CSCE 581: Introduction to Trusted AI

Lectures 19 and 20: Text Processing, LLMs

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

25TH AND 27TH MAR, 2025

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

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

Organization of Lectures 19, 20

- Introduction Section
  - Recap from Week 9 (Lectures 17 and 18)
  - Announcements and News
- Main Section
  - L19: AI - Unstructured (Text): Representation, Common NLP Tasks
  - L20: Natural Languages/ Language Models and their Impact on Text/ AI
- Concluding Section
  - About next week – Lectures 21, 22
  - Ask me anything

Introduction Section

Recap from Week 9 (Lectures 17, 18)

- We looked at
  - L17: Invited talk on Trust and Agentic AI
  - L18: Started with text processing / AI

•Trust, Human Focus and Agentic AI – invited talk
•Text processing, representation

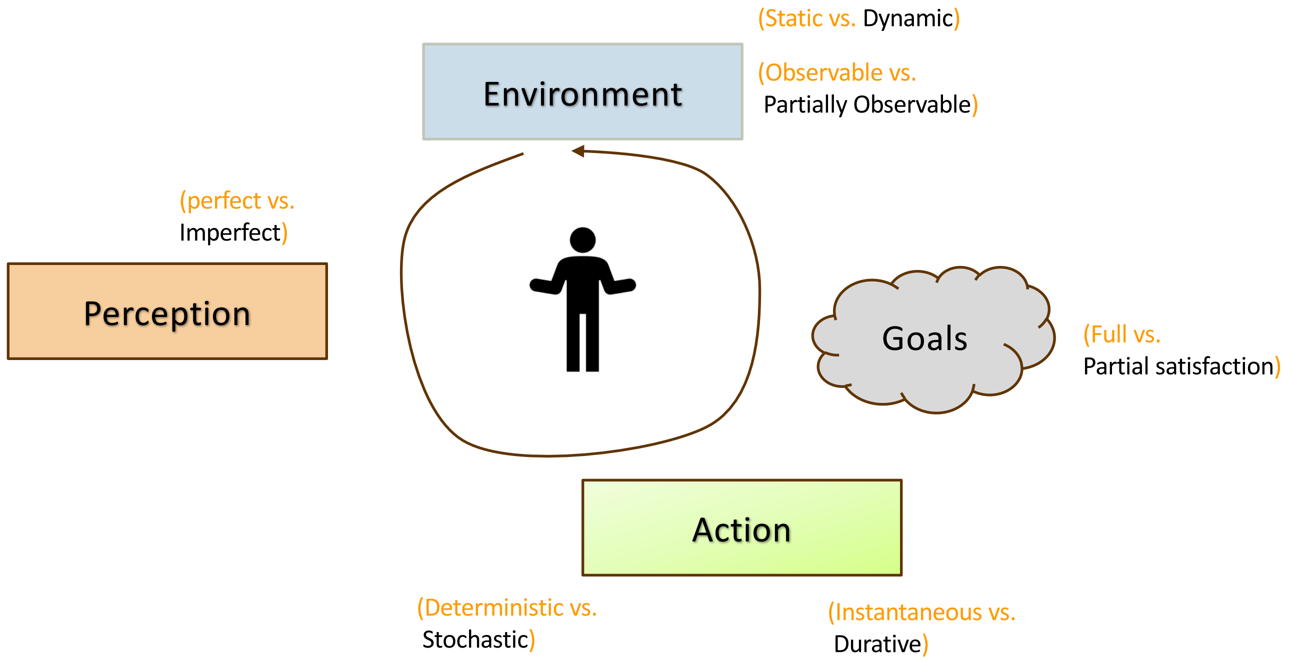
Announcement: Change to Student Assessment

A = [920-1000]
B+ = [870-919]
B = [820-869]
C+ = [770-819]
C = [720-769]
D+ = [670-719]
D = [600-669]
F = [0-599]

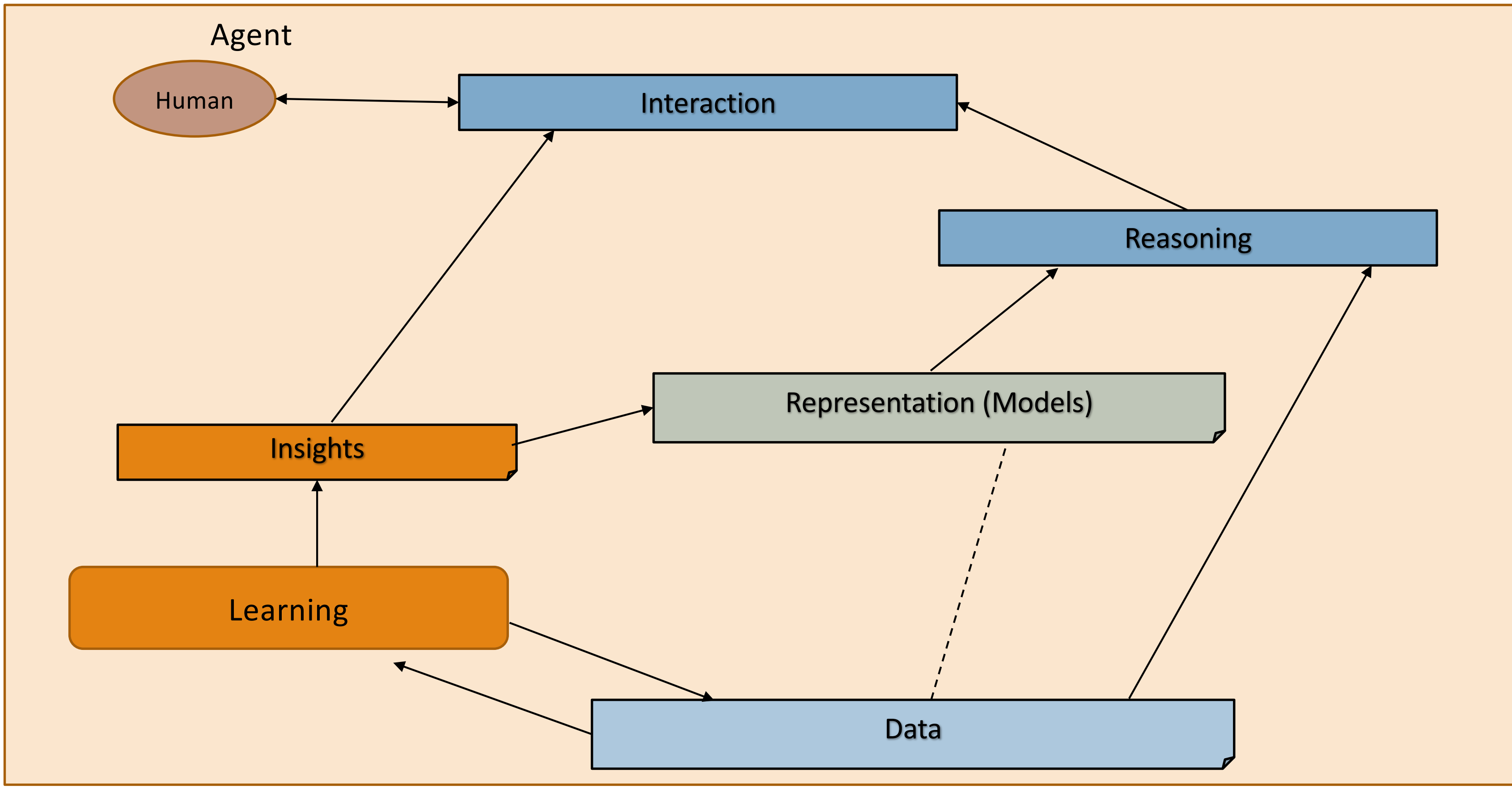
Tests	Undergrad	Grad
Course Project – report, in-class presentation	600	600
Quiz – 2 quizzes	200	200
Final Exam	200	100
Additional Final Exam – Paper summary, in-class presentation		100
Total	1000 points	1000 points

