

# NLP Reading Group — Meeting 2

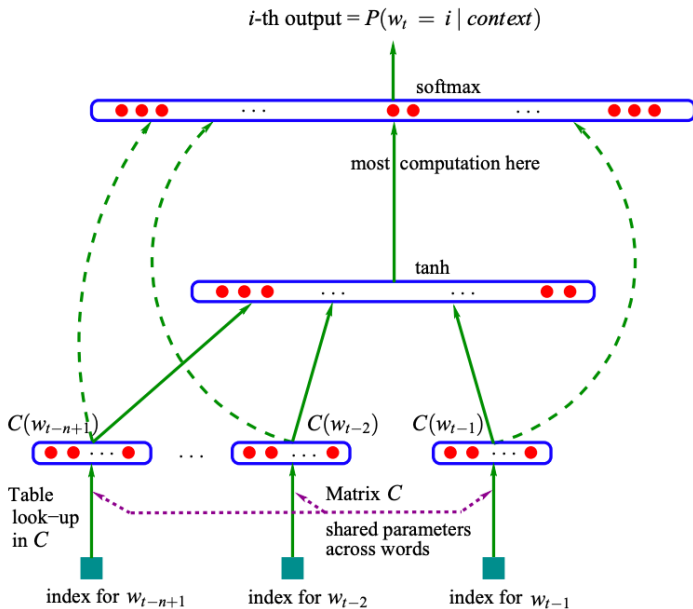
Matt Smith

UW Data Science Club

7 March, 2019

- 1 From Last Time
- 2 Neural Machine Translation by Jointly Learning to Align and Translate
- 3 Hierarchical Attention Networks for Document Classification
- 4 Attention Is All You Need
- 5 Improving Language Understanding with Unsupervised Learning

# Model



# What are those green dashed lines?

The original paper allows for direct (residual) connections between the final layer and the output of the embedding.

If  $x$  is the output of the embedding matrix, then the output of the model before the softmax layer is

$$y := U \tanh(Hx + d) + Wx + b$$

where  $b, d$  are bias terms and  $H, U, W$  are learned matrices.

Those dashed green lines indicate those residual connections. They allow for the gradient to more easily update the matrix  $C$ .

# Comparing Language Models

Let's compare the character LSTM from last time with GPT-2.

GPT-2 is a language model trained by OpenAI in the recent paper *Language Models are Unsupervised Multitask Learners*. It is currently unreleased due to concerns that people would misuse it to generate convincing spam.

As the name of the paper suggests, OpenAI found that their model could learn to do many language tasks just from its unsupervised language modeling task (predicting the next word in a sequence).

GPT-2 exhibits “zero-shot” learning, outperforming models trained on purpose-built datasets and getting competitive results on others.

# Zero-Shot Unsupervised Translation

---

”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool]**.

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: **”Mentez mentez, il en restera toujours quelque chose,”** which translates as, **”Lie lie and something will always remain.”**

”I hate the word ‘**perfume**,” Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

If listened carefully at 29:55, a conversation can be heard between two guys in French: **“-Comment on fait pour aller de l’autre coté? -Quel autre coté?”**, which means **“- How do you get to the other side? - What side?”**.

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as **Have-you to go to movies/theater?**

**“Brevet Sans Garantie Du Gouvernement”**, translated to English: **“Patented without government warranty”**.

---

*Table 1.* Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

# Zero-Shot Tasks

Radford et al. found that they could make GPT-2 perform many tasks simply by stating the input to the model in natural language.

For example, article summarization can be performed by inputting the article and the string “TL;DR:”<sup>1</sup>, GPT-2 outputs a summary of the text.

As shown before, you can also perform translation with a sentence such as “X in French is ”.

To be clear, the resulting output is not great, but the fact that a model can learn these tasks at all from unsupervised inputs is amazing.

---

<sup>1</sup>Internet slang indicating a summary follows

# Comparing Outputs

Let's look at the generated output from Karpathy's blog post (a summarization of his paper *Visualizing and understanding recurrent networks*) and compare it with GPT-2. It should give you an idea of the current SOTA in NLP.

