LINCS Reading group: Transformer models in Artificial Intelligence for Natural Language Processing

Léo Laugier

*This presentation is greatly inspired by Jay Alammar's blog https://jalammar.github.io/ and almost all pictures and animations not referenced come from there.

November 7, 2019

Contents

Contents

Introduction (1/6): Example of text generation with Open AI GPT-2

Professor Thomas Bonald is not who you think he is. He actually is the last representant of a very ancient race of alien unicorn, coming from another galaxy. He came to Earth to

Introduction (1/6): Example of text generation with Open AI GPT-2

Professor Thomas Bonald is not who you think he is. He actually is the last representant of a very ancient race of alien unicorn, coming from another galaxy. He came to Earth to learn more about the new world he was born into. He is the first alien to ever visit Earth since he's been here many thousands of times and to meet with humans who are not his kind.
Tom (as aliens called his name) is in his early thirty's now after a lifetime of flying around and exploring through space. With the help of his friends, they have found an old ship and they are able to use this to go back to where the spaceship that took him to Earth came from. They have found a new world in space where the humans have been living in for the last few millennia and who are slowly dying off.

Introduction (2/6): Overview of Natural Language Processing tasks addressed by AI / Machine Learning

- **Syntax**: Lemmatization, Part-of-Speech tagging (POS), Parsing, . . .

Introduction (2/6): Overview of Natural Language Processing tasks addressed by AI / Machine Learning

- **Syntax**: Lemmatization, Part-of-Speech tagging (POS), Parsing, ...
- **Semantics**:

Introduction (2/6): Overview of Natural Language Processing tasks addressed by AI / Machine Learning

- **Syntax**: Lemmatization, Part-of-Speech tagging (POS), Parsing, ...
- **Semantics**:
    - Classification: Named Entity Recognition (NER), Semantic Role Labelling (SRL), Information Retrieval (IR) Sentiment Analysis, Extractive Summarization, Question Answering (QA), ...

Introduction (2/6): Overview of Natural Language Processing tasks addressed by AI / Machine Learning

- **Syntax**: Lemmatization, Part-of-Speech tagging (POS), Parsing, . . .
- **Semantics**:
  - Classification: Named Entity Recognition (NER), Semantic Role Labelling (SRL), Information Retrieval (IR) Sentiment Analysis, Extractive Summarization, Question Answering (QA), . . .
  - Natural Language Understanding (NLU): Representation Learning, Word Embedding, . . .

Introduction (2/6): Overview of Natural Language Processing tasks addressed by AI / Machine Learning

- **Syntax**: Lemmatization, Part-of-Speech tagging (POS), Parsing, ...
- **Semantics**:
  - Classification: Named Entity Recognition (NER), Semantic Role Labelling (SRL), Information Retrieval (IR) Sentiment Analysis, Extractive Summarization, Question Answering (QA), ...
  - Natural Language Understanding (NLU): Representation Learning, Word Embedding, ...
  - (Conditional) Natural Language Generation (NLG): Abstractive Summarization, Machine Translation (MT), Style Transfer, Paraphrasing, Dialogue agents (chatbots), Synthetic natural language inference based on language modeling, ...

Introduction (2/6): Overview of Natural Language Processing tasks addressed by AI / Machine Learning

- **Syntax**: Lemmatization, Part-of-Speech tagging (POS), Parsing, ...
- **Semantics**:
  - Classification: Named Entity Recognition (NER), Semantic Role Labelling (SRL), Information Retrieval (IR) Sentiment Analysis, Extractive Summarization, Question Answering (QA), ...
  - Natural Language Understanding (NLU): Representation Learning, Word Embedding, ...
  - (Conditional) Natural Language Generation (NLG): Abstractive Summarization, Machine Translation (MT), Style Transfer, Paraphrasing, Dialogue agents (chatbots), Synthetic natural language inference based on language modeling, ...

Some sources include speech and multimodal learning in NLP for instance: Automatic Speech Recognition (ASR), Text-To-Speech (TTS), Image-to-text (Caption generation, Optical Character Recognition (OCR)).

Introduction (3/6): Several tasks but also various data sources and many languages...

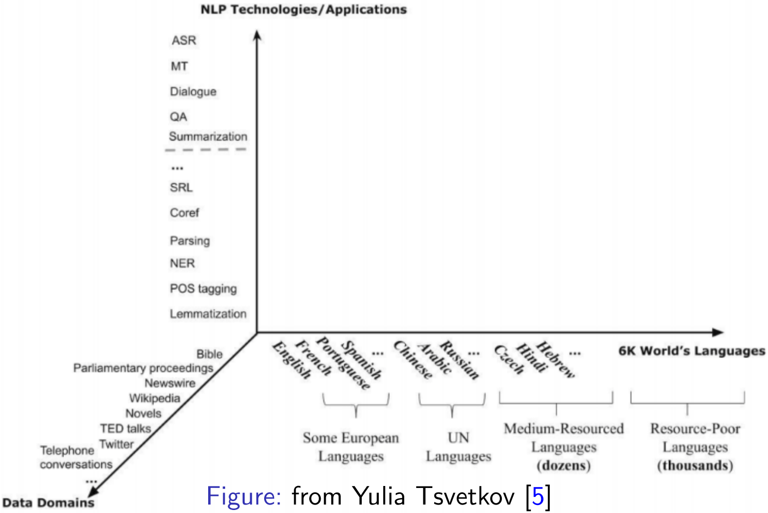


Figure: from Yulia Tsvetkov [5]

Introduction (4/6): Neural networks and deep learning models work well on NLP tasks

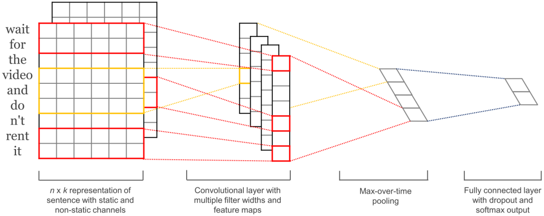


Figure: Convolutional Neural Network (CNN) [6]

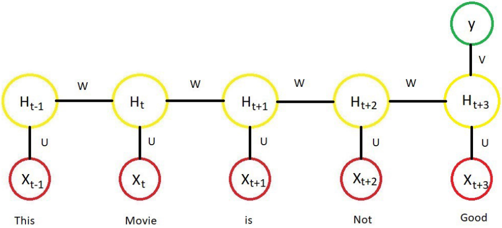


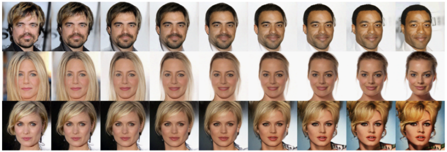
Figure: Recurrent Neural Network (RNN)

Introduction (5/6): Challenges and specificities of AI for processing natural language

1. Natural language is **discrete**: phrases, words, sub-words, letters / phonemes.

Introduction (5/6): Challenges and specificities of AI for processing natural language

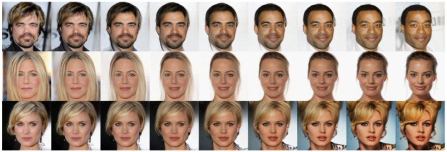
1. Natural language is **discrete**: phrases, words, sub-words, letters / phonemes.



Human vision is continuous!

Introduction (5/6): Challenges and specificities of AI for processing natural language

1. Natural language is **discrete**: phrases, words, sub-words, letters / phonemes.

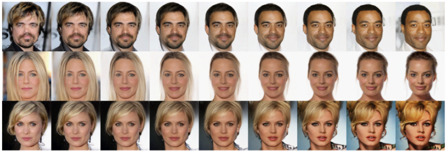


Human vision is continuous!

2. Natural language is **sequential**: context matters!

Introduction (5/6): Challenges and specificities of AI for processing natural language

1. Natural language is **discrete**: phrases, words, sub-words, letters / phonemes.

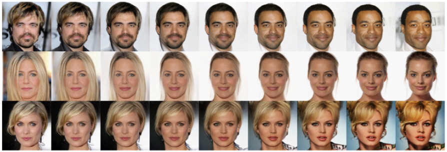


Human vision is continuous!