Change: 4 quizzes to 2; no best of 3

Intelligent Agent Model



Relationship Between Main AI Topics (Covered in Course)



High Level Semester Plan (Adapted, Approximate)

CSCE 581 –
- Week 1: Introduction
- Week 2: Background: AI - Common Methods
- Week 3: The Trust Problem
- Week 4: Machine Learning (Structured data) - Classification
- Week 5: Machine Learning (Structured data) - Classification – Trust Issues
- Week 6: Machine Learning (Structured data) – Classification – Mitigation Methods
- Week 7: Machine Learning (Structured data) – Classification – Explanation Methods
- Week 8: Machine Learning (Text data, **vision**) – Classification, **Large Language Models**
- Week 9: Machine Learning (Text data) - Classification – Trust Issues, LLMs
- Week 10: Machine Learning (Text data) – Classification – Mitigation Methods
- Week 11: Machine Learning (Text data) – Classification – Explanation Methods
- Week 12: Emerging Standards and Laws, **Real world applications**
- Week 13: Project presentations
- Week 14: Project presentations, Conclusion

Increased focus on LLMs and projects now

AI/ ML topics and with a focus on fairness, explanation, Data privacy, reliability

Main Segment

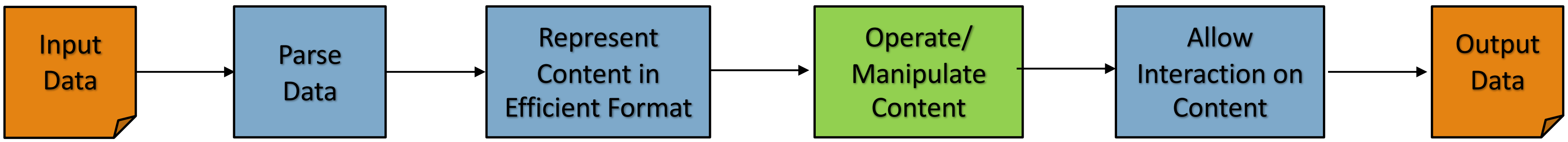
Common NLP Tasks

- 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]

