



N P T E L O N L I N E C E R T I F I C A T I O N C O U R S E S

DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING

Lecture 01 : Introduction to the Course



PROF . PAWAN GOYAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

CONCEPTS COVERED

- Course Information
- What is NLP?
- Why Deep Learning for NLP?
- Course Content

Course Information

- My Contact
 - Email: pawang@cse.iitkgp.ac.in
 - Webpage: <http://cse.iitkgp.ac.in/~pawang/>
 - Course Page: <https://sites.google.com/view/dl4nlp-nptel/home>
- Teaching Assistants (Inaugural Course)
 - Subhendu Khatuya
 - Pretam Ray

Natural Language Processing



Natural Languages: Languages that evolved naturally through human use

Source: https://en.wikipedia.org/wiki/Language_family

Natural Language Processing

What is NLP?

- *Making computers understand what we write (or speak)*
- *Making computers write (and speak)*

The field of NLP attempts to design, implement and test systems that process natural languages for practical applications

NLP Applications: NLP is everywhere!

English - Detected English Spanish French ▾

Welcome to the Natural Language Processing course X

49 / 5,000

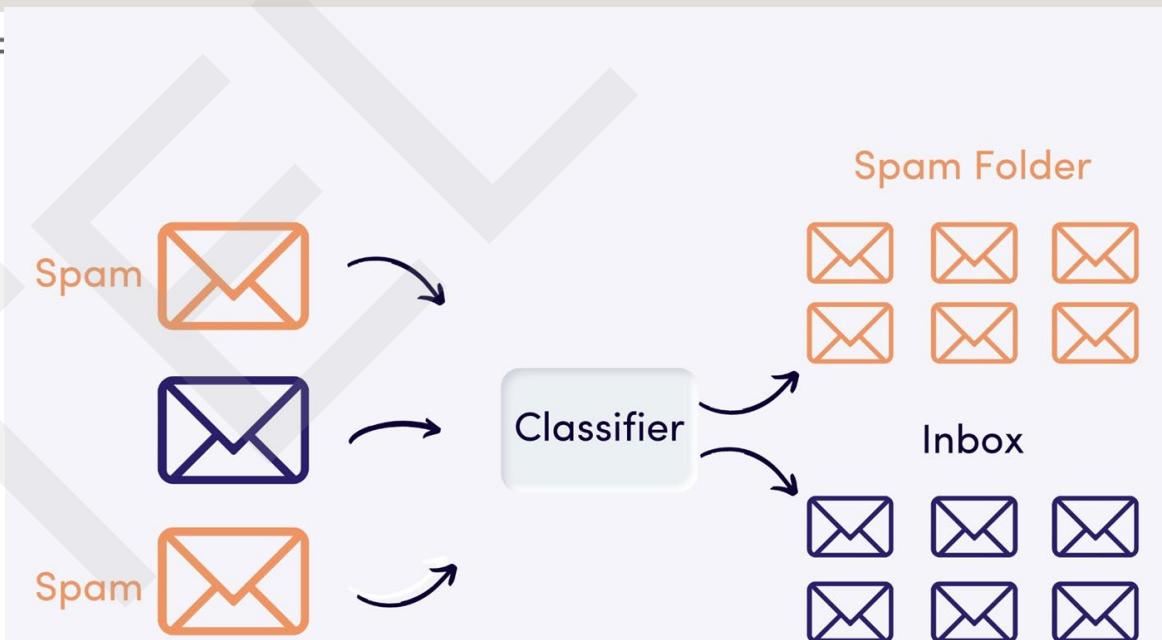
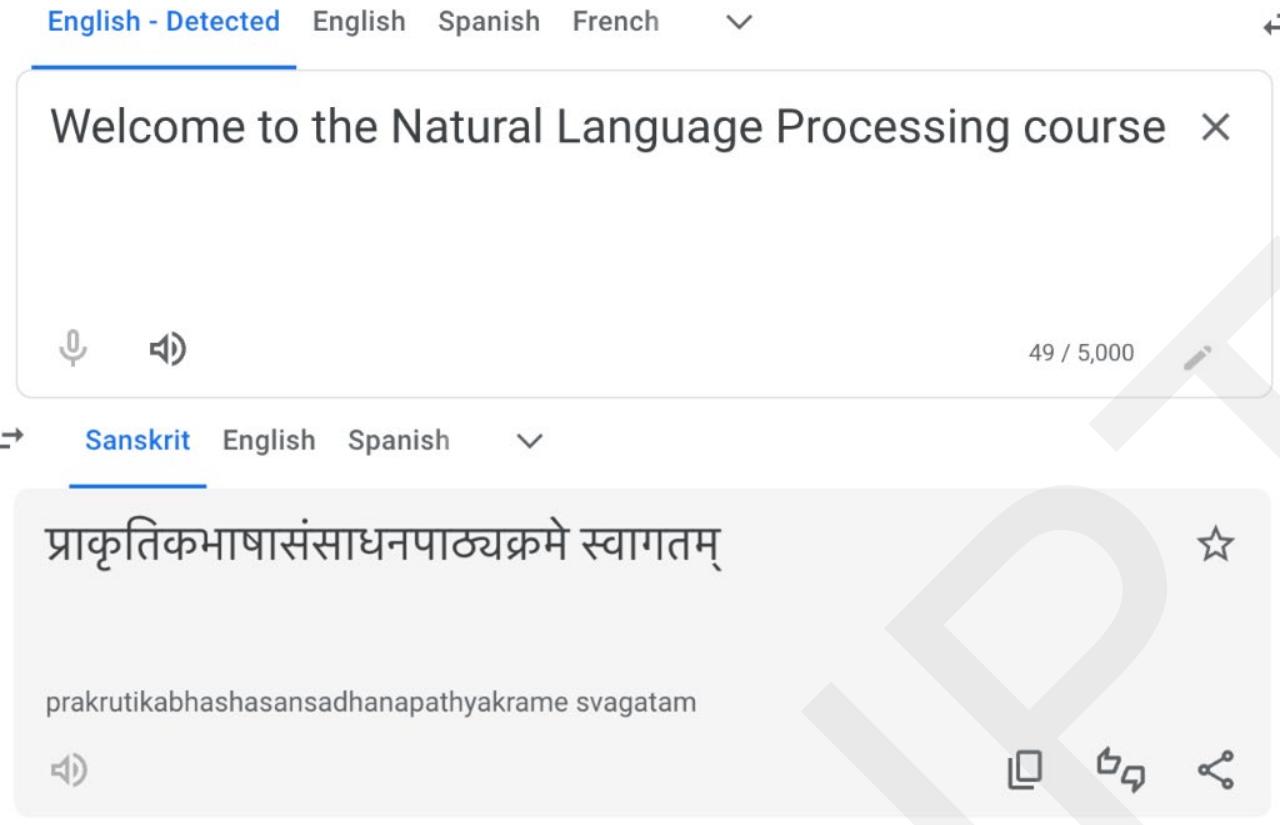
Speaker icon, Volume icon

Sanskrit English Spanish ▾

प्राकृतिकभाषासंसाधनपाठ्यक्रमे स्वागतम्

prakrutikabhashasansadhanapathyakrame svagatam

Speaker icon



NLP is everywhere!

Who is the tallest living person?

All Images Videos Shopping News Web Books More Tools

 Search Labs | AI Overview

 Listen

According to Guinness World Records, the tallest living person is [Sultan Kösen](#) from Turkey, who stands at 8 feet 2.8 inches (2.51 meters) tall. 



Key points about Sultan Kösen:

- **Nationality:** Turkish 
- **Height:** 8 feet 2.8 inches (2.51 meters) 
- **Record holder:** Guinness World Record for tallest living man 

[Guinness World Records | Tallest man - Sultan Kösen - 8 ft 2.8 in \(251 cm ...\)](#)
9 Sept 2024 — Tallest man - Sultan Kösen - 8 ft 2.8 in (251 cm)
 Instagram 

[The record holder for tallest living man in the US dies at 38 | CNN](#)
Guinness says the tallest living man in the world is Sultan Kösen from Turkey, who measures 8 feet, 2.8 in tall.
 [cnn.com](#) 

[Meet the Tallest Woman in the World, Rumeysa Gelgi | HowStuffWorks](#)
Sultan Kösen (also from Turkey) achieved the world record for tallest man living in 2011 when he measured at 8 feet, 2.8...
 [history.howstuffworks.com](#) 

NLP is everywhere!

Persona

I bought my first home.
I love to barbecue.
I live in Springfield.
I'm a writer.

Persona

I weight 300 pounds.
I am not healthy.
I am a man.
I like The Godfather.



Hello how are you, I am new to the Springfield area.



Hi! Seen any good movies lately?



I have been to the movies.



I love The Godfather, one of my favorites! Was that filmed?



I don't believe so. I don't watch movies more of a writer.



What do you write? Any diet books ? I am not very healthy.



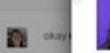
explain chatgpt like i'm five

explain chatgpt like i'm five



Sure! ChatGPT is like a really smart robot friend that can understand and talk to people using words. It has learned a lot of things by reading lots and lots of books, articles, and websites. It knows about animals, science, history, and even how to cook yummy recipes!

But remember, even though ChatGPT is really clever, it's just a computer program and doesn't have feelings or experiences like humans do. So, it's important to ask it questions and have fun with it, but also remember that it doesn't know everything and might make mistakes.



okay now explain it like i'm an adult

Certainly! ChatGPT is an advanced language model created by OpenAI called GPT-3.5. It's designed to understand and generate human-like text based on the prompts or questions it receives. It has been trained on an enormous amount of text data, including books, articles,



Certainly! ChatGPT is an advanced language model created by OpenAI called GPT-3.5. It's designed to understand and generate human-like text based on the prompts or questions it receives. It has been trained on an enormous amount of text data, including books, articles,

informative answers by drawing upon the knowledge it has acquired during training. It can assist with a wide range of topics, including science, history, mathematics, literature, and more.

However, it's important to note that ChatGPT is not a human and lacks genuine understanding or awareness. It operates purely on statistical patterns and doesn't possess real-world experiences or common sense reasoning. While it can generate impressive responses, it's always important to verify the information it provides independently, especially for critical or sensitive topics.

OpenAI continues to improve and refine its language models like ChatGPT over time, but they still have limitations. It's crucial to "Regenerate response" for general information and

Send a message

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT Max 24 Version

NLP is everywhere!

What should i keep in mind while filing taxes? Tab



37/4000



What can i eat if i have a peanut allergy? Tab



30/4000



while creating an instagram post, what hashtags should i use? Tab



44/4000



NLP is everywhere!

In fact, the Chinese NORP market has the three CARDINAL most influential names of the retail and tech space – Alibaba GPE , Baidu ORG , and Tencent PERSON (collectively touted as BAT ORG), and is betting big in the global AI GPE in retail industry space . The three CARDINAL giants which are claimed to have a cut-throat competition with the U.S. GPE (in terms of resources and capital) are positioning themselves to become the ‘future AI PERSON platforms’. The trio is also expanding in other Asian NORP countries and investing heavily in the U.S. GPE based AI GPE startups to leverage the power of AI GPE . Backed by such powerful initiatives and presence of these conglomerates, the market in APAC AI is forecast to be the fastest-growing one CARDINAL , with an anticipated CAGR PERSON of 45% PERCENT over 2018 - 2024 DATE .

To further elaborate on the geographical trends, North America LOC has procured more than 50% PERCENT of the global share in 2017 DATE and has been leading the regional landscape of AI GPE in the retail market. The U.S. GPE has a significant credit in the regional trends with over 65% PERCENT of investments (including M&As, private equity, and venture capital) in artificial intelligence technology. Additionally, the region is a huge hub for startups in tandem with the presence of tech titans, such as Google ORG , IBM ORG , and Microsoft ORG .

Domain Specific Applications



Very Long
E-Manual

CUSTOMER

SUPPORT

DOMAIN

FINANCIAL

DOMAIN

How can I encrypt my
SD Card?

Settings

Section

Encrypt or decrypt SD card

You can encrypt your optional memory card (not included) to protect its data. This only allows the SD card information to be accessed from your device with a password.

Answer

- From Settings, tap Biometrics and security > Encrypt or decrypt SD card.
- Tap Encrypt SD card and follow the prompts to encrypt all data on your memory card.

NOTE Performing a Factory data reset on your device prevents it from accessing an encrypted SD card. Before initiating a Factory data reset, make sure to decrypt the installed SD card first.

Decrypt SD card

You can decrypt an optional memory card (not included) if it was encrypted by this device. You may want to decrypt the memory card if you plan to use it with another device or before performing a Factory data

Question Answering

Cash and Cash Equivalents

Debt Instrument Convertible Conversion Price

As of December 31, 2020, we had cash equivalents of \$24.8 million and a closing stock price of \$18.20 per share.

We also acquired a business loan from the U.S. Bancorp of \$60.5 million.

**Line of Credit Facility
Maximum Borrowing Capacity**

Impairment Loss

Finally, our firm reports **no** impairment loss for this year.

NER

LEGAL DOMAIN

Clause Span Extraction

Named Entity Recognition
NCBI disease, BC2GM, ...

Relation Extraction
EU-ADR, ChemProt, ...

Question Answering
BioASQ 5b, BioASQ 6b, ...

Output

Governing Law: "This Agreement shall be governed by the laws of the State of California without giving effect to conflict or choice of law principles." (Page 2)

⚠️ Covenant Not to Sue: "In addition, Company shall not now or in the future contest the validity of Investor's ownership of its Intellectual Property." (Page 30)

▶ Perpetual / Irrevocable License: "Company grants to Investor a worldwide, royalty-free, exclusive, irrevocable license (with the right to grant sublicenses)." (Page 151)

BIOMEDICAL DOMAIN

the adult renal failure cause ...
► O O B I O ...