2. Natural language is **sequential**: context matters!
3. Natural language is **ambiguous** ("Let's eat, Grandma!"): homonyms, paronyms, polysemy, metonymy, metaphors, zeugmas, syllepsis, synonyms, irony, sarcasm, litotes, . . .

Introduction (5/6): Challenges and specificities of AI for processing natural language

1. Natural language is **discrete**: phrases, words, sub-words, letters / phonemes.



Human vision is continuous!

2. Natural language is **sequential**: context matters!
3. Natural language is **ambiguous** ("Let's eat, Grandma!"): homonyms, paronyms, polysemy, metonymy, metaphors, zeugmas, syllepsis, synonyms, irony, sarcasm, litotes, . . .

Images are less often ambiguous than natural language.

Introduction (6/6): Food for thought...

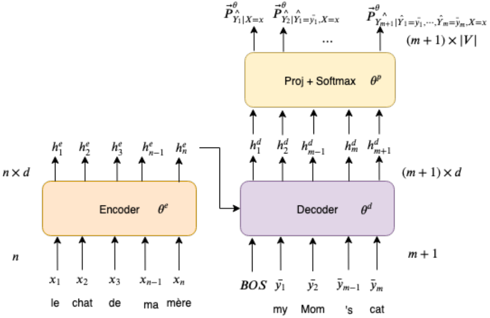
Herbert H. Clark & Michael F. Schober, 1992

The common misconception is that language has to do with **words** and what they mean.
It doesn't.
It has to do with **people** and what they mean.

Contents

Seq2Seq (e.g. MT) (1/3): Model

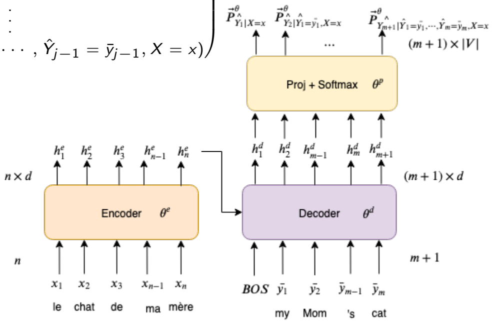
$$\bar{y}_j = \{ \; y_j \text{ if training}$$



Seq2Seq (e.g. MT) (1/3): Model

$$\bar{y}_j = \{ \; y_j \text{ if training}$$

$$\vec{P}^{\theta}_{\hat{Y}_j|\hat{Y}_1=\bar{y}_1,\cdots,\hat{Y}_{j-1}=\bar{y}_{j-1},X=x} = \begin{pmatrix} P^{\theta}(\hat{Y}_j = w_1 | \hat{Y}_1 = \bar{y}_1, \cdots, \hat{Y}_{j-1} = \bar{y}_{j-1}, X = x) \\ P^{\theta}(\hat{Y}_j = w_2 | \hat{Y}_1 = \bar{y}_1, \cdots, \hat{Y}_{j-1} = \bar{y}_{j-1}, X = x) \\ \vdots \\ P^{\theta}(\hat{Y}_j = w_{|V|} | \hat{Y}_1 = \bar{y}_1, \cdots, \hat{Y}_{j-1} = \bar{y}_{j-1}, X = x) \end{pmatrix}$$

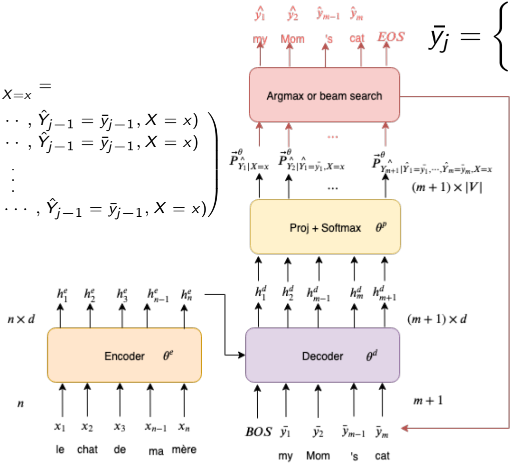


Seq2Seq (e.g. MT) (1/3): Model

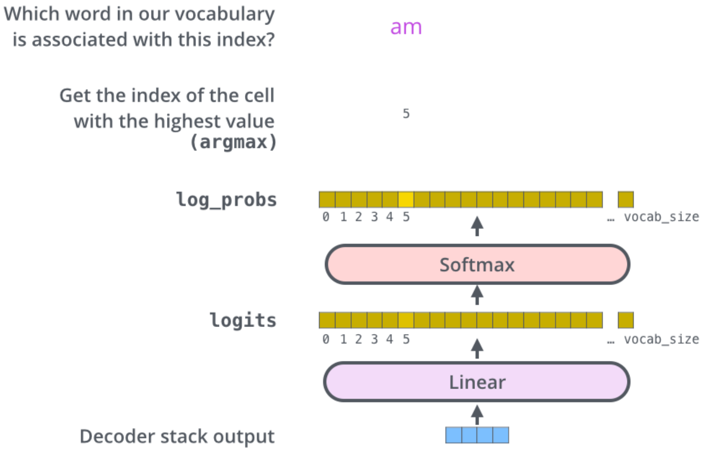
Red: Only at inference time

$$\bar{y}_j = \begin{cases} y_j & \text{if training} \\ \hat{y}_j & \text{if inference} \end{cases}$$

$$\vec{P}^{\theta}_{\hat{Y}_j|\hat{Y}_1=\bar{y}_1,\cdots,\hat{Y}_{j-1}=\bar{y}_{j-1},X=x} = \begin{pmatrix} P^{\theta}(\hat{Y}_j = w_1|\hat{Y}_1 = \bar{y}_1, \cdots, \hat{Y}_{j-1} = \bar{y}_{j-1}, X = x) \\ P^{\theta}(\hat{Y}_j = w_2|\hat{Y}_1 = \bar{y}_1, \cdots, \hat{Y}_{j-1} = \bar{y}_{j-1}, X = x) \\ \vdots \\ P^{\theta}(\hat{Y}_j = w_{|V|}|\hat{Y}_1 = \bar{y}_1, \cdots, \hat{Y}_{j-1} = \bar{y}_{j-1}, X = x) \end{pmatrix}$$



Seq2Seq (e.g. MT) (2/3): The Final Linear and Softmax Layer



Seq2Seq (e.g. MT) (3/3): Training with a maximum likelihood objective (loss)

Maximum likelihood

$\mathbb{P}^{\theta}\left[\hat{Y}=(y_1,\ldots,y_m)|X=(x_1,\ldots,x_n)\right]=\prod_{j=1}^{m}\mathbb{P}^{\theta}\left[\hat{Y}_j=y_j|\hat{Y}_{1..j-1}=(y_1,\ldots,y_{j-1}),X=(x_1,\ldots,x_n)\right]$

$L(\theta)=\prod_{(x_i,y_i)}\mathbb{P}^{\theta}\left[\hat{Y}=y^i|X=x^i\right]=\prod_{i=1}^{p}\prod_{j=1}^{m_i}\mathbb{P}^{\theta}\left[\hat{Y}_j=y_j^i|\hat{Y}_{1..j-1}=(y_1^i,\ldots,y_{j-1}^i),X=x^i\right]$

$\hat{\theta}=\arg\max_{\theta}L(\theta)=\arg\min_{\theta}-\log L(\theta)=\arg\min_{\theta}-\sum_{i=1}^{p}\sum_{j=1}^{m_i}\mathbb{P}^{\theta}\left[\hat{Y}_j=y_j^i|\hat{Y}_{1..j-1}=(y_1^i,\ldots,y_{j-1}^i),X=x^i\right]$

Seq2Seq (e.g. MT) (3/3): Training with a maximum likelihood objective (loss)

