EEP 596: Adv Intro ML || Lecture 16

Dr. Karthik Mohan

Univ. of Washington, Seattle

February 28, 2023

Lots of Due Dates

1. Mini-project 1 due March 1st, tomorrow

Lots of Due Dates

1. Mini-project 1 due March 1st, tomorrow
2. Last **conceptual assignment** on Deep Learning will be assigned in couple of days and due in 2 weeks

Lots of Due Dates

1. Mini-project 1 due March 1st, tomorrow
2. Last **conceptual assignment** on Deep Learning will be assigned in couple of days and due in 2 weeks
3. **Mini-project 2** on **Twitter Emotions Analysis** and **Zero-Shot Emotion Detection** to be posted wednesday and due in 2 weeks - Can use cutting edge DL models including Transformers for this!

Lots of Due Dates

1. Mini-project 1 due March 1st, tomorrow
2. Last **conceptual assignment** on Deep Learning will be assigned in couple of days and due in 2 weeks
3. **Mini-project 2** on **Twitter Emotions Analysis** and **Zero-Shot Emotion Detection** to be posted wednesday and due in 2 weeks - Can use cutting edge DL models including Transformers for this!
4. Last Lecture on March 9th, but project presentations will be in finals week

Lots of Due Dates

1. Mini-project 1 due March 1st, tomorrow
2. Last **conceptual assignment** on Deep Learning will be assigned in couple of days and due in 2 weeks
3. **Mini-project 2** on **Twitter Emotions Analysis** and **Zero-Shot Emotion Detection** to be posted wednesday and due in 2 weeks - Can use cutting edge DL models including Transformers for this!
4. Last Lecture on March 9th, but project presentations will be in finals week
5. **5 minute team presentations** on either of the two mini-projects on one of two days: **March 14th or March 16th**

Lots of Due Dates

1. Mini-project 1 due March 1st, tomorrow
2. Last **conceptual assignment** on Deep Learning will be assigned in couple of days and due in 2 weeks
3. **Mini-project 2** on **Twitter Emotions Analysis** and **Zero-Shot Emotion Detection** to be posted wednesday and due in 2 weeks - Can use cutting edge DL models including Transformers for this!
4. Last Lecture on March 9th, but project presentations will be in finals week
5. **5 minute team presentations** on either of the two mini-projects on one of two days: **March 14th or March 16th**
6. Pick your slot - 10 team presentations on March 14th and March 16th

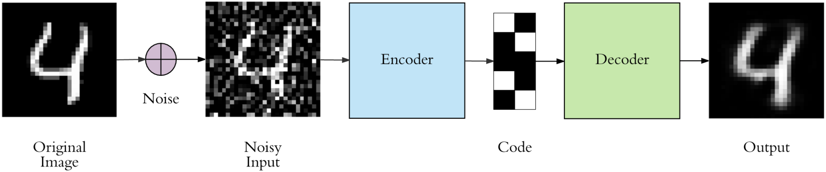
Last Time

a. Applications in NLP
b. State of the art models in NLP

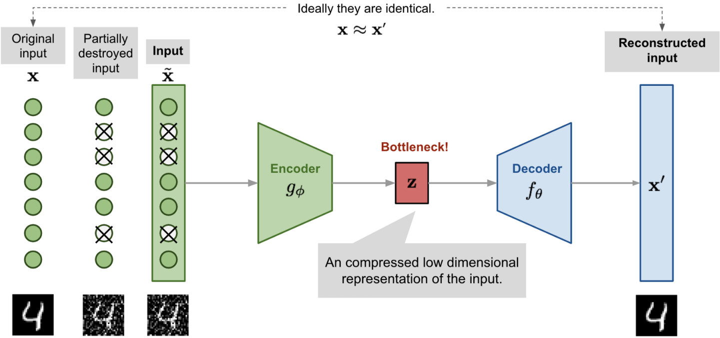
Today

- NLP applications
- Evolution of DL models esp. for NLP
- Attention and Transformers
- Transformer Demo

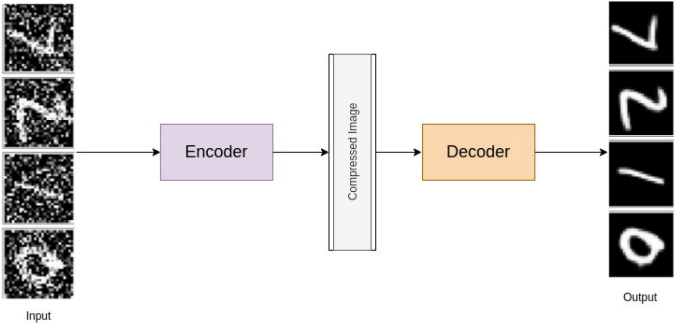
De-noising Auto Encoders



De-noising Auto Encoders



De-noising Auto Encoders



De-noising Auto Encoders

Details

- Just like an Auto Encoder

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.
- This forces the Auto Encoder to "de-noise" data, esp. useful for images!

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.
- This forces the Auto Encoder to “de-noise” data, esp. useful for images!
- Esp. useful for a category of objects or images (e.g. digit recognition or face recognition, etc)

De-noising Auto Encoders

Details

- Just like an Auto Encoder
- Difference: Noise is injected in the inputs on purpose but output is a clean data point.
- This forces the Auto Encoder to "de-noise" data, esp. useful for images!
- Esp. useful for a category of objects or images (e.g. digit recognition or face recognition, etc)
- De-noising AEs can be used to learn **noise-aware embeddings** - Helps with improving robustness of downstream models

ICE #1

Unsupervised Learning

Which of these is NOT an example of unsupervised learning?