Variants in the @GENE\$ region contribute to @DISEASE\$ susceptibility.
► True

What does mTOR stands for?
► mammalian target of rapamycin

Several sentence and token level tasks

Why is NLP Hard? Language Ambiguity

background | 'bak,ground |

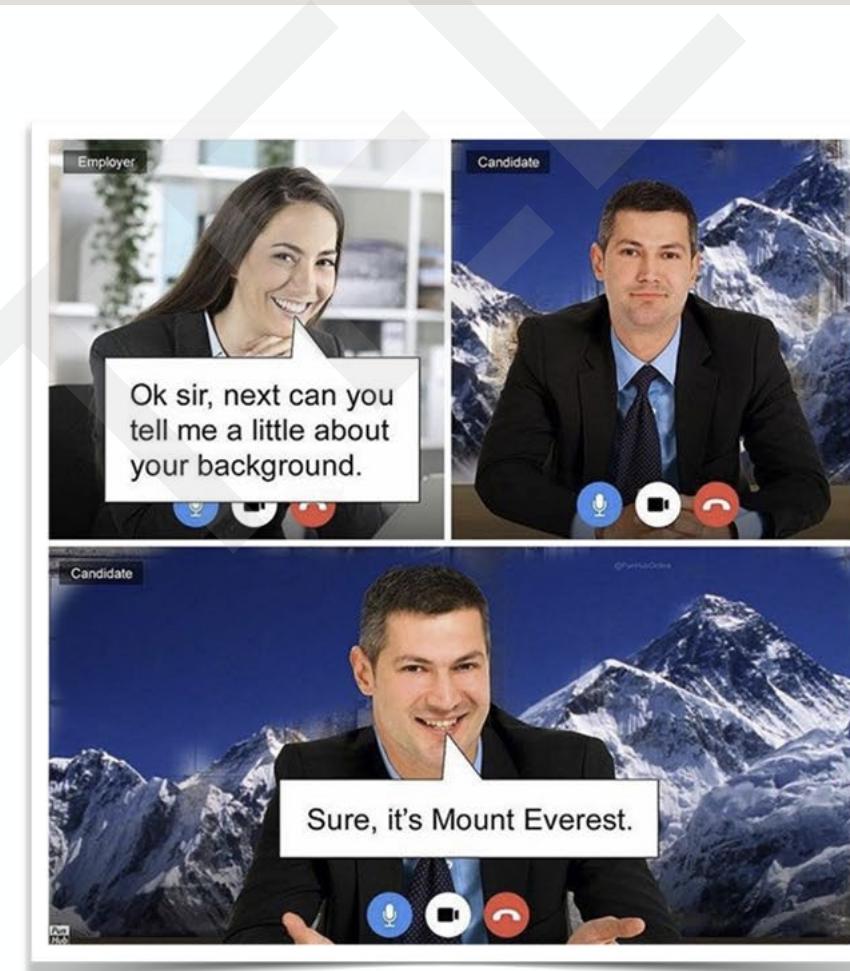
noun

1 [in singular] the area or scenery behind the main object of contemplation, especially when perceived as a framework for it: *the house stands against a background of sheltering trees.*

- the part of a picture or design that serves as a setting to the main figures or objects, or that appears furthest from the viewer: *the background shows a landscape of domes and minarets | the word is written in white on a red background.*
- a position or function that is not prominent or conspicuous: *after that evening, Athens remained **in the background**.*
- Computing used to describe tasks or processes running on a computer that do not need input from the user: *programs can be left running **in the background**.*
- Physics low-intensity radiation from radioisotopes present in the natural environment.
- unwanted signals, such as noise in the reception or recording of sound.

2 the circumstances or situation prevailing at a particular time or underlying a particular event: *the political and economic background | [as modifier] : background information.*

- a person's education, experience, and social circumstances: *she has a background in nursing | a mix of students from many different backgrounds.*



Source: <https://courses.cs.cornell.edu/courses/cs5740>

Why is NLP Hard? Language Ambiguity



Source: <https://courses.cs.cornell.edu/courses/cs5740>

Why is NLP Hard? Language Ambiguity

Let's try to decipher this weird conversation!

Rahul: *I saw a monkey with a banana.*

Computer: *That's gruesome!*

Rahul: *Why? What's so gruesome about seeing a monkey?*

Computer: *Oh I see! What else did you see with the banana?*

In Natural Languages, ambiguity is the rule, not an exception

Example: Courtesy Dr. Monojit choudhury

NLP: Levels of Linguistic Structure

Discourse

Semantics

CommunicationEvent(e)
Agent(e, Alice)
Recipient(e, Bob)

SpeakerContext(s)
TemporalBefore(e, s)

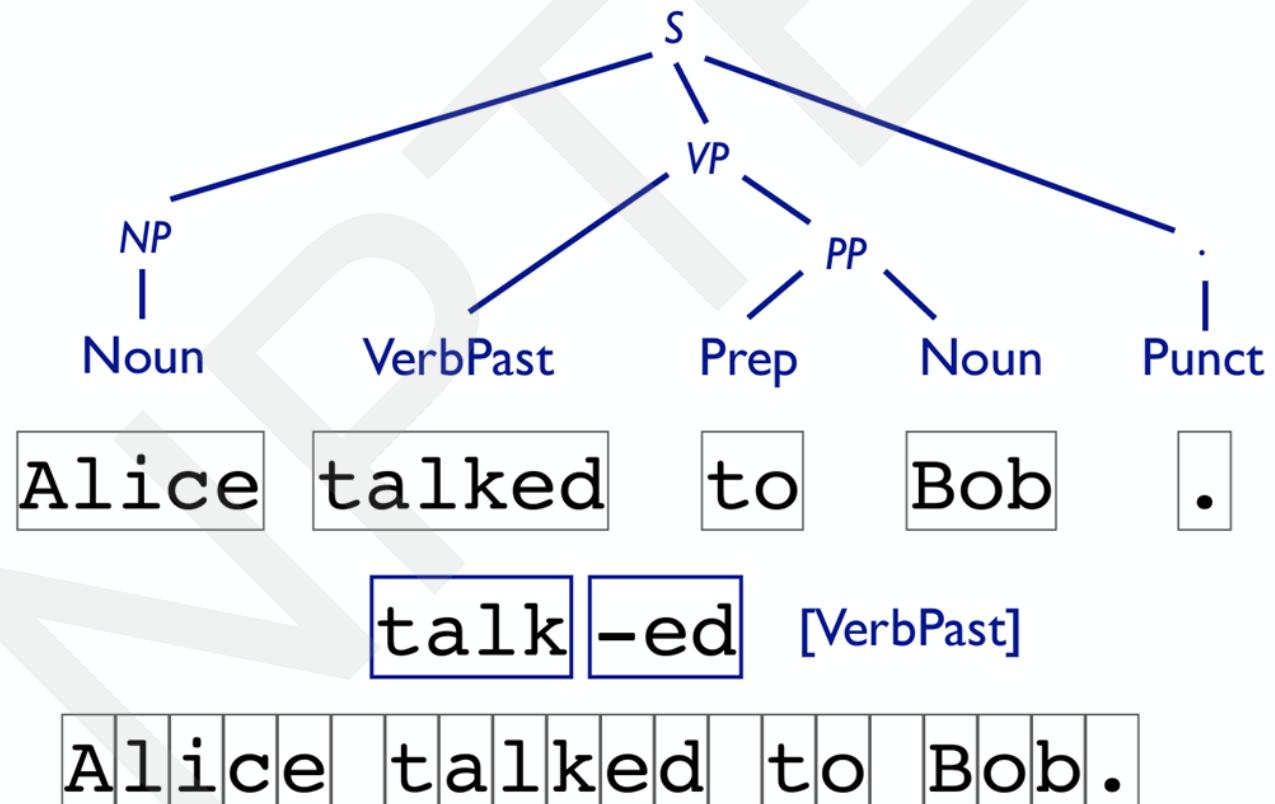
Syntax: Constituents

Syntax: Part of Speech

Words

Morphology

Characters



NLP Paradigms

We generally try to map problems to various (ML) paradigms

- Sentiment Analysis, news article groupings, etc. → text classification
- Named entity recognition, code-mixing, etc. → Sequence Labeling
- Machine Translation, summarization, chatbots, etc. → Text generation

*Expert Systems
and Statistical
Models*



1. Rules and Ontology based Systems
2. Statistical Models
3. N-Grams combined with Machine learning algorithms

1 1950s-1980s

2 1980s-2000s

3 2000s-2010s

4 2010s-2020s

5 2020s-now

*Syntactic and
Grammar-based*



1. Syntactic Structures by Noam Chomsky
2. ELIZA Chatbot
3. SHRDLU rule-based system

*Neural Models
and Dense
Representations*



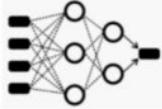
1. Bengio et al.'s Dense Vector Representation
2. Mikolov et al.'s language Models based on Recurrent Networks
3. Pre-Trained Word Embeddings

Era of LLMs

LLM

1. OpenAI releases GPT-2, GPT-3.5 and GPT 4
2. RLHF for alignment towards human values such as safety, groundedness , etc.
3. Open source LLMs and frameworks

*The Deep
Learning
Revolution*



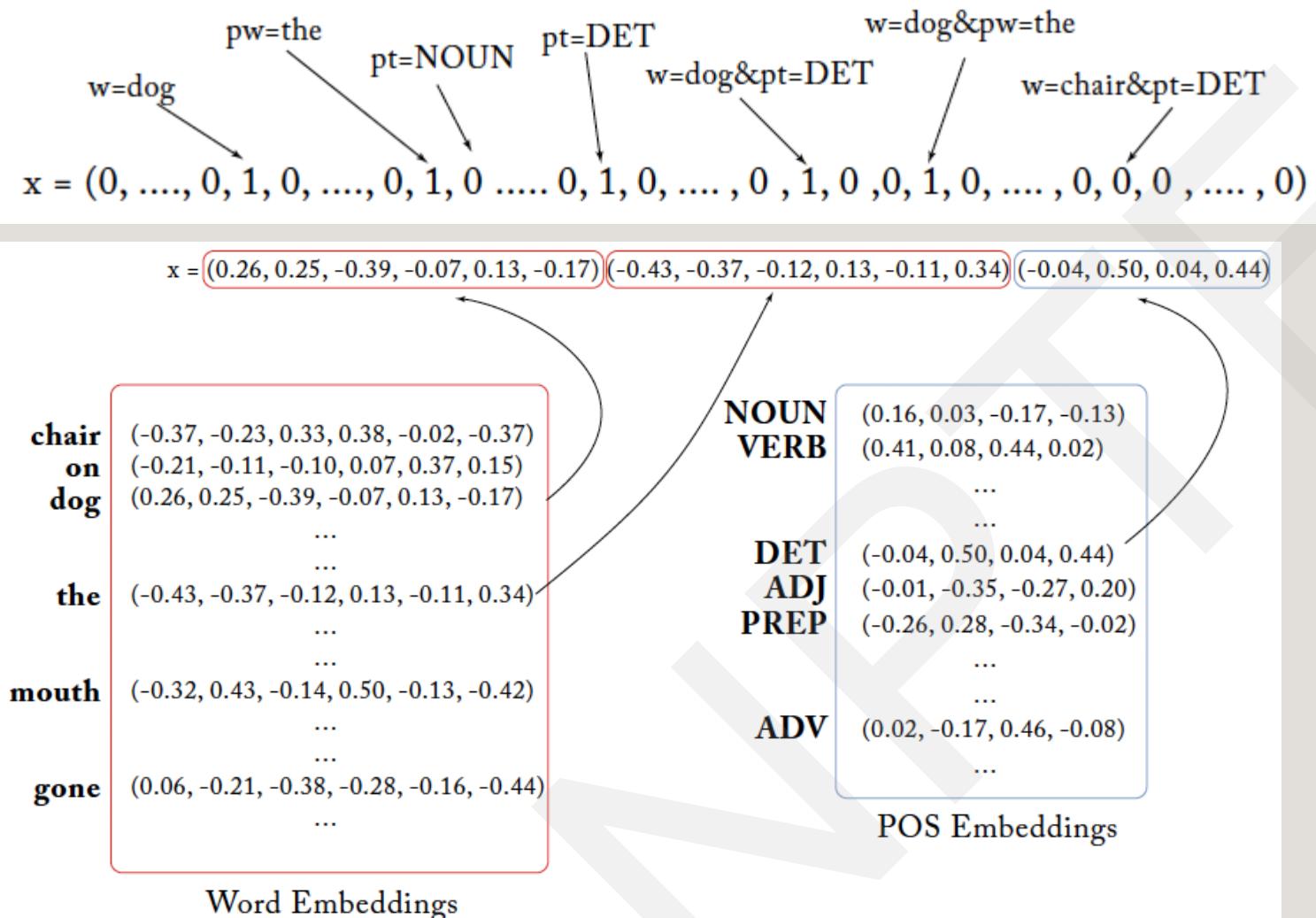
1. Word2Vec, GLoVe, etc. word embeddings
2. Transfer Learning through pre-trained and fine-tuning
3. Attention Mechanism by Bahdanau et al.
4. Transformers by Vaswani et al.
5. BERT, GPT, and other models

*Timeline illustrating the progression
of NLP from the 1950s*

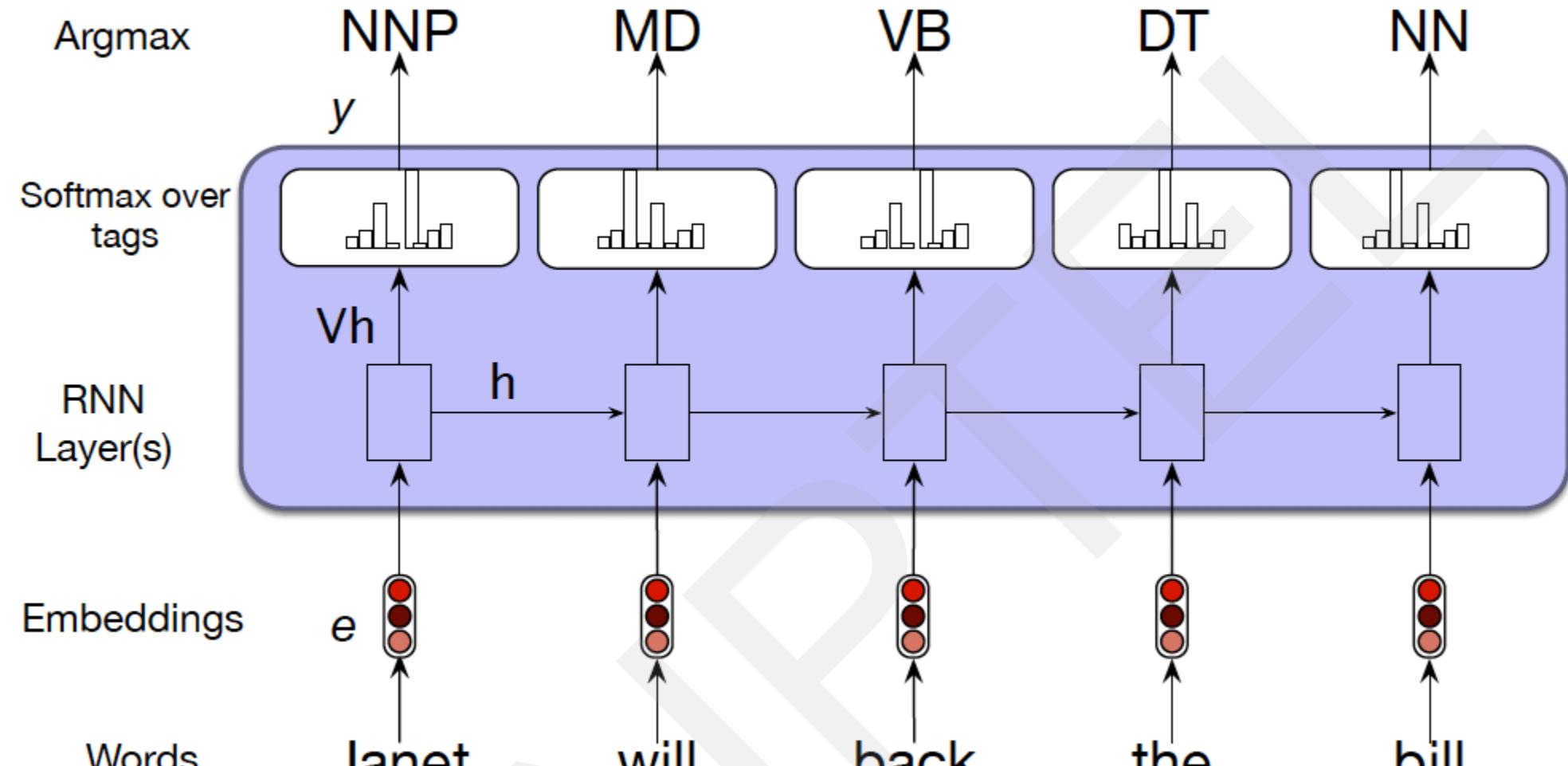
Source: Kamath, Uday, et al. "Large Language Models: A Deep Dive." (2024).



