CSCE 580: Introduction to AI
CSCE 581: Trusted AI

Lecture 21 & 22: Text, Language Models

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

7TH NOV & 9TH NOV, 2023

Carolinian Creed: "I will practice personal and academic integrity."

Credits: **Copyrights of all material reused acknowledged**

Organization of Lectures 21 & 22

- Introduction Segment
  - Recap of Lectures 19 and 20
- Main Segment
  - Text Processing
  - Language Models (LMs)
  - Learning for LMs with NN
  - Large LMs
- Concluding Segment
  - Course Project Discussion
  - About Next Lecture – Lecture 23
  - Ask me anything

Introduction Section

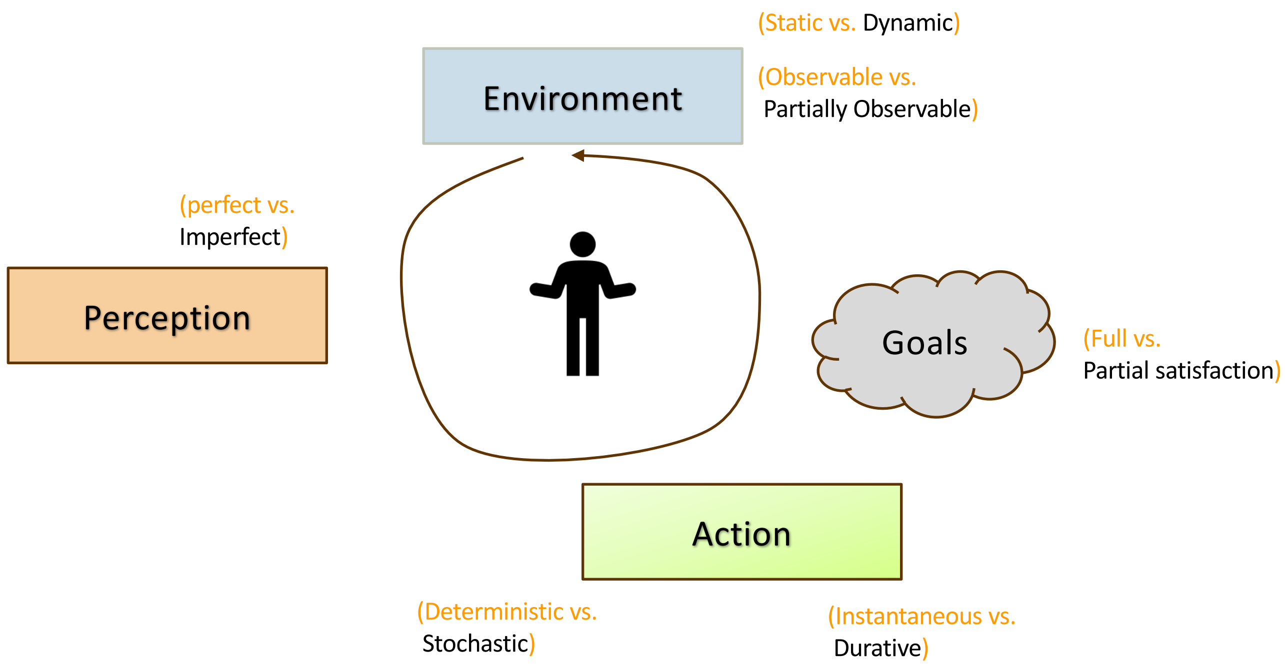
Recap of Lecture 19 and 20

- Topic discussed
  - Neural Networks
  - Deep Learning
  - Adversarial attacks
  - Trust Issues

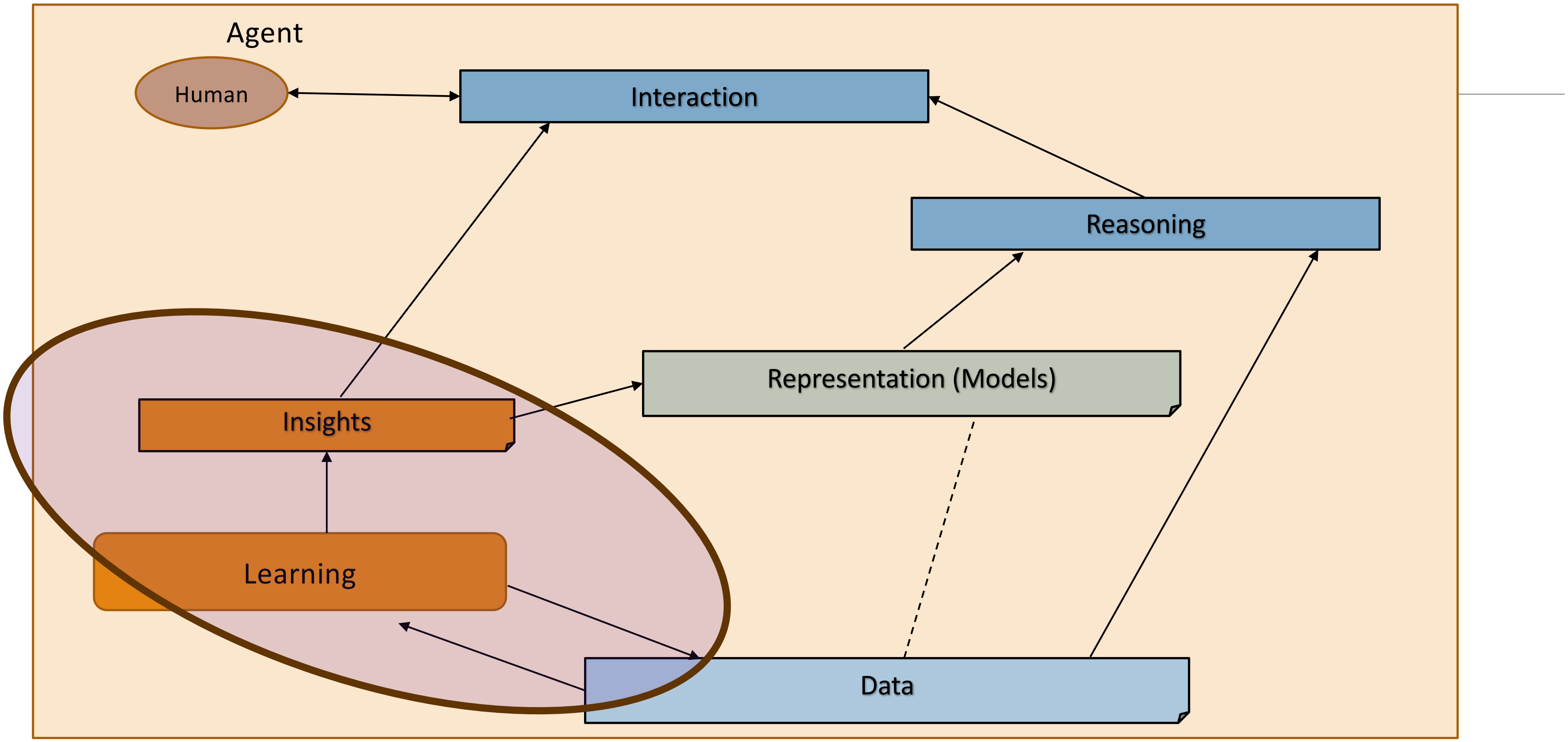
Graduate Paper Presentation

- Papers between 2021-2023 (last 3 years)
- At top AI venues: AAAI, Neurips, IJCAI, ICML, ICLR, or discuss with instructor
- Guideline on presentation
  - Summary of the paper
  - Critique (+ves/ -ves)
  - Relevance to your and anyone else's project in the class