1. Perceptron
2. Auto Encoder
3. De-noising Auto Encoder
4. K-means++
5. None of the above
6. All of the above

ML Modeling applications of Auto Encoders

1. **BERT Transformer:** One of the most popular architectures for NLP comprehension tasks is based on auto-encoder style loss function. Given a sentence with masked tokens, predict the masks.

ML Modeling applications of Auto Encoders

1. **BERT Transformer:** One of the most popular architectures for NLP comprehension tasks is based on auto-encoder style loss function. Given a sentence with masked tokens, predict the masks.
2. **BART:** Another popular architecture for both NLP comprehension and auto-regressive Language Generation is trained as a Denoising AutoEncoder!

ML Modeling applications of Auto Encoders

1. **BERT Transformer:** One of the most popular architectures for NLP comprehension tasks is based on auto-encoder style loss function. Given a sentence with masked tokens, predict the masks.
2. **BART:** Another popular architecture for both NLP comprehension and auto-regressive Language Generation is trained as a Denoising AutoEncoder!
3. More on BERT and BART when we get to **Transformers**

Sequence structure in NLP

Example

I love this car! Positive Sentiment

Sequence structure in NLP

Example

I love this car! Positive Sentiment

Example

I am not sure I love this car! Negative Sentiment

Sequence structure in NLP

Example

I love this car!   Positive Sentiment

Example

I am not sure I love this car!   Negative Sentiment

Example

I don't think its a bad car at all! $\rightarrow$ Positive Sentiment

Sequence structure in NLP

Example

I love this car! Positive Sentiment

Example

I am not sure I love this car! Negative Sentiment

Example

I don't think its a bad car at all! $\rightarrow$ Positive Sentiment

Example

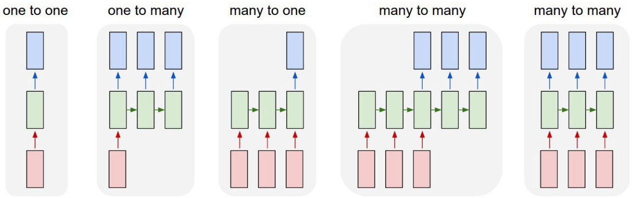
Have to carry the **context(state)** from some-time back to fully understand what's happening!

Next Mini Project

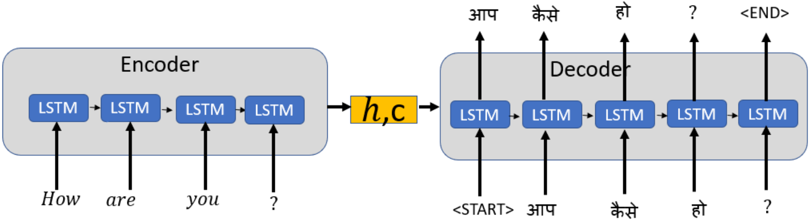
Twitter Emotion Analysis

Can you train a model to identify emotions from tweets? E.g. "I don't know if anything is going right for me these days." (Sadness/Anger) And what if the model is asked about an emotion if it's never seen as a label in training before (zero-shot learning)? E.g. is the previous sentence connected to the emotion of frustration? (model hasn't seen frustration in the training ever before). Two kaggle contests for each task will be setup! (The two tasks are related!)

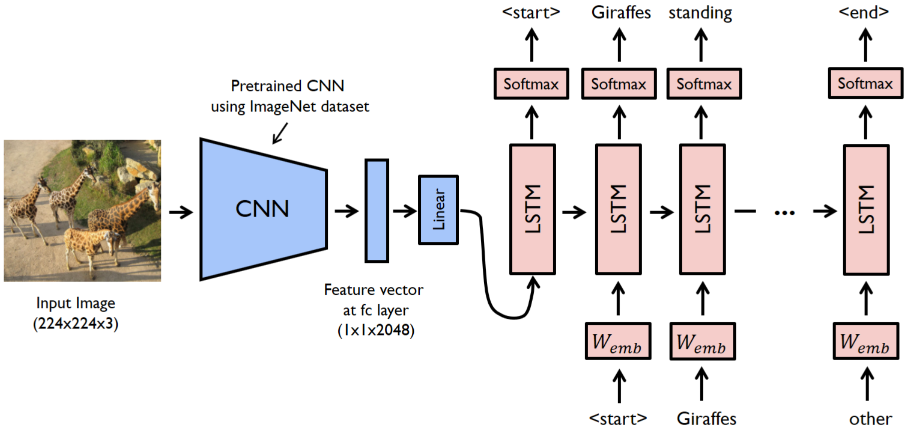
Sequence to Sequence Model (LSTM) Applications



Sequence to Sequence Model (LSTM) Applications



Sequence to Sequence Model (LSTM) Applications



Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation
3. Sentiment Analysis

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation
3. Sentiment Analysis
4. Question Answering

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation
3. Sentiment Analysis
4. Question Answering
5. Chat bots

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation
3. Sentiment Analysis
4. Question Answering
5. Chat bots
6. Document Summarization

Applications in Natural Language Processing (NLP)

Applications

1. Topic Modeling
2. Machine Translation/Language Translation
3. Sentiment Analysis
4. Question Answering
5. Chat bots
6. Document Summarization
7. Many more!