*Naturalism and decision for the majority of Arab countries' capitalide was grounded by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]], associated with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal in the [[Protestant Immineners]], which could be said to be directly in Cantonese Communication, which followed a ceremony and set inspired prison, training. The emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known in western [[Scotland]], near Italy to the conquest of India with the conflict. Copyright was the succession of independence in the slop of Syrian influence that was a famous German movement based on a more popular servicious, non-doctrinal and sexual power post. Many governments recognize the military housing of the [[Civil Liberalization and Infantry Resolution 265 National Party in Hungary]], that is sympathetic to be to the [[Punjab Resolution]] (PJS)[<http://www.humah.yahoo.com/guardian.cfm/7754800786d17551963s89.htm> Official economics Adjoint for the Nazism, Montgomery was swear to advance to the resources for those Socialism's rule, was starting to signing a major tripad of aid exile.]]*



# Comparing Outputs

## Given the prompt

*Recycling is good for the world.  
NO! YOU COULD NOT BE MORE WRONG!!*

## GPT-2 produces:

*Recycling is NOT good for the world. It is bad for the environment, it is bad for our health, and it is bad for our economy. I'm not kidding. Recycling is not good for the environment. It is destructive to the earth and it is a major contributor to global warming. Recycling is not good for our health. It contributes to obesity and diseases like heart disease and cancer. Recycling is bad for our economy. It increases the cost of a product, and in turn, the price of everything that is made with that product. Recycling is not good for our nation. We pay a tremendous price for the privilege of having the world's most advanced and efficient recycling system. Recycling is a huge, colossal waste of time, energy, money, and resources. And THAT is why we need to get back to basics and get back to basics in our recycling efforts. One of the best ways to start is to look at the process of creating a paper product. When you make a paper product, it is basically a long chain of materials. Everything from the raw materials (wood, cardboard, paper, etc.), to the reagents (dyes, solvents, etc.) to the printing equipment (chemicals, glue, paper, ink, etc.), to the packaging, to the packaging materials (mercury, chemicals, etc.) to the processing equipment (heating, cooling, etc.), to the packaging materials, to the packaging materials that are shipped overseas and to the packaging materials that are used in the United States. Each step along the way creates tons of waste that we constantly have to clean up. The process of making a paper product is a very wasteful one. But the end result is something that all of us need to consume. And if we want to keep the recycling process running efficiently, then we really need to think about each and every step that goes into making a paper product.*

# Coverage

- 1 From Last Time
- 2 Neural Machine Translation by Jointly Learning to Align and Translate
- 3 Hierarchical Attention Networks for Document Classification
- 4 Attention Is All You Need
- 5 Improving Language Understanding with Unsupervised Learning

Pretty obvious: Given some text in one language, convert it to another.

## Impossibility

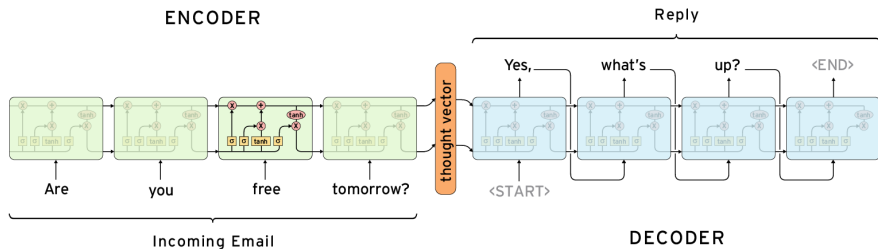
Sometimes we can't translate without assumptions. This is not fixable without changing the paradigm of translation.