Intelligent Agent Model



Relationship Between Main AI Topics



Where We Are in the Course

CSCE 580/ 581 – In This Course

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 4-5: Search, Heuristics - Decision Making
- Week 6: Constraints, Optimization – Decision Making
- Week 7: Classical Machine Learning – Decision Making, Explanation
- Week 8: Machine Learning - Classification
- Week 9: Machine Learning - Classification – Trust Issues and Mitigation Methods
- Topic 10: Learning neural network, deep learning, Adversarial attacks
- Week 11: Large Language Models – Representation, Issues
- Topic 12: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 13: Planning, Reinforcement Learning – Sequential decision making
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

Main Section

Credit: Retrieved from internet

Text Processing

Common Textual Data Processing Steps for ML

- Input: strings / documents/ corpus
- Processing steps (task dependent / optional - *)
  - Parsing
  - Word pre-processing
    - Tokenization – getting tokens for processing
    - Normalization* - making into canonical form
    - Case folding* – handling cases
    - Lemmatization* – handling variants (shallow)
    - Stemming* – handling variants (deep)
  - Semantic parsing – representations for reasoning with meaning *
  - Embedding – creating vector representation*

CSCE 771 goes into details

Common NLP Tasks

- Extracting entities [Entity Extraction]
- Finding sentiment [Sentiment Analysis]
- Generating a summary [Text Summarization]
- Translating to a different language [Machine translation]
- Natural Language Interface to Databases [NLI]
- Natural Language Generation [NLG]

CSCE 771 goes into details

Language Models (LMs)

Language Model

Problem:
Given a sentence fragment, predict what word(s) come next

Language Model:
estimate probability of substrings of a sentence

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) = \frac{P(w_1, w_2, \ldots, w_{i-1}, w_i)}{P(w_1, w_2, \ldots, w_{i-1})}$$

Applications:
- Spelling correction
- speech recognition
- machine translation,
- …

Bigram approximation

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) \approx \frac{P(w_{i-1}, w_i)}{P(w_{i-1})}$$

From Jurafsky & Martin

Language Model

Markovify library
https://github.com/jsvine/markovify

Language Model:
estimate probability of substrings of a sentence

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) = \frac{P(w_1, w_2, \ldots, w_{i-1}, w_i)}{P(w_1, w_2, \ldots, w_{i-1})}$$

See code samples with Markovify library on Github

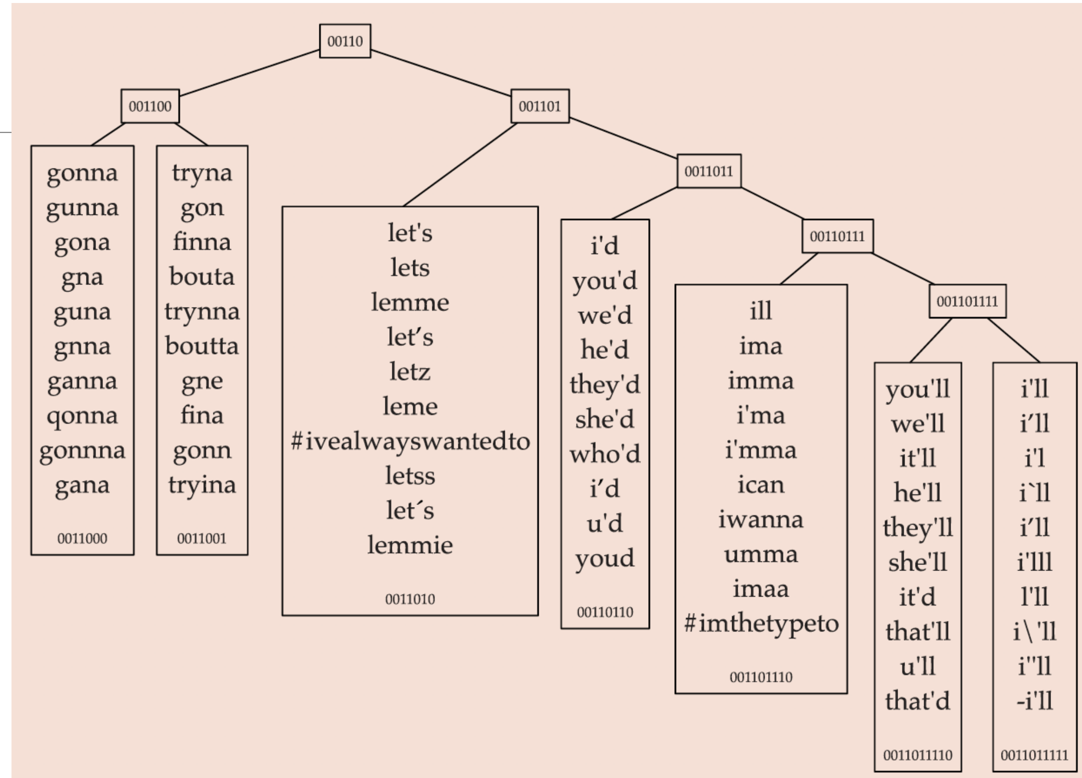
- *Prepare data – two datasets shown*
- *Try generator:*
    - *https://github.com/biplav-s/course-nl/blob/master/l7-language/code/TryMarkovifyLangModel.ipynb*

Contextual Word Embeddings

- Words as discrete
- Words with distributional assumptions:
  - Context: given a word, its nearby words or sequences of words
  - Words used in similar ways are likely to have related meanings; i.e., words used in the same (similar) context have related meanings
    - No claim about meaning except relative similarity v/s dis-similarity of words

Contextual Representation by Clustering

- Cluster words by context
- Compare with words in a manually-created taxonomy, e.g., Wordnet



The 10 most frequent words in clusters in the section of the hierarchy with prefix bit string 00110.

Owoputi, O., O’Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N.A. Improved part-ofspeech tagging for online conversational text with word clusters. In Proceedings of 2013 NAACL.