Maximum likelihood

$$\mathbb{P}^{\theta}\left[\hat{Y} = (y_1, \ldots, y_m) | X = (x_1, \ldots, x_n)\right] = \prod_{j=1}^{m} \mathbb{P}^{\theta}\left[\hat{Y}_j = y_j | \hat{Y}_{1..j-1} = (y_1, \ldots, y_{j-1}), X = (x_1, \ldots, x_n)\right]$$

$$L(\theta) = \prod_{(x_i, y_i)} \mathbb{P}^{\theta}\left[\hat{Y} = y^i | X = x^i\right] = \prod_{i=1}^{p} \prod_{j=1}^{m_i} \mathbb{P}^{\theta}\left[\hat{Y}_j = y_j^i | \hat{Y}_{1..j-1} = (y_1^i, \ldots, y_{j-1}^i), X = x^i\right]$$

$$\hat{\theta} = \arg\max_{\theta} L(\theta) = \arg\min_{\theta} - \log L(\theta) = \arg\min_{\theta} - \sum_{i=1}^{p} \sum_{j=1}^{m_i} \mathbb{P}^{\theta}\left[\hat{Y}_j = y_j^i | \hat{Y}_{1..j-1} = (y_1^i, \ldots, y_{j-1}^i), X = x^i\right]$$

Loss = negative log-likelihood

$$\mathcal{L} = \ell = -\log L$$

Seq2Seq (e.g. MT) (3/3): Training with a maximum likelihood objective (loss)

Maximum likelihood

$\mathbb{P}^\theta\left[\hat{Y} = (y_1, \dots, y_m) | X = (x_1, \dots, x_n)\right] = \prod_{j=1}^m \mathbb{P}^\theta\left[\hat{Y}_j = y_j | \hat{Y}_{1..j-1} = (y_1, \dots, y_{j-1}), X = (x_1, \dots, x_n)\right]$

$L(\theta) = \prod_{(x_i, y_i)} \mathbb{P}^\theta\left[\hat{Y} = y^i | X = x^i\right] = \prod_{i=1}^p \prod_{j=1}^{m_i} \mathbb{P}^\theta\left[\hat{Y}_j = y_j^i | \hat{Y}_{1..j-1} = (y_1^i, \dots, y_{j-1}^i), X = x^i\right]$

$\hat{\theta} = \arg\max_\theta L(\theta) = \arg\min_\theta - \log L(\theta) = \arg\min_\theta - \sum_{i=1}^p \sum_{j=1}^{m_i} \mathbb{P}^\theta\left[\hat{Y}_j = y_j^i | \hat{Y}_{1..j-1} = (y_1^i, \dots, y_{j-1}^i), X = x^i\right]$

Loss = negative log-likelihood

$\mathcal{L} = \ell = -\log L$

Cross-entropy (equivalent)

$$p : \mathcal{X} \longrightarrow [0,1] \qquad x^i \longmapsto \mathbb{P}\left[Y = y^i | X = x^i\right] = 1$$

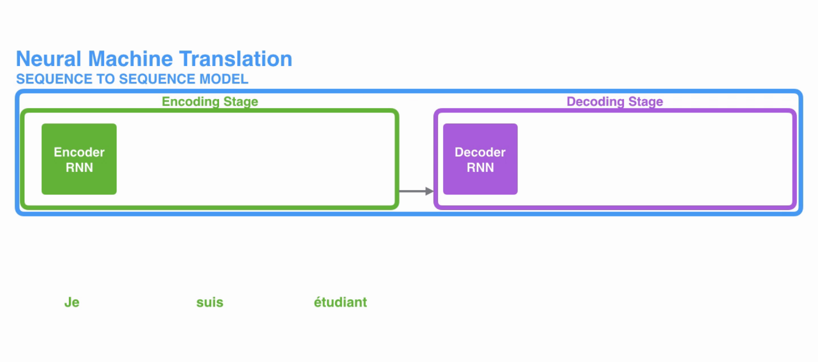
$$q^\theta : \mathcal{X} \longrightarrow [0,1] \qquad x^i \longmapsto \mathbb{P}^\theta\left[\hat{Y} = y^i | X = x^i\right]$$

$H(p, q^\theta) = \mathbb{E}_p[-\log(q)] = -\sum_{x^i \in \mathcal{X}} \mathbb{P}\left[Y = y^i | X = x^i\right] \log \mathbb{P}^\theta\left[\hat{Y} = y^i | X = x^i\right]$

$\hat{\theta} = \arg\min_\theta H(p, q^\theta)$

Contents

Attention is All You Need (1/9): The fall of RNN



Before transformers: Hidden states, RNNs, Long Short-Term Memory (LSTM), Gated Recurrent Units (GRUs).

Issues:

- Not parallelizable
- Long-term information has to sequentially travel through all cells but vanishing gradients problems.

Attention is all you need (2/9): Attention Model Motivation

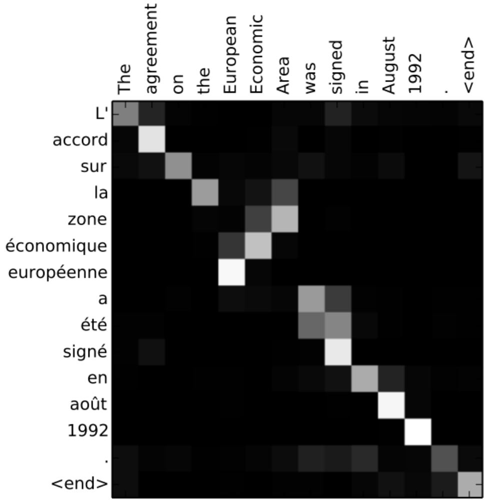
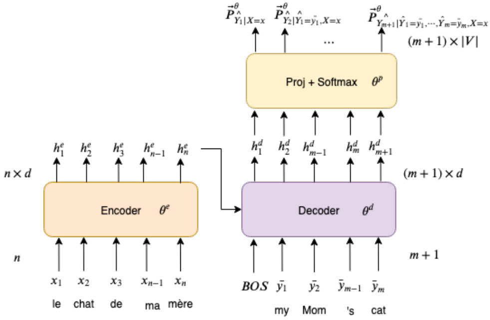
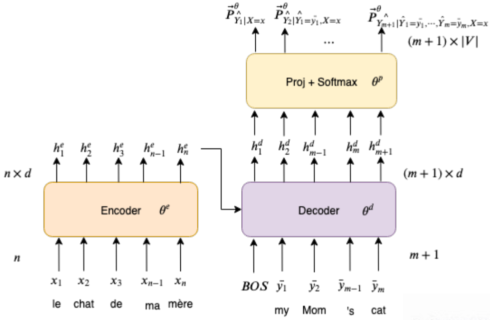


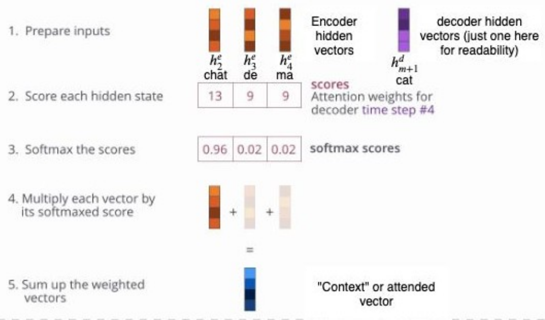
Figure: from Bahdanau *et al.* [7] (2015)

Attention is all you need (3/9): Attention Model Intuition

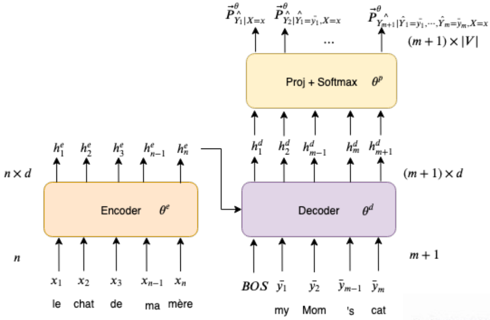


Attention is all you need (3/9): Attention Model Intuition

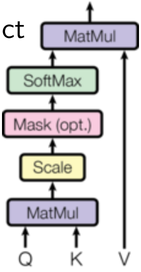


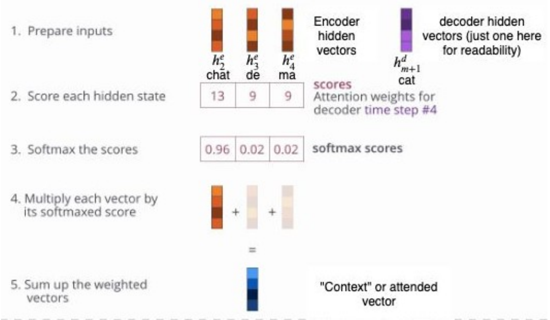


Attention is all you need (3/9): Attention Model Intuition



Scaled dot-product attention





Attention is all you need (4/9): Before making robots ruled by Asimov's Laws, let's make robots understand the laws with **self-attention**