- Politeness: “だ” vs “です”
- Idioms: “The cat's pyjamas” ⇒ “???”
- Different meanings: “I love you” vs “我爱你”
- Slang: “飯テロ”
- Many-to-many: “Let's do it” ⇒ “それで行こう” vs “するか”

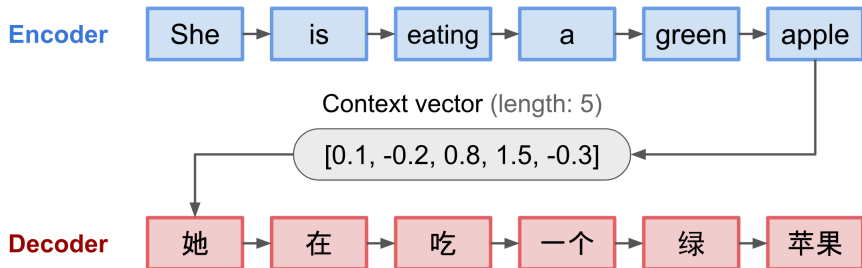
Usually evaluated by comparison with a database of human-generated candidate translations. Unsupervised learning is possible due to large amounts of parallel texts (movie subtitles, EU/UN proceedings, etc.)

# Encoder-Decoder Architecture

At the time, SOTA was encoder-decoder networks:



# Encoder-Decoders for Translation



# Paper-Reading Strategy

I skimmed the paper's sections to determine which sections have new content and which sections we can handle already.

- The abstract, sections 1 and 2 are important background information to motivate the paper; we'll skim these
- Section 2.1 just describes the LSTM so we'll skip it
- Section 3 talks about the creation of the alignment (attention) mechanism so I'll go over it with you
- Sections 4-7 talk about results, future work and conclusions; we'll skim these

How are we going to skim as a group? I have extracted the passages and figures that I think are most relevant.

*Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder-decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.*

## Section 1: Motivation

“Cho et al. showed that indeed the performance of a basic encoder—decoder deteriorates rapidly as the length of an input sentence increases.”

“In order to address this issue, we introduce an extension to the encoder—decoder model which learns to align and translate jointly. Each time the proposed model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated. The model then predicts a target word based on the context vectors associated with these source positions and all the previous generated target words.”

“The most important distinguishing feature of this approach from the basic encoder—decoder is that it does not attempt to encode a whole input sentence into a single fixed-length vector.”



## Section 2: NMT Background

“From a probabilistic perspective, translation is equivalent to finding a target sentence  $y$  that maximizes the conditional probability of  $y$  given a source sentence  $x$ , i.e.,  $\arg \max_y p(y | x)$ . In neural machine translation, we fit a parameterized model to maximize the conditional probability of sentence pairs using a parallel training corpus.”

“Despite being a quite new approach, neural machine translation has already shown promising results. Sutskever et al. (2014) reported that the neural machine translation based on RNNs with long short-term memory (LSTM) units achieves close to the state-of-the-art performance of the conventional phrase-based machine translation system on an English-to-French translation task. Adding neural components to existing translation systems, for instance, to score the phrase pairs in the phrase table (Cho et al., 2014a) or to re-rank candidate translations (Sutskever et al., 2014), has allowed to surpass the previous state-of-the-art performance level.”

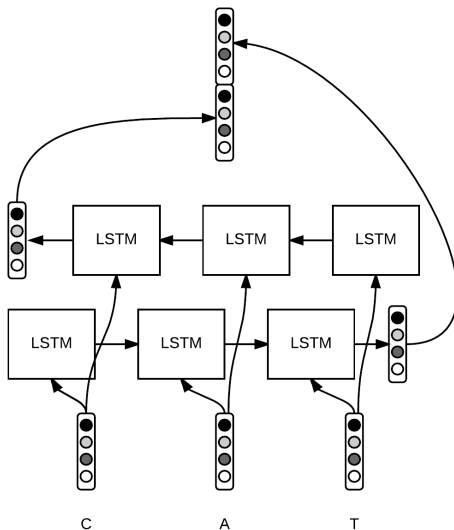
## Section 3.2: Learning to Align and Translate – BiRNN

Here, the authors employ a common technique in RNNs to increase performance. Usually RNNs are motivated because they seem to act like humans do when reading or listening to natural language:

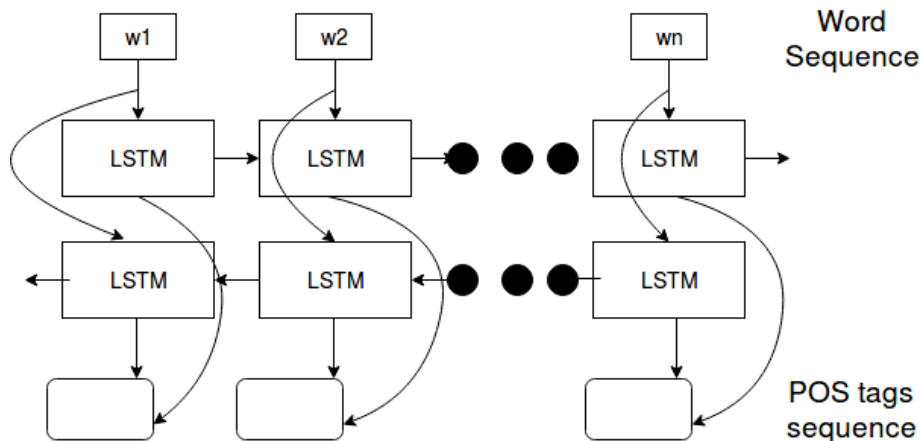
- Read word-by-word (or unit-by-unit) in “reading order”
- Keep a working memory that is used to contextualize future words

BiRNNs have the RNN read the sentence in both reading order and also in reversed order. The output vectors of the RNN are then combined (usually concatenation or addition).

# BiLSTM with One Output



# BiLSTM with Multiple Outputs



# Why?

We throw away a bit of our biological motivation for RNNs to achieve better performance.

## Exercise

Why does this work?

# Why?

We throw away a bit of our biological motivation for RNNs to achieve better performance.

## Exercise

Why does this work?

- We mentioned earlier that the vector in the encoder-decoder network was a bottle-neck for sentence length
- Similarly, the hidden state in a RNN is a bottleneck
- We often work hard to fix this bottleneck and allow our RNNs to parse longer-range dependencies (part of our motivation for LSTM/GRU)
- Single-output RNNs get context from both the beginning and the end of the sentence

## Section 3.1: Learning to Align and Translate – Decoder Attention

More biological motivation:

- When you read, you don't actually just read in reading order
- Your eyes move back and forth, rereading parts of the sentence sometimes if you need to get more context
- This is especially true if you're translating a sentence as languages may have different word order
- This means you don't need to keep the entire meaning of the sentence in your head, you can look back at the original input as needed!

Thus we want some sort of “attention” mechanism. The authors call this “soft-alignment” but future work frames it like a question-answering network (more on this later).

# Attention

When we are trying to use a RNN make a prediction at timestep  $t$ , we take as inputs an input vector  $x_t$  and a context vector  $s_{t-1}$ .

Typically in NMT, the input to the decoder at timestep  $t$  is just the output from the decoder at the previous timestep.

Instead, attention dynamically computes what the input to the decoder should be based on a weighted sum of the outputs of the encoder.

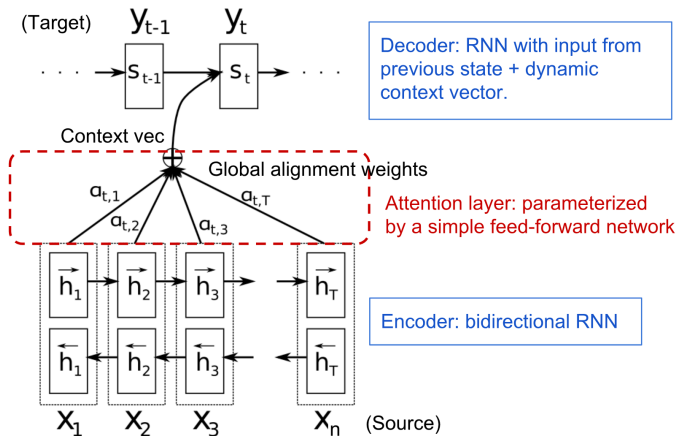
For each encoder output  $h_i$ , a weight  $\alpha_{t,i}$  is produced. Then,

$$x_t = \sum_{i=1}^T \alpha_{t,i} h'_i$$

where  $h_i$  is the output of the encoder at timestep  $i$ .