Credit:
Contextual Word Representations: Putting Words into Computers”, by Noah Smith, CACM June 2020

Contextual Representation by Dimensionality Reduction

- Creating word vectors in which each dimension corresponds to the frequency the word type occurred in some context.
- Strategy 1: select contexts
  - Examples
    - Custom methods
    - TF-IDF
  - Approach
    - Use words
      - Words in the neighborhood
      - Words of specific types
    - Build vectors
    - Use vector operations to derive meaning

context words	v(astronomers)	v(bodies)	v(objects)
't			1
,		2	1
.	1		1
1			1
And			1
Belt			1
But	1		
Given			1
Kuiper			1
So	1		
and		1	
are		2	1
between			1
beyond		1	
can			1
contains		1	
from	1		
hypothetical			1
ice		1	
including		1	
is	1		
larger		1	
now	1		
of	1		

$$\text{cosine\_similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}$$

	astronomers	bodies	objects
astronomers	$\frac{14}{\sqrt{14} \cdot \sqrt{14}} = 1$	$\frac{0}{\sqrt{24} \cdot \sqrt{14}} = 0$	$\frac{1+1}{\sqrt{14} \cdot \sqrt{16}} \approx 0.134$
bodies		$\frac{24}{\sqrt{24} \cdot \sqrt{24}} = 1$	$\frac{2+2+2}{\sqrt{24} \cdot \sqrt{16}} \approx 0.306$
objects			$\frac{16}{\sqrt{16} \cdot \sqrt{16}} = 1$

Bodies and **objects** are **most similar** (0.306) than
- **Bodies** and **astronomers** (0)
- **Objects** and **astronomers** (0.134)

Credit:
Contextual Word Representations: Putting Words into Computers", by Noah Smith, CACM June 2020

TF-IDF based Word Representation -1

- Given N documents

- **Term frequency (TF):** for term (word) t in document d = tf(t, d)

Variants to reduce bias due to document length

Sources:
(a) sci-kit documentation
(b) Wikipedia: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

Variants of term frequency (tf) weight

weighting scheme	tf weight
binary	$0, 1$
raw count	$f_{t,d}$
term frequency	$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$
log normalization	$\log(1 + f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K) \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

TF-IDF based Word Representation -2

- Given N documents
- Term frequency (TF): for term (word) t in document d = tf(t, d)
- **Inverse document frequency IDF**(t)

  = log [N / **DF**(t)] + 1

DF(t) = **document frequency**, the number of documents in the document set that contain the term t.

- **TF-IDF**(t, d) = TF(t, d) * IDF(t),

Variants of inverse document frequency (idf) weight

weighting scheme	idf weight ($n_t = \lvert\{d \in D : t \in d\}\rvert$)
unary	1
inverse document frequency	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
inverse document frequency smooth	$\log \left(\frac{N}{1 + n_t} \right) + 1$
inverse document frequency max	$\log \left(\frac{\max_{\{t' \in d\}} n_{t'}}{1 + n_t} \right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

Sources:
(a) sci-kit documentation
(b) Wikipedia: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

TF-IDF Example Calculation

See sample code on GitHub:
https://github.com/biplav-s/course-nl-f22/sample-code/l5-wordrepresent/Word%20Representations%20-%20Vectors.ipynb

Contextual Representation by Dimensionality Reduction - 1

- Strategy 2: learn contexts from documents. Vector size is given as input
- Train a neural network to learn vector representation
  - value placed in each dimension of each word type's vector is a parameter that will be optimized
  - Selection of parameter values is done using iterative algorithms / gradient descent
  - **Hope** is that ***different senses*** in which a word is used will be captured through the learning procedure as long as the dataset is large enough to represent all senses. Paper quotes: 30 meanings of **get**
- **Optionally**: Sometime task specific inputs are given during pre-processing, processing or post-processing

Disadvantage: individual dimensions are no longer interpretable

Contextual Representation by Dimensionality Reduction -2

- Strategy 2: learn contexts from documents. Vector size is given as input

Sometime task specific inputs are given during pre-processing, processing or post-processing

- Pre-processing
  - Vector initialization by pre-training. Called **finetuning**
- Processing
  - **Knowledge-infusion** (emerging area)
- Post-processing
  - Adjust output vectors so that word types that are related in reference taxonomy (like WordNet) are closer to each other in vector space. Called **retrofitting**.

Credit:
Contextual Word Representations: Putting Words into Computers", by Noah Smith, CACM June 2020

Where are We

- Learning representation
  - Approach 1: count-based
    - Creating word vectors in which each dimension corresponds to the frequency the word type occurred in some context.
    - Example: TF-IDF
  - Approach 2: learning-based
    - learn contexts from documents. Vector size is given as input
    - Examples: Word2Vec, Glove, RNN/LSTM (arc), Transformers

Reading

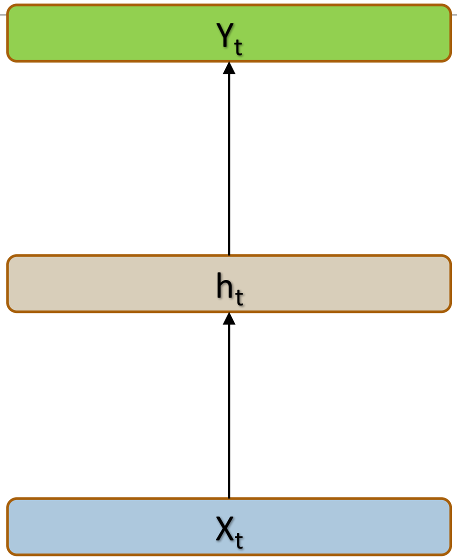
- Contextual Word Representations: Putting Words into Computers", by Noah Smith, CACM June 2020
- Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep Learning--based Text Classification: A Comprehensive Review. ACM Comput. Surv. 54, 3, Article 62 (April 2022), 40 pages. https://doi.org/10.1145/3439726
- Hang Li, Language Models: Past, Present, and Future, Communications of the ACM, July 2022, Vol. 65 No. 7, Pages 56-63 10.1145/3490443

Learning for LMs with NN

Recall: (Feed forward) NN

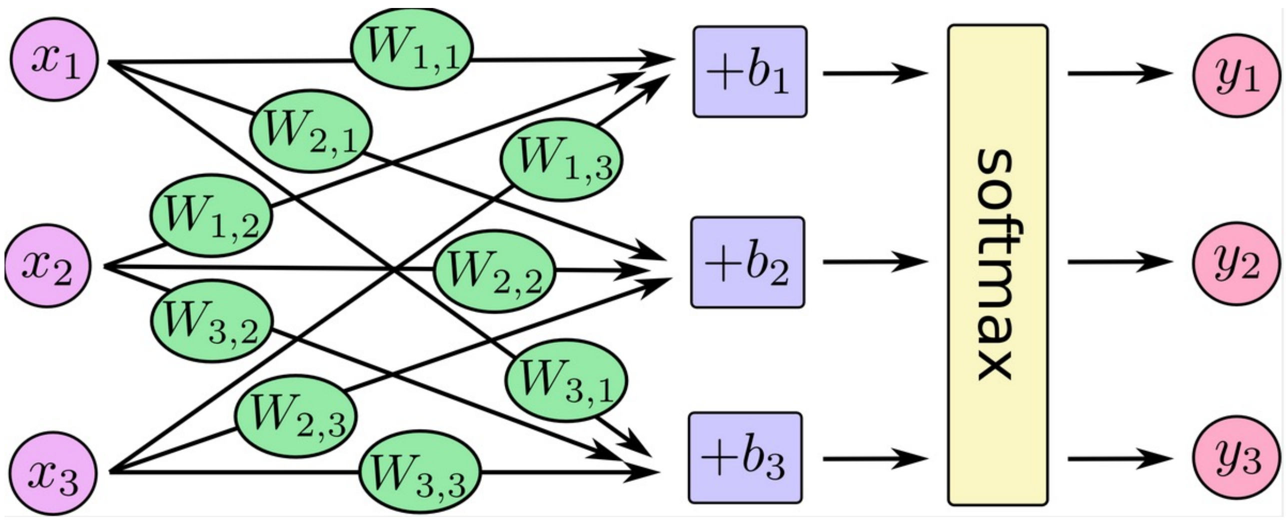
Propagation

$$f(X_j) = X_j W + b$$



Intuitive Description: https://jalammar.github.io/visual-interactive-guide-basics-neural-networks/, https://jalammar.github.io/feedforward-neural-networks-visual-interactive