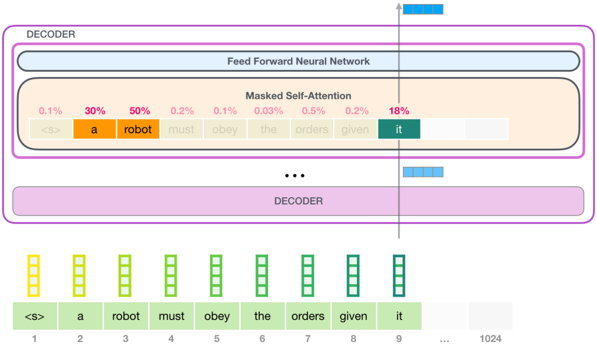
Second Law of Robotics

A robot must obey the orders given *it* by human beings except where *such orders* would conflict with *the First Law*.

Attention is all you need (4/9): Before making robots ruled by Asimov's Laws, let's make robots understand the laws with **self-attention**

Second Law of Robotics

A robot must obey the orders given *it* by human beings except where *such orders* would conflict with *the First Law*.



Attention is all you need (4/9): Before making robots ruled by Asimov's Laws, let's make robots understand the laws with **self-attention**

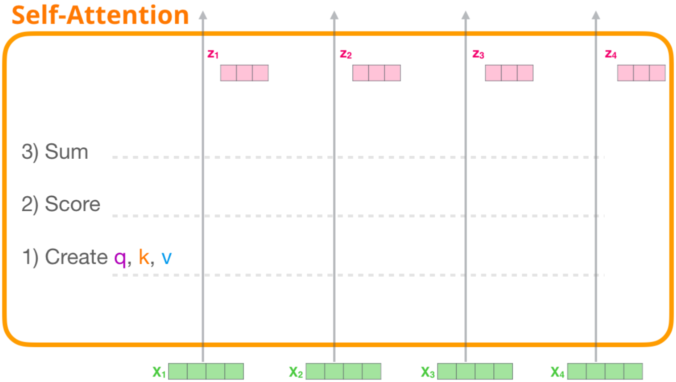
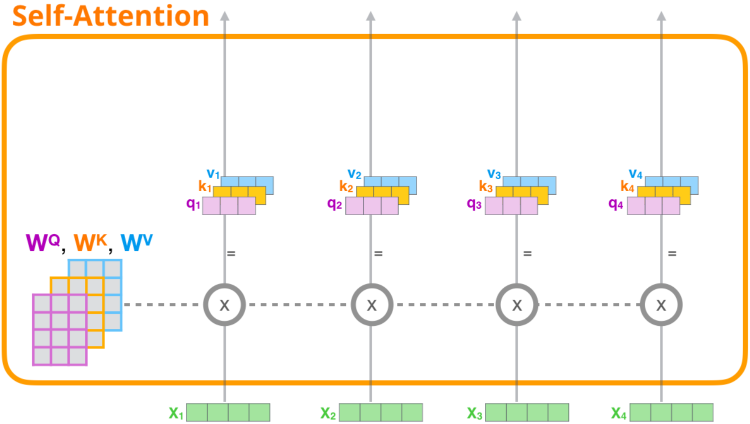


Figure: Self-attention computation principles

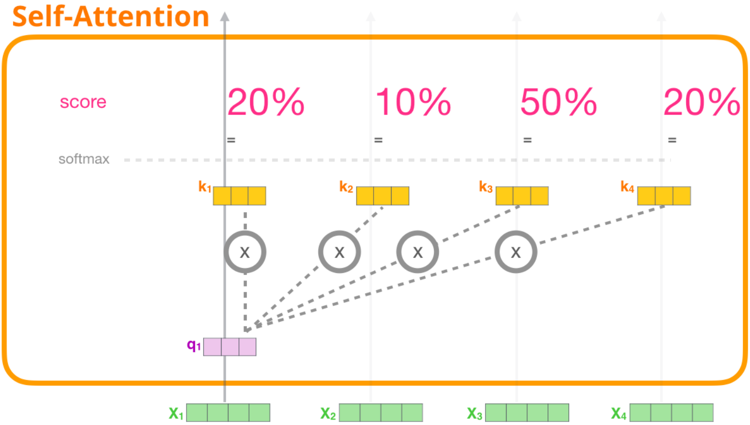
Attention is all you need (4/9): Before making robots ruled by Asimov's Laws, let's make robots understand the laws with **self-attention**

1) For each input token, create a query vector, a key vector, and a value vector by multiplying by weight Matrices W^Q, W^K, W^V



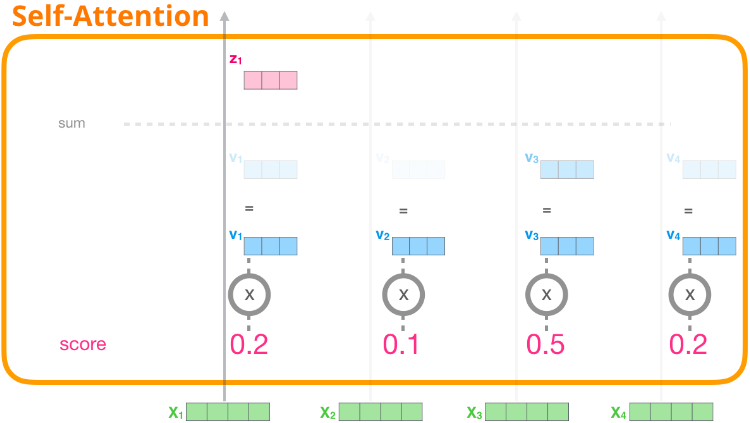
Attention is all you need (4/9): Before making robots ruled by Asimov's Laws, let's make robots understand the laws with **self-attention**

2) Multiply (dot product) the current query vector, by all the key vectors, to get a score of how well they match



Attention is all you need (4/9): Before making robots ruled by Asimov's Laws, let's make robots understand the laws with **self-attention**

3) Multiply the value vectors by the scores, then sum up



Attention is all you need (5/9): Masked Self-Attention to enforce the decoder to attend "past" words

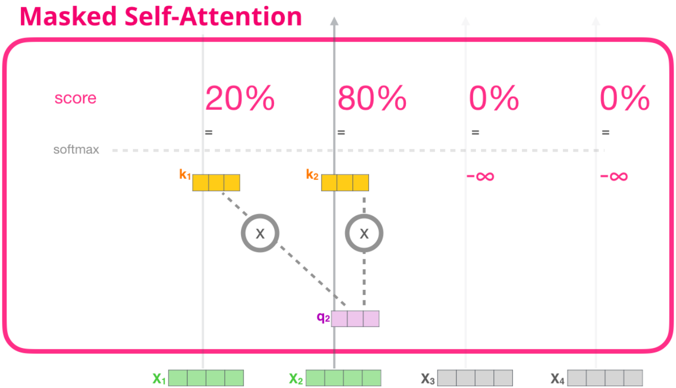


Figure: At inference time, only words previously generated will be available to predict the next word

Attention is all you need (5/9): Masked Self-Attention to enforce the decoder to attend “past” words

	Features				**Labels**
Example: / **position:**	1	2	3	4	
1	robot	must	obey	orders	must
2	robot	must	obey	orders	obey
3	robot	must	obey	orders	orders
4	robot	must	obey	orders	<eos>

Attention is all you need (5/9): Masked Self-Attention to enforce the decoder to attend "past" words