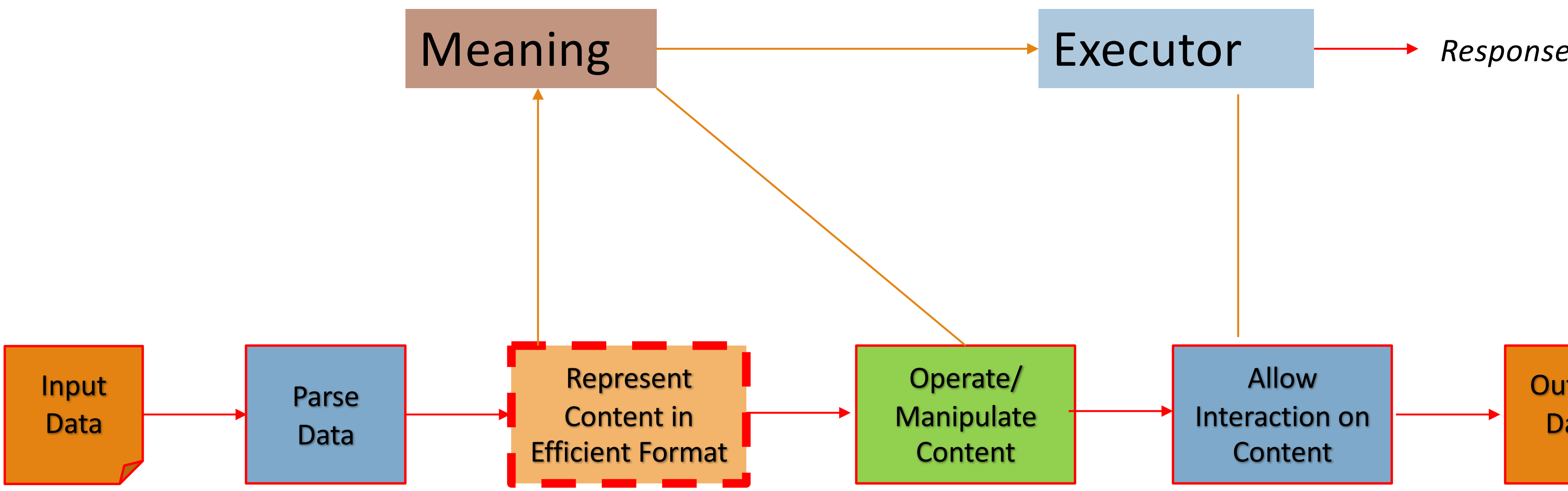
CSCE 771 goes into details

Class 19: Text Processing, Common NLP Tasks

Document Processing Pipeline



Semantics, Parsing and Representation

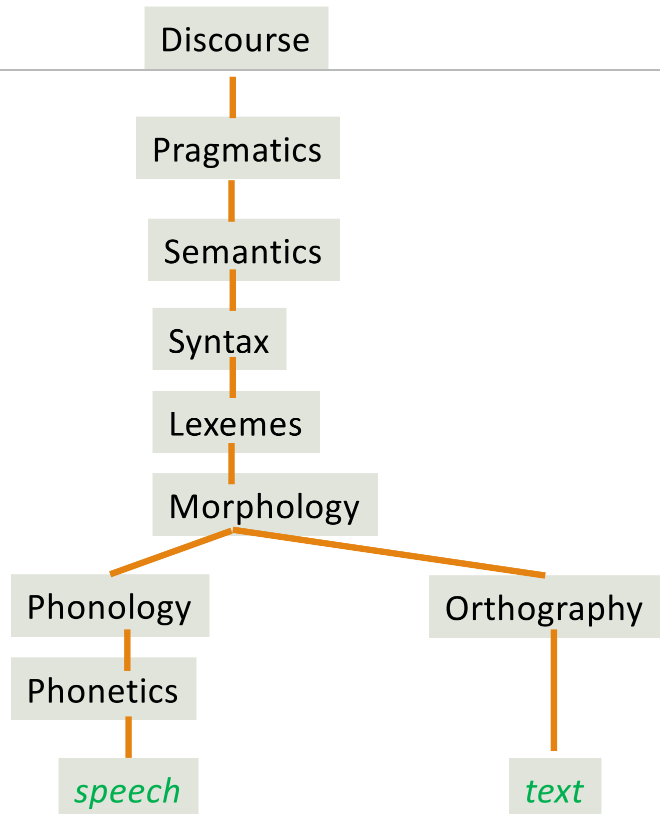


Common Textual Data Processing Steps for ML

- 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

Levels of Linguistic Studies



- **Discourse:** study of group of sentences
- **Pragmatics:** how context contributes to meaning of sentences
- **Semantics:** meaning of words and combinations of words
- **Syntax:** rules for combining and using words/ phonemes.
- **Lexemes:** a set of words that are related through inflection (fly: verb, fly: noun)
- **Morphology**—rules that govern morphemes - the minimal meaningful units of language (lemmas and affixes)
- **Orthography**: convention for writing a language. E.g., spelling
- **Phonology:** organization of speech sound (i.e., phoneme)
- **Phonetics**: study of how sound is made and received

Key Step: Parsing

- Recognizing legal inputs from illegal
- Usage of parse representation - parse tree
  - Grammar checking
  - Semantic analysis
  - Machine translation
  - Question answering
  - Information extraction
  - Speech recognition
  - …

Adapted from material by
Robert C. Berwick

Simple Example Using CFGs

N	a set of **non-terminal symbols** (or **variables**)
Σ	a set of **terminal symbols** (disjoint from N)
R	a set of **rules** or productions, each of the form $A \rightarrow \beta$,
	where A is a non-terminal,
	β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$
S	a designated **start symbol** and a member of N

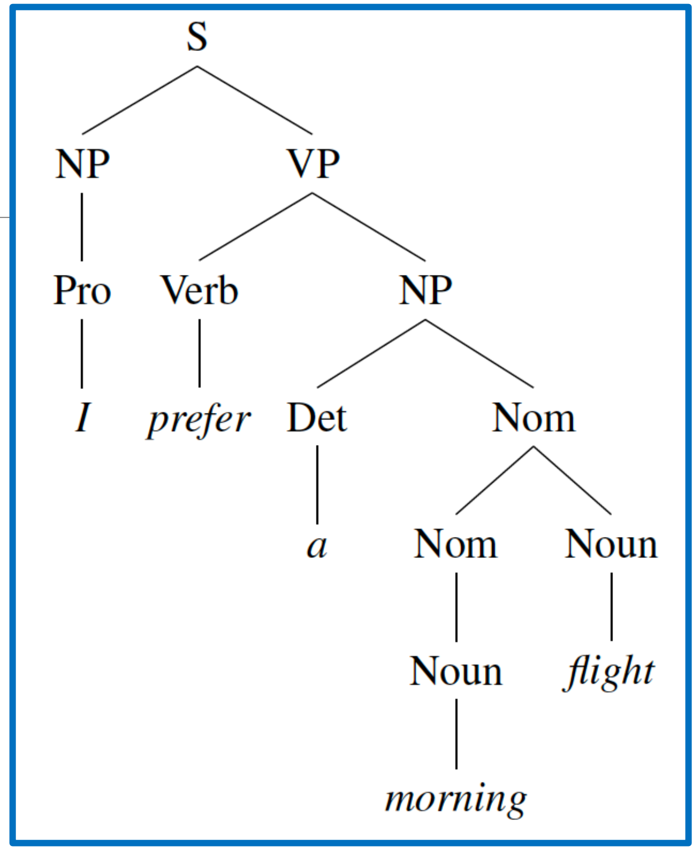
Grammar Rules			Examples
S	$\rightarrow$	$NP\ VP$	I + want a morning flight
NP	$\rightarrow$	$Pronoun$	I
	$\mid$	$Proper\text{-}Noun$	Los Angeles
	$\mid$	$Det\ Nominal$	a + flight
$Nominal$	$\rightarrow$	$Nominal\ Noun$	morning + flight
	$\mid$	$Noun$	flights
VP	$\rightarrow$	$Verb$	do
	$\mid$	$Verb\ NP$	want + a flight
	$\mid$	$Verb\ NP\ PP$	leave + Boston + in the morning
	$\mid$	$Verb\ PP$	leaving + on Thursday
PP	$\rightarrow$	$Preposition\ NP$	from + Los Angeles

$$
\begin{aligned}
Noun &\rightarrow flights \mid breeze \mid trip \mid morning \\
Verb &\rightarrow is \mid prefer \mid like \mid need \mid want \mid fly \\
Adjective &\rightarrow cheapest \mid non\text{-}stop \mid first \mid latest \\
&\quad \mid other \mid direct \\
Pronoun &\rightarrow me \mid I \mid you \mid it \\
Proper\text{-}Noun &\rightarrow Alaska \mid Baltimore \mid Los\ Angeles \\
&\quad \mid Chicago \mid United \mid American \\
Determiner &\rightarrow the \mid a \mid an \mid this \mid these \mid that \\
Preposition &\rightarrow from \mid to \mid on \mid near \\
Conjunction &\rightarrow and \mid or \mid but
\end{aligned}
$$