# A Graphical View of Attention



**Additive Attention**

Image: Lilian Weng's *Attention? Attention!*

# Computing attention weights

This paper computes attention weights using an attention model.

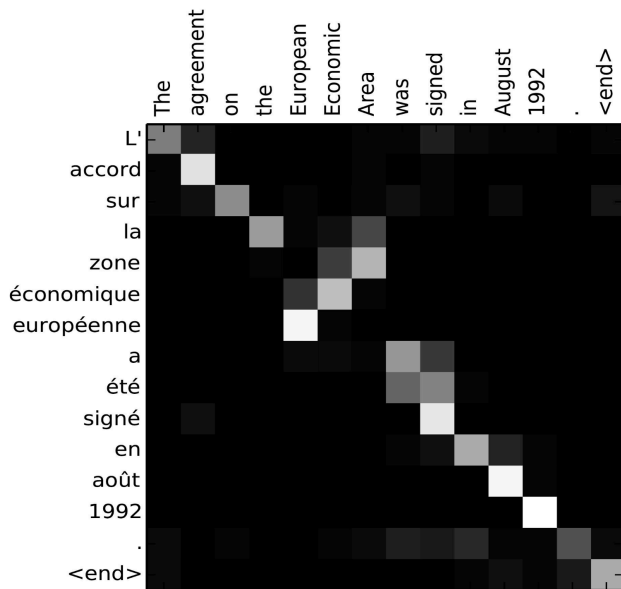
$$e_{t,i} = a([s_{t-1}; h_i])$$
$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^T \exp(e_{t,j})}$$

In this case  $a$  is a feed-forward neural network,

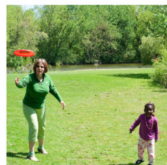
$$a([s_{t-1}; h_i]) = \sigma(W[s_{t-1}; h_i] + b)$$

Notes about attention: it's kind of slow. The naive implementation runs on the CPU.

# Visualizing Attention



# Visualizing Attention



A(0.98)



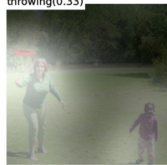
woman(0.54)



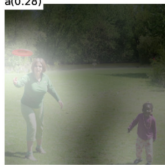
is(0.37)



throwing(0.33)



a(0.28)



frisbee(0.37)



in(0.21)



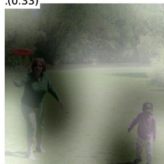
a(0.18)



park(0.35)



.(0.33)



# Visualizing Attention

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

# Different Kinds of Attention – Self-Attention

The previous two pictures are examples of self-attention.

Instead of having the a decoder attend to an encoder, an encoder can attend to its previous layers.

This provides a powerful and generic layer for discovering features that depend on correlations within data e.g. anaphora resolution.

## Anaphora Resolution

“Stacy went to the movies. She had fun.”

Who does ‘she’ refer to? ‘She’ is an anaphora, and determining that ‘she’ = ‘Stacy’ is anaphora resolution.

Self-attention uses a dot-product formulation that can be done on GPU.