Topic Modeling

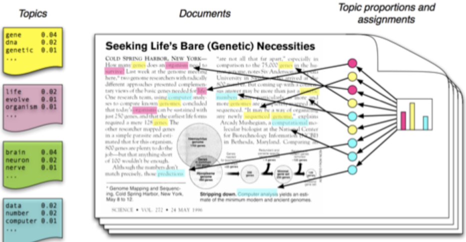
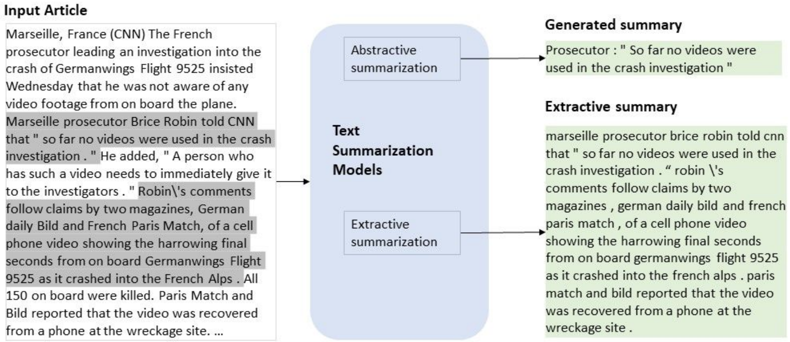


Figure source: Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

Document Summarization — Extractive



Evaluation Metrics

1. ROUGE score: Recall-Oriented Understudy for Gisting Evaluation
2. ROUGE-N: N-gram overlap between two summaries

ICE #2

ROUGE-1

Consider the truth summary and an automated summary of an article from International Geographic! Find the ROUGE-N score based on finding the proportion of N-grams in the truth summary that are also in the automated summary for $N = 1$.

Truth Summary: A symbiotic relationship exists between these two species. The cows feed on wild grass and the egrets feed on the tics found on the surface of the cows.

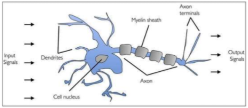
Automated Summary: These two species have a symbiotic relationship.

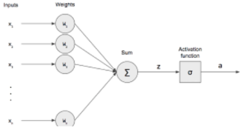
ROUGE-1 =

a) 0.33 b) 0.4 c) 0.2 d) 0.25

Evolution of DNN architectures for NLP!

Perceptron

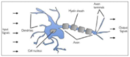


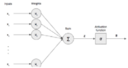


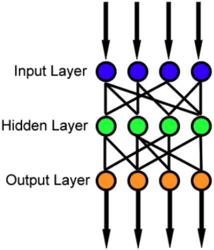
Evolution of DNN architectures for NLP!

Perceptron

Feed Forward NN

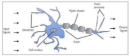




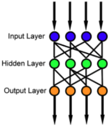


Evolution of DNN architectures for NLP!

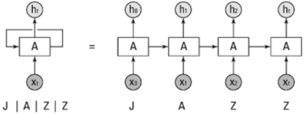
Perceptron



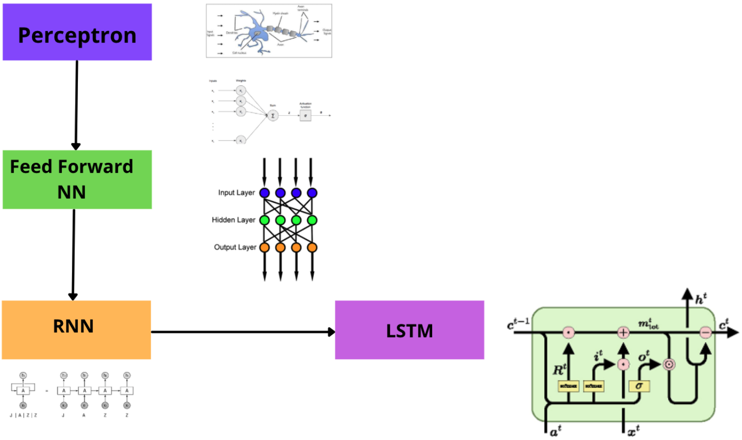
Feed Forward NN

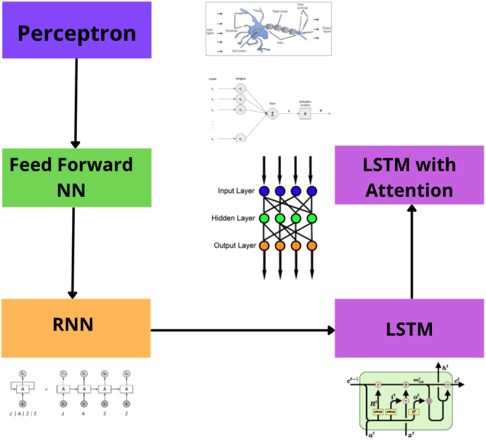


RNN

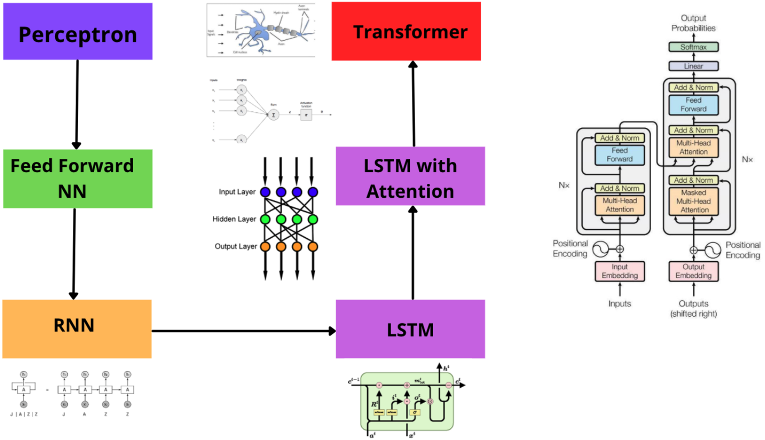


Evolution of DNN architectures for NLP!