From Jurafsky & Martin

An Example Using CFGs

Grammar Rules			Examples
S	→	*NP VP*	I + want a morning flight
NP	→	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	→	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	→	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	→	*Preposition NP*	from + Los Angeles



From Jurafsky & Martin

$[_S\ [_{NP}\ [_{Pro}\ \text{I}]]\ [_{VP}\ [_V\ \text{prefer}]\ [_{NP}\ [_{Det}\ \text{a}]\ [_{Nom}\ [_N\ \text{morning}]\ [_{Nom}\ [_N\ \text{flight}]]]]]]$

Bracketed Notation

Interpretation of Parsing Rules

- generation (production): $S \rightarrow NP\ VP$
- parsing (comprehension): $S \leftarrow NP\ VP$
- verification (checking): $S = NP\ VP$
- CFGs are declarative – tell us *what* the well-formed structures & strings are
- Parsers are procedural – tell us *how* to compute the structure(s) for a given string

From Robert C. Berwick

Types of Parsing

- **Phrase structure** / **Constituency Parsing**: find phrases and their recursive structure.
Constituency - groups of words behaving as single units, or constituents.
    - **Shallow Parsing/ Chunking**: identify the flat, non-overlapping segments of a sentence: noun phrases, verb phrases, adjective phrases, and prepositional phrases.

- **Dependency Parsing**: find relations in sentences

- **Probabilistic Parsing**: given a sentence X, predict the most probable parse tree Y

Semantics

- ***lexical semantics****:* studies word meanings and word relations, and
- ***formal semantics****:* studies the logical aspects of meaning, such as sense, reference, implication, and logical form
- ***conceptual semantics****:* studies the cognitive structure of meaning

Source: Jurafsky & Martin,
Wikipedia (https://en.wikipedia.org/wiki/Semantics)

Review: Common Definitions

- **Corpus** (plural corpora): a computer-readable corpora collection of text or speech.
- **Lemma**: A lemma is a set of lexical forms having the same stem, the same major part-of-speech, and the same word sense. Example: Cat and cats have same lemma.
- **Word form**: The word form is the full inflected or derived form of the word. Example: Cat and cats have different word forms.
- **Word type**: Types are the number of distinct words in a corpus. if the set of words is V, the number of types is the word token vocabulary size |V|.
- **Word tokens**: The total number N of running words in the sentence / document of interest.
- **Code switching**: use multiple languages in a code switching single communicative act – Example: Hindlish (Hindi English), Spanish (Spanish English)

"They picnicked by the pool, then lay back on the grass and looked at the stars."

- 16 tokens, 14 word types

Source: Jurafsky & Martin

From Text to Meaning

- Shallow semantics
  - Input: text
  - Output: ***lexical semantics***
- Deep semantics
  - Input: text
  - Output: ***formal semantics***

LOGIC format:

$\exists$ w, b, g:
instance(w, want-01) $\wedge$ instance(g, go-01) $\wedge$
instance(b, boy) $\wedge$ arg0(w, b) $\wedge$
arg1(w, g) $\wedge$ arg0(g, b)

AMR format (based on PENMAN):

```
(w / want-01
  :arg0 (b / boy)
  :arg1 (g / go-01
          :arg0 b))
```

GRAPH format:

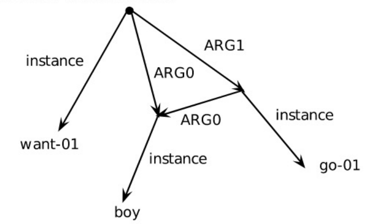
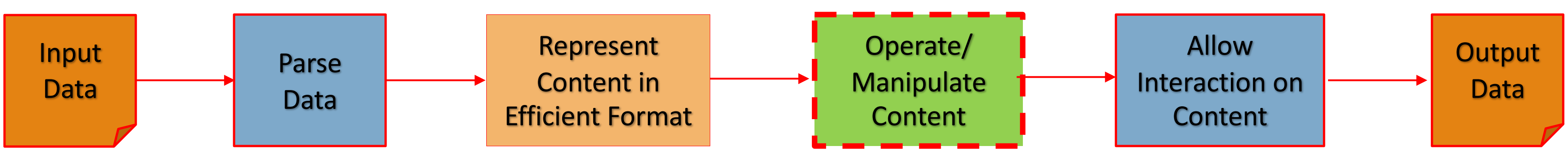


Figure 1: Equivalent formats for representating the meaning of "The boy wants to go".

Source: Abstract Meaning Representation for Sembanking, https://amr.isi.edu/a.pdf

Entity Extraction



What is an Entity?

- Definition
  - Oxford: “a thing with distinct and independent existence”
  - Practical: Any mention in text of interest
- Types
  - Physical: Person, animal, mountain
  - Abstract: Emotion, nation, money
- Heuristic: Entities are often nouns

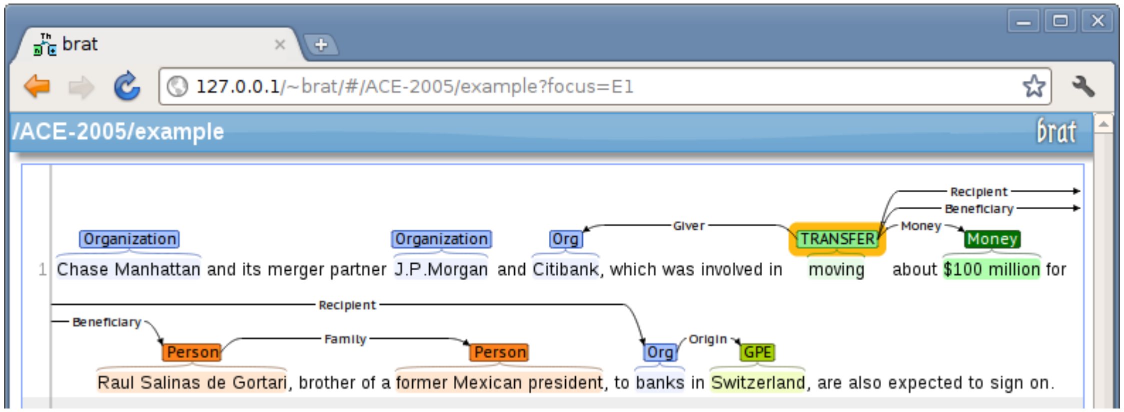
The **Nobel Peace Prize** is one of the five **Nobel Prizes** established by the **will** of Swedish industrialist, inventor, and armaments manufacturer **Alfred Nobel**, along with the prizes in Chemistry, Physics, Physiology or Medicine, and Literature

