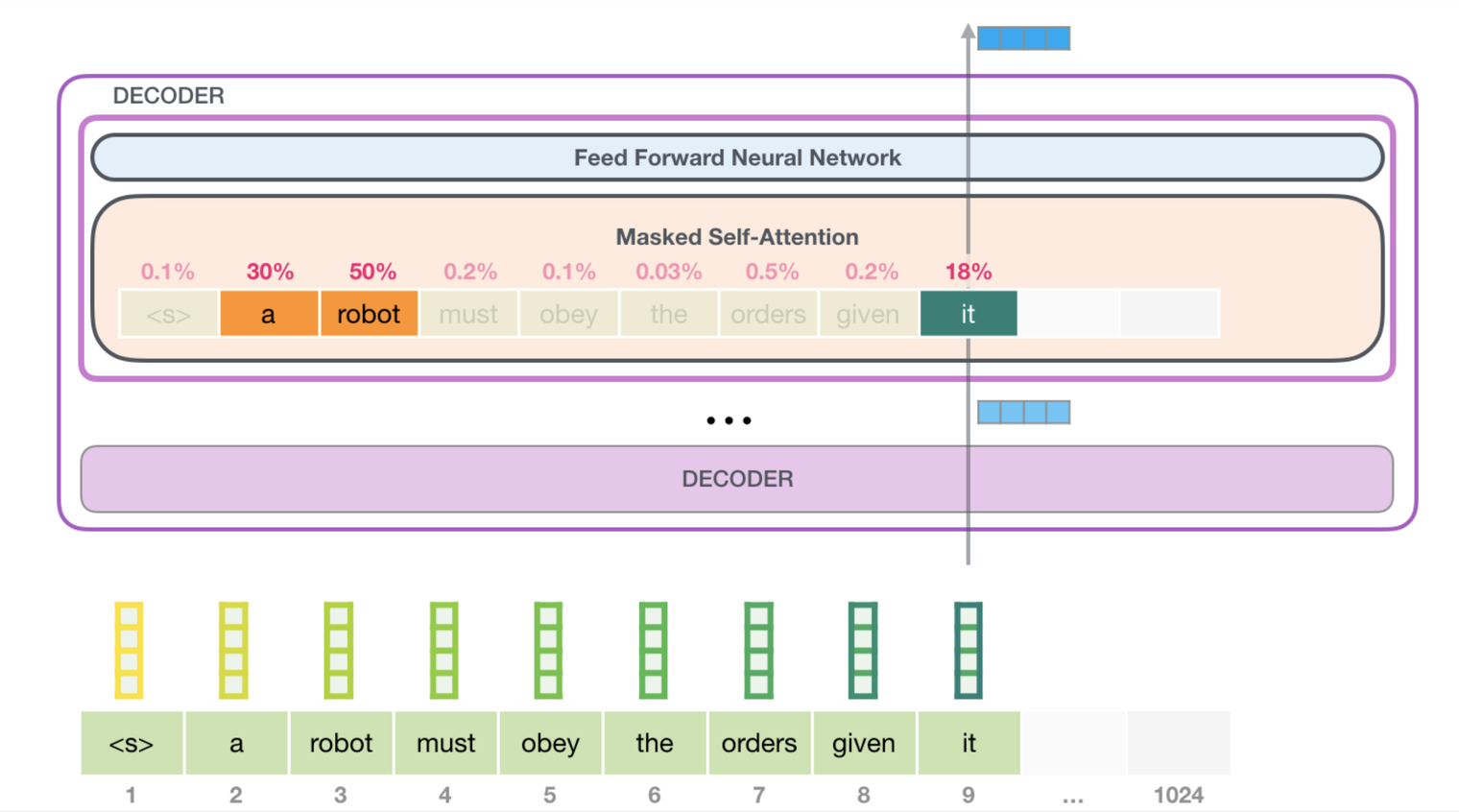
Transformers in NLP

attention is all you need!



Transformers in NLP

attention is all you need!