Using (Feed forward) NN



Softmax

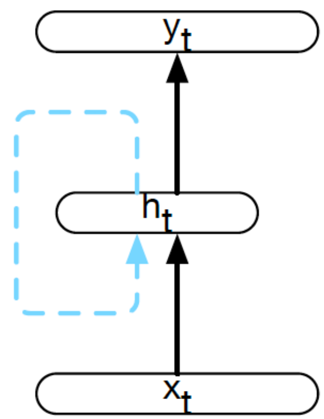
$$f(x) = \frac{1}{1+e^{-x}}$$

Source; see also : https://jalammar.github.io/visual-interactive-guide-basics-neural-networks/, https://jalammar.github.io/feedforward-neural-networks-visual-interactive

RNN - Recurrent Neural Networks

- **Recurrence**: A *recurrence* relation is an equation that defines a sequence based on a rule that gives the next term as a *function* of the previous term(s).
[https://mathinsight.org/definition/recurrence_relation]

- Simple Recurrent NN or Elman Network

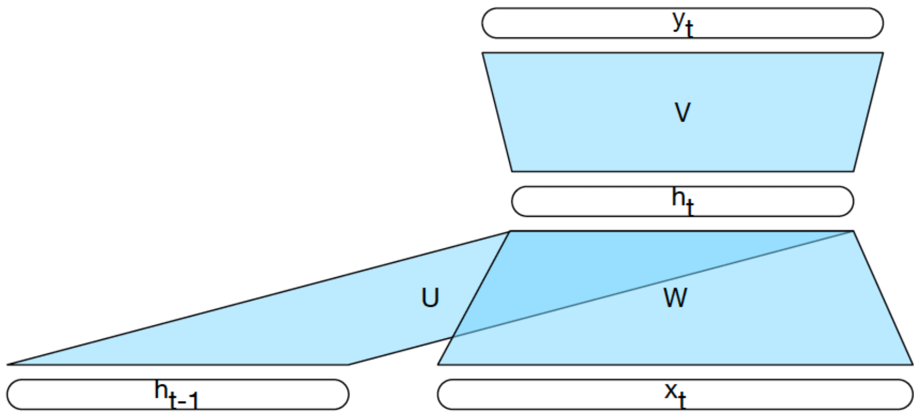


Source: Jurafsky and Martin

RNN

Recurrence unrolled

U, W, V are Weights to be learned



$$h_t = g(Uh_{t-1} + Wx_t)$$
$$y_t = f(Vh_t)$$

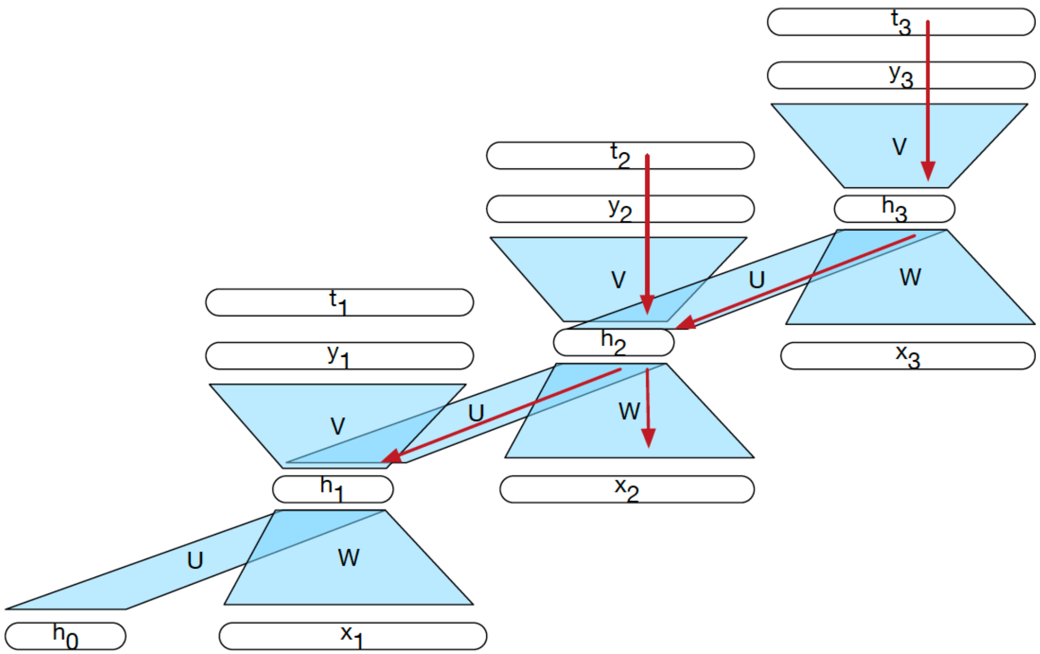
$$y_t = softmax(Vh_t)$$

Source: Jurafsky and Martin

RNN Backpropagation of Errors

Recurrence unrolled

U, W, V are Weights to be learned



Source: Jurafsky and Martin

RNN-based Language Model

- Based on characters or words
- At each step (i.e., character or word)
  - the network retrieves a word embedding for the current word as input
  - combines it with the hidden layer from the previous step to
    - compute a new hidden layer
    - generate an output layer which is passed through a softmax layer to generate a probability distribution over the entire vocabulary.

$$
\begin{aligned}
P(w_n|w_1^{n-1}) &= y_n \\
&= softmax(Vh_n)
\end{aligned}
$$

Prob. of a word

$$
\begin{aligned}
P(w_1^n) &= \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \\
&= \prod_{k=1}^{n} y_k
\end{aligned}
$$