Queries

robot	must	obey	orders

X

Keys

robot	must	obey	orders
robot	must	obey	orders
robot	must	obey	orders
robot	must	obey	orders

=

Scores (before softmax)

0.11	0.00	0.81	0.79
0.19	0.50	0.30	0.48
0.53	0.98	0.95	0.14
0.81	0.86	0.38	0.90

Attention is all you need (5/9): Masked Self-Attention to enforce the decoder to attend "past" words

Scores
(before softmax)

0.11	0.00	0.81	0.79
0.19	0.50	0.30	0.48
0.53	0.98	0.95	0.14
0.81	0.86	0.38	0.90

Apply Attention Mask →

Masked Scores
(before softmax)

0.11	-inf	-inf	-inf
0.19	0.50	-inf	-inf
0.53	0.98	0.95	-inf
0.81	0.86	0.38	0.90

Attention is all you need (5/9): Masked Self-Attention to enforce the decoder to attend "past" words

Masked Scores
(before softmax)

0.11	-inf	-inf	-inf
0.19	0.50	-inf	-inf
0.53	0.98	0.95	-inf
0.81	0.86	0.38	0.90

Softmax
(along rows) →

Scores

1	0	0	0
0.48	0.52	0	0
0.31	0.35	0.34	0
0.25	0.26	0.23	0.26

Attention is all you need (6/9): Building blocks of the transformer architecture

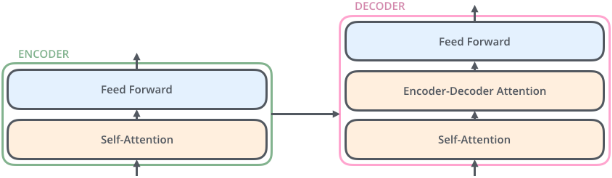


Figure: Encoder-Decoder architecture: each module is made of an ((masked) Self / "Cross") attention block and a Feed Forward Neural Network

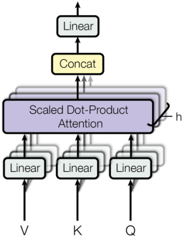
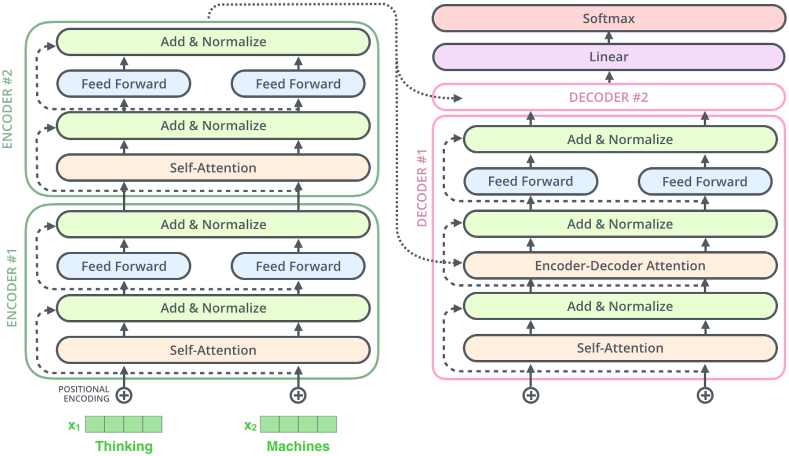
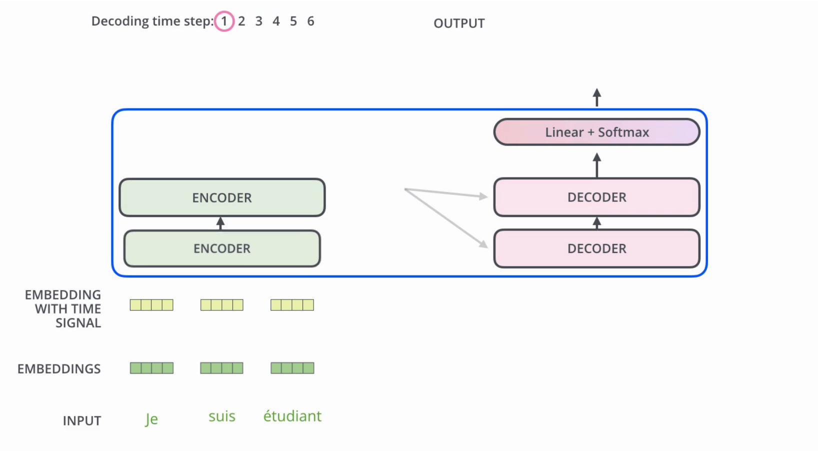


Figure: Actually it is not one attention block but a Multi-Head Attention block used for learning different text features. From Vaswani *et al.* [2]

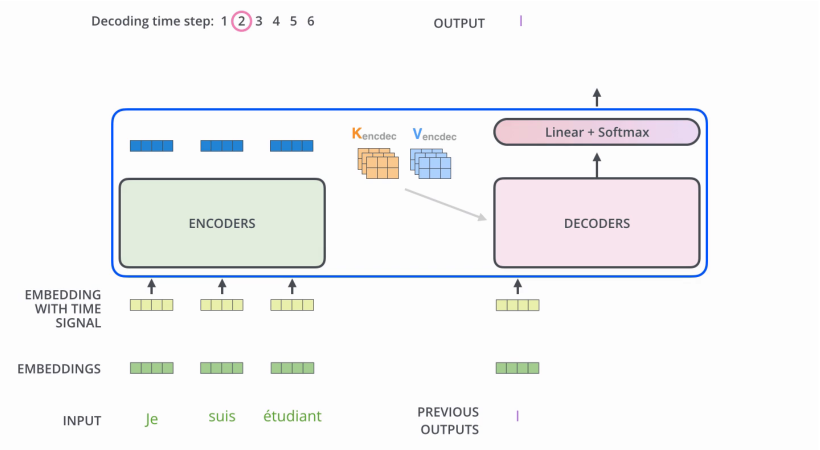
Attention is all you need (7/9): The full transformer architecture is deep *i.e.* made of N layers of blocks



Attention is all you need (8/9): Inference time - where the Natural Language Generation happens



Attention is all you need (8/9): Inference time - where the Natural Language Generation happens



Attention is all you need (9/9): Inference time - where the Natural Language Generation happens

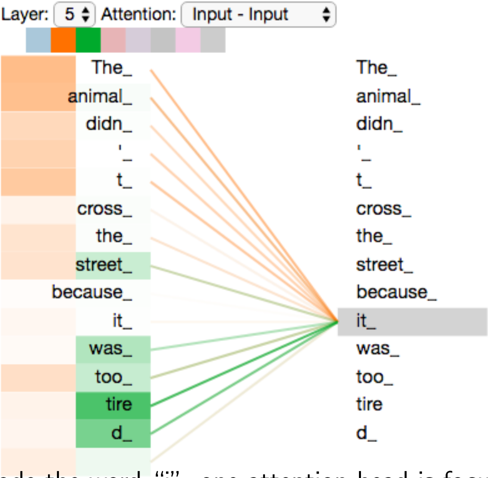


Figure: As we encode the word "i", one attention head is focusing most on "the animal", while another is focusing on "tired".

Contents

BERT (1/6): How NLP Cracked Transfer Learning

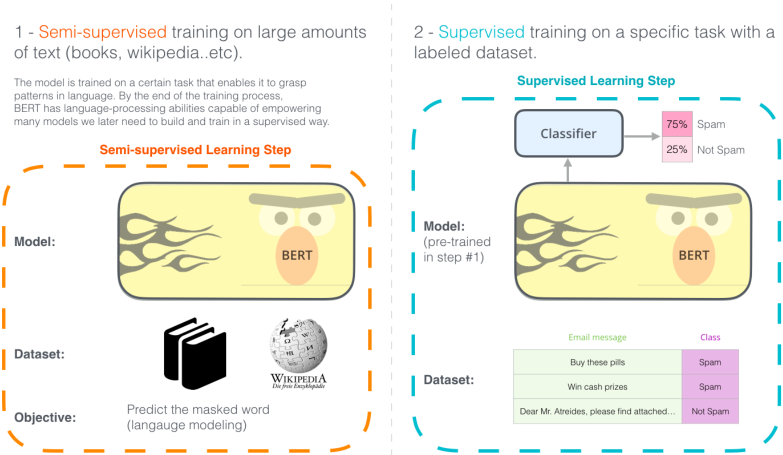


Figure: The two steps of how BERT is developed. You can download the model pre-trained in step 1 (trained on un-annotated data, namely BooksCorpus (800M words) and English Wikipedia (2.5G words)), and only worry about fine-tuning it for step 2.