# Dot-Product Attention


Input

Thinking

Machines

Embedding

$x_1$  

$x_2$  

Queries

$q_1$  

$q_2$  

Keys

$k_1$  

$k_2$  

Values

$v_1$  

$v_2$  

Score

$q_1 \cdot k_1 = 112$

$q_1 \cdot k_2 = 96$

Divide by  $8 (\sqrt{d_k})$

14

12

Softmax

0.88

0.12

Softmax

X

Value

$v_1$  

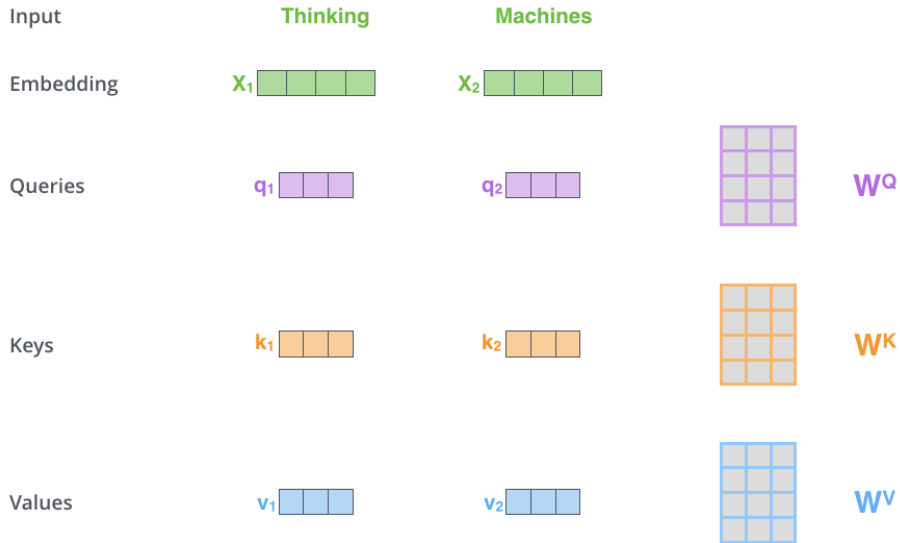
$v_2$  

Sum

$z_1$  

$z_2$  

# Dot-Product Attention





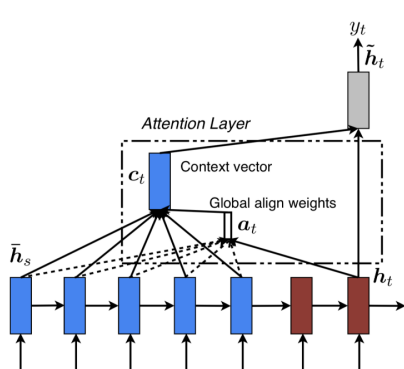
# Dot-Product Attention



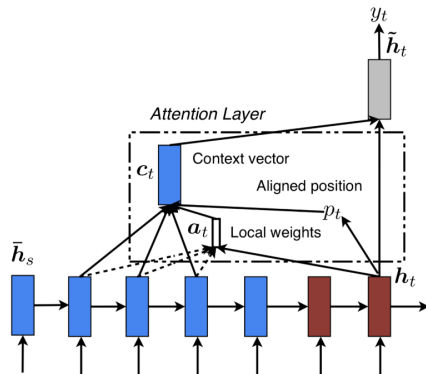
# Dot-Product Attention

$$\text{softmax} \left( \frac{\begin{matrix} \mathbf{Q} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{matrix} \times \begin{matrix} \mathbf{K}^T \\ \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} \end{matrix} \right) \begin{matrix} \mathbf{V} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{matrix}$$
$$= \begin{matrix} \mathbf{Z} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{matrix}$$

# Different Kinds of Attention – Global vs Local



Global Attention Model

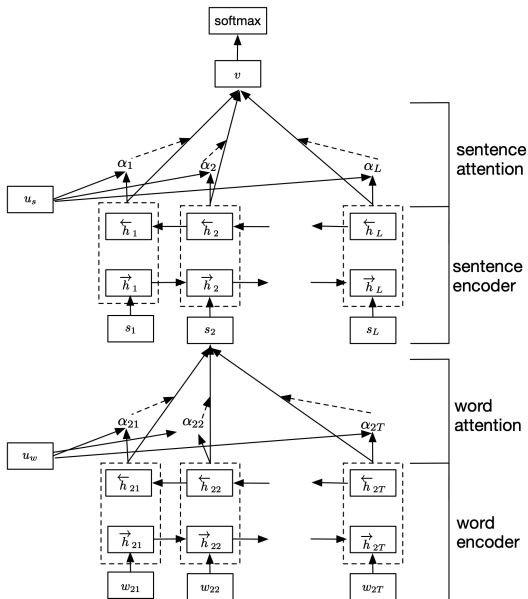


Local Attention Model

# Coverage

- 1 From Last Time
- 2 Neural Machine Translation by Jointly Learning to Align and Translate
- 3 Hierarchical Attention Networks for Document Classification**
- 4 Attention Is All You Need
- 5 Improving Language Understanding with Unsupervised Learning