Why Deep Learning?

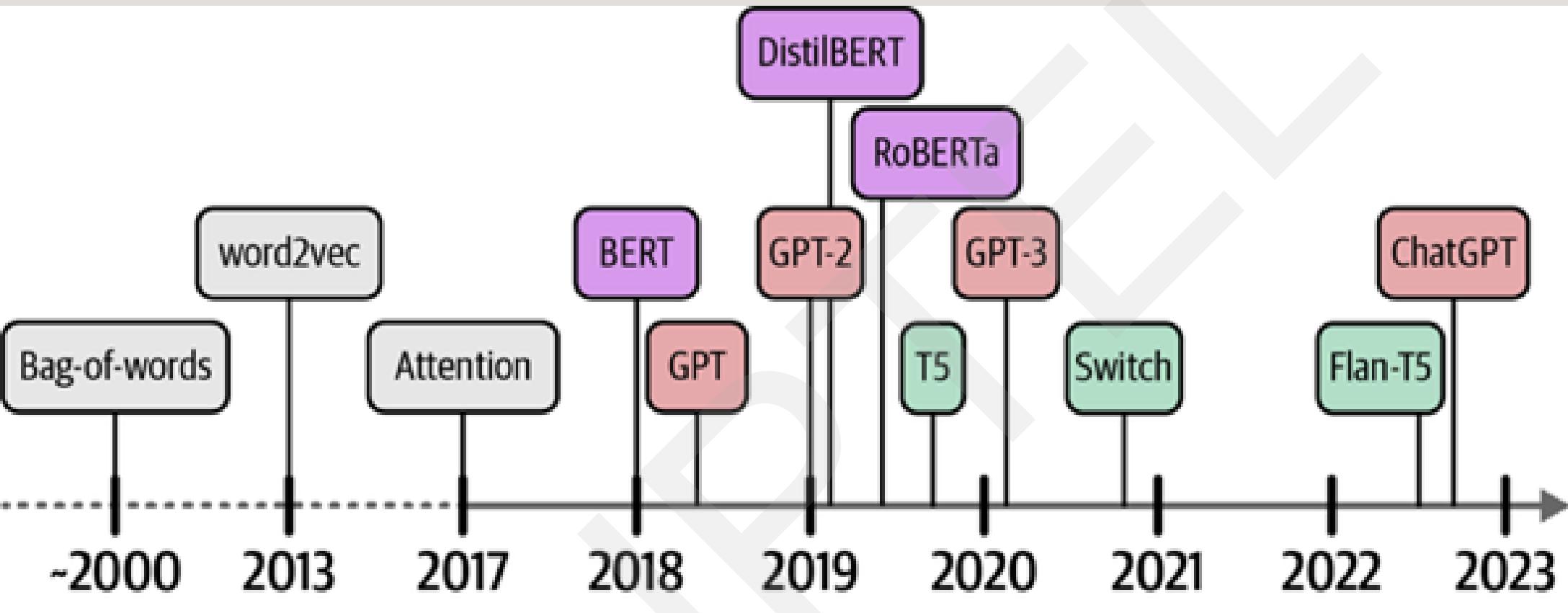


Sparse vs. dense feature representations. Two encodings of the information: current word is “dog;” previous word is “the” previous pos-tag is “DET.”



These dense feature representations are used with various deep-learning architectures

A timeline of the recent developments



Source: Alammar, J., & Grootendorst, M. (2024). Hands-On Large Language Models. O'Reilly.

Change of NLP paradigms: Just use generation!

Summarization

The picture appeared on the wall of a Poundland store on Whymark Avenue [...] How would you rephrase that in a few words?

Sentiment Analysis

Review: We came here on a Saturday night and luckily it wasn't as packed as I thought it would be [...] On a scale of 1 to 5, I would give this a

Question Answering

I know that the answer to "What team did the Panthers defeat?" is in "The Panthers finished the regular season [...]" . Can you tell me what it is?

Multi-task training

Zero-shot generalization

Natural Language Inference

Suppose "The banker contacted the professors and the athlete". Can we infer that "The banker contacted the professors"?

T₀

Graffiti artist Banksy is believed to be behind [...]

4

Arizona Cardinals

Yes

Course Content (Weeks 1-6)

Background

- Introduction to NLP
- Introduction to Deep Learning and Representation Learning
- Word Representation: Word2Vec, Glove, FastText, Multilingual

Models and Architectures

- Recurrent Neural Networks: RNNs, LSTMs, Sequence to Sequence
- Attention Mechanism and Transformers: Attention in RNNs, Self-Attention in Transformers

Methods

- Pretraining: Self-supervised Learning objectives for Pretraining, ELMo, BERT, GPT, T5, BART, Fine-tuning

Course Content (Weeks 7-12)

Tasks

- Question Answering, Text Summarization, Dialogs
- Domain and language-specific applications and challenges

Methods (LLMs)

- Towards building LLMs as chat assistants: Instruction Fine-tuning, Reinforcement learning from human feedback, Alignment techniques
- In-content learning, chain-of-thought prompting, Various LLMs
- Parameter Efficient Fine-tuning (PEFT), LoRA, QLoRA
- Handling Long Context, Retrieval Augmented Generation (RAG)

Conclusion

- Analysis and Interpretability, ethical considerations

REFERENCES

- Daniel Jurafsky and James H. Martin. 2024. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. Online manuscript released August 20, 2024. <https://web.stanford.edu/~jurafsky/slp3>.
- Alammar, J., & Grootendorst, M. (2024). Hands-On Large Language Models. O'Reilly.
- Yoav Goldberg, Graeme Hirst. Neural Network Methods in Natural Language Processing, Morgan & Claypool Publishers (2017).
- Kamath, Uday, et al. "Large Language Models: A Deep Dive." (2024).



THANK YOU



N P T E L O N L I N E C E R T I F I C A T I O N C O U R S E S

DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING

Lecture 02 : Text Processing Basics, Tokenization



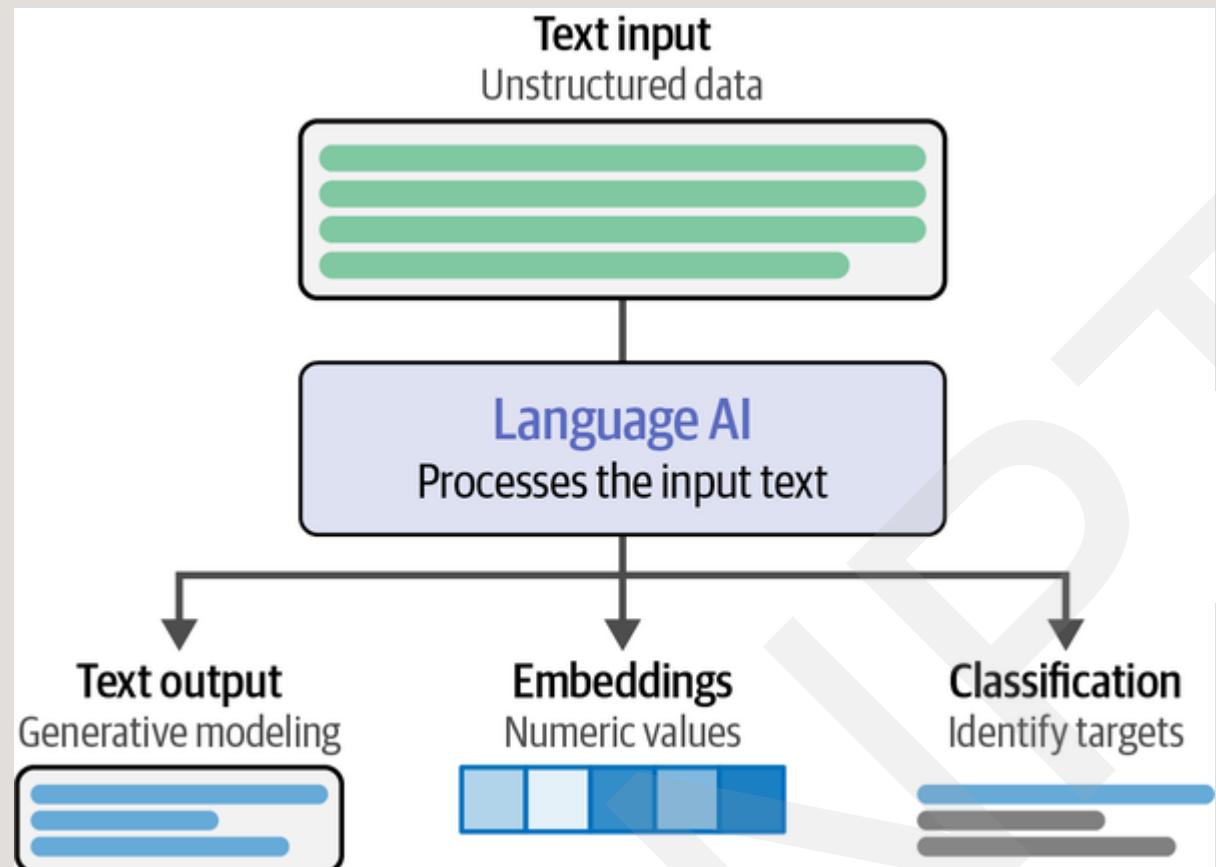
PROF . PAWAN GOYAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

CONCEPTS COVERED

- Processing Text Input
- Whitespace Tokenizer
- Byte-Pair Encoding

Processing Text Input



For any NLP application, the input text needs to be processed first.

The first step in processing text is *tokenization*.

Input text: students opened their books

Input token IDs: **11** **298** **34** **567**

Tokenization: How many words in a sentence?

they lay back on the San Francisco grass and looked at the stars and their

Type: an element of the vocabulary.

Token: an instance of that type in running text.

How many?

- 15 tokens
- 13 types

Source: Speech and Language Processing, 3rd Ed.

How many words in a corpus?

N = number of tokens

V = vocabulary = set of types, $|V|$ is size of vocabulary

Heaps Law = Herdan's Law = $|V| = kN^\beta$ where often $.67 < \beta < .75$

i.e., vocabulary size grows with $>$ square root of the number of word tokens

	Tokens = N	Types = V
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
COCA	440 million	2 million
Google N-grams	1 trillion	13+ million

Source: Speech and Language Processing, 3rd Ed.

Corpora: Where do the words come from?

Words don't appear out of nowhere!

A text is produced by

- a specific writer(s),
- at a specific time,
- in a specific variety,
- of a specific language,
- for a specific function.

Corpora vary along dimensions like

- **Language:** 7097 languages in the world
- **Variety,** like African American Language varieties.
Twitter posts might include forms like "*iont*" (*I don't*)
- **Code switching,** e.g., Spanish/English, Hindi/English:
S/E: Por primera vez veo a @username actually being hateful! It was beautiful:)
[*For the first time I get to see @username actually being hateful! it was beautiful:)*]
H/E: dost tha or rahega ... dont worry
["*he was and will remain a friend ... don't worry*"]
- **Genre:** newswire, fiction, scientific articles, Wikipedia
- **Author Demographics:** writer's age, gender, ethnicity

Whitespace tokenization

Tokens are implied to be *words*

Example:

Input text: students opened their books

Input token IDs: **11** **298** **34** **567**

Whitespace tokenizer issues

- *conjunctions*: isn't ⇒ is, n't
- *hyphenated phrases*: prize-winning ⇒ prize, -, winning
- *punctuation*: great movie! ⇒ great, movie, !

(Word tokenizers require lots of specialized rules about how to handle specific inputs)

What if a new (or infrequent) word appears?

Out-of-vocabulary (OOV): Words that were seen very rarely during training or not even at all

Closed-vocabulary models: Unable to produce word forms unseen in training data

<UNK> tokens:

- Historically rare word types were replaced with a new word type UNK (unknown) at training time
- At test time, any token that was not part of the model's vocabulary could then be replaced by UNK
- But you should not generate UNK when generating text
- UNKs don't give features for novel words that maybe useful anchors of meaning
- In languages other than English, in particular those with more productive morphology, removing rare words is infeasible

Limitations of <UNK>

We lose lots of information about texts with a lot of rare words / entities

The chapel is sometimes referred to as "Hen Gapel Lligwy" ("hen" being the Welsh word for "old" and "capel" meaning "chapel").

The chapel is sometimes referred to as " Hen <unk> <unk> " (" hen " being the Welsh word for " old " and "<unk> " meaning " chapel ").

Source: <https://people.cs.umass.edu/~miyyer/cs685>

Maximal Decomposition into Characters

But deciding what counts as a word in Chinese is complex. For example, consider the following sentence:

- (2.4) 姚明进入总决赛
“Yao Ming reaches the finals”

As Chen et al. (2017) point out, this could be treated as 3 words ('Chinese Treebank' segmentation):

- (2.5) 姚明 进入 总决赛
YaoMing reaches finals

or as 5 words ('Peking University' segmentation):

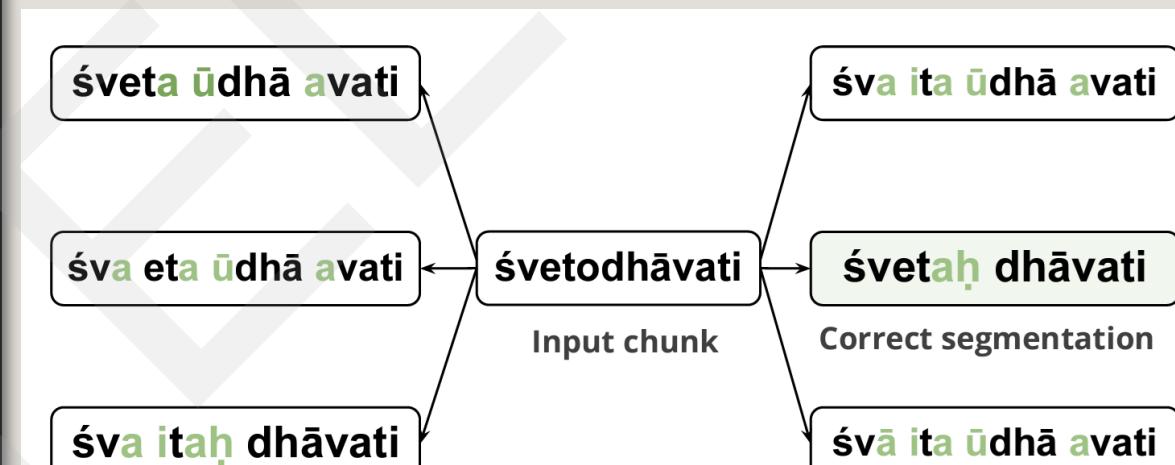
- (2.6) 姚 明 进 入 总 决 赛
Yao Ming reaches overall finals

Finally, it is possible in Chinese simply to ignore words altogether and use characters as the basic elements, treating the sentence as a series of 7 characters:

- (2.7) 姚 明 进 入 总 决 赛
Yao Ming enter enter overall decision game

In fact, for most Chinese NLP tasks it turns out to work better to take characters rather than words as input, since characters are at a reasonable semantic level for most applications, and since most word standards, by contrast, result in a huge vocabulary with large numbers of very rare words (Li et al., 2019).