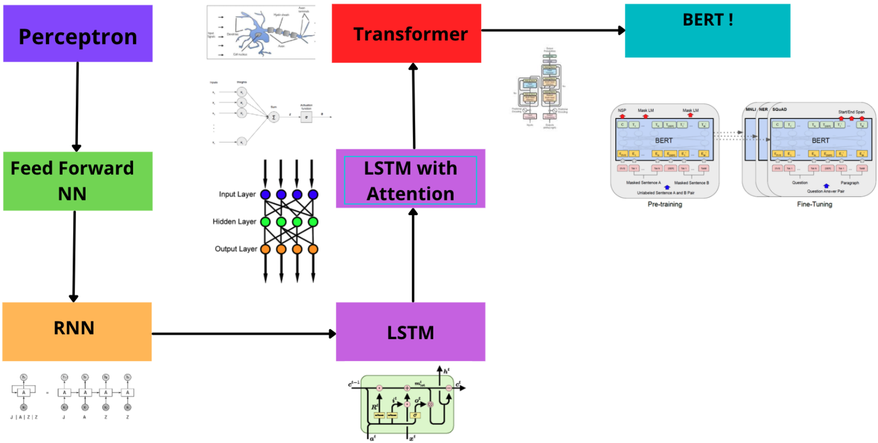



Perceptron
Feed Forward NN
RNN
LSTM
LSTM with Attention
Input Layer
Hidden Layer
Output Layer

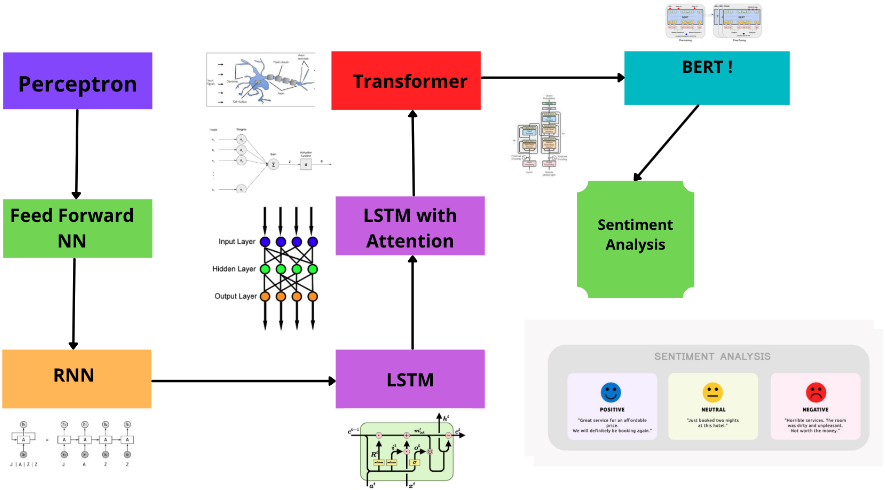
Evolution of DNN architectures for NLP!



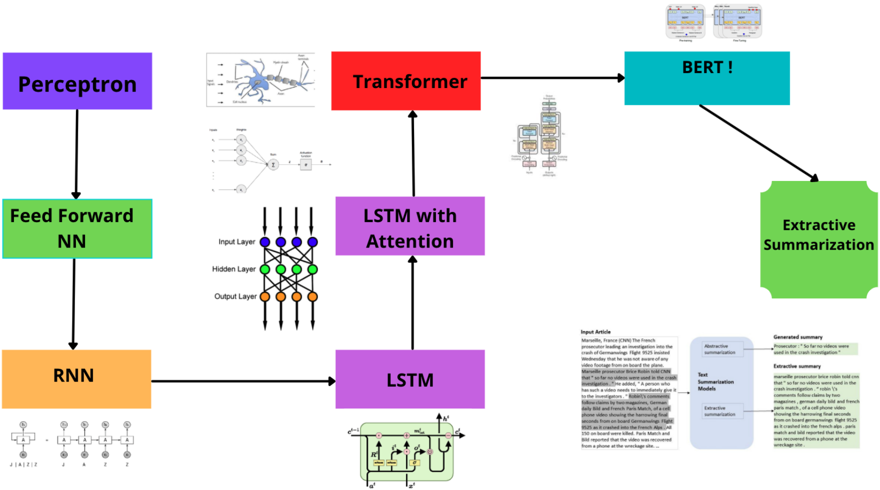
Evolution of DNN architectures for NLP!



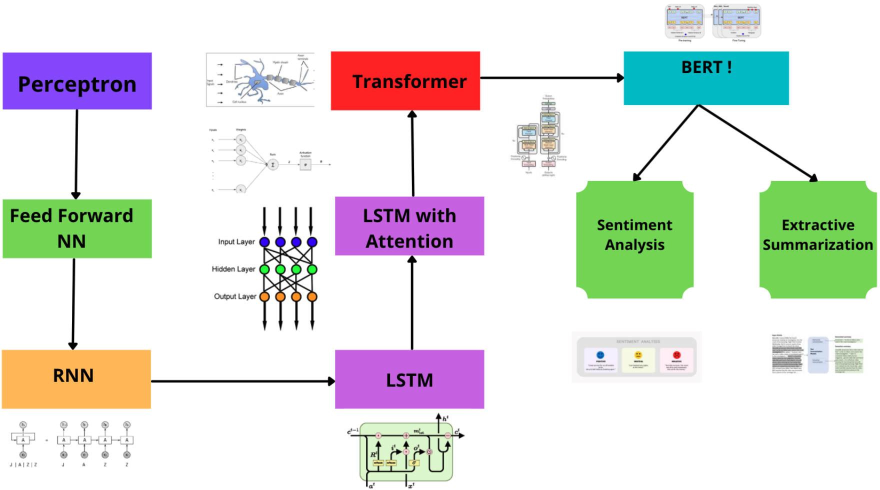
Evolution of DNN architectures for NLP!