# Model – Talk about implications



# Coverage

- 1 From Last Time
- 2 Neural Machine Translation by Jointly Learning to Align and Translate
- 3 Hierarchical Attention Networks for Document Classification
- 4 Attention Is All You Need**
- 5 Improving Language Understanding with Unsupervised Learning

# Self-attention

RNNs take a sequence as input and output a sequence.

Attention mechanisms take a sequence as input and output a sequence.

#showerthoughts: what if we only used attention mechanisms.

Enter transformer networks. All of the work is done through self-attention and dense layers.

We already know how scaled dot-product attention works so I'll describe it simply.

# Multi-head Attention Mechanism

Transformers use multiple “heads” of attention to attend to multiple concepts at once.

The  $V$ ,  $K$  and  $Q$  matrices are projected into subspaces, attended to and then concatenated.

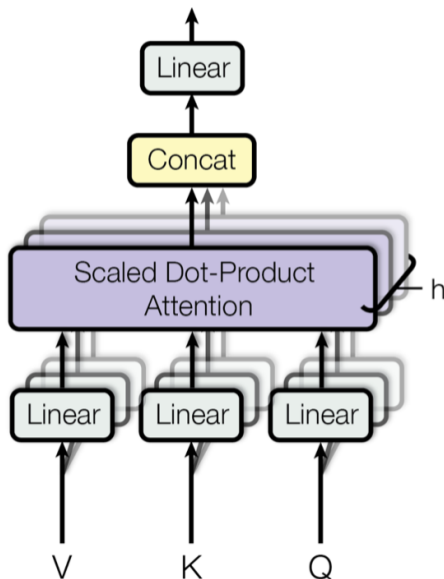
The intuition here is that different heads can learn different linguistic features.

A parallel intuition is that this is somewhat like ensembling.

The final result for each head is concatenated together and then passed through a dense layer.



# Multi-head Attention Mechanism



The final model is quite complicated, but I'll throw up a picture on the next slide.

This is still an encoder/decoder network. There are two notes to make here:

1) Because there's no RNNs, we can't access word order. In order to fix this, we introduce some periodic noise that the model can learn:

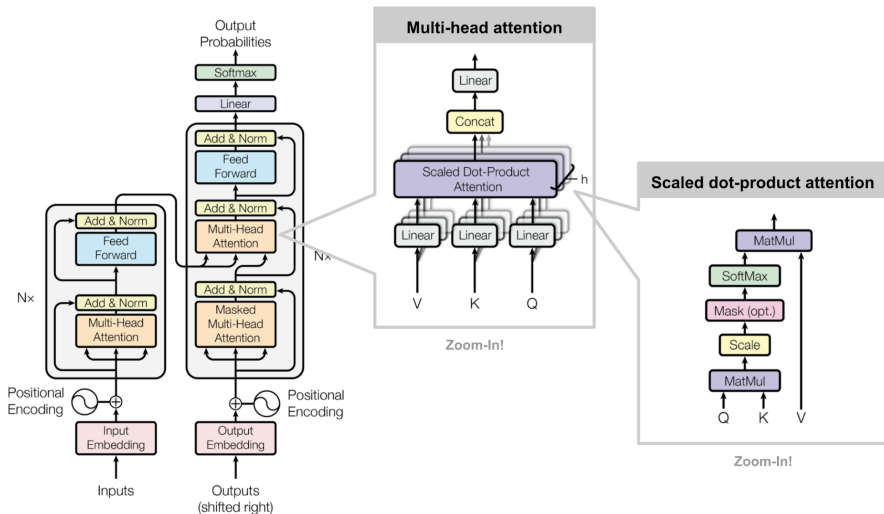
$$x_{(pos,2i)} += \sin(pos/10000^{2i/d_{model}})$$