Source: <https://utah-cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>



Challenges due to *sandhi* phenomena for Sanskrit Word Segmentation

Preprocessing / Text normalization

- **Lemmatization:** determining that two words have the same root, despite their surface differences
 - sang, sung, and sings are forms of sing
- **Stemming:** strip suffixes from the end of the word
- **Sentence segmentation:** Breaking up a text into individual sentences
- **Stopword removal:** Remove commonly used words in a language
 - a, the, is, are
- **Casing:** Lowercase all words or not

With pretrained language models, besides casing, we do none of the other steps

After text normalization, most tokenizers are **irreversible**

we cannot recover the raw text definitively from the tokenized output

Source: <https://utah-cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>

A redefinition of the notion of tokenization

Due to:

- Scientific results: The impact of sub-word segmentation on machine translation performance in 2016
- Technical requirements: A fixed-size vocabulary for neural language models

...in current NLP, the notion of token and tokenization changed

“Tokenization” is now the task of segmenting a sentence into non-typographically (and non-linguistically) motivated units, which are often smaller than classical tokens, and therefore often called **sub-words**

Typographic units (the “old” tokens) are now often called “**pre-tokens**”, and what used to be called “tokenization” is therefore called “**pre-tokenization**”

- https://github.com/huggingface/tokenizers/tree/main/tokenizers/src/pre_tokenizers

Source: <https://utah-cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>

Subwords are expected to be meaningful units

Subwords can be arbitrary substrings...

...but subwords can be meaning-bearing units like the morphemes -est or -er

- A **morpheme** is the smallest meaning-bearing unit of a language
 - “unlikeliest” has the morphemes {un-, likely, -est}
- **Morphology** is the study of the way words are built up from morphemes
- **Word forms** are the variations of a word that express different grammatical categories (tense, case, number, gender, etc) and thus help convey the specific meaning and function of the word in a sentence

Unseen word like lower can thus be represented by

some sequence of known subword units, such as {low, er}

Source: <https://utah-cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>

Byte-Pair-Encoding (BPE)

Main idea: Use data to automatically tell what the tokens should be

Token learner

Raw train corpus \Rightarrow Vocabulary (a set of tokens)

Token segmenter

Raw sentences \Rightarrow Tokens in the vocabulary

Source: <https://utah -cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>
[coined by [Gage et al., 1994](#); adapted to the task of word segmentation by [Sennrich et al., 2016](#); see [Gallé \(2019\)](#) for more]

Byte-Pair-Encoding (BPE) – Token learner

Raw train corpus \Rightarrow Vocabulary (a set of tokens)

- Pre-tokenize the corpus in words & append a special end-of-word symbol `_` to each word
- Initialize vocabulary with the set of all individual characters
- Choose 2 tokens that are most frequently adjacent (“A”, “B”)
 - Respect word boundaries
- Add a new merged symbol (“AB”) to the vocabulary
- Change the occurrence of the 2 selected tokens with the new merged token in the corpus
- Continues doing this until k merges are done

All k new symbols and initial characters are the final vocabulary

What's k ? Open research question

Byte-Pair-Encoding (BPE) – Example

corpus

5	l o w _
2	l o w e s t _
6	n e w e r _
3	w i d e r _
2	n e w _

vocabulary

_, d, e, i, l, n, o, r, s, t, w

Source: <https://utah-cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>

Byte-Pair-Encoding (BPE) – Example

corpus

5 l o w _
2 l o w e s t _
6 n e w **er** _
3 w i d **er** _
2 n e w _

vocabulary

_, d, e, i, l, n, o, r, s, t, w, **er**

Source: <https://utah -cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>

Byte-Pair-Encoding (BPE) – Example

corpus

5 l o w _
2 l o w e s t _
6 n e w er_
3 w i d er_
2 n e w _

vocabulary

, d, e, i, l, n, o, r, s, t, w, er, **er**

Byte-Pair-Encoding (BPE) – Example

corpus

5 l o w _

2 l o w e s t _

6 ne w er_

3 w i d er_

2 ne w _

vocabulary

, d, e, i, l, n, o, r, s, t, w, er, er, ne

Source: <https://utah -cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>

Byte-Pair-Encoding (BPE) – Example

merge	current vocabulary
(ne, w)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new
(l, o)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo
(lo, w)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low
(new, er_)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_
(low, _)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_, low_

Source: <https://utah-cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>

Byte-Pair-Encoding (BPE) – Token segmenter



Just runs on the test data the merges we have learned from the training data, greedily, in the order we learned them

First we segment each test sentence word into characters

Then we apply the first merge rule

- E.g., replace every instance of “e”, “r” in the test corpus with “er”

Then the second merge rule

- E.g., replace every instance of “er”, “_” in the test corpus with “er_”

And so on

Source: <https://utah -cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>

Byte-Pair-Encoding (BPE) Vocabulary

Model	Tokenizer	Vocabulary Size
BERT base (uncased) [2018]	WordPiece	30,522
BERT base (cased) [2018]	WordPiece	28,996
GPT-2 [2019]	BPE	50,257
Flan-T5 [2022]	SentencePiece	32,100
GPT-4 [2023]	BPE	> 100,000
StarCoder2 [2024]	BPE	49,152
Llama2 [2023]	BPE	32,000

You can play with different tokenizers here: <https://tiktoktokenizer.vercel.app/>

Subwords - Example

GPT-3.5 & GPT-4 GPT-3 (Legacy)

Have the bards who preceded me left any theme unsung?

Clear

Show example

Tokens

13

Characters

53

Have the bards who preceded me left any theme unsung?

Text

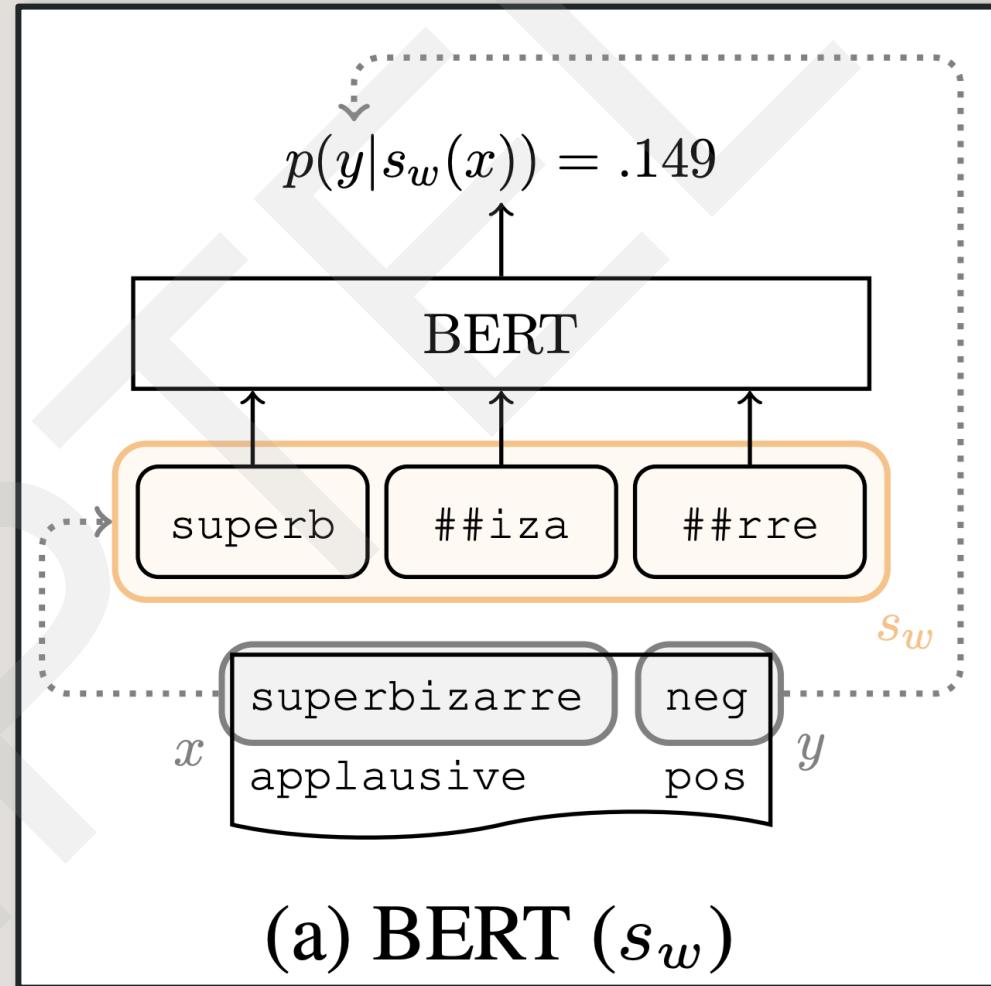
Token IDs

Source: Alammar, J., & Grootendorst, M. (2024). Hands-On Large Language Models. O'Reilly.

Byte-Pair-Encoding (BPE) Implications

[Hofmann et al., 2021]

BERT thinks the sentiment of "superbizarre" is positive because its tokenization contains the token "superb"



Source: <https://utah-cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>

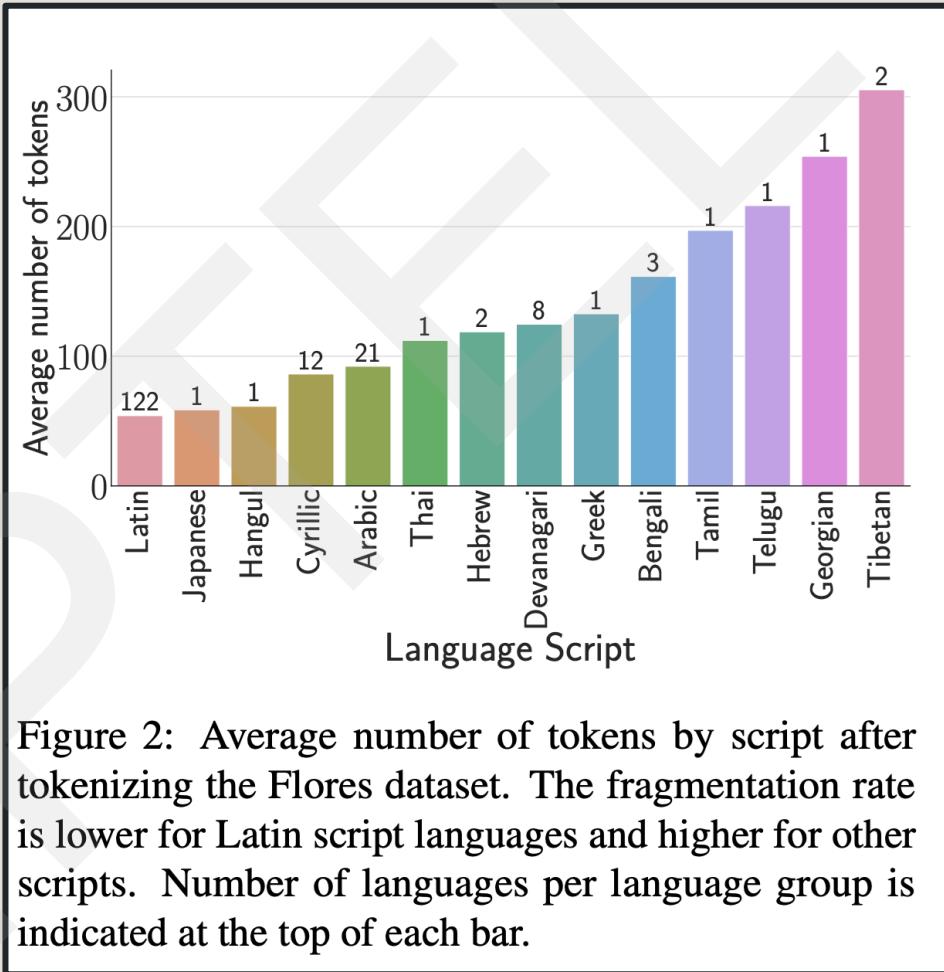
Byte-Pair-Encoding (BPE) Implications – Do All languages cost the same?

[Ahia et al., 2023]

Proprietary models, as GPT4, are accessible only through paid APIs

API cost is measured by the number of tokens processed or generated

Subword tokenizers lead to disproportionate fragmentation rates for different languages and writing scripts



Source: <https://utah -cs6340-nlp.notion.site/Natural-Language-Processing-bd1a2ca290fc44f69556908ad8d25c70>

Other subword encoding schemes

WordPiece (Schuster et al., ICASSP 2012): merge by likelihood as measured by language model, not by frequency

SentencePiece (Kudo et al., 2018): can do subword tokenization without pretokenization (good for languages that don't always separate words w/ spaces), although pretokenization usually improves performance

REFERENCES

- Daniel Jurafsky and James H. Martin. 2024. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. Online manuscript released August 20, 2024. <https://web.stanford.edu/~jurafsky/slp3>. [Chapter 2]



THANK YOU



N P T E L O N L I N E C E R T I F I C A T I O N C O U R S E S

DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING

Lecture 03 : N-gram Language Models: Part 1



PROF . PAWAN GOYAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

CONCEPTS COVERED

- What is Language Modeling? (LM in LLMs!!)
- N-gram Language Models
- Some Practical Issues

Predicting words

- The water of Walden Pond is beautifully ...

blue
green
clear

*refrigerator
*that

Language Models

Systems that can predict upcoming words

- Can assign a probability to each potential next word
- Can assign a probability to a whole sentence

Source: <https://web.stanford.edu/~jurafsky/slp3> .

Why word prediction?

It's a helpful part of language tasks

- Grammar or spell checking

Their are two midterms

Their There are two midterms

Everything has improve

Everything has improve improved

- Speech recognition

I will be back soonish

I will be bassoon dish

Why word prediction?

It's how **large language models (LLMs)** work!