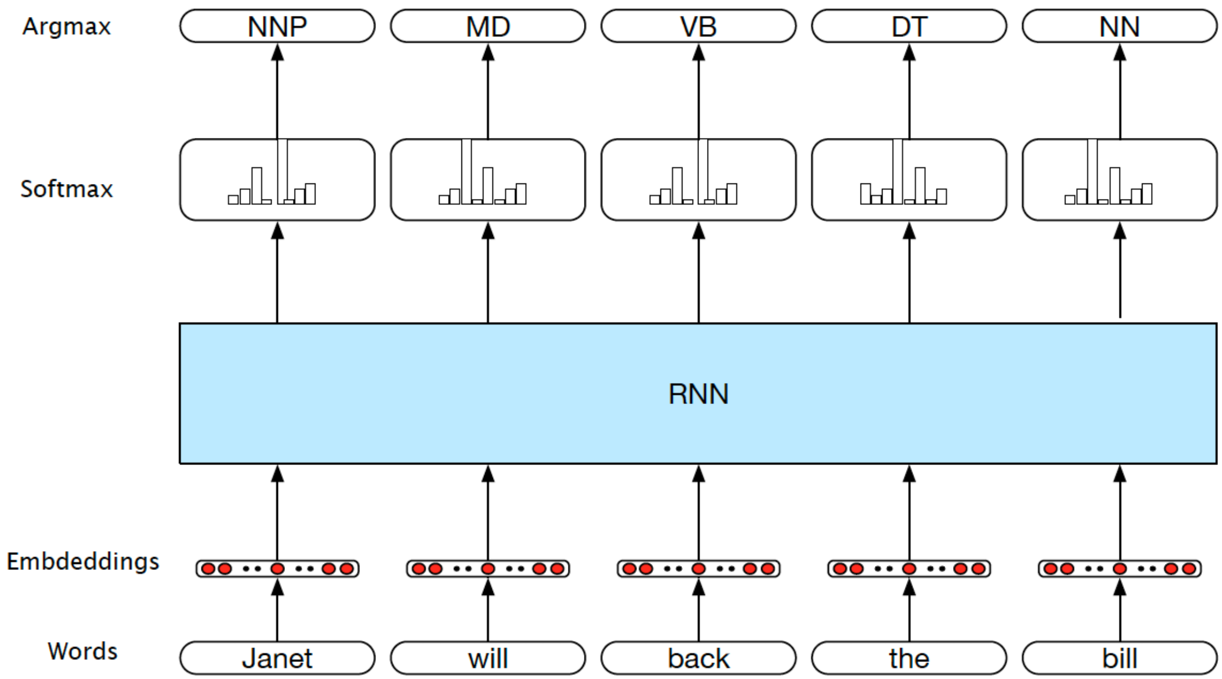
Prob. of a sequence

Source: Jurafsky and Martin

RNN Discussion

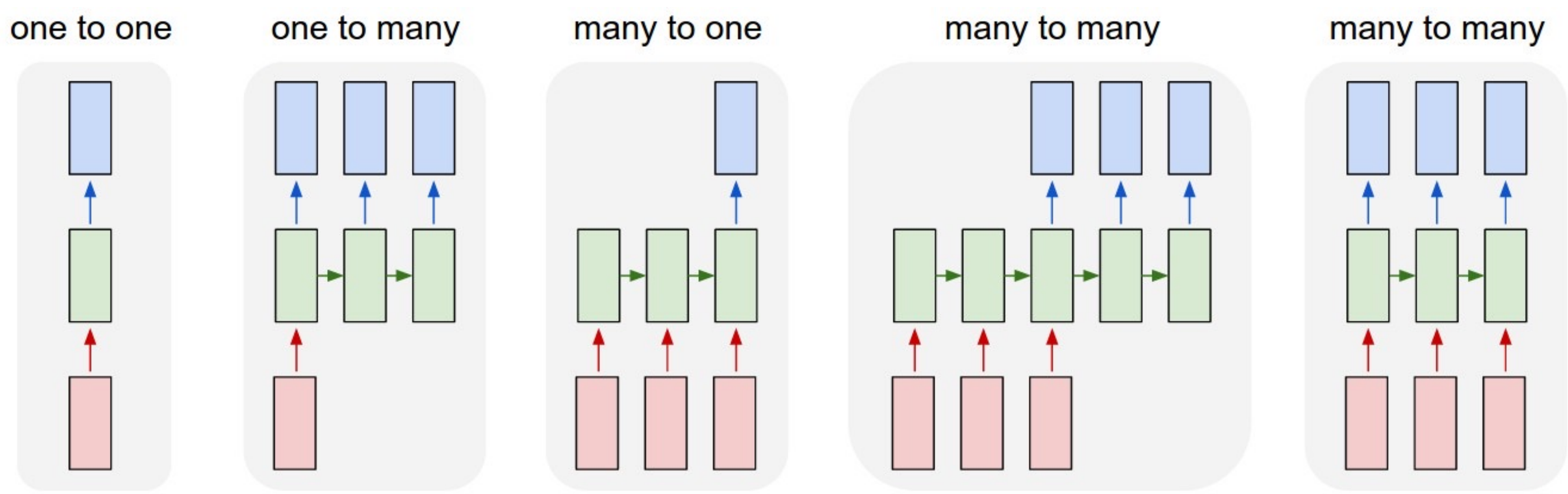
- Language model
  - Not dependent on N-gram boundaries
  - Whole sequence is the context
- Program generation
  - Complexity is Turing-complete
  - In practical terms: On the Practical Computational Power of Finite Precision RNNs for Language Recognition, Gail Weiss, Yoav Goldberg, Eran Yahav, ACL 2018, https://www.aclweb.org/anthology/P18-2117/

