AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

UNIVERSITY OF SOUTH CAROLINA

CSCE 771: Computer Processing of Natural Language
Lecture 16: Using LLMs in NLP Tasks, Entities

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

10TH OCTOBER, 2024

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

Organization of Lecture 16

- Opening Segment
  - Recap of recent classes
- Main Lecture
- Concluding Segment
  - About Next Lecture – Lecture 17

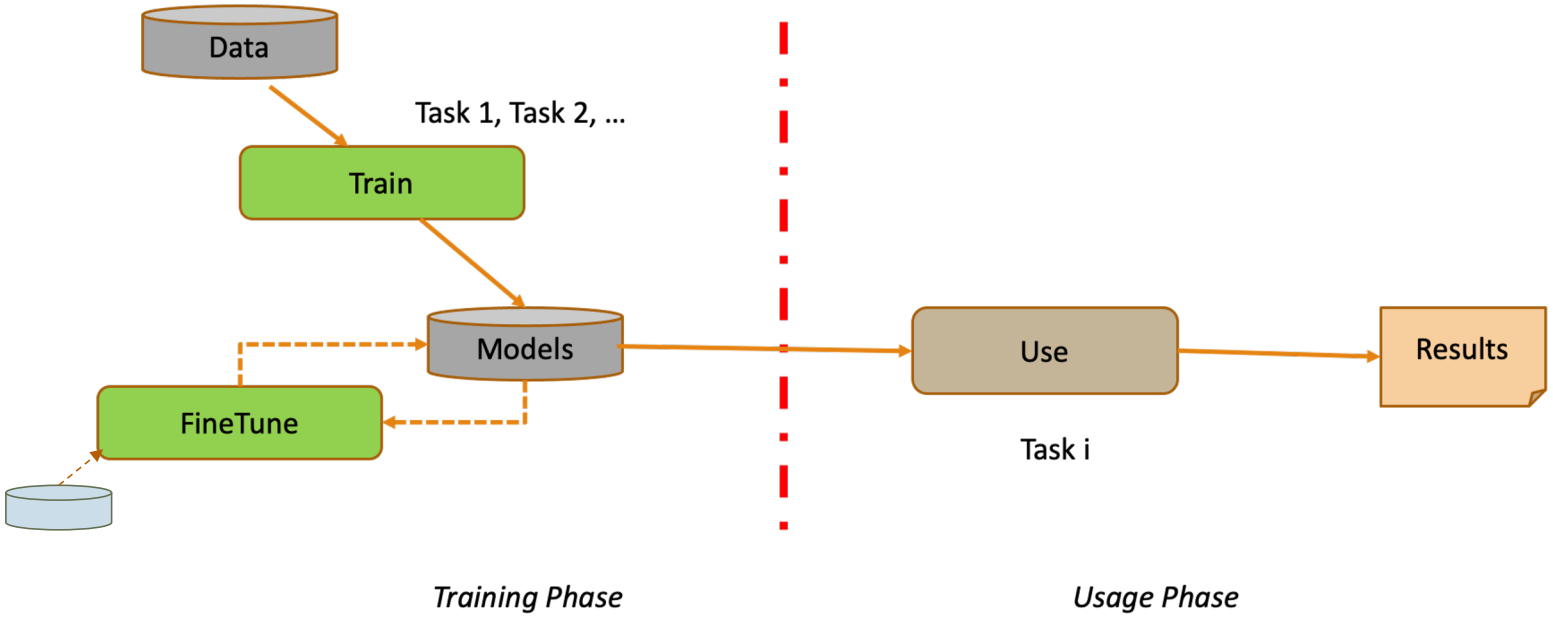
Main Section
- Review LLM key concepts
- Using LLMs in practice
- NLP tasks
  - Methods learned
  - Using LLMs
  - Comparing results
- Entity Extraction
- Paper Reading

Recap of Lecture 15

- Milestone 1 of project
  - Good range of states for election Q/A analysis
  - Gear up for detailed analysis with language/ NLP methods discussed
  - As well as LLM background used

Main Lecture

LM Basics


Data
Task 1, Task 2, …
Train
Models
FineTune
Use
Results
Task i
Training Phase
Usage Phase

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.