BERT (1/6): How NLP Cracked Transfer Learning

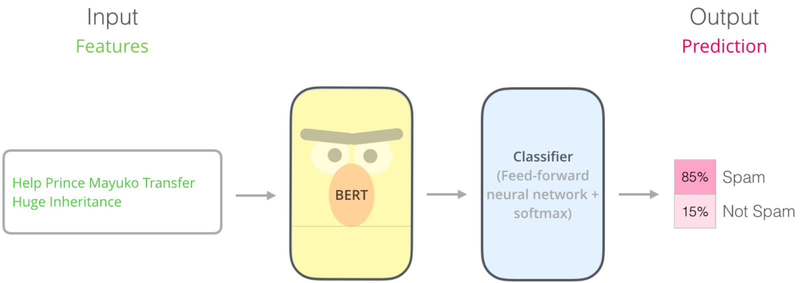


Figure: Fine-tuning BERT with a basic feed-forward neural network ("*NLP's ImageNet moment*")

BERT (1/6): How NLP Cracked Transfer Learning

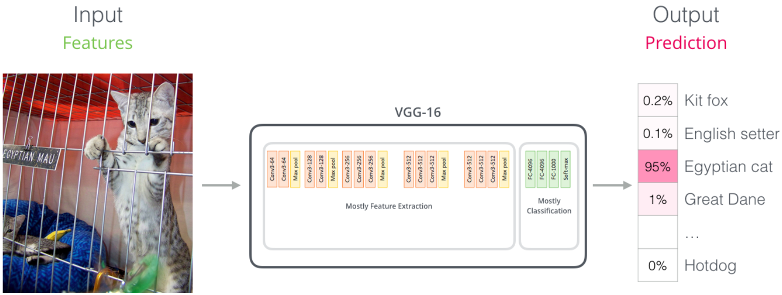


Figure: Parallels with Convolutional Neural Networks

BERT (2/6): Bidirectional Encoder Representations from Transformers

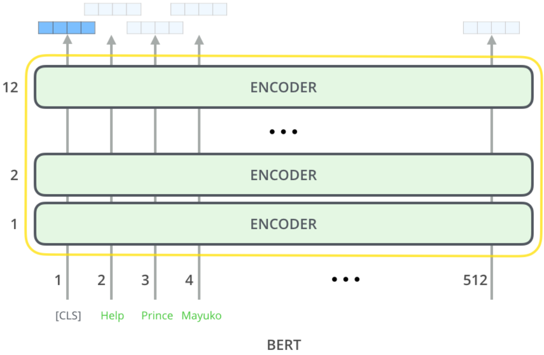


Figure: BERT is a pre-trained Transformer Encoder Stack

BERT (2/6): Bidirectional Encoder Representations from Transformers

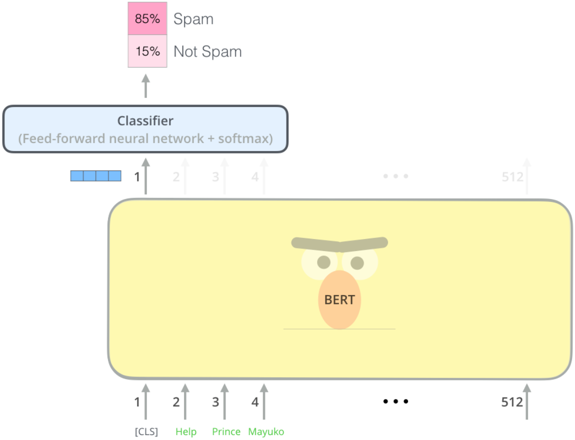


Figure: Leveraging BERT's output ("representation") to fine-tune downstream NLU tasks.

BERT (3/6): Word2Vec limit = Context matters!

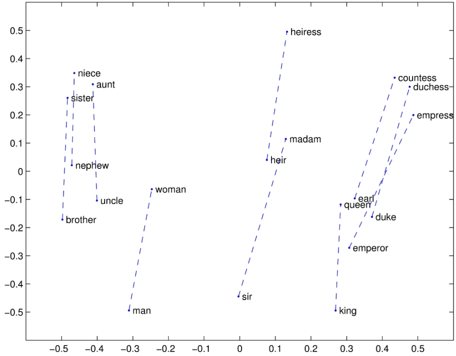
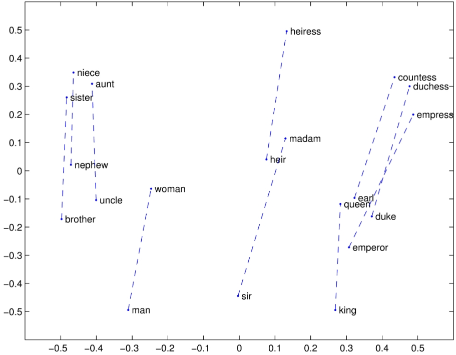
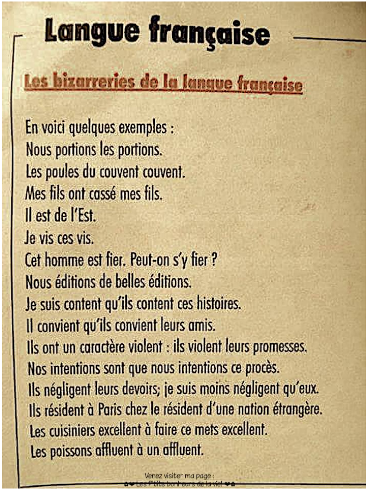


Figure: Fixed GloVe [8] embeddings projected on a plan

BERT (3/6): Word2Vec limit = Context matters!



(a) Fixed GloVe [8] embeddings projected on a plan



Langue française

Les bizarreries de la langue française

En voici quelques exemples :
Nous portions les portions.
Les poules du couvent couvent.
Mes fils ont cassé mes fils.
Il est de l'Est.
Je vis ces vis.
Cet homme est fier. Peut-on s'y fier ?
Nous éditions de belles éditions.
Je suis content qu'ils content ces histoires.
Il convient qu'ils convient leurs amis.
Ils ont un caractère violent : ils violent leurs promesses.
Nos intentions sont que nous intentions ce procès.
Ils négligent leurs devoirs; je suis moins négligent qu'eux.
Ils résident à Paris chez le résident d'une nation étrangère.
Les cuisiniers excellent à faire ce mets excellent.
Les poissons affluent à un affluent.

Venez visiter ma page :

(b) French Language Oddities...

BERT (4-1/6): Pre-training a Masked Language Model

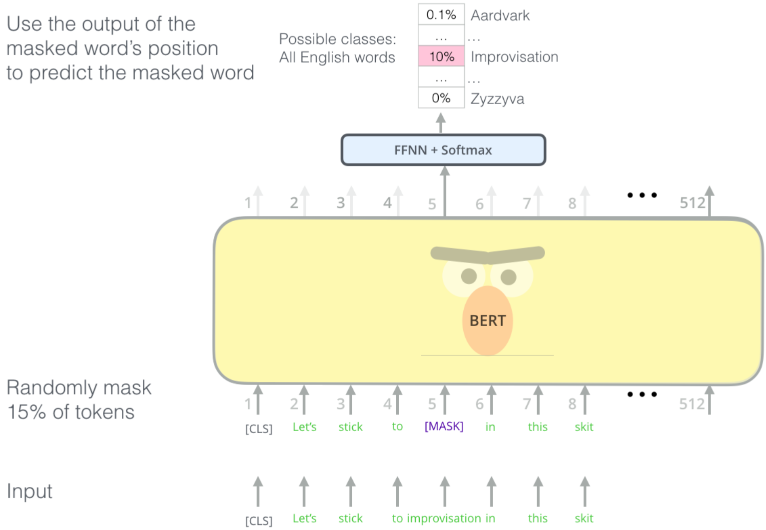


Figure: BERT's clever **language modeling** task masks 15% of words in the input and asks the model to predict the missing word.