RNN Usage Example: Sentence Labeling



Source: Jurafsky and Martin

RNN - Many Applications



Language model

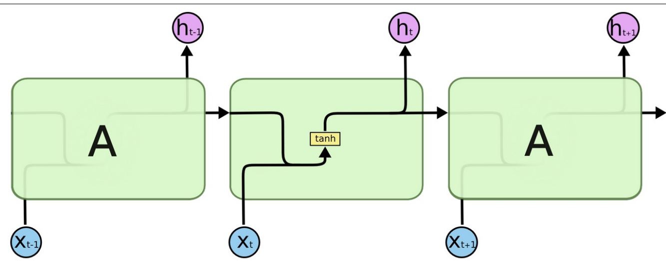
Caption generation

Sentiment detection

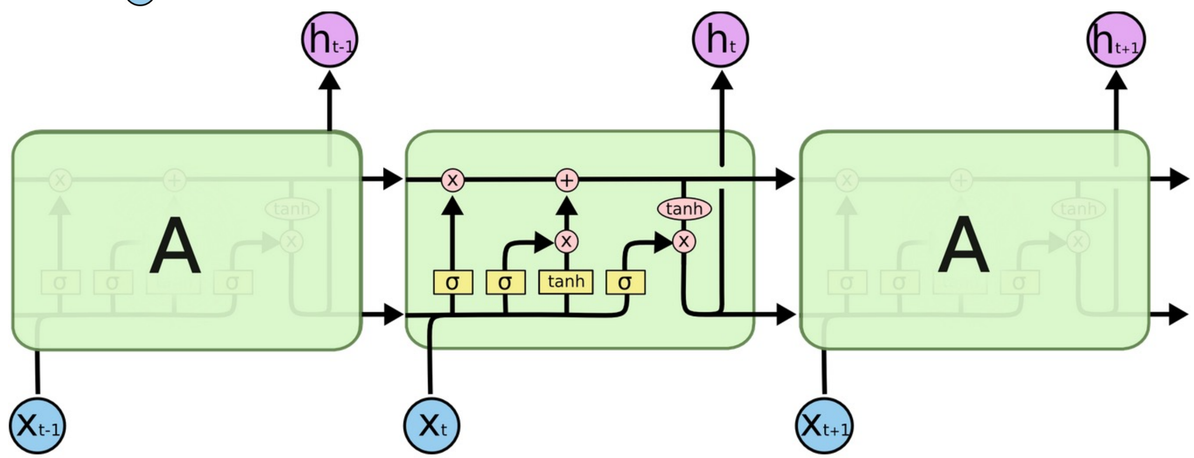
Machine translations

Source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

RNN and LSTM - Long Short Term Memory



RNN



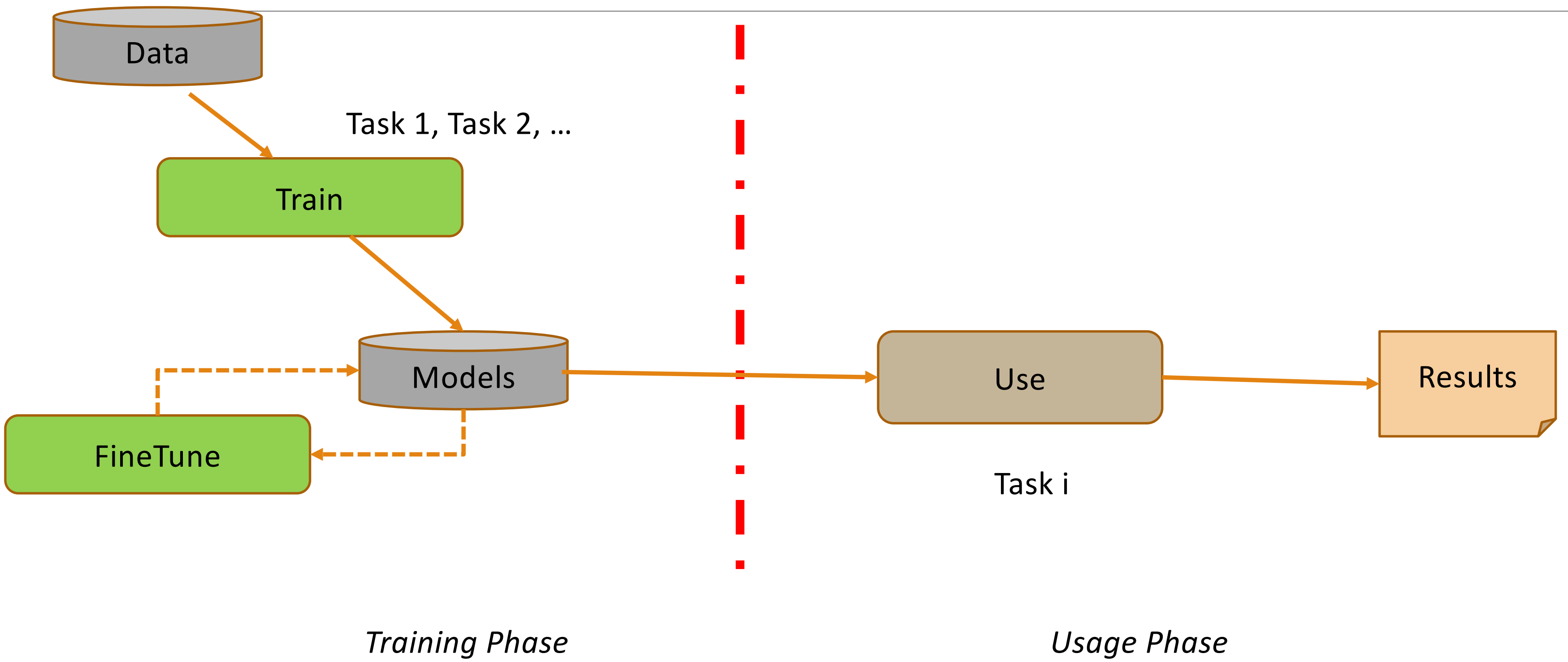
LSTM

To learn long term relationships; has 4 NNs

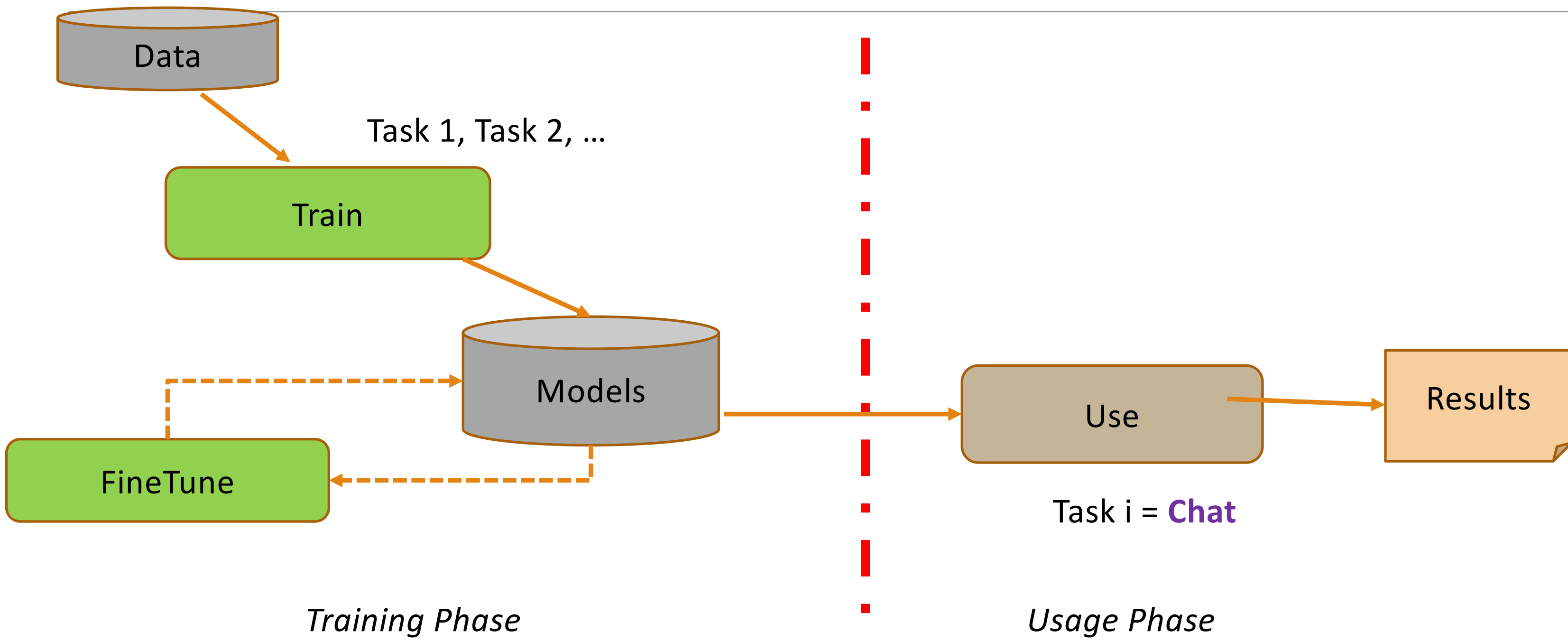
Source and details: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Large LMs (LLMs)

Large Language Models (LLMs) Basics



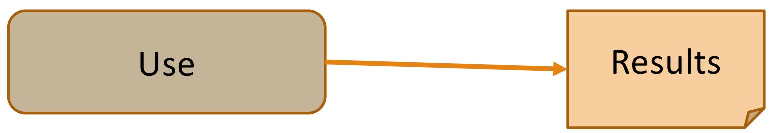
ChatGPT: Large Language Models (LLMs) based Chatbot



Another “Turning Point” Moment In Technology

Raised interest about Chatbots among public

- Excitement about new use-cases
- Concerns about social impact – cheating, jobs, misinformation
- Renewed calls for regulations



Task i = **Generally speaking: content generation – text, image, video, audio, …**

Usage Phase

Transformer

- RNN/ LSTM with
  - Attention
    - attention layer can access all previous states and weighs them according to some learned measure of relevancy to the current token, providing sharper information about far-away relevant tokens
    - **Query** vector, **Key** vector, and **Value** vectors introduced during encoding and decoding phase
  - Parallelization of learning
  - See Dr. Amitava Das's slide for Attention/ BERT video
    - https://prezi.com/view/amx5hBo8UhMOn1rPyJ02/

Source and details: https://en.wikipedia.org/wiki/Transformer_(machine_learning_model), http://jalammar.github.io/illustrated-transformer/

BERT - **B**idirectional **E**ncoder **R**epresentations from **T**ransformers

Learns with two tasks

- Predicting missing words in sentences
  - mask out 15% of the words in the input, predict the masked words.
- Given two sentences A and B, is B the actual next sentence that comes after A, or just a random sentence from the corpus?