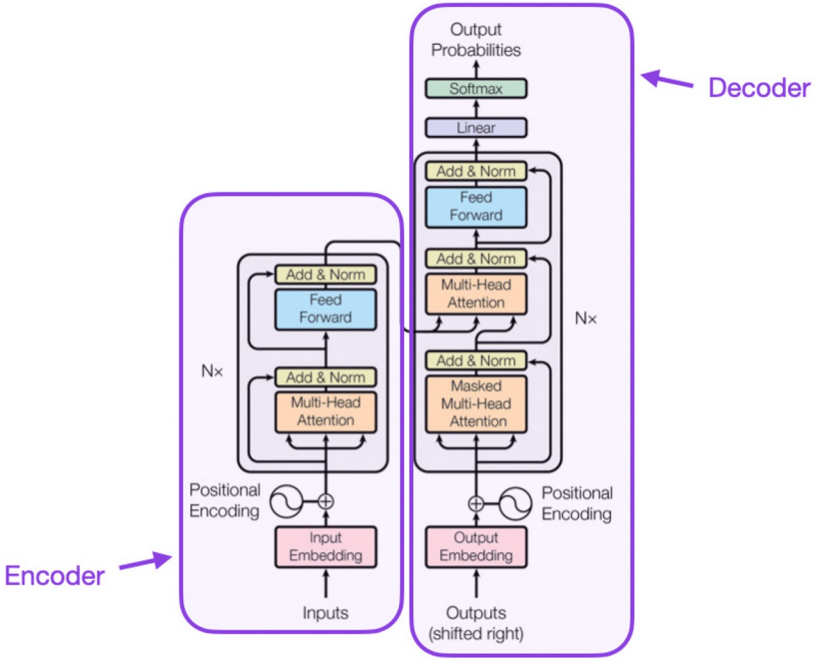


Figure. The Transformer - model architecture.

Review: 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
  - Kausik Roy's last four lectures

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?

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

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

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

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

Contextual Representations

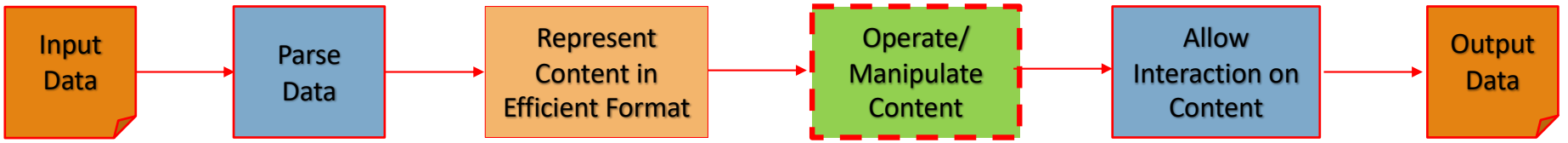
	Name	Description	URL, References
1.	TF-IDF		Ramos, J. (1999). Using TF-IDF to Determine Word Relevance in Document Queries
2	Word2Vec	Word-based, prediction focus	*Mikolov, Tomas; et al. (2013). "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781 [cs.CL]. Mikolov, Tomas (2013). "Distributed representations of words and phrases and their compositionality". Advances in Neural Information Processing Systems. arXiv:1310.4546.*
3	Glove	Word-based, count	https://nlp.stanford.edu/projects/glove/, Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. [pdf] [bib]
4	BERT / DistilBERT		**BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding** J. Devlin, M. Chang, K. Lee, and K. Toutanova. (*2018*)*cite arxiv:1810.04805*
5	Elmo (embeddings from language models)	Contextual, deep, character-based	https://allennlp.org/elmo; Deep contextualized word representations, Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. NAACL 2018.
6	Fasttext	Variation of word2vec, works with N-gram, words not in voacbulary	https://fasttext.cc/ , P. Bojanowski*, E. Grave*, A. Joulin, T. Mikolov, *Enriching Word Vectors with Subword Information*

Commentaries:
https://jalammar.github.io/illustrated-bert/ , https://cai.tools.sap/blog/glove-and-fasttext-two-popular-word-vector-models-in-nlp/

To Do:

- Read papers on language models. (Full pdf available from reading list page on GitHub: https://github.com/biplav-s/course-nl-f24/blob/main/reading-list/Readme-LLMs.md
  - Hang Li, Language Models: Past, Present, and Future, Communications of the ACM, July 2022, Vol. 65 No. 7, Pages 56-63 10.1145/3490443
  - A Primer in BERTology: What We Know About How BERT Works (Rogers et al., TACL 2020)
- Try LLM finetuning using project (election) data
- Will be asked in Quiz3 (Oct 15)