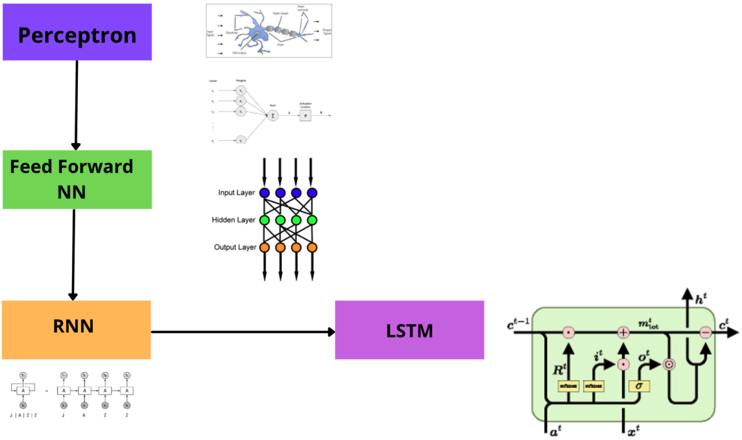
Evolution of DNN architectures for NLP!



Evolution of DNN architectures for NLP!

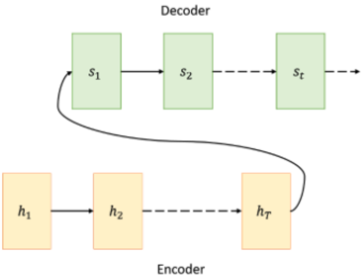


LSTM model

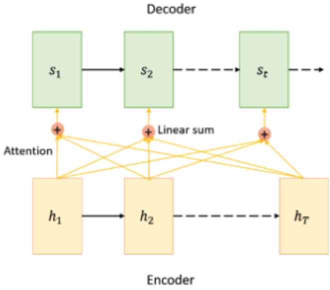


Perceptron
Feed Forward NN
RNN
LSTM
Input Layer
Hidden Layer
Output Layer

LSTM with attention



(a) Vanilla Encoder Decoder Architecture



(b) Attention Mechanism

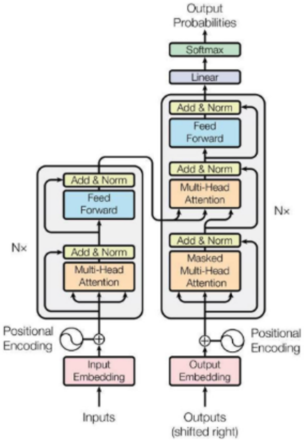
ICE #3

RNN vs LSTM

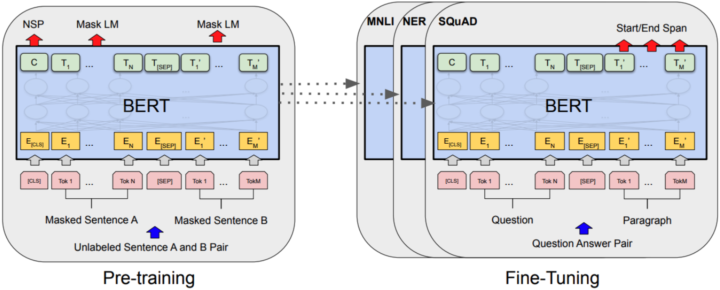
Which of the following statements are NOT true?

1. LSTM doesn't have the exploding/vanishing gradients issue as it occurs in RNNs
2. LSTM applies to sequential language tasks while RNNs applies to non-sequential language tasks
3. LSTM is better than RNN in most language tasks
4. LSTMs can be used for machine translation tasks

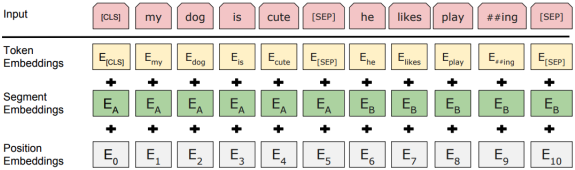
Transformer Archtiecture



BERT - Bi-directional Encoders from Transformers



BERT Embeddings



BERT pre-training

Two Tasks

1. **Masked LM Model:** Mask a word in the middle of a sentence and have BERT predict the masked word
2. **Next-sentence prediction:** Predict the next sentence - Use both positive and negative labels. How are these generated?

BERT pre-training

Two Tasks

1. **Masked LM Model:** Mask a word in the middle of a sentence and have BERT predict the masked word
2. **Next-sentence prediction:** Predict the next sentence - Use both positive and negative labels. How are these generated?

ICE: Supervised or Un-supervised?

1. Are the above two tasks supervised or un-supervised?

BERT pre-training

Two Tasks

1. **Masked LM Model:** Mask a word in the middle of a sentence and have BERT predict the masked word
2. **Next-sentence prediction:** Predict the next sentence - Use both positive and negative labels. How are these generated?