Input: the man went to the [MASK1] . he bought a [MASK2] of milk.
Labels: [MASK1] = store; [MASK2] = gallon

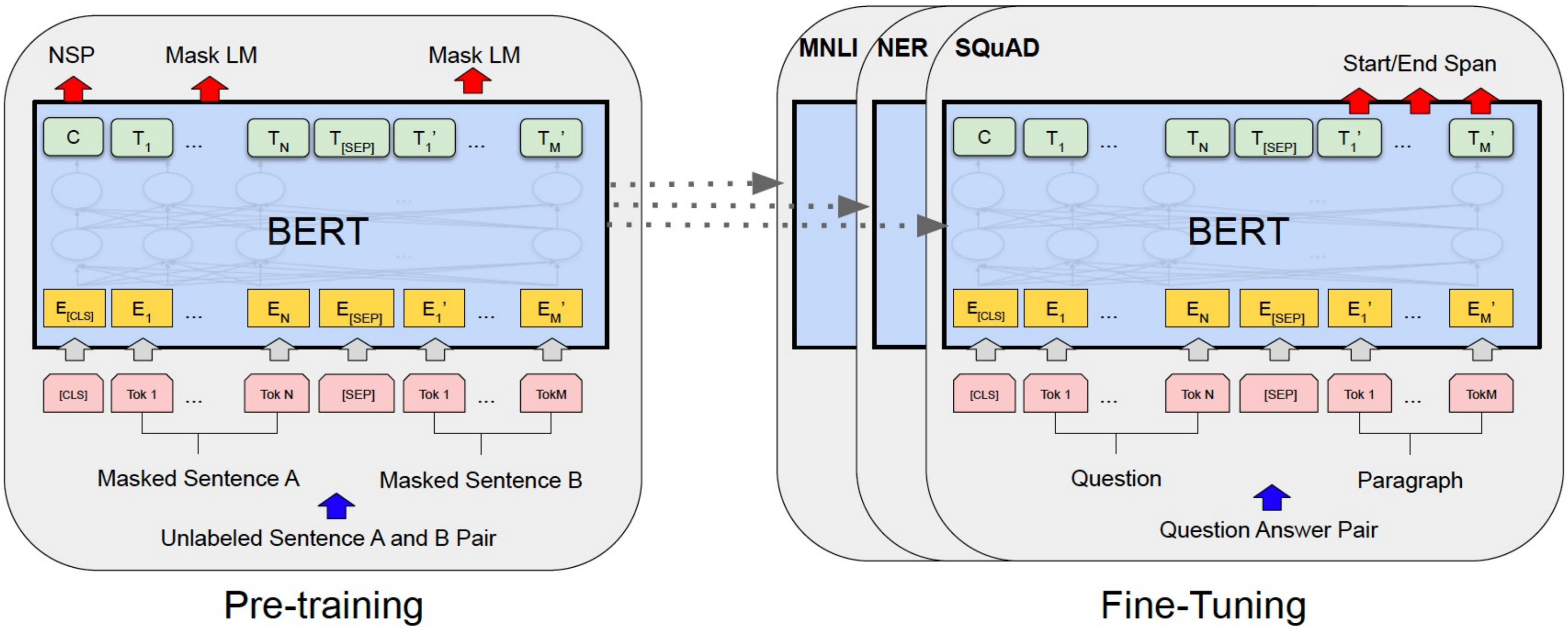
Sentence A: the man went to the store .
Sentence B: he bought a gallon of milk .
Label: IsNextSentence

Sentence A: the man went to the store .
Sentence B: penguins are flightless .
Label: NotNextSentence

(12-layer to 24-layer Transformer)
on (Wikipedia + [BookCorpus]())

Credit and details: https://github.com/google-research/bert

BERT: Before and During Usage



Credit and details: **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, 2018

Using BERT in Practice – Huggingface Libraries

- Transformers – https://github.com/huggingface/transformers
- APIs to download and use pre-trained models, fine-tune them on own datasets and tasks
  - Code Sample

```
# Loading BERT
model_class, tokenizer_class, pretrained_weights =  (ppb.DistilBertModel, ppb.DistilBertTokenizer, 'distilbert-base-uncased')

# Load pretrained model/tokenizer
tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
model = model_class.from_pretrained(pretrained_weights)
```

- Provides pretrained models in 100+ languages.
- Use with popular deep learning libraries, PyTorch and TensorFlow,
  - Possible to train / fine-tune models with one, and load it for inference with another

Using BERT in Practice – Huggingface Libraries

- DistilBERT
  - Details: https://medium.com/huggingface/distilbert-8cf3380435b5
  - Teacher-student learning, also called model distillation
    - Teacher: bert-base-uncased
    - Student: dstilBERT - BERT without *the token-type embeddings and the pooler* , and half the layers
  - **“DistilBERT,** has **about half** the total number of parameters of BERT base and retains 95% of BERT’s performances on the language understanding benchmark GLUE”

- Sample code of usage for sentiment classification: https://github.com/biplav-s/course-nl/blob/master/l12-langmodel/UsingLanguageModel.ipynb