Methods to Extract Content



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

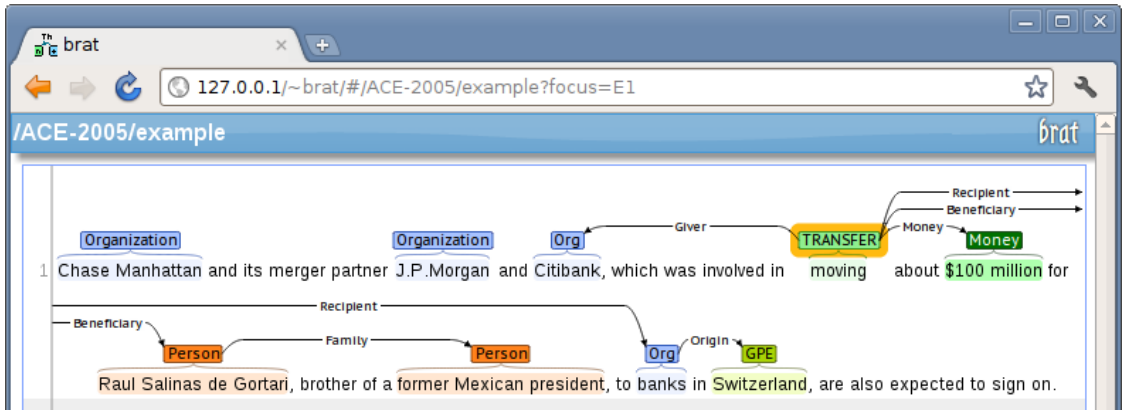
The **Nobel Peace Prize** is one of the five **Nobel Prizes** established by the **will** of Swedish industrialist, inventor, and armaments manufacturer **Alfred Nobel**, along with the prizes in Chemistry, Physics, Physiology or Medicine, and Literature

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

Lecture 16: Concluding Comments

- Summarized LLMs/ transformers and their motivation
- Compliment with reading
- Started with Entity extraction

Concluding Segment

Course Project

Discussion: Course Project

Theme: Analyze quality of official information available for elections in 2024 [in a state]

- Take information available from
  - Official site: State Election Commissions
  - Respected non-profits: League of Women Voters
- Analyze information
  - State-level: Analyze quality of questions, answers, answers-to-questions
  - Comparatively: above along all states (being done by students)
- Benchmark and report
  - Compare analysis with LLM
  - Prepare report

- Process and analyze using NLP
  - Extract entities
  - Assess quality – metrics
    - Content – *Englishness*
    - Content – *Domain* -- election
  - … other NLP tasks
  - Analyze and communicate overall

Major dates for project check

- Sep 10: written – project outline
- Oct 8: in class
- Oct 31: in class // LLM
- Dec 5: in class // Comparative

Obtaining Election Data

Here are a few things to do:
A) **Official data** backed by laws: state election commission
a) Find the state's election commission
b) Find the Q/As they provide. They may be as FAQs or on different web pages.
c) Collect the Q/A programmatically

B) **Secondary data** sources: non-profit
a) Find Q/As from Vote 411 which is supported by the non-profit: LWV.

For reference, for SC,
A) Official - https://scvotes.gov/voters/voter-faq/
B) Secondary - https://www.vote411.org/south-carolina

For extraction, one or more approaches:
- Manually annotating
- BeautifulSoup,
- Tika
- or other open source libraries.

Election Q/A for Your State

- Format in .json; name file as "**xy**_qa.json", where xy is the two-character US state acronym
- Fixed attributes in .json
  - state: xy
  - num_questions: **a**, where a is the number of questions
  - num_answers: **b**, where b is the number of answers
  - contributor: student name
- List of Q/As with attributes for each it:
  - q // question
  - a // answer
  - s // source url from where the information is taken
  - t // time when the information is taken
- Store it in your github repo; put in sub-dir like "project/data"
- Instructor will keep it in common place inside course github repo and share.

Election Q/As for Multiple States

- Instructor will keep it common place inside course github repo and share.
- You will be able to access Q/As of all states from common location
  - To compare data across all states

Discussion

- How will you use a LLMs for election data analysis ?
- When and Why? (conversely, not)

<Student Name>

CSCE 771 Fall 2024: Milestone#1

1. State Selected:
2. Election data sites:
   - Official site (e.g.,State Election Commission) url
   - Secondary site (e.g., League of Women Voters) url
1. Report how data collected and Q/A statistics

1. Take on NLP methods you will use and why for Q/A analysis
   1. State-level (right)
   2. Comparatively: above along states being done by peers

Initial analysis of questions (Q)

*

Initial analysis of answers (A)

*

Initial analysis of an answer (a_i) for a question (q_i)

*

About Next Lecture – Lecture 17

Lecture 17 Outline

- Quiz 2
- Lecture 18 (after Fall break)
  - Entity extraction
  - Entity linking
  - Representation of extracted entities and links

13	Oct 1 (Tu)	Language model – comparing arch, finetuning - **Guest Lecture**
14	Oct 3 (Th)	Language model – comparison of results, discussion, ongoing trends– **Guest Lecture**
15	Oct 8 (Tu)	PROJ REVIEW
16	Oct 10 (Th)	Using lang models to solve NLP tasks
17	Oct 15 (Tu)	QUIZ 3
	Oct 17 (Th)	
18	Oct 22 (Tu)	Entity extraction, linking
19	Oct 24 (Th)	Events extraction, spatio-temporal analysis
20	Oct 29 (Tu)	Topic Analysis