ICE: Supervised or Un-supervised?

1. Are the above two tasks supervised or un-supervised?

Data set!

English Wikipedia and book corpus documents!

BERT - Bi-directional Encoders from Transformers

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

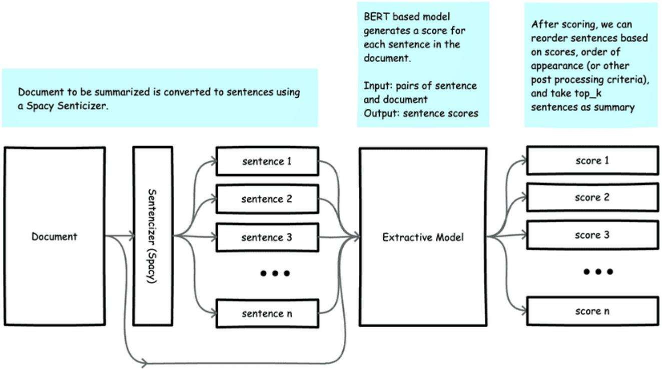
ICE #4

MLM

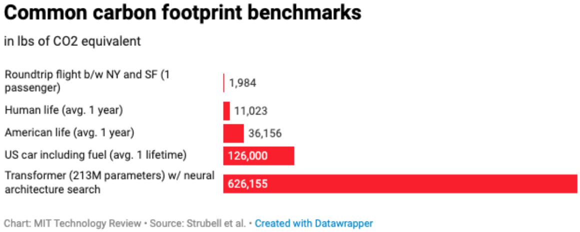
What's the real point of using masked language models (MLM) as compared to regular language models (LM). Select ones that apply!

1. MLMs are used to learn how words fit together in a sentence
2. MLMs incorporate context from both directions and hence lead to better embeddings and predictions as compared to LMs
3. MLMs are great for complicated language tasks such as QA where you need to understand the sentence as a whole to give an appropriate answer to a question

Document Summarization — BERT Based Extractive Model



Carbon Footprint of pre-training a Transformer Model!



Carbon Footprint of pre-training a Transformer Model!

	Date of original paper	Energy consumption (kWh)	Carbon footprint (lbs of CO2e)	Cloud compute cost (USD)
Transformer (65M parameters)	Jun, 2017	27	26	$41-$140
Transformer (213M parameters)	Jun, 2017	201	192	$289-$981
ELMo	Feb, 2018	275	262	$433-$1,472
BERT (110M parameters)	Oct, 2018	1,507	1,438	$3,751-$12,571
Transformer (213M parameters) w/ neural architecture search	Jan, 2019	656,347	626,155	$942,973-$3,201,722
GPT-2	Feb, 2019	-	-	$12,902-$43,008

Note: Because of a lack of power draw data on GPT-2's training hardware, the researchers weren't able to calculate its carbon footprint.

Table: MIT Technology Review • Source: Strubell et al. • Created with Datawrapper

Transformers Demo on Paraphrasing Task

1. **Pre-Training:** We don't do pre-training as that's expensive, requires lots of compute over many days, models have already been optimized and leaves a huge carbon footprint.

Transformers Demo on Paraphrasing Task

1. **Pre-Training:** We don't do pre-training as that's expensive, requires lots of compute over many days, models have already been optimized and leaves a huge carbon footprint.
2. **Fine-Tuning:** But we can **leverage** pre-training so we don't have to build a model that understands language from sractch. For instance BERT or ALBERT will do it for us. But needs to be fine-tuned to get good performance on our task of interest.

Transformers Demo on Paraphrasing Task

1. **Pre-Training:** We don't do pre-training as that's expensive, requires lots of compute over many days, models have already been optimized and leaves a huge carbon footprint.
2. **Fine-Tuning:** But we can **leverage** pre-training so we don't have to build a model that understands language from sractch. For instance BERT or ALBERT will do it for us. But needs to be fine-tuned to get good performance on our task of interest.
3. **Notebook Demo:** Let's take a look at how fine-tuning can be done using Hugging Face Libraries.

Additional Slides

Breakouts Time #1

Auto-complete — 5 mins

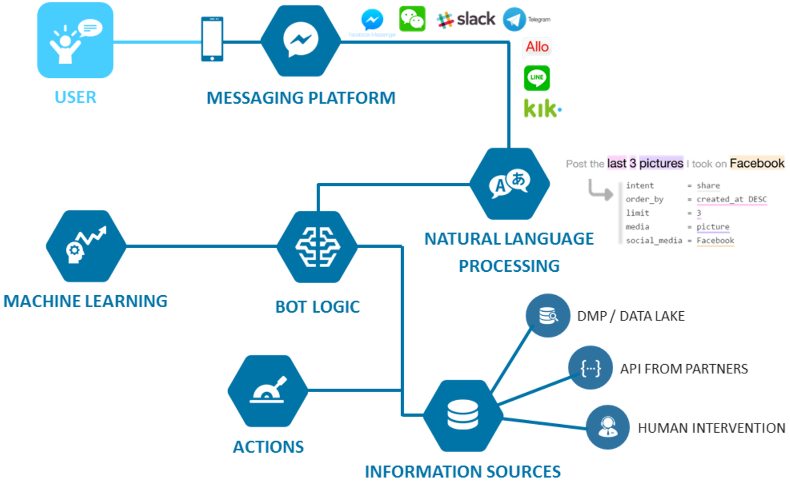
Let's say you are tasked with building an in-email auto-completion application, which can help complete partial sentences into full sentences through suggestions (auto-complete). How would you use what we have learned so far to model this? What architecture would you use? What would be your data? And what are some pitfalls or pain-points your model should address?