Write With Transformer

transformer.huggingface.co

https://transformer.huggingface.co/

Intro to Hugging Face 🤗

your Keras for state-of-the-art NLP

```python
import tensorflow as tf
import tensorflow_datasets
from transformers import *

# Load dataset, tokenizer, model from pretrained model/vocabulary
tokenizer = BertTokenizer.from_pretrained('bert-base-cased')
model = TFBertForSequenceClassification.from_pretrained('bert-base-cased')
data = tensorflow_datasets.load('glue/mrpc')

# Prepare dataset for GLUE as a tf.data.Dataset instance
train_dataset = glue_convert_examples_to_features(data['train'], tokenizer, max_length=128, task='mrpc')
valid_dataset = glue_convert_examples_to_features(data['validation'], tokenizer, max_length=128, task='mrpc')
train_dataset = train_dataset.shuffle(100).batch(32).repeat(2)
valid_dataset = valid_dataset.batch(64)
```

Intro to Hugging Face 🤗

your Keras for state-of-the-art NLP

```
# Prepare training: Compile tf.keras model with optimizer, loss and learning rate schedule
optimizer = tf.keras.optimizers.Adam(learning_rate=3e-5, epsilon=1e-08, clipnorm=1.0)
loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
metric = tf.keras.metrics.SparseCategoricalAccuracy('accuracy')
model.compile(optimizer=optimizer, loss=loss, metrics=[metric])

# Train and evaluate using tf.keras.Model.fit()
history = model.fit(train_dataset, epochs=2, steps_per_epoch=115,
                    validation_data=valid_dataset, validation_steps=7)
```

Main training parameters

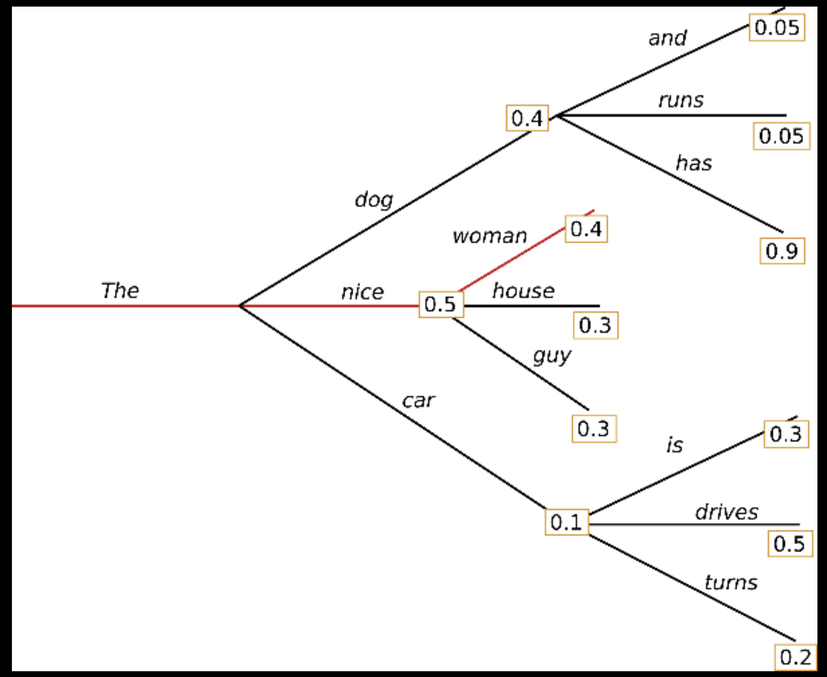
brr-brr go!

- model_type (gpt-2, xlnet, ctrl)
- per_gpu_train_batch_size (usually 1-4)
- learning_rate (small since we fine-tune, 1e-5)
- num_train_epochs (3-5 already can lead to overfitting)