Credit: From Wikipedia

Entity Extraction Methods

- Regular expression: find patterns in content
  - Why: if pattern known, easy, fast and cheap to implement
  - Why not: pattern has to be known
- Manual annotation: tag entities and store in a repository; runtime - match in content and retrieve tags
  - Tool: BRAT - https://brat.nlplab.org/introduction.html
  - Why: use information when available
  - Why not: cost of annotation is high, time-consuming

Also called: Entity identification, entity chunking, Named entity recognition (NER)



Annotate entity types and relationships; Credit: https://brat.nlplab.org/introduction.html

Reference: https://lionbridge.ai/articles/the-essential-guide-to-entity-extraction/

Entity Extraction – Methods Continued

- Learning based – many varieties
  - Why: Pretrained available, domain-specific models, alignment with standards
  - Why not: needs large compute resources, may not be explainable

Reference: https://lionbridge.ai/articles/the-essential-guide-to-entity-extraction/

Which Learning-based Method

- Conditional random field (CRF): learn probability of entities based on defined features over inputs
  - Requires labeled data about text and entities, needs features, learns entity labels
  - Articles: https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html#let-s-use-conll-2002-data-to-build-a-ner-system; https://www.depends-on-the-definition.com/named-entity-recognition-conditional-random-fields-python/

- LSTM-based: predict labels (entities) over text sequences.
  - Requires labeled data about text and entities, models forward and backward neighborhood, learns entity labels
  - Blog: https://www.depends-on-the-definition.com/named-entity-recognition-with-residual-lstm-and-elmo/

- Deep learning based models
  - A Survey on Recent Advances in Named Entity Recognition from Deep Learning models, Vikas Yadav, Steven Bethard, ACL 2018 https://www.aclweb.org/anthology/C18-1182.pdf

Benchmarks – Oct 2022

https://paperswithcode.com/task/named-entity-recognition-ner/codeless

Benchmarks

These leaderboards are used to track progress in Named Entity Recognition (NER)

Trend	Dataset	Best Model
	CoNLL 2003 (English)	ACE + document-context
	Ontonotes v5 (English)	BERT-MRC+DSC
	NCBI-disease	Spark NLP
	WNUT 2017	CL-KL
	ACE 2005	Ours: cross-sentence ALB
	JNLPBA	KeBioLM
	BC5CDR	BINDER
	GENIA	DeepStruct multi-task w/ finetune
	BC5CDR-chemical	Spark NLP
	BC2GM	Spark NLP

Oct 2024

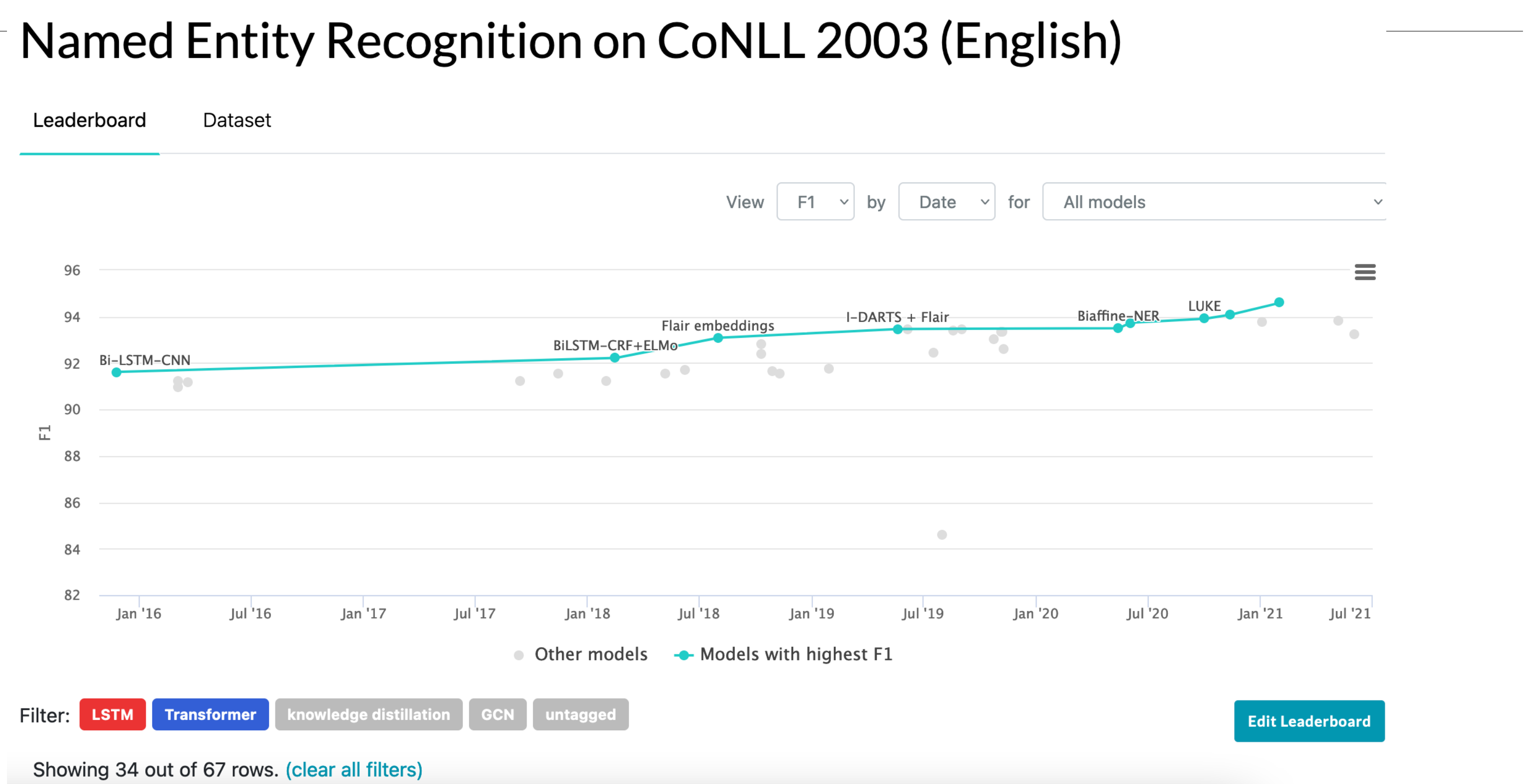
Trend	Dataset	Best Model	Paper	Code	Compare
	CoNLL 2003 (English)	ACE + document-context			See all
	Ontonotes v5 (English)	BERT-MRC+DSC			See all
	NCBI-disease	Spark NLP			See all
	WNUT 2017	CL-KL			See all
	ACE 2005	Ours: cross-sentence ALB			See all
	JNLPBA	KeBioLM			See all
	BC5CDR	CL-L2			See all
	SLUE	W2V2-L-LL60K (pipeline approach, uses LM)			See all
	BC5CDR-chemical	Spark NLP			See all
	GENIA	Biaffine-NER			See all