BERT (4-2/6): Pre-training on "next sentence prediction"

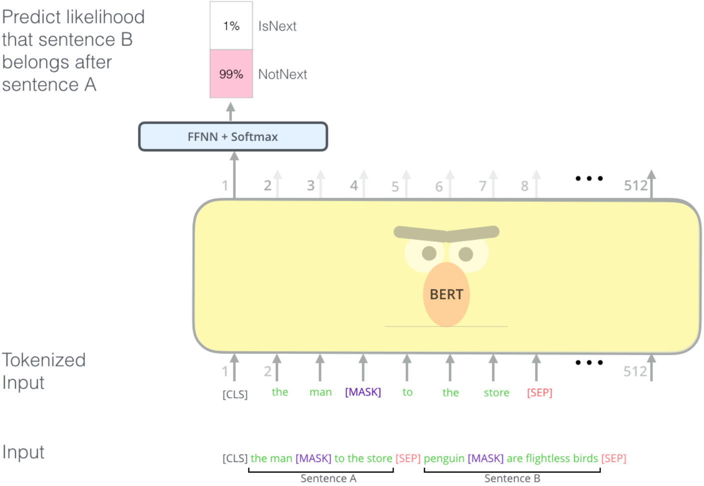


Figure: The second task BERT is pre-trained on is a two-sentence classification task.

BERT (5/6): Fine-tuning by adapting the model on specific tasks

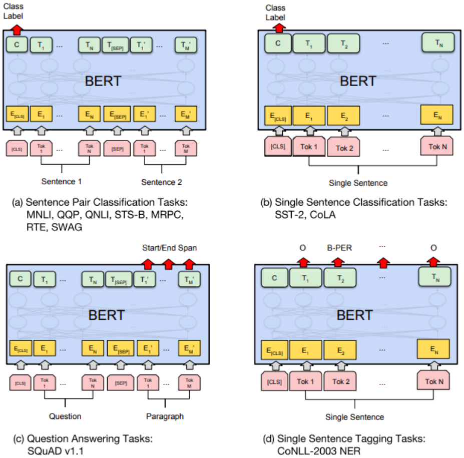


Figure: The BERT paper [3] shows a number of ways to use BERT for different tasks.

BERT (6/6): Results on NLU tasks

BERT Size:

- BERTBASE (L=12, H=768, A=12, Total Parameters=110M)
- BERTLARGE (L=24, H=1024, A=16, Total Parameters=340M).

General Language Understanding Evaluation (GLUE) benchmark: Leaderboard

- Grammaticallity: CoLA
- Sentiment Analysis: SST-2
- Similarity and paraphrase: MRPC, STS-B, QQP
- Inference: MNLI, QNLI, RTE, WNLI

SuperGLUE: Leaderboard.

The Stanford **Question Answering** Dataset (SQuAD 2.0): Leaderboard

BERT (6/6): Results on NLU tasks

The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, Il milione (or, The Million, known in English as the Travels of Marco Polo), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge through contact with Persian traders since many of the places he named were in Persian.

How did some suspect that Polo learned about China instead of by actually visiting it?

Answer: through contact with Persian traders

Figure: Example of a question in SQuAD 2.0

Contents

GPT-2 (1/7): Introduction to Language Modeling

What is a Language Model?

A statistical language model is a probability distribution over sequences of words.

Predicting the next word: $p(w_t|w_1, \cdots, w_{t-1})$

Predicting a sentence of $s = (w_1, \cdots, w_n)$ length n:
$p(s) = \prod_{i=1}^{n} p(w_i|w_1, \cdots, w_{i-1})$

Example of direct application in real life: word suggestion - Gmail "Smart Compose"

GPT-2 (2/7): Language Models are Unsupervised Multitask Learners

Generative **P**re-**T**raining

Training set : WebText, 40 GB of text data crawled from the Internet.

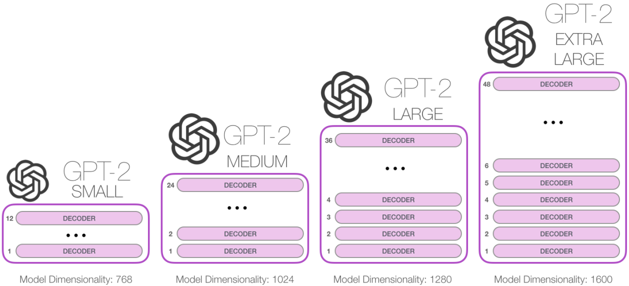
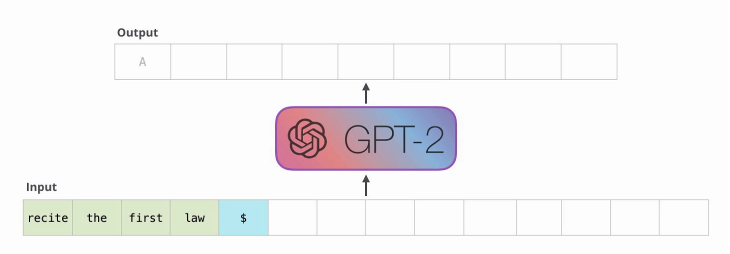


Figure: Architecture hyperparameters for the 4 model sizes.

Memory sizes:

- GPT-2 Small :117M parameters, 500 MBs, released
- GPT-2 Extra Large: 1.542B parameters **6.5 GBs**, not released because of "deep fakes"

GPT-2 (3/7): The "auto-regression" generation process



GPT-2 (4/7): The masked self-attention in the transformer decoder

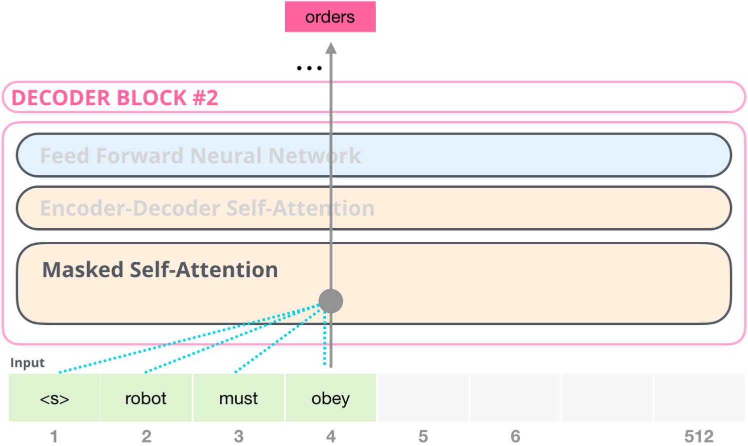


Figure: One key difference in the self-attention layer here, is that it masks future tokens – not by changing the word to [mask] like BERT, but by interfering in the self-attention calculation blocking information from tokens that are to the right of the position being calculated.