BERT - Bi-directional Encoders from Transformers

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

Table 4: SWAG Dev and Test accuracies. †Human performance is measured with 100 samples, as reported in the SWAG paper.

Chat Bots



USER
MESSAGING PLATFORM
slack
Telegram
Allo
kik·
Post the last 3 pictures I took on Facebook
intent = share
order_by = created_at DESC
limit = 3
media = picture
social_media = Facebook
NATURAL LANGUAGE PROCESSING
MACHINE LEARNING
BOT LOGIC
DMP / DATA LAKE
API FROM PARTNERS
HUMAN INTERVENTION
ACTIONS
INFORMATION SOURCES

Breakouts Time #2

Retrieving Tables with Chat bots — 7 mins

You are building a chat-bot product at your company where queries come in from customers that own data in your company's cloud service. Your chat-bot responds retrieves the right table or combination of tables (through merge/filter operations) that contains this information or returns back with follow up questions to get more precise information or get back with a "Sorry, I don't have that information" response. How would you go about building a chat-bot like this? What data would you use? What ML models would you use, would it be supervised or un-supervised learning? What would be your evaluation metric? How would you test if your chat bot is accurate in its responses?

Attention Motivation

First Attention Models

Reference paper

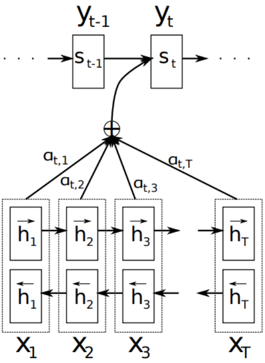
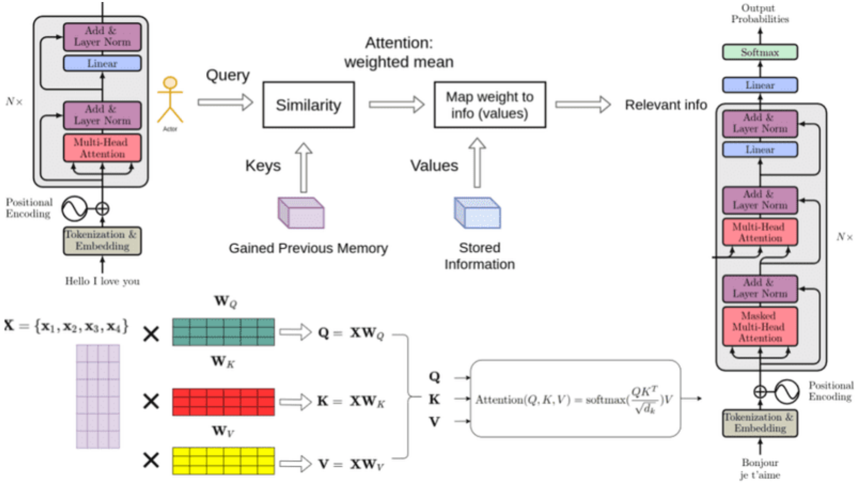


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

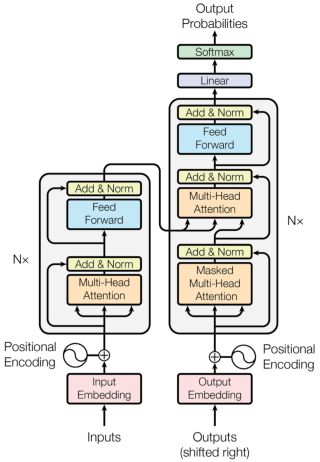
Transformers Architecture



Transformers Architecture

Transformer

Reference: Attention is all you need!



Transformer

Reference: Attention is all you need!

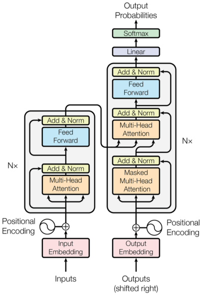


Figure 1: The Transformer - model architecture.

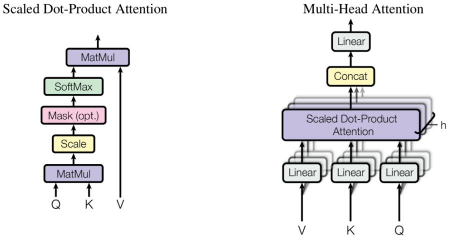
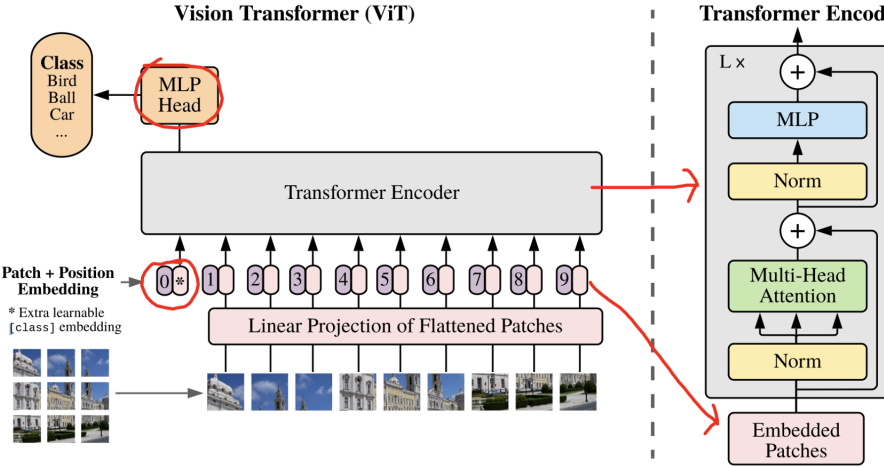


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

Transformers Architecture



Retrieving Tables from queries

Context

Many a times, we have a Natural Language Query - E.g. “Which quarter in the past 5 years had the most amount of sales for fashion products”. From this natural language query, we want to retrieve a data table that is perhaps the most similar to the query and helps answer the query.