all 65 benchmarks

Oct 2022

Benchmark on a Dataset – Oct 2022

- https://paperswithcode.com/task/named-entity-recognition-ner/codeless



Annotation of Entities for Interchange

- IOB stands for inside-outside-beginning
- Standoff format

Named Entity types
- person names (PER),
- organizations (ORG),
- locations (LOC) and
- Times
- Quantities
- Miscellaneous names (MISC)

CONLL shared tasks
2002: https://www.aclweb.org/anthology/W02-2024/
2003: https://www.aclweb.org/anthology/W03-0419.pdf

U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

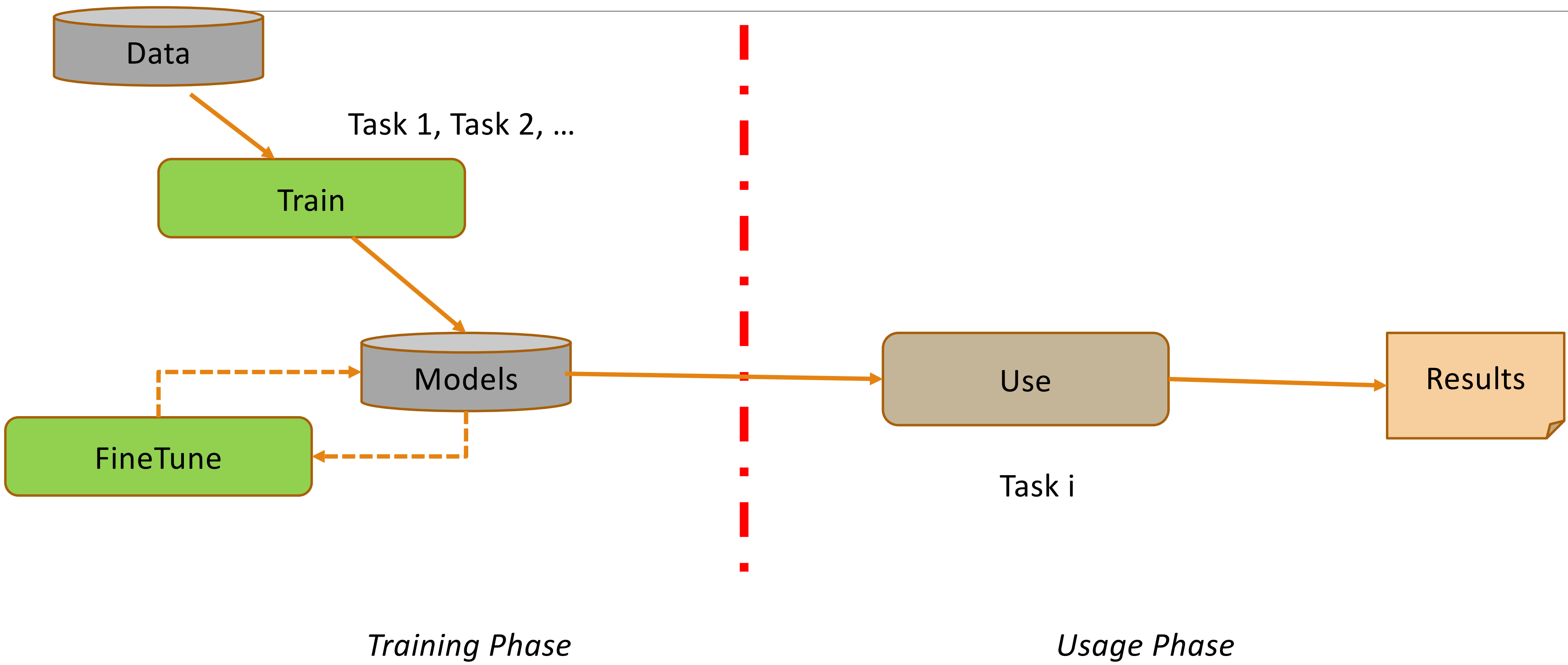
Source: https://lionbridge.ai/articles/the-essential-guide-to-entity-extraction/