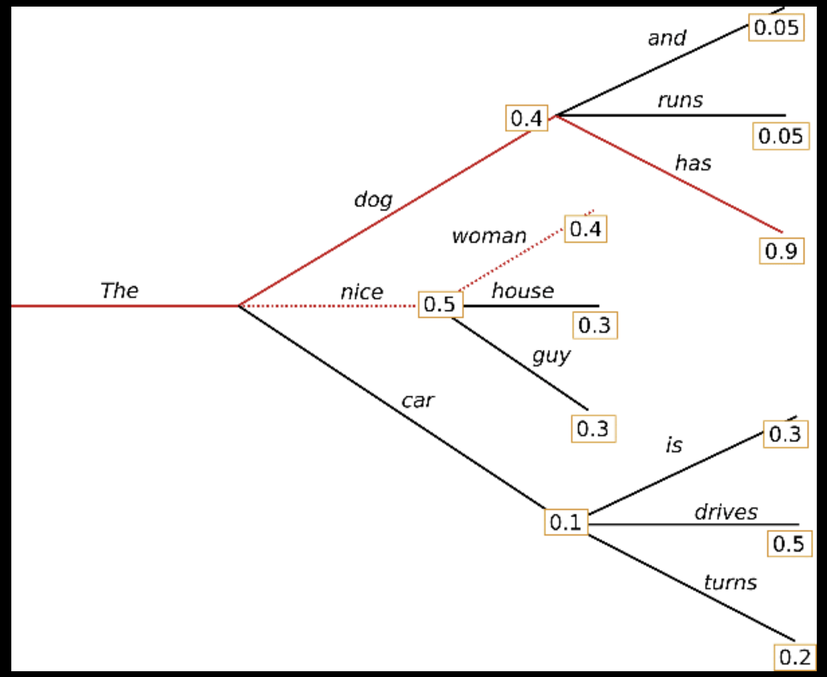
train time

Inference schemes

greedy search vs beam search



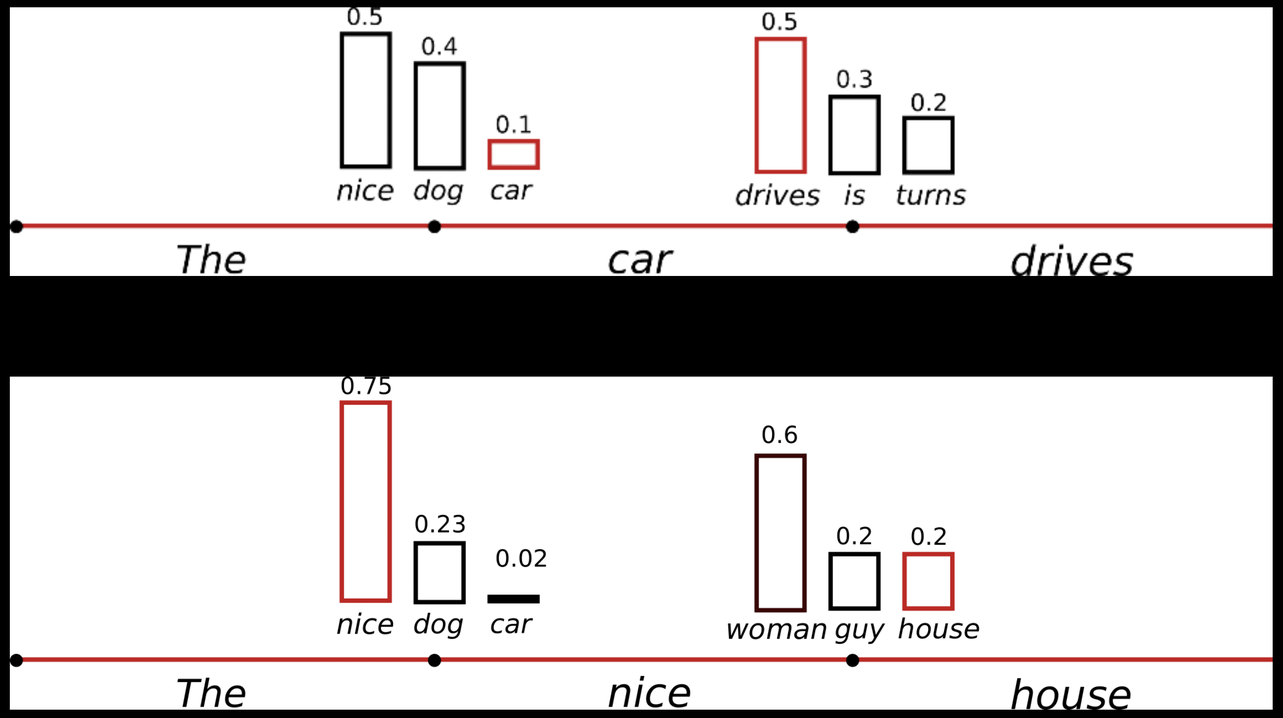
Greedy search



Beam Search

Inference schemes

random sampling



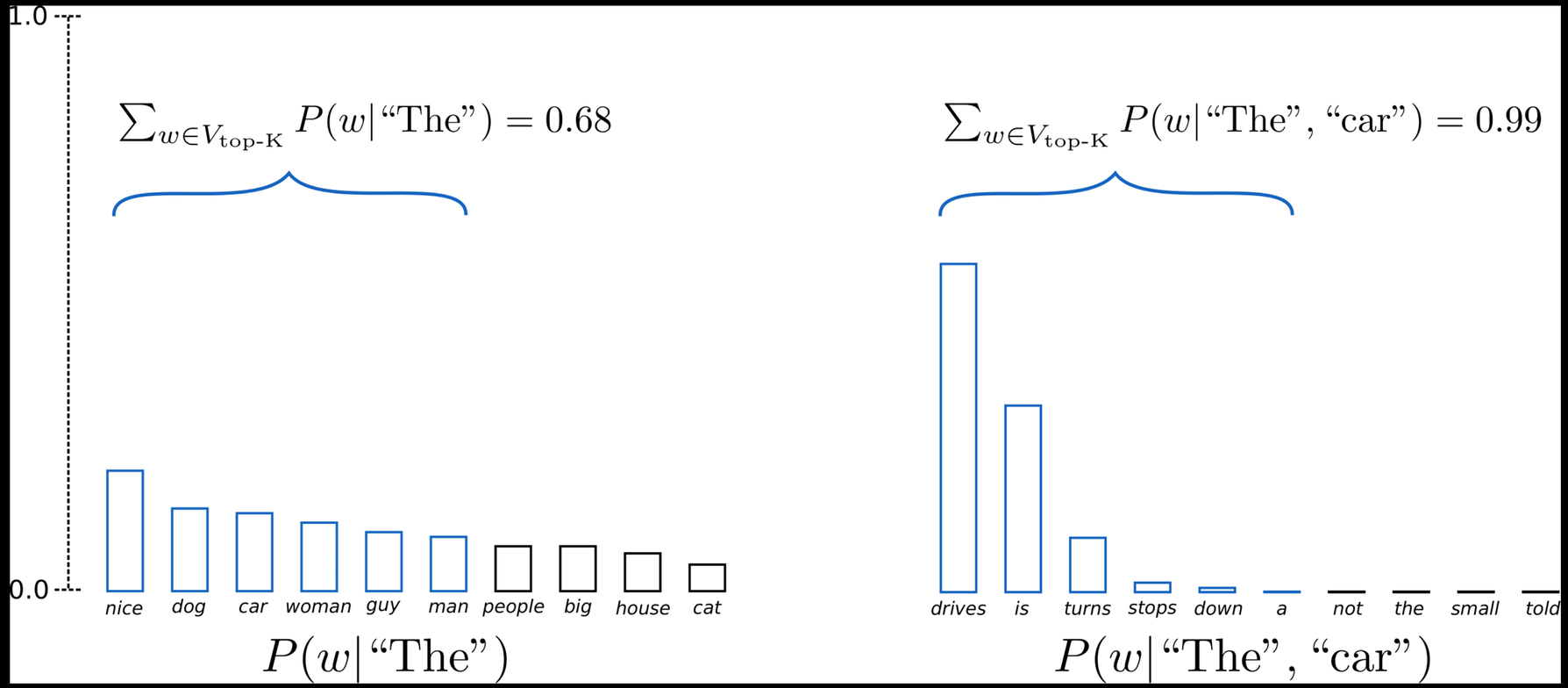
High temperature of softmax

$$w_t \sim P(w|w_{1:t-1})$$

Low temperature of softmax

Inference schemes

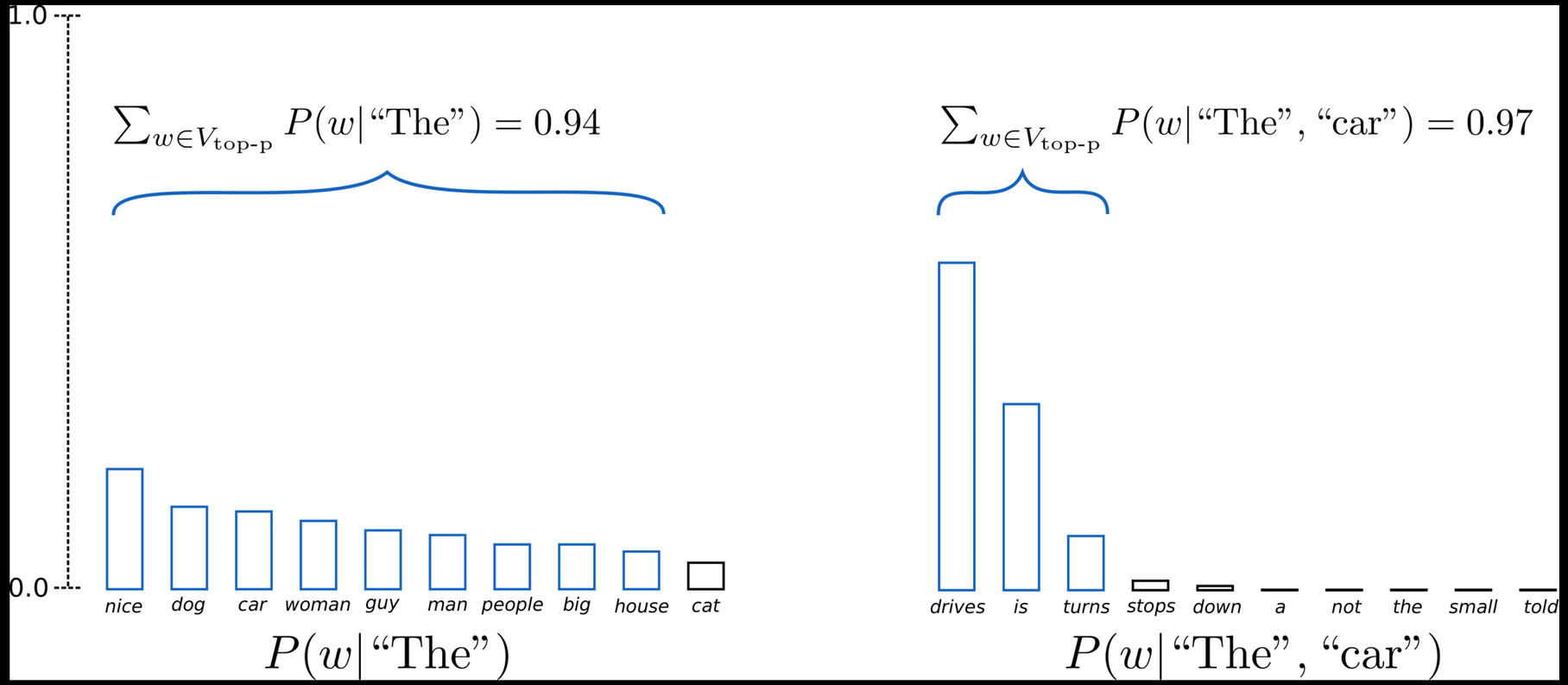
top-k vs top-p sampling?



Top-K sampling scheme

Inference schemes

top-k vs top-p sampling?



Top-P sampling scheme

rap time

Next steps to improve results

rhyme and rhythm

- Get more data, MUCH MORE DATA
- Run deeper models (GPT-2 Extra Large)
- Add prefixes for defining styles and new texts
- Add special tokens to define rhyme words
- Fine-tune on all rap first, then on the target rapper
- Additional quality / creativity losses
- While sampling, force to the rhymes with vocabulary
- Generate texts backwards (?)

Conclusions

if someone asks you what you have learned today

- You understand the idea of autoregressive text modeling
- You know how to run training of GPT-2 on the custom datasets
- You know how to control sampling process from trained GPT-2 model

Resources and references

homework

- Modern NLP theory: https://cs224d.stanford.edu/
- Transformers basics: http://jalammar.github.io/illustrated-gpt2/
- Hugging Face: http://huggingface.co/
- Notes of practitioners:
  - https://www.gwern.net/GPT-2
  - https://spectrum.ieee.org/artificial-intelligence/machine-learning/this-ai-poet-mastered-rhythm-rhyme-and-natural-language-to-write-like-shakespeare

FB: @data.science.ua

FB: @rachnogstyle
MEDIUM: @alexrachnog

GITHUB:
https://github.com/Rachnog/Rap_Generation