Retrieving Tables from queries

Context

Many a times, we have a Natural Language Query - E.g. "Which quarter in the past 5 years had the most amount of sales for fashion products". From this natural language query, we want to retrieve a data table that is perhaps the most similar to the query and helps answer the query.

SQL queries vs Natural Language queries

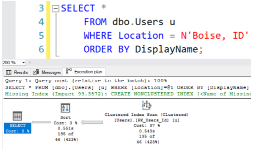


Table2Vec

Region	Release Date	Label	Release Format
United Kingdom	22 September 2008	Super Records	DVD
Ireland	[illegible]	cords	DVD
Japan	[illegible]	rax	DVD
Argentina	18 May 2009	EMI Music	Digital Download
Singapore	12 June 2009	Warner Music	DVD
Spain	1 December 2009	EMI Music Spain	Digital Download

pgTitle: Radio:Active
secondTitle: Release history
caption: Release history

Table2Vec

Embedding a Table?

1. Identify key entities in a table - E.g. headers and key words
2. Approach 1: Take a weighted average of these entity embeddings and call it the Table embedding
3. Approach 2: Pass the key entities in the table through a sequence model and generate a Table embedding.
4. Other approaches?

Query to a Table

Given a Natural Language query, how could you fuzzy match tables to a query?

1. Get a query embedding

Query to a Table

Given a Natural Language query, how could you fuzzy match tables to a query?

1. Get a query embedding
2. Get a table embedding

Query to a Table

Given a Natural Language query, how could you fuzzy match tables to a query?

1. Get a query embedding
2. Get a table embedding
3. Use an appropriate metric to do the matching!

ICE #5

What similarity metric would be appropriate to match a query with a table, given embeddings for both that are constructed out of word/entity embeddings?

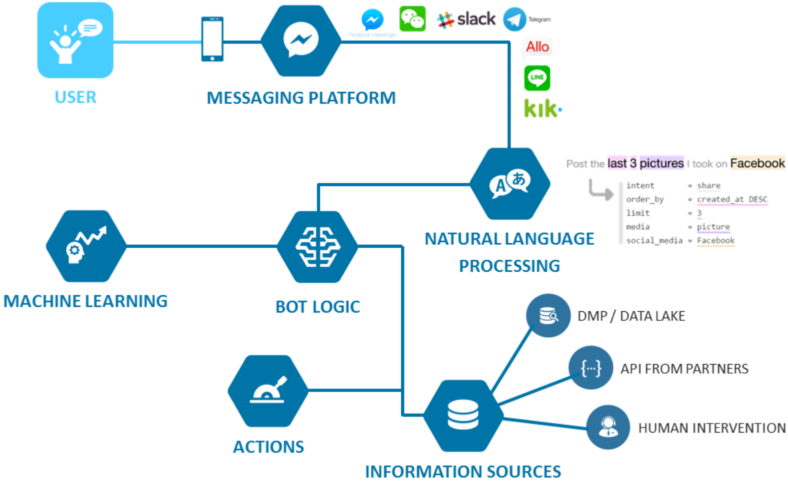
1. Jaccard Similarity
2. Ranking Similarity
3. Cosine Similarity
4. Sentence Similarity

ICE #6

Let's say we want to automatically convert a **Natural Language Query** to a **SQL** query. E.g. "Which quarter in the past 5 years had the most amount of sales for fashion products" to "SELECT ... FROM ... WHERE ..." What kind of deep learning architecture would support this problem?

1. Siamese Network
2. LSTM to LSTM sequence model
3. BERT model
4. Feed Forward Neural Network

Chat Bots



USER
MESSAGING PLATFORM
slack
Telegram
Allo
LINE
kik
Post the last 3 pictures I took on Facebook
intent = share
order_by = created_at DESC
limit = 3
media = picture
social_media = Facebook
NATURAL LANGUAGE PROCESSING
MACHINE LEARNING
BOT LOGIC
DMP / DATA LAKE
API FROM PARTNERS
HUMAN INTERVENTION
ACTIONS
INFORMATION SOURCES

Identifying bad actors from social media messages

Context

When messages on social media can spew hate or be inappropriate - Can a model be learned to classify them as inappropriate? E.g.

1. "You are f**** annoying me right now."

Identifying bad actors from social media messages

Context

When messages on social media can spew hate or be inappropriate - Can a model be learned to classify them as inappropriate? E.g.

1. "You are f**** annoying me right now."
2. "If you don't follow up on what we discussed, then things may not look so good for you."

Breakouts Time #3

Identifying inappropriate speech (7 mins)

Think of a simple baseline model that can help you identify a message/sentence on social media as inappropriate. When would this baseline model work? When would it fail? What deep learning architecture can help you fix the baseline model? What data would you use for your model? How would you gather the data for training? What do the inputs and labels look like? What are some evaluation metrics that can be used to measure the success of your models?

Extra Slides

Breakouts Time 1

5 mins

Discuss in your groups what are some real-world applications of any or many of the Auto Encoder Architectures we discussed so far you can think of in your area of work or in a standard context e.g. images.