Coding Resource

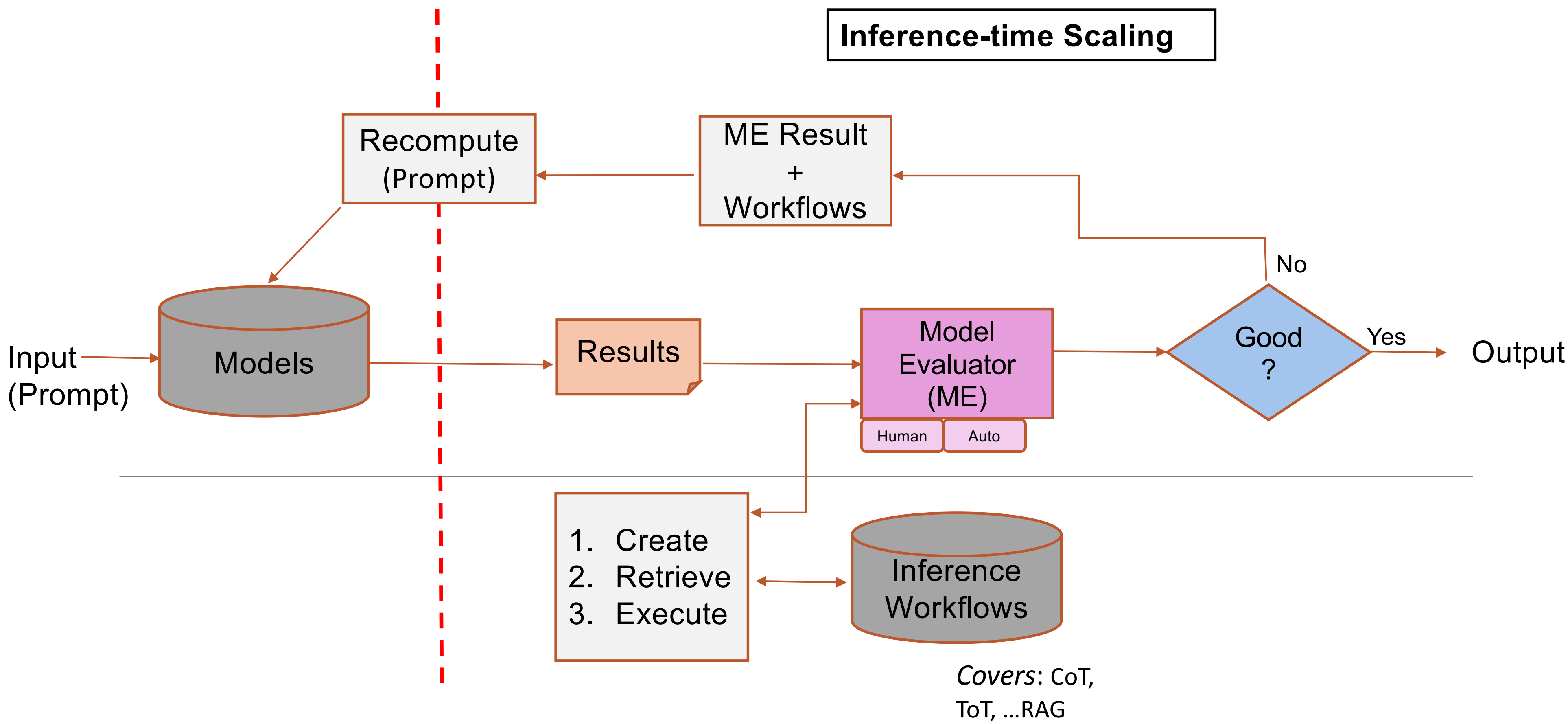
- **Notebook**: https://github.com/biplav-s/course-nl-f22/blob/main/sample-code/l17-eventextr/SimpleEntitySearch.ipynb

Class 20: LLMs and Common NLP Tasks

Large Language Models (LLMs) Basics



Inference Time with LLMs



BERT - Bidirectional Encoder Representations from Transformers

Learns with two tasks

- Predicting missing words in sentences
  - mask out 15% of the words in the input, predict the masked words.

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

- Given two sentences A and B, is B the actual next sentence that comes after A, or just a random sentence from the corpus?

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

Major LM Types

- Large
  - Large training dataset
  - Large number of parameters
- General purpose
  - Commonality of human languages
  - Resource restriction
- Pre-trained and fine-tuned

Credits: Google Cloud Skills Boost

LLMs have three different architectures - (a) encoder-only, (b) decoder-only, and (c) encoder-decoder, each with their own benefits.

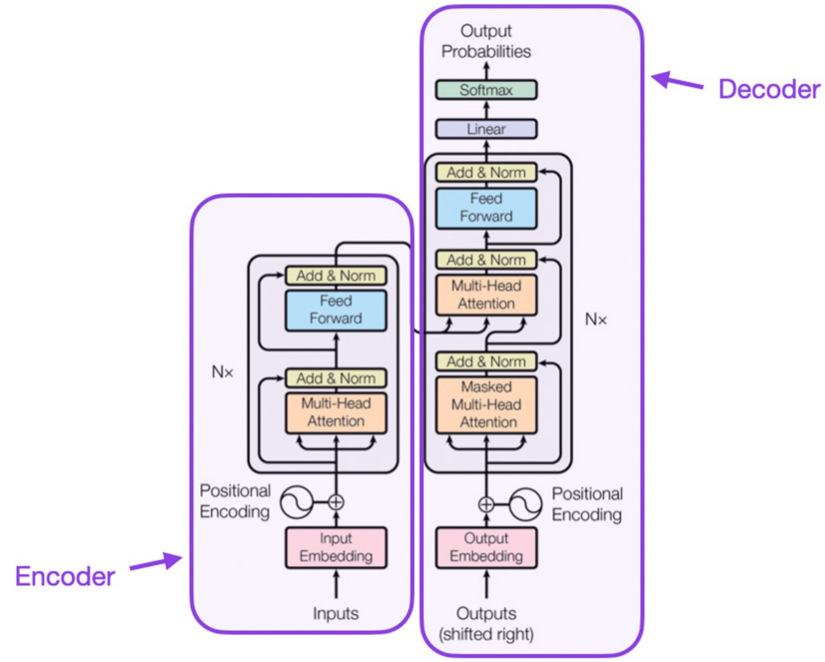
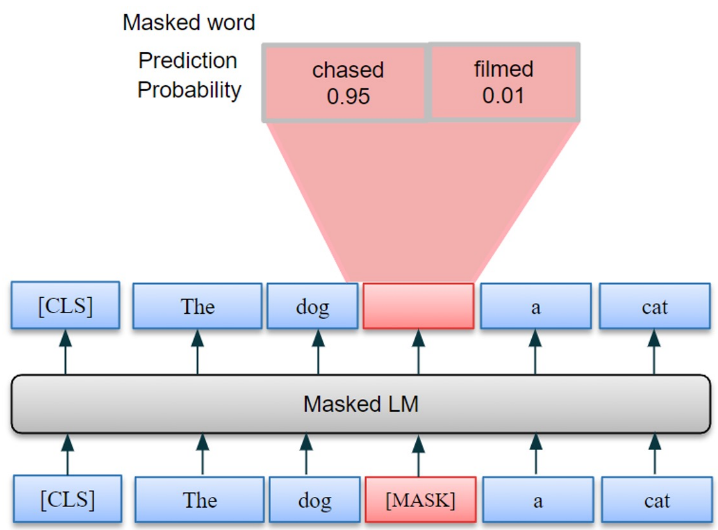


Figure. The Transformer - model architecture.

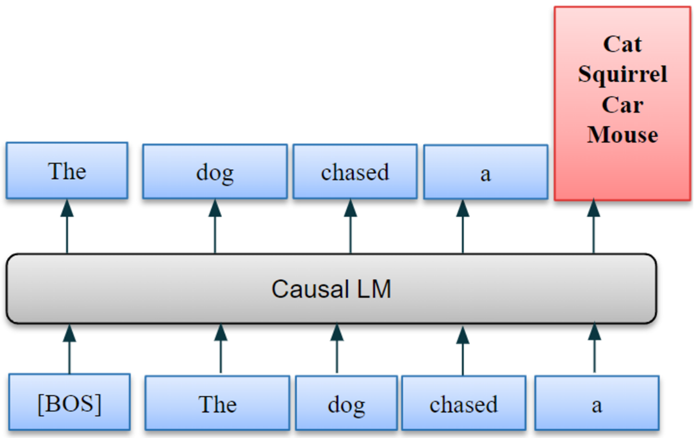
Encoder-only

- Encoder-only architectures are trained to understand the bidirectional context by predicting words randomly masked in a sentence.
- **Example:** BERT
- **Effective for:** sentiment analysis, classification, entailment.