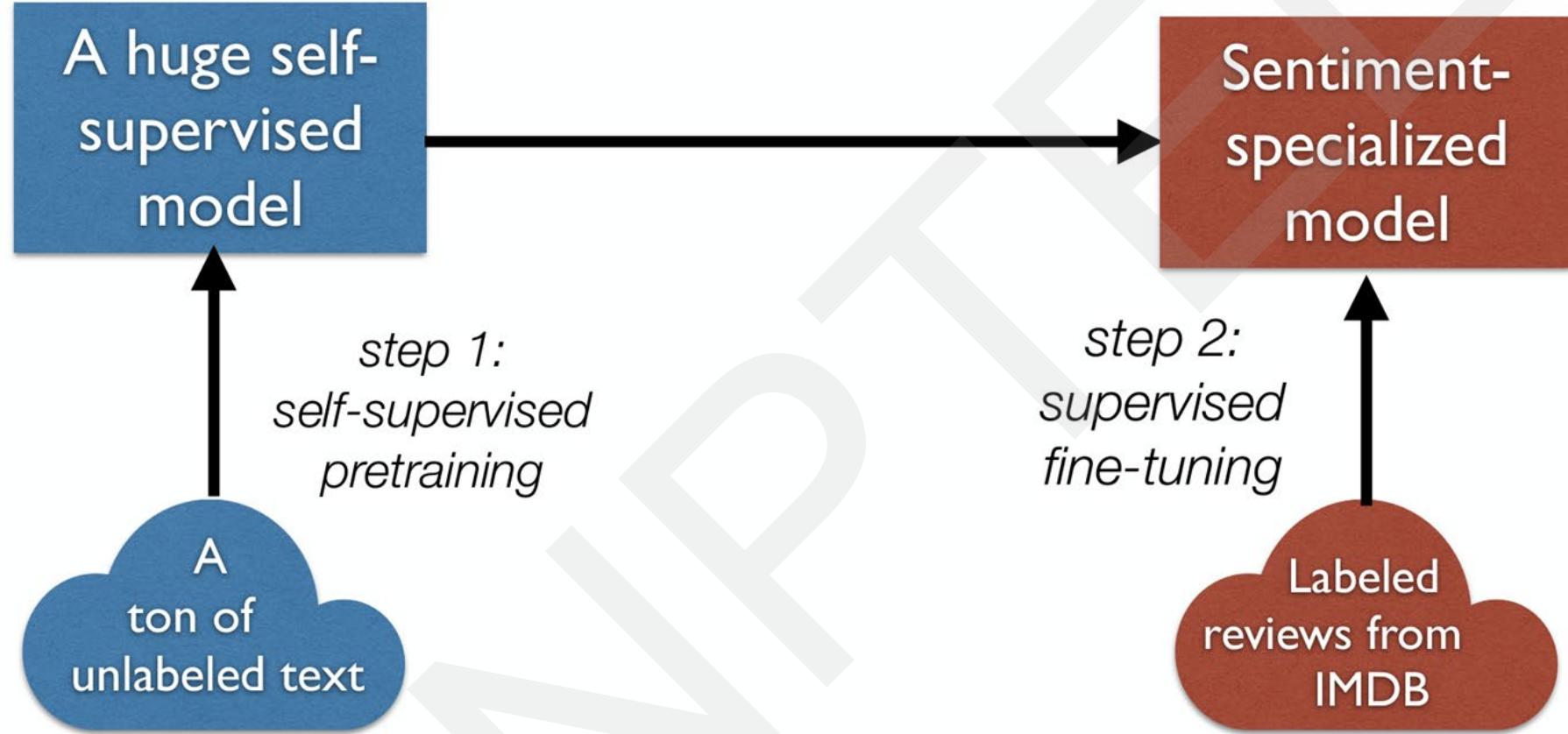
LLMs are **trained** to predict words

- Left-to-right (autoregressive) LMs learn to predict next word

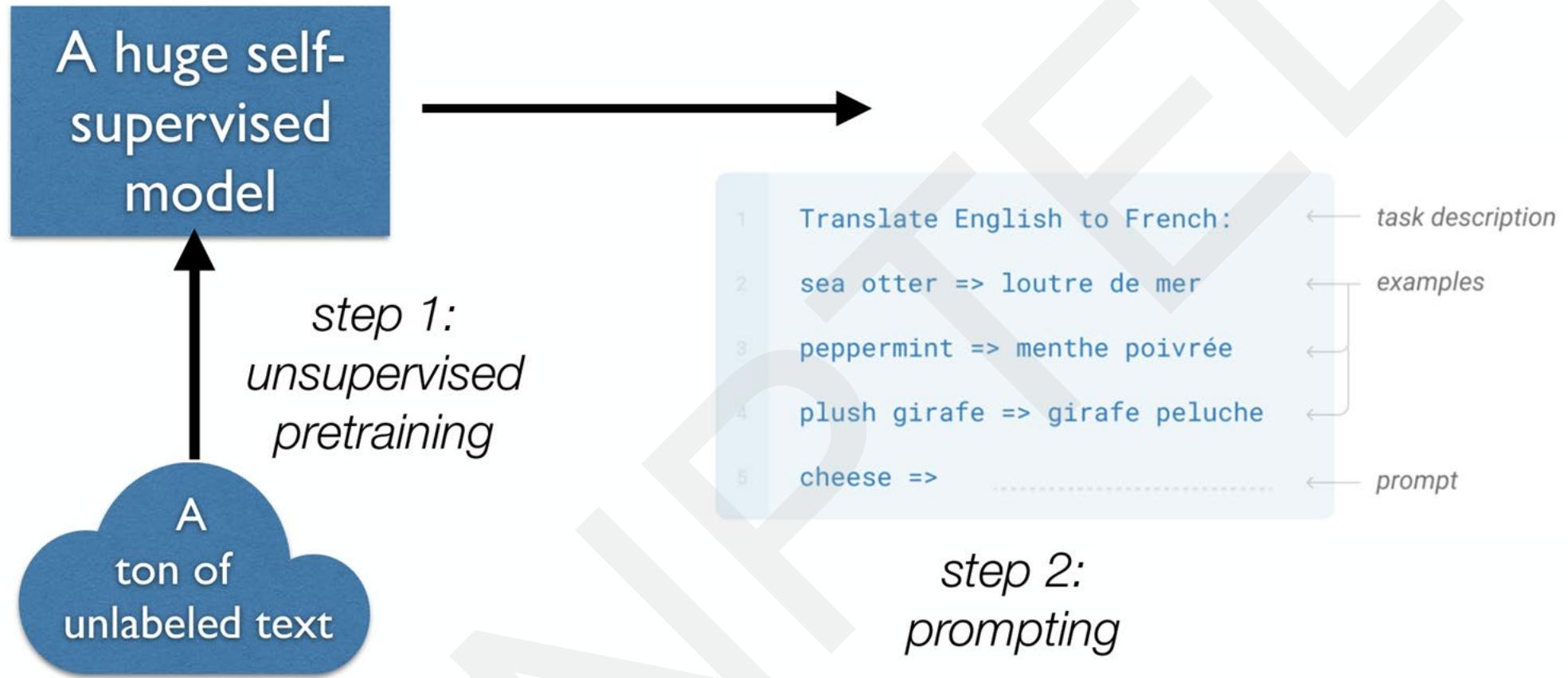
LLMs **generate** text by predicting words

- By predicting the next word over and over again

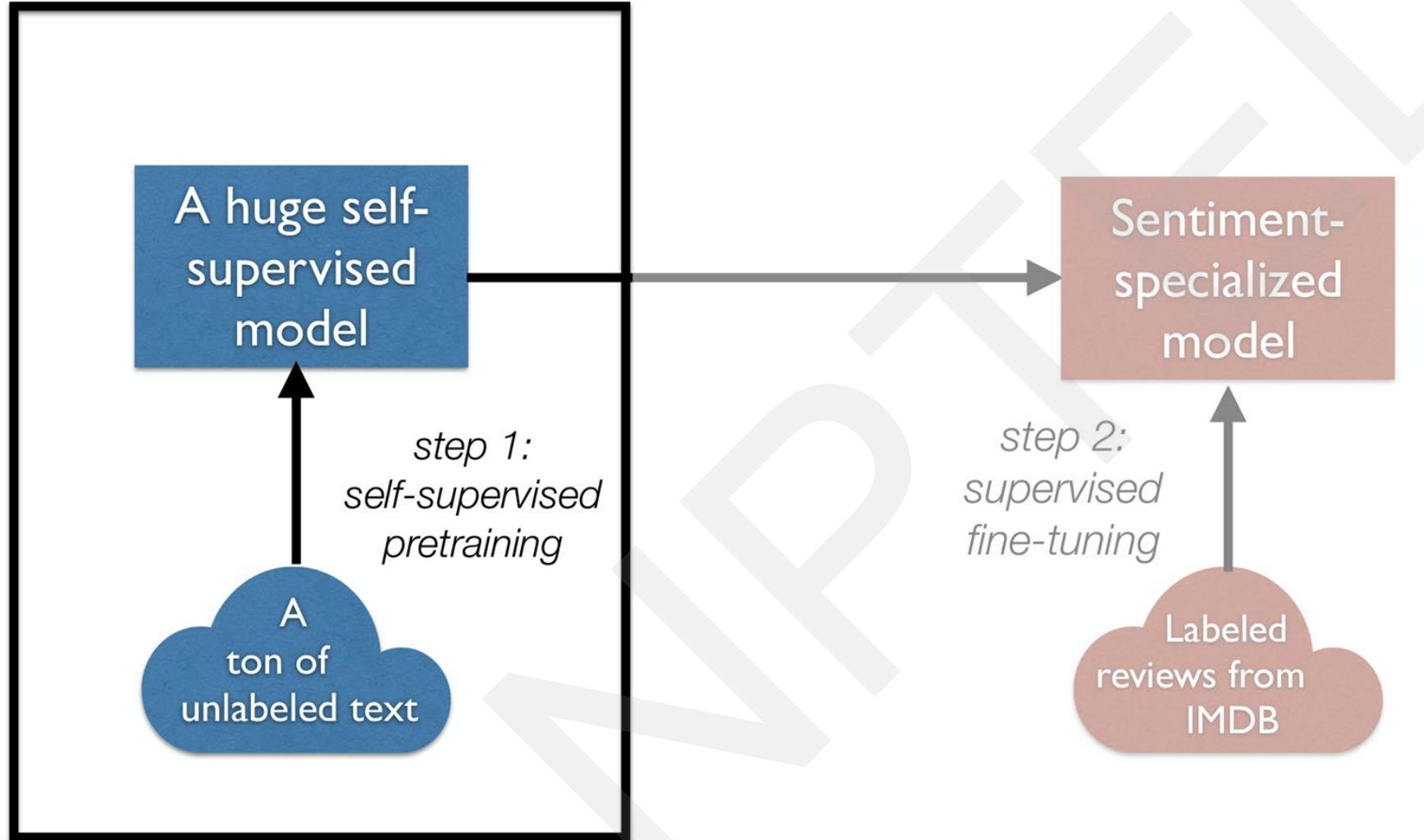
Pretrain-then-finetune paradigm



Pretrain-then-Prompt paradigm

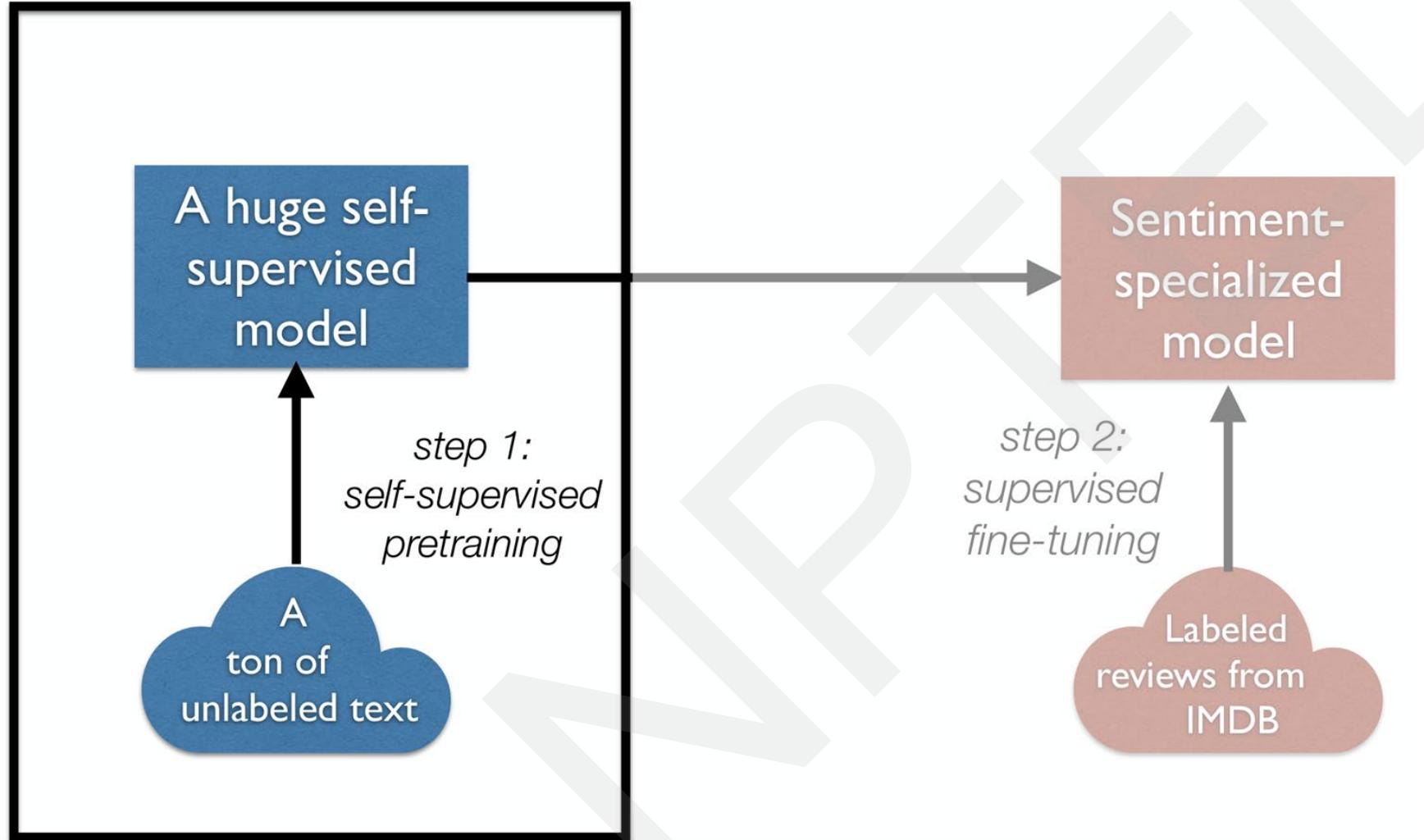


Language modeling forms the core of most self-supervised NLP approaches



Source: <https://people.cs.umass.edu/~miyyer/cs685>

Language modeling forms the core of most self-supervised NLP approaches



Source: <https://people.cs.umass.edu/~miyyer/cs685>

Language Modeling: More Formally

- **Goal:** Compute the probability of a sentence or sequence of words:

$$P(W) = P(w_1, w_2, w_3, \dots, w_n)$$

- **Related Task:** probability of an upcoming word:

$$P(w_4 | w_1, w_2, w_3)$$

- A model that computes either of these is called a **language model**

How to compute $P(W)$ or $P(w_n | w_1, \dots w_{n-1})$

- How to compute the joint probability $P(W)$:

$P(\text{The, water, of, Walden, Pond, is, so, beautifully, blue})$

- Intuition: let's rely on the Chain Rule of Probability

Source: <https://web.stanford.edu/~jurafsky/slp3> .