Example Pre-Trained Models

1. ALBERT (from Google Research and the Toyota Technological Institute at Chicago) released with the paper ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, by Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut.
2. BART (from Facebook) released with the paper BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension by Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov and Luke Zettlemoyer.
3. BERT (from Google) released with the paper BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding by Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova.
4. BERT For Sequence Generation (from Google) released with the paper Leveraging Pre-trained Checkpoints for Sequence Generation Tasks by Sascha Rothe, Shashi Narayan, Aliaksei Severyn.
5. CamemBERT (from Inria/Facebook/Sorbonne) released with the paper CamemBERT: a Tasty French Language Model by Louis Martin*, Benjamin Muller*, Pedro Javier Ortiz Suárez*, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah and Benoît Sagot.
6. CTRL (from Salesforce) released with the paper CTRL: A Conditional Transformer Language Model for Controllable Generation by Nitish Shirish Keskar*, Bryan McCann*, Lav R. Varshney, Caiming Xiong and Richard Socher.
7. DeBERTa (from Microsoft Research) released with the paper DeBERTa: Decoding-enhanced BERT with Disentangled Attention by Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen.
8. DialoGPT (from Microsoft Research) released with the paper DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation by Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, Bill Dolan.
9. DistilBERT (from HuggingFace), released together with the paper DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter by Victor Sanh, Lysandre Debut and Thomas Wolf. The same method has been applied to compress GPT2 into DistilGPT2, RoBERTa into DistilRoBERTa, Multilingual BERT into DistilmBERT and a German version of DistilBERT.
10. DPR (from Facebook) released with the paper Dense Passage Retrieval for Open-Domain Question Answering by Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih.
11. ELECTRA (from Google Research/Stanford University) released with the paper ELECTRA: Pre-training text encoders as discriminators rather than generators by Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning.
12. FlauBERT (from CNRS) released with the paper FlauBERT: Unsupervised Language Model Pre-training for French by Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, Didier Schwab.
13. Funnel Transformer (from CMU/Google Brain) released with the paper Funnel-Transformer: Filtering out Sequential Redundancy for Efficient Language Processing by Zihang Dai, Guokun Lai, Yiming Yang, Quoc V. Le.
14. GPT (from OpenAI) released with the paper Improving Language Understanding by Generative Pre-Training by Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever.
15. GPT-2 (from OpenAI) released with the paper Language Models are Unsupervised Multitask Learners by Alec Radford*, Jeffrey Wu*, Rewon Child, David Luan, Dario Amodei** and Ilya Sutskever**.
16. LayoutLM (from Microsoft Research Asia) released with the paper LayoutLM: Pre-training of Text and Layout for Document Image Understanding by Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou.
17. Longformer (from AllenAI) released with the paper Longformer: The Long-Document Transformer by Iz Beltagy, Matthew E. Peters, Arman Cohan.
18. LXMERT (from UNC Chapel Hill) released with the paper LXMERT: Learning Cross-Modality Encoder Representations from Transformers for Open-Domain Question Answering by Hao Tan and Mohit Bansal.
19. MarianMT Machine translation models trained using OPUS data by Jörg Tiedemann. The Marian Framework is being developed by the Microsoft Translator Team.
20. MBart (from Facebook) released with the paper Multilingual Denoising Pre-training for Neural Machine Translation by Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, Luke Zettlemoyer.
21. MMBT (from Facebook), released together with the paper a Supervised Multimodal Bitransformers for Classifying Images and Text by Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Davide Testuggine.
22. Pegasus (from Google) released with the paper PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization> by Jingqing Zhang, Yao Zhao, Mohammad Saleh and Peter J. Liu.
23. Reformer (from Google Research) released with the paper Reformer: The Efficient Transformer by Nikita Kitaev, Łukasz Kaiser, Anselm Levskaya.
24. RoBERTa (from Facebook), released together with the paper a Robustly Optimized BERT Pretraining Approach by Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. ultilingual BERT into DistilmBERT and a German version of DistilBERT.
25. SqueezeBert released with the paper SqueezeBERT: What can computer vision teach NLP about efficient neural networks? by Forrest N. Iandola, Albert E. Shaw, Ravi Krishna, and Kurt W. Keutzer.
26. T5 (from Google AI) released with the paper Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer by Colin Raffel and Noam Shazeer and Adam Roberts and Katherine Lee and Sharan Narang and Michael Matena and Yanqi Zhou and Wei Li and Peter J. Liu.
27. Transformer-XL (from Google/CMU) released with the paper Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context by Zihang Dai*, Zhilin Yang*, Yiming Yang, Jaime Carbonell, Quoc V. Le, Ruslan Salakhutdinov.
28. XLM (from Facebook) released together with the paper Cross-lingual Language Model Pretraining by Guillaume Lample and Alexis Conneau.
29. XLM-RoBERTa (from Facebook AI), released together with the paper Unsupervised Cross-lingual Representation Learning at Scale by Alexis Conneau*, Kartikay Khandelwal*, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer and Veselin Stoyanov.
30. XLNet (from Google/CMU) released with the paper XLNet: Generalized Autoregressive Pretraining for Language Understanding by Zhilin Yang*, Zihang Dai*, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le.

https://github.com/huggingface/transformers

Preq-requisites for Understanding Advanced Language Models

- Advanced language models need pre-requisites to understand
  - BERT, Transformers, GPT-2, GPT-3, GPT-4, …

- Understand word representation
- Understand context representation
- Understand machine learning/ neural methods

Commentary: http://jalammar.github.io/illustrated-gpt2/

Course Project

Project Discussion: What Problem Fascinates You ?

- Data
  - Water
  - Finance
  - …
- Analytics
  - Search, Optimization, Learning, Planning, …
- Application
  - Building chatbot
- Users
  - Diverse demographics
  - Diverse abilities
  - Multiple human languages

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

Project Discussion: Dates and Deliverables

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
- Nov 30, 2023
  - Project report due
- Dec 5 / 7, 2023
  - In-class presentation

Skeleton: A Basic Chatbot

- Run in an infinite loop until the user wants to quit
- Handle any user response
  - User can quit by typing “Quit” or “quit” or just “q”
  - User can enter any other text and the program has to handle it. The program should write back what the user entered and say – “I do not know this information”.
- Handle known user query types   // Depends on your project
  - “Tell me about N-queens”, “What is N ?”
  - “Solve for N=4?”
  - “Why is this a solution? ”
- Handle chitchat // Support at least 5, extensible from a file
  - “Hi” => **“Hello”**
  - …
- ***Store session details in a file***

Illustrative Project

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Discussion: Illustration

1. Create a private Github repository called “CSCE58x-Fall2023-<studentname>-Repo”. Share with Instructor (biplav-s) and TA (kausik-l)
2. Create Google folder called “CSCE58x-Fall2023-<studentname>-SharedInfo”. Share with Instructor (prof.biplav@gmail.com) and TA (lakkarajukausik90@gmail.com)
3. Create a Google doc in your Google repo called “Project Plan” and have the following by next class (Sep 5, 2023)

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Illustration: N-Queens

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
    - Method 1: Random solution
    - Method 2: Search – BFS
    - Method 3: Search - …
  - Human interaction: Develop a basic chatbot (no AI) as outlined
  - Deliverable
    - Code structure in Github
      - ./data
      - ./code
      - ./docs
      - ./test
    - Presentation: Make sprint presentation on Oct 12, 2023

Reference: Project Rubric - NEW

- **Project report** – 60%
  - Project description: problem, related work, approach, evaluation – 40%
  - Working system demo/ video – 10%
    - Well organized Github with code (./data, ./code, ./docs, ./test) – 10%
- **Project presentation** – 40%
  - Evaluation by peers, instructor and TA
- **Bonus**
  - Instructor discretion – 10%
- **Penalty**
  - Lack of timeliness as per announced policy (right) - up to 30%

Milestones and **Penalties**

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
  - **Penalty: presentation not ready by Oct 10, 2023 [-10%]**
- Nov 30, 2023
  - Project report due
  - **Project report not ready by date [-10%]**
- Dec 5 / 7, 2023
  - In-class presentation
  - **Project presentations not ready by Dec 4, 2023 [-10%]**

Evaluation of Presentation

1. An online form will be available during presentation
2. During a presentation, three students will be assigned to review along with instructor and TA
3. They will enter following survey questions:
    1. Their name
    2. Presentation number
    3. How useful is the system – will you use it? [1-5 scale]
    4. How well have you understood the project from the presentation? [1-5 scale]
4. Top and bottom scores will be removed. Average of remaining three will be used for final presentation marks

Lecture 21 & 22: Summary

- We talked about
  - Text Processing
  - Language Models (LMs)
  - Learning for LMs with NN
  - Large LMs

Concluding Section

About Next Lecture – Lecture 23

Lecture 23-24: Decision Problems

- Making simple decisions
  - Maximum Expected Utility (MEU)
- Making complex decisions
  - Markov Decision Processes (MDPs)