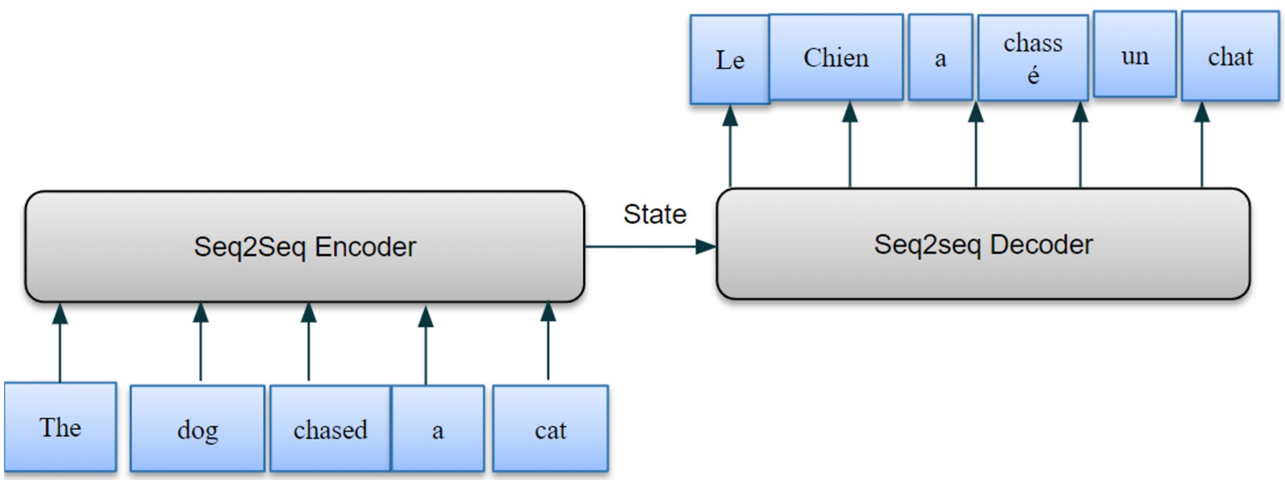
Decoder-only

- Decoder-only architectures are designed for tasks where text generation is sequential and dependent on the preceding context.
- They predict each subsequent word based on the preceding words, modeling the probability of a word sequence in a forward direction.
- **Example:** GPT-4, Llama series, Claude, Vicuna.
- **Effective for:** Content generation.



Encoder-Decoder

- Encoder-Decoder architectures are designed to transform an input sequence into a related output sequence.
- **Example:** T5, CodeT5, FlanT5
- **Effective for:** summarization, language translation.



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

- Also see: https://huggingface.co/blog/sentiment-analysis-python

Options

- Assumption: Already tried a pre-trained model and know performance on one's tasks
- Option 1: Fine-tune a pretrained model
- Option 2: Use someone-else's pretrained model
  - Creating mini-GPT (OpenAI): https://help.openai.com/en/articles/8554397-creating-a-gpt
- Option 3: Build one's own on specialized data, tasks and optimizing performance metrics of interest

Project Discussion

Course Project

- **Framework**
  1. (Problem) Think of a problem whose solution may benefit people (e.g., health, water, air, traffic, safety)
  2. (User) Consider how the primary user (e.g., patient, traveler) may be solving the problem today
  3. (AI Method) Think of what the solution will do to help the primary user
     1. Solution => ML task (e.g. classification), recommendation, text summarization, …
     2. Use a foundation model (e.g., LLM-based) solution as the baseline
  4. (Data) Explore the data for a solution to work
  5. (Reliability: Testing) Think of the evaluation metric we should employ to establish that the solution will works? (e.g., 20% reduction in patient deaths)
  6. (Holding Human Values) Discuss if there are fairness/bias, privacy issues?
  7. (Human-AI) Finally, elaborate how you will explain the primary user that your solution is trustable to be used by them

Project Discussion: What to Focus on ?

- Problem: you should care about it
- Data: should be available
- Method: you need to be comfortable with it. Have at least two – one serves as baseline
- Trust issue
  - Due to Users
    - Diverse demographics
    - Diverse abilities
    - Multiple human languages
  - Or other impacts
- What one does to mitigate trust issue

Rubric for Evaluation of Course Project

Project

- Project plan along framework introduced (7 points)
- Challenging nature of project
- Actual achievement
- Report
- Sharing of code

Presentation

- Motivation
- Coverage of related work
- Results and significance
- Handling of questions

Concluding Section

Week 10 (L19 and 20): Concluding Comments

- We looked at
  - Solving common NLP tasks
  - Impact of LLMs on Text/ AI tasks

About Next Week – Lectures 21, 22

Lectures 21, 22

- Supervised ML with Text
- Trust Issues with ML/ Text

13	Feb 25 (Tu)	AI - Supervised ML: Explanation Tools
14	Feb 27 (Th)	AI Trust - Mitigation method (Trust rating) – Kausik Lakkaraju
15	Mar 4 (Tu)	Large Language Models (LLMs), Machine Learning – Trust Issues (Explainability)
16	Mar 6 (Th)	Student presentations - project
	Mar 11 (Tu)	
	Mar 12 (Th)	
17	Mar 18 (Tu)	**Invited Guest** – Kush Varshney
18	Mar 20 (Th)	AI - Unstructured (Text): Processing and Representation
19	Mar 25 (Tu)	AI - Unstructured (Text): Representation, Common NLP Tasks, Large Language Models (LLMs)
20	Mar 27 (Th)	Natural Languages/ Language Models and their Impact on AI
21	Apr 1 (Tu)	AI - Unstructured (Text): Analysis – Supervised ML – Trust Issues
22	Apr 3 (Th)	AI - Unstructured (Text): Analysis – Supervised ML – Mitigation Methods
23	Apr 8 (Tu)	AI - Unstructured (Text): Analysis – Rating and Debiasing Methods