Reminder: The Chain Rule

- Recall the definition of conditional probabilities

$$P(B|A) = P(A,B)/P(A) \quad \text{Rewriting: } P(A,B) = P(A) P(B|A)$$

- More variables:

$$P(A,B,C,D) = P(A) P(B|A) P(C|A,B) P(D|A,B,C)$$

- The Chain Rule in General

$$P(x_1, x_2, x_3, \dots, x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)\dots P(x_n|x_1, \dots, x_{n-1})$$

Source: <https://web.stanford.edu/~jurafsky/slp3> .

The Chain Rule applied to compute joint probability of words in sentence

$$\begin{aligned} P(w_{1:n}) &= P(w_1)P(w_2|w_1)P(w_3|w_{1:2}) \dots P(w_n|w_{1:n-1}) \\ &= \prod_{k=1}^n P(w_k|w_{1:k-1}) \end{aligned}$$

$P(\text{"The water of Walden Pond is so beautifully blue"}) =$
 $P(\text{The}) \times P(\text{water}|\text{The}) \times P(\text{of}|\text{The water})$
 $\quad \times P(\text{Walden}|\text{The water of}) \times$
 $P(\text{Pond}|\text{The water of Walden}) \times \dots$

Source: <https://web.stanford.edu/~jurafsky/slp3> .

How to estimate these probabilities

- Could we just count and divide?

$P(\text{blue} | \text{The water of Walden Pond is so beautifully blue})$

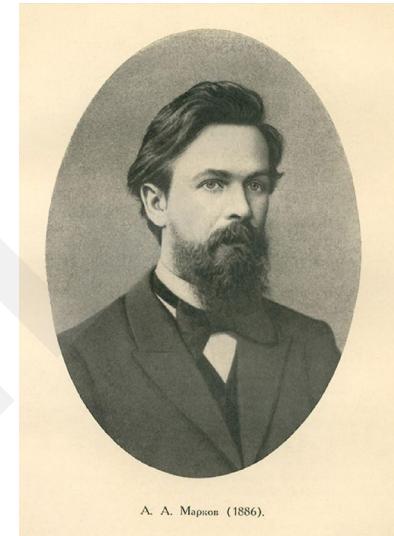
=

$$\frac{C(\text{The water of Walden Pond is so beautifully blue})}{C(\text{The water of Walden Pond is so beautifully not blue})}$$

- *We'll never see enough data for estimating these!!*

Markov Assumption

- Simplifying assumption:



Andrei Markov

$P(\text{blue} | \text{The water of Walden Pond is so beautifully})$

$$\approx P(\text{blue} | \text{beautifully})$$

$$P(w_n | w_{1:n-1}) \approx P(w_n | w_{n-1})$$

Source: <https://web.stanford.edu/~jurafsky/slp3> .

Bigram Markov Assumption

$$P(w_{1:n}) \approx \prod_{k=1}^n P(w_k | w_{k-1})$$

Instead of:

$$\prod_{k=1}^n P(w_k | w_{1:k-1})$$

More generally, we approximate each component in the product

$$P(w_n | w_{1:n-1}) \approx P(w_n | w_{n-N+1:n-1})$$

Simplest case: Unigram model

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i)$$

Some automatically generated sentences from two different unigram models

To him swallowed confess hear both . Which . Of save on trail
for are ay device and rote life have

Hill he late speaks ; or ! a more to leg less first you enter

Months the my and issue of year foreign new exchange's
September

were recession exchange new endorsed a acquire to six
executives

Source: <https://web.stanford.edu/~jurafsky/slp3> .

Bigram model

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

Some automatically generated sentences from two different unigram models

Why dost stand forth thy canopy, forsooth; he is this
palpable hit the King Henry. Live king. Follow.

What means, sir. I confess she? then all sorts, he is trim,
captain.

Last December through the way to preserve the Hudson
corporation N. B. E. C. Taylor would seem to complete the
major central planners one gram point five percent of U. S.
E. has already old M. X. corporation of living

Approximating Shakespeare

Unigram

To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have
Every enter now severally so, let
Hill he late speaks; or! a more to leg less first you enter
Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like

Bigram

What means, sir. I confess she? then all sorts, he is trim, captain.
Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.
What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?

Trigram

Sweet prince, Falstaff shall die. Harry of Monmouth's grave.
This shall forbid it should be branded, if renown made it empty.
Indeed the duke; and had a very good friend.
Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.

Quadrigram

King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;
Will you not tell me who I am?
It cannot be but so.
Indeed the short and the long. Marry, 'tis a noble Lepidus.

Problems with N-gram models

- N-grams can't handle **long-distance dependencies**:

"The soups that I made from that new cookbook I bought yesterday were amazingly delicious."

- N-grams don't do well at modeling new sequences with similar meanings

The solution: **Large language models**

- can handle much longer contexts
(because of using embedding spaces)
- can model synonymy better

Why N-gram models?

A nice clear paradigm that lets us introduce many of the important issues for **large language models**

- **training** and **test** sets
- the **perplexity** metric
- **sampling** to generate sentences
- ideas like **interpolation** and **backoff**

Source: <https://web.stanford.edu/~jurafsky/slp3> .

Estimating n-gram probabilities

Maximum Likelihood Estimate

Value that makes the observed data the “most probable”

$$P(w_i|w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

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

Estimating n-gram probabilities: an Example



Given a corpus C, the bigram probability of “paper | question” is 0.3 and the count of occurrences of the word “question” is 600. What will be the frequency of the pair (question, paper) in the corpus C?

$$P(\text{paper} \mid \text{question}) = \text{freq}(\text{question}, \text{paper}) / \text{freq}(\text{question})$$

$$\text{freq}(\text{question}, \text{paper}) = 180$$

An Example

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

< s > I am here < /s >
< s > who am I < /s >
< s > I would like to know < /s >

Estimating bigrams

$$P(I|< s >) = 2/3$$

$$P(< /s >|here) = 1$$

$$P(would | I) = 1/3$$

$$P(here | am) = 1/2$$

$$P(know | like) = 0$$

Computing Sentence Probabilities

$P(< s > \text{ I want english food } </ s >)$

$$= P(\text{I} | < s >) \times P(\text{want} | \text{I}) \times P(\text{english} | \text{want}) \times P(\text{food} | \text{english}) \times P(</ s > | \text{food})$$

Practical Issues

Everything in log space

- Avoids underflow
- Adding is faster than multiplying

$$\log(p_1 \times p_2 \times p_3 \times p_4) = \log p_1 + \log p_2 + \log p_3 + \log p_4$$

REFERENCES

- Daniel Jurafsky and James H. Martin. 2024. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. Online manuscript released August 20, 2024. [Chapter 3]



THANK YOU



N P T E L O N L I N E C E R T I F I C A T I O N C O U R S E S

DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING

Lecture 04 : N-gram Language Models: Part 2



PROF . PAWAN GOYAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

CONCEPTS COVERED

- Smoothing in n-gram LMs
- Evaluation: Perplexity
- Sampling from the distribution for generation
- Larger n-grams

Some Important Points with N-gram LMs

- For unigram counts, $P(w)$ is always non-zero
 - if our dictionary is derived from the document collection
- This won't be true of $P(w_k | w_{k-1})$. Let's take an example below.

Training set

- ... denied the allegations
- ... denied the reports
- ... denied the claims
- ... denied the request

Test Data

- ... denied the offer
- ... denied the loan

$P(\text{offer} | \text{denied the}) = 0$, the test sentence will be assigned a probability of 0!

Smoothing

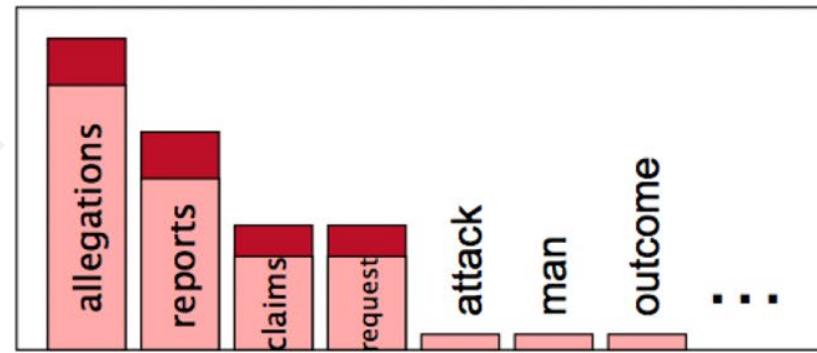
With sparse statistics

$P(w | \text{denied the})$
3 allegations
2 reports
1 claims
1 request
7 total



Steal probability mass to generalize better

$P(w | \text{denied the})$
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other
7 total



Add-1 Smoothing

- Pretend as if we saw each word (N-gram) one more time than we actually did
- Just add one to all the counts!
- MLE estimate for bigram: $P_{MLE}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$
- Add-1 estimate: $P_{Add-1}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + V}$

Add-1 Smoothing: Example

Given a corpus C, the bigram probability of “paper | question” is 0.3 and the count of occurrences of the word “question” is 600. What will be the frequency of the pair (question, paper) in the corpus C? → *We got this as 180*

Now, suppose the vocabulary size is 1210. What will be the probability of “paper | question” after add-1 smoothing?

$$\begin{aligned} P_{add_1}(\text{"paper | question"}) &= \\ &(freq(\text{question}, \text{paper})+1)/(freq(\text{question})+V) \\ &= (180+1)/(600+1210) \\ &= 0.1 \end{aligned}$$

More General Formulations

$$P_{Add-k}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + k}{c(w_{i-1}) + kV}$$

$$P_{Add-k}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + m(\frac{1}{V})}{c(w_{i-1}) + m}$$

Interpolate with unigram probabilities instead

$$P_{UnigramPrior}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + mP(w_i)}{c(w_{i-1}) + m}$$

Backoff and Interpolation

- Sometimes it helps to use **less context**
 - Condition on less context for contexts you know less about
- **Backoff:**
 - use trigram if you have good evidence,
 - otherwise bigram, otherwise unigram
- **Interpolation:**
 - mix unigram, bigram, trigram
- Interpolation works better

How to evaluate N-gram models

- "Extrinsic (in-vivo) Evaluation"

To compare models A and B

1. Put each model in a real task
 - Machine Translation, speech recognition, etc.
2. Run the task, get a score for A and for B
 - How many words translated correctly
 - How many words transcribed correctly
3. Compare accuracy for A and B

Intrinsic (in-vitro) evaluation

- Extrinsic evaluation not always possible
 - Expensive, time-consuming
 - Doesn't always generalize to other applications
- Intrinsic evaluation: **perplexity**
 - Directly measures language model performance at predicting words.
 - Doesn't necessarily correspond with real application performance
 - But gives us a single general metric for language models
 - Useful for large language models (LLMs) as well as n-grams

Training sets and test sets

We train parameters of our model on a **training set**.

We test the model's performance on data we haven't seen.

- A **test set** is an unseen dataset; different from training set.
 - Intuition: we want to measure generalization to unseen data
- An **evaluation metric** (like **perplexity**) tells us how well our model does on the test set.

Choosing training and test sets

- If we're building an LM for a specific task
 - The test set should reflect the task language we want to use the model for
- If we're building a general-purpose model
 - We'll need lots of different kinds of training data
 - We don't want the training set or the test set to be just from one domain or author or language.

Training on the test set

We can't allow test sentences into the training set

- Or else the LM will assign that sentence an artificially high probability when we see it in the test set
- And hence assign the whole test set a falsely high probability.
- Making the LM look better than it really is

This is called “**Training on the test set**”

Bad science!

Dev sets

- If we test on the test set many times we might implicitly tune to its characteristics
 - Noticing which changes make the model better.
- So we run on the test set only once, or a few times
- That means we need a third dataset:
 - A **development test set** or, **devset**.
 - We test our LM on the devset until the very end
 - And then test our LM on the **test set** once

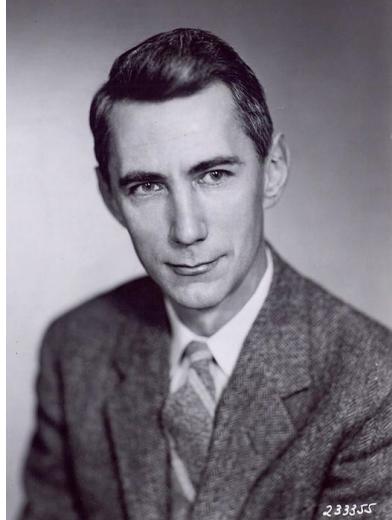
Intuition of perplexity as evaluation metric: How good is our language model?

Intuition: A good LM prefers "real" sentences

- Assign higher probability to “real” or “frequently observed” sentences
- Assigns lower probability to “word salad” or “rarely observed” sentences?

Intuition of perplexity 2:

Predicting upcoming words



Claude Shannon

The Shannon Game: **How well can we predict the next word?**

- Once upon a _____
- That is a picture of a _____
- For breakfast I ate my usual _____

time 0.9
dream 0.03
midnight 0.02
...
and 1e-100

Unigrams are terrible at this game (Why?)

A good LM is one that assigns a higher probability to the next word that actually occurs

Intuition of perplexity 3: The best language model is one that best predicts the entire unseen test set

- We said: a good LM is one that assigns a higher probability to the next word that actually occurs.
- Let's generalize to all the words!
 - The best LM assigns high probability to the entire test set.
- When comparing two LMs, A and B
 - We compute $P_A(\text{test set})$ and $P_B(\text{test set})$
 - The better LM will give a higher probability to
(=be less surprised by) the test set than the other LM.

Intuition of perplexity 4: Use perplexity instead of raw probability

- Probability depends on size of test set
 - Probability gets smaller the longer the text
 - Better: a metric that is **per-word**, normalized by length
- **Perplexity** is the inverse probability of the test set, normalized by the number of words

$$\begin{aligned} PP(W) &= P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}} \end{aligned}$$

Intuition of perplexity 5: the inverse

Perplexity is the **inverse** probability of the test set, normalized by the number of words

$$\begin{aligned} PP(W) &= P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}} \end{aligned}$$

(The inverse comes from the original definition of perplexity from cross-entropy rate in information theory)

Probability range is $[0,1]$, perplexity range is $[1, \infty]$

Minimizing perplexity is the same as maximizing probability

Intuition of perplexity 6: N-grams

$$PP(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}}$$

Chain rule:

$$PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i | w_1 \dots w_{i-1})}}$$

Bigrams:

$$PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i | w_{i-1})}}$$

Source: <https://web.stanford.edu/~jurafsky/slp3> .