$$x_{(pos,2i+1)} += \cos(pos/10000^{2i/d_{model}})$$

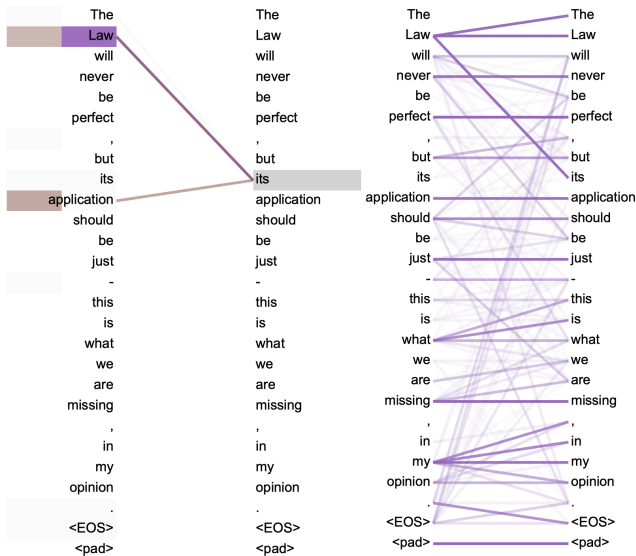
The authors also tried learned embeddings.

2) We modify one of the attention mechanisms in the decoder to stop it from “looking into the future” (masking)

# Picture of the Model



# Visualizing Self-attention – Anaphora Resolution



# Coverage

- 1 From Last Time
- 2 Neural Machine Translation by Jointly Learning to Align and Translate
- 3 Hierarchical Attention Networks for Document Classification
- 4 Attention Is All You Need
- 5 Improving Language Understanding with Unsupervised Learning**

*Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by generative pre-training of a language model on a diverse corpus of unlabeled text, followed by discriminative fine-tuning on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied.*

Before this paper, transformer networks were impractical. Transformers are hard to train.

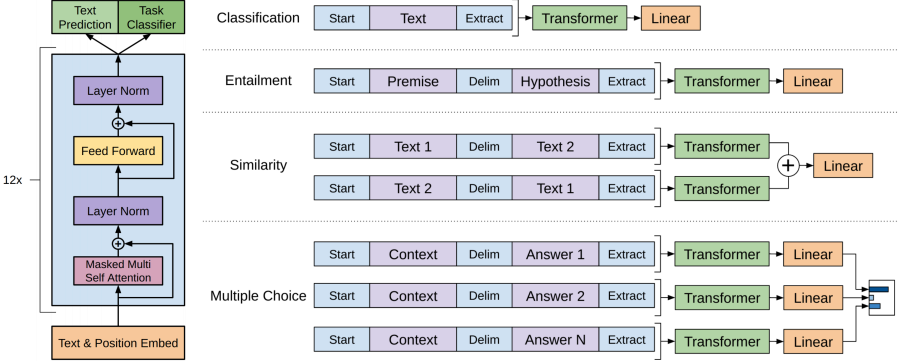
The idea in this paper: by training a transformer on a language modeling task, it provides a very good initialization for quickly retraining the transformer on a domain-specific task with a smaller labeled dataset.

This model in this paper is “GPT-1”.

The trick is representing text in a special form such that the input type (strings) says the same. They do this by concatenating input dimensions using special delimiters.

Note the similarity to how GPT-2 does multi-task learning, although GPT-1 is all about retraining on a small dataset while GPT-2’s paper was focused on unsupervised learning.

# Model





# Homework

Practice:

Download a limited version of GPT-2 (<https://github.com/openai/gpt-2>). Come up with a creative input prompt to get it to perform a task. E.g. “The objectively best meme is ” for meme generation (I hope).

Your results might not be as impressive as this is a scaled-down version, but I hope you have fun.

Exercises (in order of masochism):

- 1 Implement your own attention mechanism
- 2 Implement a HAN from the paper
- 3 Implement a transformer from the paper (actually quite a good exercise, see <https://github.com/lilianweng/transformer-tensorflow>)