GPT-2 (4/7): The masked self-attention in the transformer decoder

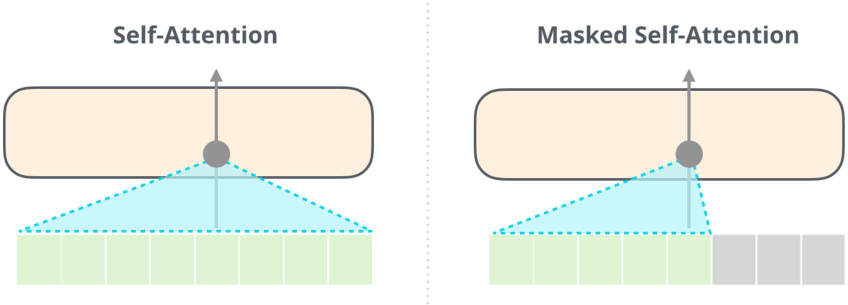
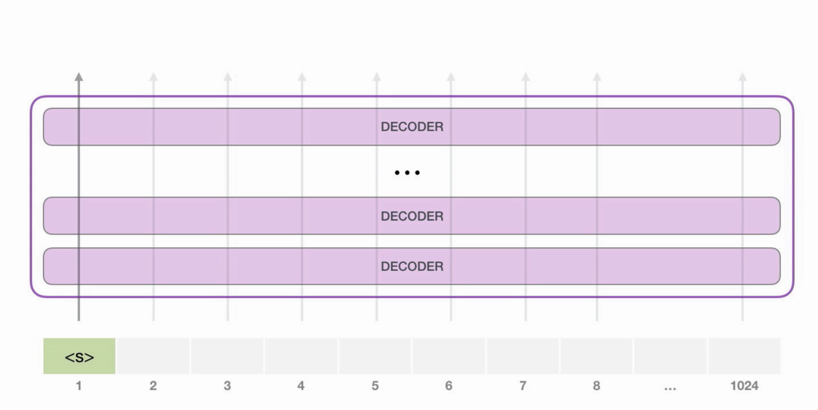
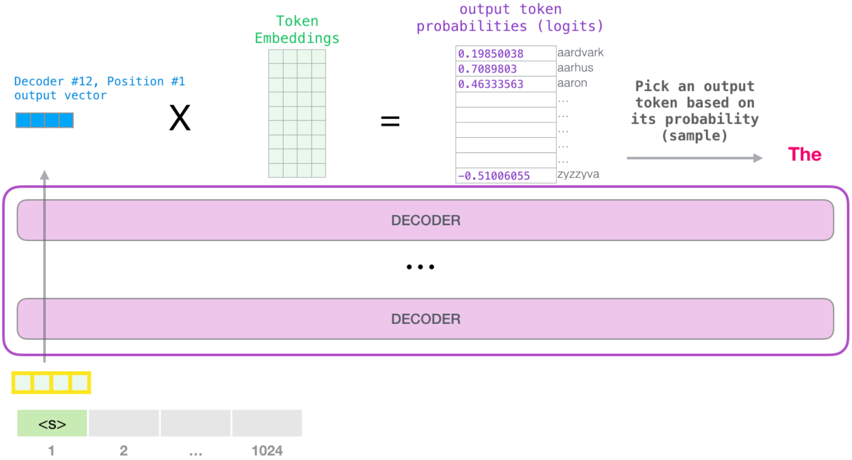


Figure: Distinction between self-attention (what BERT uses) and masked self-attention (what GPT-2 uses)

GPT-2 (5/7): Generating unconditional samples with a Start of Sentence token



GPT-2 (6/7): Generating words from their representation output by the transformer



GPT-2 (7/7): Beyond Language Modeling

Even though results on other language tasks such as Question Answering are far from the SOTA using supervised learning, GPT-2 achieves decent results for a model trained **unsupervisingly**, because the unlabeled training set (and also the model) is huge (40 GB).

Training Dataset

I	am	a	student	<to-fr>	je	suis	étudiant
let	them	eat	cake	<to-fr>	Qu'ils	mangent	de
good	morning	<to-fr>	Bonjour				

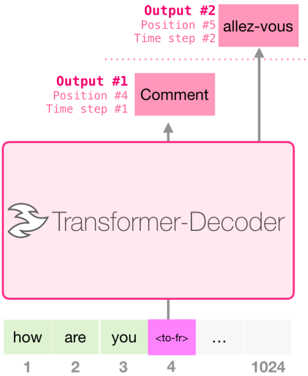


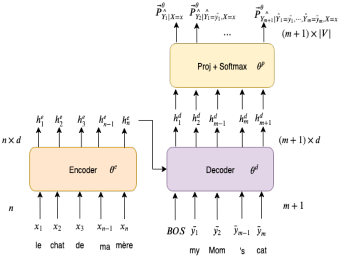
Figure: Example of prompting the trained model in the right way to perform Machine Translation

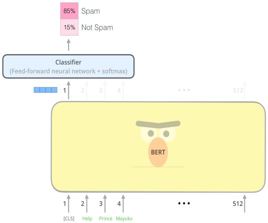
Contents

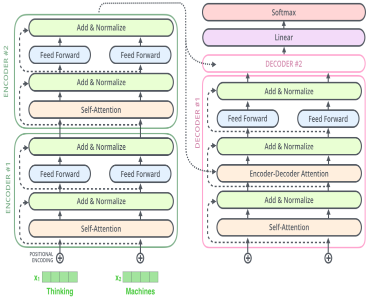
Conclusion (1/4): Important details not mentioned before

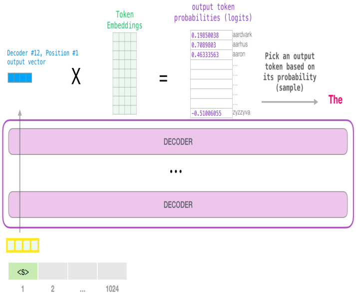
- A positional signal (sine wave) encodes the position of words in the sentence
- Word are not directly tokenized, instead sub-words are tokenized: See WordPiece (Google) and Byte-Pair Encoding (Facebook)
- Transformers use a lot of layer normalization, which is pretty important.
- In NLG, beam search algorithm can be used at inference time to generate top k most probable words per step instead of top 1, then generate several candidate sentences and eventually keep the most probable candidate sentence overall.

Conclusion (2/4): Wrap up









Conclusion (3/4): More recent models based on or inspired from BERT

- XL-Net (Yang *et al.*[9], 2019), based on the Transformer XL (Dai *et al.*[10], 2019)
- RoBERTa (Liu *et al.*[11], 2019)
- AlBERT

Conclusion (4/4): Implementation and packages (in Python)

Original implementations (TensorFlow 1):

- Transformer: https://github.com/tensorflow/tensor2tensor
- BERT: https://github.com/google-research/bert
- GPT-2: https://github.com/openai/gpt-2

Official TensorFlow 2.x: Transformer, BERT, XL-Net, GPT-2 (coming soon)

Hugging Face (https://huggingface.co/transformers/)
TensorFlow 2.0 and PyTorch 1.0.0+: BERT, GPT, GPT-2, Transformer-XL, XLNet, XLM, RoBERTa, DistilBERT

Texar (https://texar.io/): BERT, GPT2, XLNet

Conclusion (4/4): Implementation and packages (in Python)

Demos

- Allen Institute for Artificial Intelligence: https://gpt2.apps.allenai.org GPT-2 Medium and Large
- Talk to Transformer: https://talktotransformer.com/ GPT-2 Large
- Write With Transformer: https://transformer.huggingface.co/ GPT-2 Small, Medium and Large, XL-Net

References I

Ilya Sutskever, Oriol Vinyals, and Quoc V Le.
Sequence to sequence learning with neural networks.
In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
CoRR, abs/1706.03762, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.
Bert: Pre-training of deep bidirectional transformers for language understanding.
arXiv preprint arXiv:1810.04805, 2018.

References II

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever.
Language models are unsupervised multitask learners.
2019.

Yulia Tsvetkov.
Towards personalized adaptive nlp: Modeling output spaces in continuous-output language generation.
2019.

Yoon Kim.
Convolutional neural networks for sentence classification.
In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

References III

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio.
Neural machine translation by jointly learning to align and translate.
In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning.
Glove: Global vectors for word representation.
In *In EMNLP*, 2014.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le.
Xlnet: Generalized autoregressive pretraining for language understanding.
CoRR, abs/1906.08237, 2019.

References IV

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov.
Transformer-xl: Attentive language models beyond a fixed-length context.
CoRR, abs/1901.02860, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov.
Roberta: A robustly optimized BERT pretraining approach.
CoRR, abs/1907.11692, 2019.

LINCS Reading group: Transformer models in Artificial Intelligence for Natural Language Processing

Léo Laugier

*This presentation is greatly inspired by Jay Alammar's blog https://jalammar.github.io/ and almost all pictures and animations not referenced come from there.

November 7, 2019