Intuition of perplexity 7: Weighted average branching factor

Perplexity is also the **weighted average branching factor** of a language.

Branching factor: number of possible next words that can follow any word

Example: Deterministic language $L = \{\text{red, blue, green}\}$

Branching factor = 3 (any word can be followed by red, blue, green)

Now assume LM A where each word follows any other word with equal probability $\frac{1}{3}$

Given a test set $T = \text{"red red red red blue"}$

$$\text{Perplexity}_A(T) = P_A(\text{red red red red blue})^{-1/5} = 3$$

- But now suppose red was very likely in training set, such that for LM B:

- $P(\text{red}) = .8 \quad p(\text{green}) = .1 \quad p(\text{blue}) = .1$

- We would expect the probability to be higher, and hence the perplexity to be smaller:

$$\text{Perplexity}_B(T) = P_B(\text{red red red red blue})^{-1/5}$$

$$= (.8 * .8 * .8 * .8 * .1)^{-1/5} = .04096^{-1/5} = .527^{-1} = 1.89$$

Holding test set constant: Lower perplexity = better language model

- Training 38 million words, test 1.5 million words, WSJ

N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109

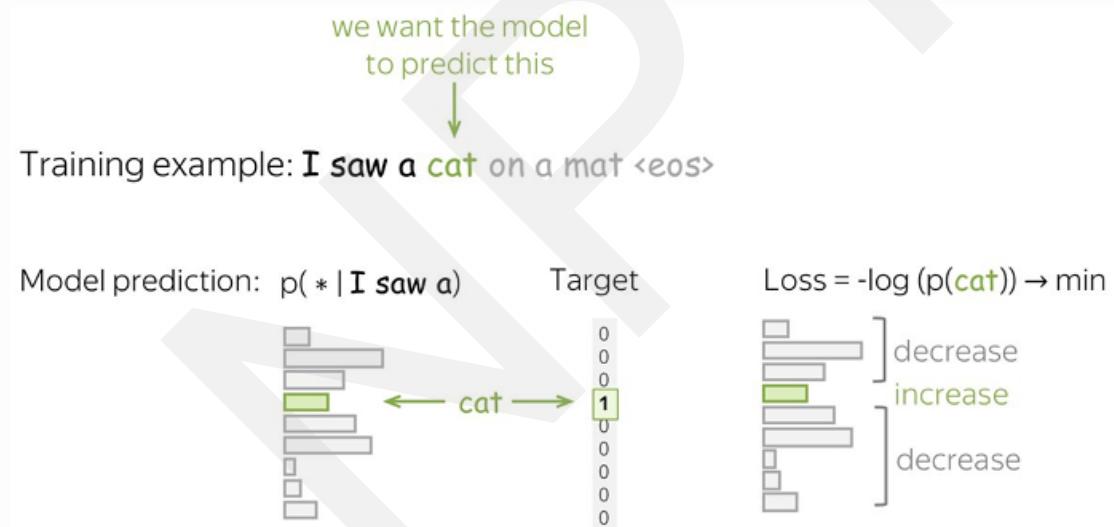
Heavily abbreviated history of LMs

1948: Claude Shannon models English

1948-2017: 

$$Loss(p^*, p) = -\log(p_{y_t}) = -\log(p(y_t|y_{<t})).$$

At each step, we maximize the probability a model assigns to the correct token. Look at the illustration for a single timestep.



The Shannon (1948) Visualization Method

Sample words from an LM



Claude Shannon

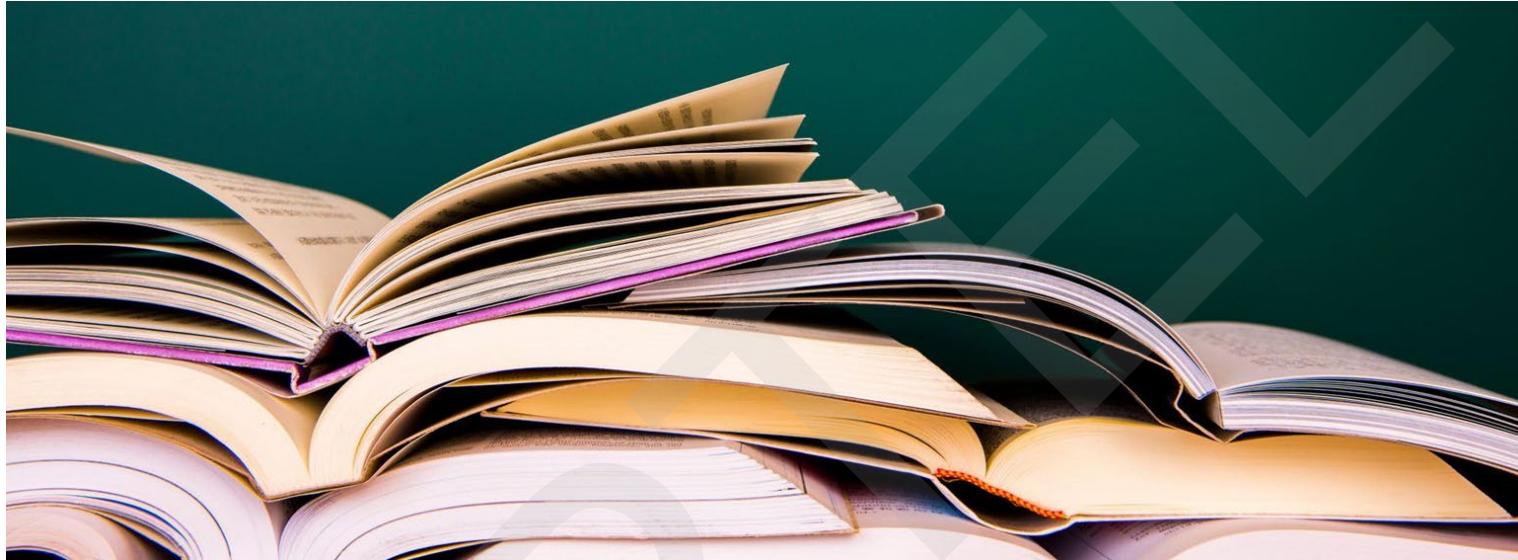
- **Unigram:**

REPRESENTING AND SPEEDILY IS AN GOOD APT OR
COME CAN DIFFERENT NATURAL HERE HE THE A IN
CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES
THE LINE MESSAGE HAD BE THESE.

- **Bigram:**

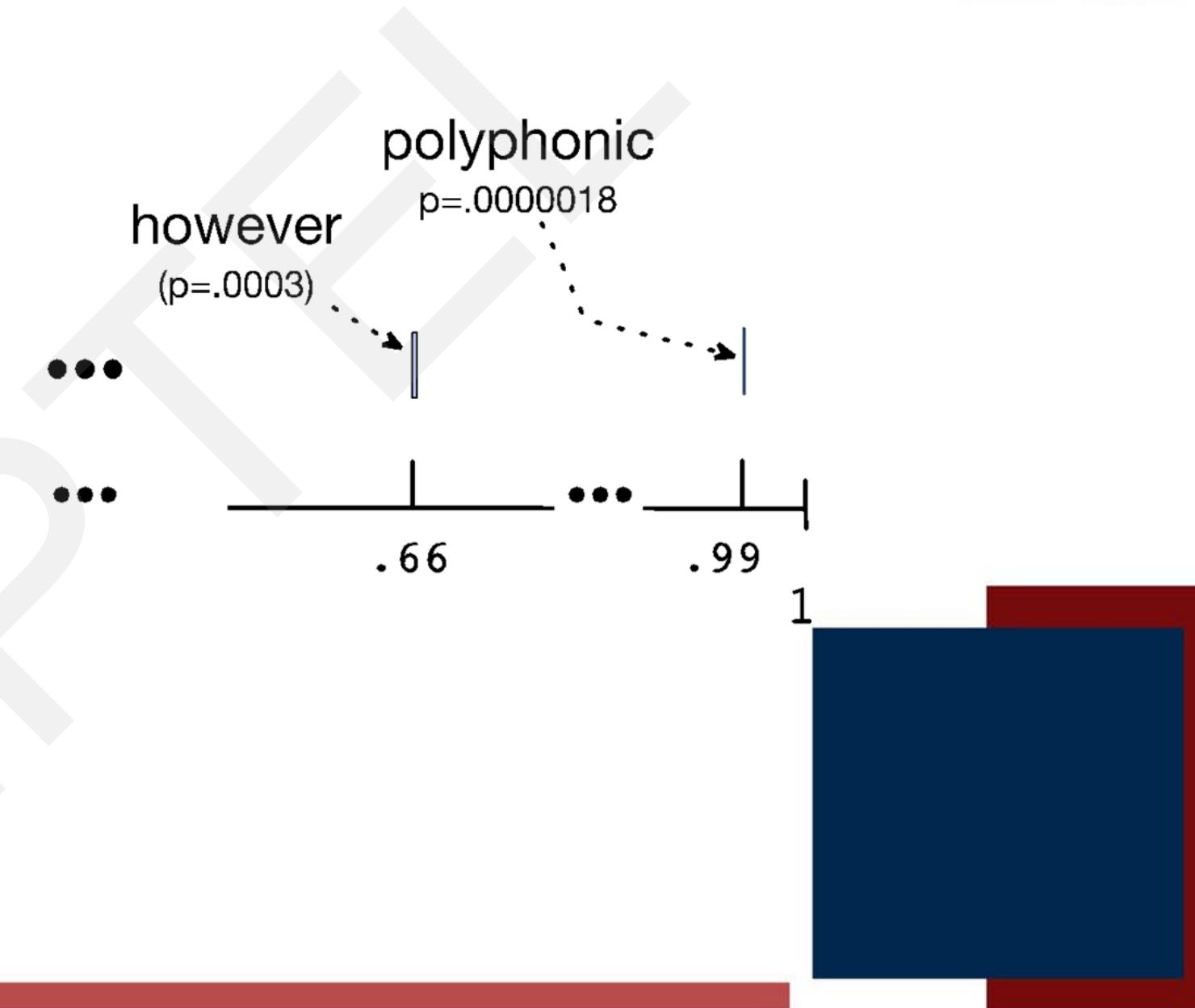
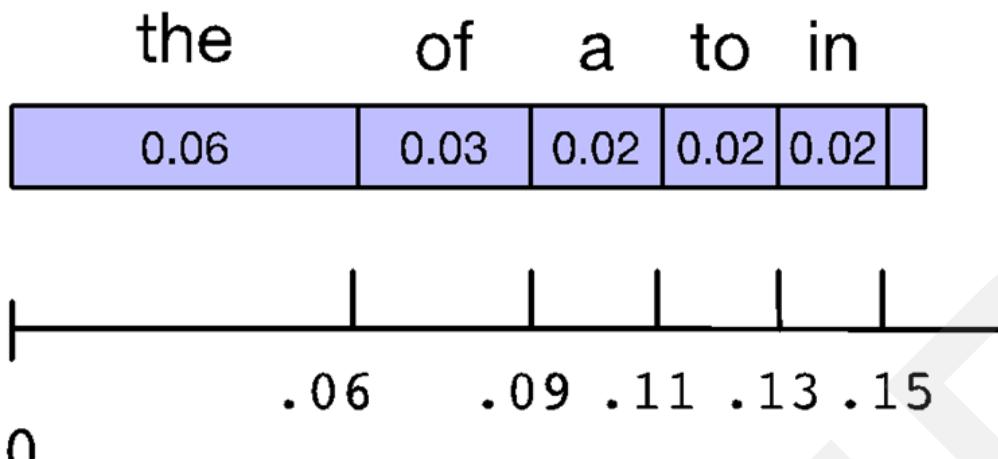
THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH
WRITER THAT THE CHARACTER OF THIS POINT IS
THEREFORE ANOTHER METHOD FOR THE LETTERS THAT
THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN
UNEXPECTED.

How Shannon sampled those words in 1948



"Open a book at random and select a letter at random on the page. This letter is recorded. The book is then opened to another page and one reads until this letter is encountered. The succeeding letter is then recorded. Turning to another page this second letter is searched for and the succeeding letter recorded, etc."

Sampling a word from a distribution



The Shannon Visualization Method

Use the language model to generate word sequences

- Choose a random bigram ($< s >$, w) as per its probability
- Choose a random bigram (w, x) as per its probability
- And so on until we choose $< /s >$

$< s >$ I
I want
want to
to eat
eat Chinese
Chinese food
food $< /s >$

I want to eat Chinese food

Note: there are other sampling methods

Used for neural language models

Many of them avoid generating words from the very unlikely tail of the distribution

We'll discuss when we get to neural LM decoding:

- Temperature sampling
- Top-k sampling
- Top-p sampling

Larger ngrams

- 4-grams, 5-grams
 - Large datasets of large n-grams have been released
 - N-grams from Corpus of Contemporary American English (COCA) 1 billion words (Davies 2020)
 - Google Web 5-grams (Franz and Brants 2006) 1 trillion words)
 - Efficiency: quantize probabilities to 4-8 bits instead of 8-byte float
- Newest model: infini-grams (∞ -grams) (Liu et al 2024)
- No precomputing! Instead, store 5 trillion words of web text in **suffix arrays**.
 - Can compute n-gram probabilities with any n!

∞ -grams

Prompt

... conducts research at the Paul G. Allen School of Computer Science and Engineering, University of

5-gram LM $(n = 5)$

$\text{cnt}(\text{Engineering, University of}) = 274644$

$P(*) | \text{Engineering, University of} =$



∞ -gram LM $(n = 16 \text{ for this case})$

$\text{cnt}(\text{research at the Paul G. Allen School of Computer Science and Engineering, University of}) = 0$

$\text{cnt}(\text{at the Paul G. Allen School of Computer Science and Engineering, University of}) = 10$

$P(*) | \text{at the Paul G. Allen School of Computer Science and Engineering, University of} =$

$\text{Washington (10 / 10) } 100\%$



Figure 1: An example where a 5-gram LM gives an incorrect prediction but the ∞ -gram gives the correct prediction by using the longest suffix of the prompt that has a non-zero count in the corpus. The counting and distribution estimate in ∞ -gram LM are powered by our infini-gram engine.

Source: <https://arxiv.org/pdf/2401.17377>

∞ -grams as a variant of back-off

Prompt

... conducts research at the Paul G. Allen School of Computer Science and Engineering, University of

5-gram LM ($n = 5$)

cnt(Engineering, University of) = 274644

$P(* | \text{Engineering, University of}) =$

_California (20896 / 274644)	8%
_Illinois (10631 / 274644)	4%
_Michigan (9094 / 274644)	3%
_Colorado (6438 / 274644)	2%
_Southern (6340 / 274644)	2%
_Washington (6340 / 274644)	2%
...	

∞ -gram LM ($n = 16$ for this case)

cnt(research at the Paul G. Allen School of Computer Science and Engineering, University of) = 0

$P(* | \text{at the Paul G. Allen School of Computer Science and Engineering, University of}) =$

_Washington (10 / 10) 100%



$$P_{\infty}(w_i | w_{1:i-1}) = \frac{\text{cnt}(w_{i-(n-1):i-1} w_i | \mathcal{D})}{\text{cnt}(w_{i-(n-1):i-1} | \mathcal{D})}$$

where $w_{1:i-1}$ are all tokens preceding w_i in the document, and

$$n = \max\{n' \in [1, i] \mid \text{cnt}(w_{i-(n'-1):i-1} | \mathcal{D}) > 0\}$$

Source: <https://arxiv.org/pdf/2401.17377>

N-gram LM Toolkits

- SRILM
 - <http://www.speech.sri.com/projects/srilm/>
- KenLM
 - <https://kheafield.com/code/kenlm/>

REFERENCES

- Daniel Jurafsky and James H. Martin. 2024. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. Online manuscript released August 20, 2024. [Chapter 3]



THANK YOU



N P T E L O N L I N E C E R T I F I C A T I O N C O U R S E S

DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING

Lecture 05 : NLP Tasks and Paradigms



PROF . PAWAN GOYAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

CONCEPTS COVERED

- Paradigms in NLP
- Text Classification, Sequence Labeling, Text Generation, Structured Prediction
- Some NLP Tasks

NLP Paradigms

We generally try to map NLP problems to various (ML) paradigms

- Sentiment Analysis, news article groupings, etc. → **Text Classification**
- Named entity recognition, code-mixing, etc. → **Sequence Labeling**
- Machine Translation, summarization, chatbots, etc. → **Text Generation**

Other popular paradigm: **Structured Prediction**

Example of NLP tasks

Word / Span Level: Word sense disambiguation, Entity Linking

Sentence Level: Sentence Similarity, Natural Language Inference

Paragraph / Document Level: Question Answering

NLP Paradigms: Classification

Classification : Positive or negative review?

- + ...zany characters and richly applied satire, and some great plot twists
- It was pathetic. The worst part about it was the boxing scenes...
- + ...awesome caramel sauce and sweet toasty almonds. I love this place!
- ...awful pizza and ridiculously overpriced...

Why sentiment analysis?

Movie: is this review positive or negative?

Products: what do people think about the new iPhone?

Politics: what do people think about this candidate or issue?

Prediction: predict election outcomes or market trends from sentiment

Text Classification: formal definition

Input:

- a document d
- a fixed set of classes $C = \{c_1, c_2, \dots, c_J\}$

Output: a predicted class $c \in C$

Classification Methods: Machine Learning

Input:

- a document d
- a fixed set of classes $C = \{c_1, c_2, \dots, c_J\}$
- A training set of m hand-labeled documents $(d_1, c_1), \dots, (d_m, c_m)$

Output:

- a learned classifier $\gamma: d \rightarrow c$

Any kind of classifier

- Naïve Bayes
- Support Vector Machines
- Neural networks
- k-Nearest Neighbors
- ...

Evaluation for Text Classification

Let's consider just binary text classification tasks

Imagine you're the CEO of Delicious Pie Company

You want to know what people are saying about your pies

So you build a "Delicious Pie" tweet detector

- Positive class: tweets about Delicious Pie Co
- Negative class: all other tweets

The 2-by-2 confusion matrix

		<i>gold standard labels</i>		
		gold positive	gold negative	
<i>system output labels</i>	system positive	true positive	false positive	$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$
	system negative	false negative	true negative	
		$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$	$\text{accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{fp} + \text{tn} + \text{fn}}$	

Evaluation: Accuracy

*Why don't we use **accuracy** as our metric?*

Imagine we saw 1 million tweets

- 100 of them talked about Delicious Pie Co.
- 999,900 talked about something else

We could build a dumb classifier that just labels every tweet "not about pie"

- It would get 99.99% accuracy!!! Wow!!!!
- But useless! Doesn't return the comments we are looking for!
- That's why we use **precision** and **recall** instead

Why Precision and recall

Our dumb pie-classifier

- Just label nothing as "about pie"

Accuracy=99.99%

but

Recall = 0

- (it doesn't get any of the 100 Pie tweets)

Precision and recall, unlike accuracy, emphasize true positives:

- finding the things that we are supposed to be looking for.

A combined measure: F

F measure: a single number that combines P and R:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2P + R}$$

We almost always use balanced F_1 (i.e., $\beta = 1$)

$$F_1 = \frac{2PR}{P + R}$$

Confusion Matrix for 3-class classification

		<i>gold labels</i>			
		urgent	normal	spam	
<i>system output</i>	urgent	8	10	1	$\text{precision}_u = \frac{8}{8+10+1}$
	normal	5	60	50	$\text{precision}_n = \frac{60}{5+60+50}$
	spam	3	30	200	$\text{precision}_s = \frac{200}{3+30+200}$
		$\text{recall}_u = \frac{8}{8+5+3}$	$\text{recall}_n = \frac{60}{10+60+30}$	$\text{recall}_s = \frac{200}{1+50+200}$	

How to combine P/R from 3 classes to get one metric

Macro-averaging:

- compute the performance for each class, and then average over classes

Micro-averaging:

- collect decisions for all classes into one confusion matrix
- compute precision and recall from that table.

Macro-averaging and Micro-averaging

Class 1: Urgent

	true	true
urgent	urgent	not
system	8	11
not	8	340

Class 2: Normal

	true	true
normal	normal	not
system	60	55
not	40	212

Class 3: Spam

	true	true
spam	spam	not
system	200	33
not	51	83

Pooled

	true	true
yes	yes	no
system	268	99
no	99	635

$$\text{precision} = \frac{8}{8+11} = .42$$

$$\text{precision} = \frac{60}{60+55} = .52$$

$$\text{precision} = \frac{200}{200+33} = .86$$

$$\text{microaverage precision} = \frac{268}{268+99} = .73$$

$$\text{macroaverage precision} = \frac{.42+.52+.86}{3} = .60$$

NLP Paradigms: Sequence Labeling

Parts-of-Speech Tagging

Sequence Labeling: Parts of Speech

From the earliest linguistic traditions (Yaska and Panini 5th C. BCE, Aristotle 4th C. BCE), the idea that words can be classified into grammatical categories

- part of speech, word classes, POS, POS tags

8 parts of speech attributed to Dionysius Thrax of Alexandria (c. 1st C. BCE):

- noun, verb, pronoun, preposition, adverb, conjunction, participle, article

Open vs. Closed Class

Open class ("content") words

Nouns

Proper

Janet

Italy

Common

cat, cats

mango

Verbs

Main

eat

went

Adjectives

old green tasty

Adverbs

slowly yesterday

Numbers

122,312

one

Interjections

Ow hello

... more

Closed class ("function")

Determiners

the some

Conjunctions

and or

Pronouns

they its

Auxiliary

can

had

Prepositions

to with

Particles

off up

... more

Part-of-Speech Tagging

Assigning a part-of-speech to each word in a text.

Words often have more than one POS.

book:

- VERB: (*Book that flight*)
- NOUN: (*Hand me that book*).

Popular tag-set: Penn Treebank

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coordinating conjunction	<i>and, but, or</i>	PDT	predeterminer	<i>all, both</i>	VBP	verb non-3sg present	<i>eat</i>
CD	cardinal number	<i>one, two</i>	POS	possessive ending	's	VBZ	verb 3sg pres	<i>eats</i>
DT	determiner	<i>a, the</i>	PRP	personal pronoun	<i>I, you, he</i>	WDT	wh-determ.	<i>which, that</i>
EX	existential ‘there’	<i>there</i>	PRP\$	possess. pronoun	<i>your, one's</i>	WP	wh-pronoun	<i>what, who</i>
FW	foreign word	<i>mea culpa</i>	RB	adverb	<i>quickly</i>	WP\$	wh-possess.	<i>whose</i>
IN	preposition/ subordin-conj	<i>of, in, by</i>	RBR	comparative adverb	<i>faster</i>	WRB	wh-adverb	<i>how, where</i>
JJ	adjective	<i>yellow</i>	RBS	superlatv. adverb	<i>fastest</i>	\$	dollar sign	\$
JJR	comparative adj	<i>bigger</i>	RP	particle	<i>up, off</i>	#	pound sign	#
JJS	superlative adj	<i>wildest</i>	SYM	symbol	<i>+, %, &</i>	"	left quote	‘ or “
LS	list item marker	<i>1, 2, One</i>	TO	“to”	<i>to</i>	"	right quote	’ or ”
MD	modal	<i>can, should</i>	UH	interjection	<i>ah, oops</i>	(left paren	[, (, {, <
NN	sing or mass noun	<i>llama</i>	VB	verb base form	<i>eat</i>)	right paren],), }, >
NNS	noun, plural	<i>llamas</i>	VBD	verb past tense	<i>ate</i>	,	comma	,
NNP	proper noun, sing.	<i>IBM</i>	VBG	verb gerund	<i>eating</i>	.	sent-end punc	. ! ?
NNPS	proper noun, plu.	<i>Carolinas</i>	VBN	verb past part.	<i>eaten</i>	:	sent-mid punc	: ; ... --

Methods and Evaluation

Methods:

Hidden Markov Models

Maximum Entropy Markov Models

Conditional Random Fields

RNNs, Transformers

Evaluation:

Accuracy

Macro-F1 (giving equal importance to each tag)

NLP Paradigms: Text Generation

Dialogs

Example: Dialogs

- Generating responses
 - That are **consistent** and **coherent** with the dialog history
 - That are **interesting** and **engaging**
 - That meaningfully progress the dialog towards a goal

Two kind of conversational agents

1. Chatbots

- mimic informal human chatting
- for fun, or even for therapy

2. (Task-based) Dialogue Agents

- interfaces to personal assistants
- cars, robots, appliances
- booking flights or restaurants

Chatbot Architectures

Rule-based

Pattern-action rules (**ELIZA**)
+ A mental model (**PARRY**):

The first system to pass the Turing Test!

Corpus-based

Information Retrieval (**Xiaolce**)
Neural encoder-decoder (**BlenderBot**)

Response by generation

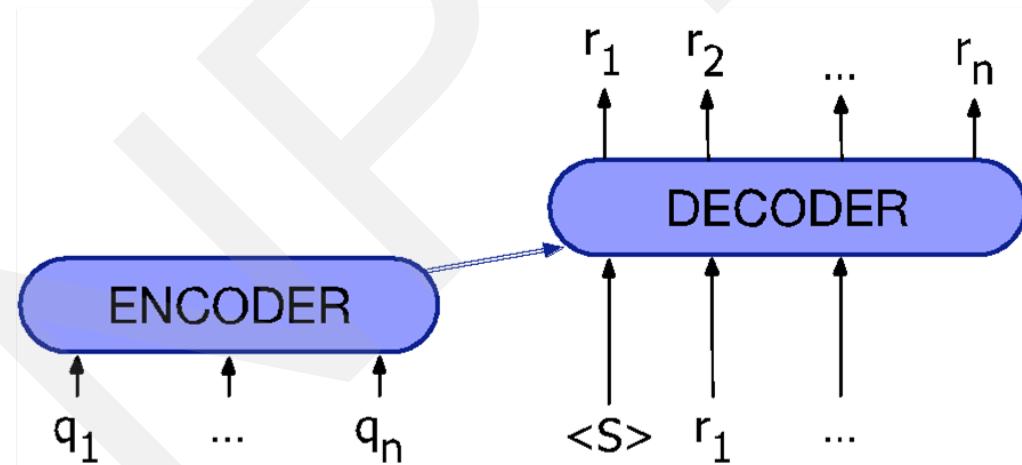
Think of response production as an encoder-decoder task

Generate each token r_t of the response by conditioning on the encoding of the entire query q and the response so far $r_1 \dots r_{t-1}$

$$\hat{r}_t = \operatorname{argmax}_{w \in V} P(w | q, r_1 \dots r_{t-1})$$

Conditional LM

Evaluation is tricky



NLP Paradigms: Structured Prediction

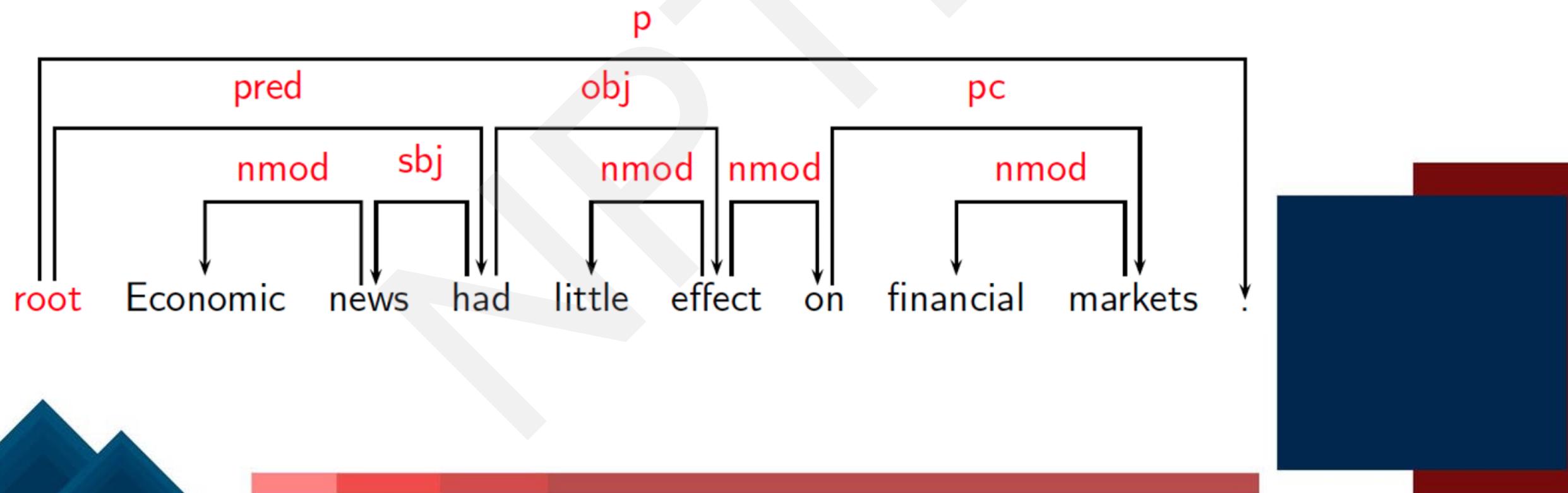
Dependency Parsing



Dependency Parsing

Dependency Parsing

- **Input:** Sentence $x = w_1, \dots, w_n$
- **Output:** Dependency graph G



Other NLP Tasks: Some Examples

Word Sense Disambiguation (WSD)

Sense ambiguity

- Many words have several meanings or senses
- The meaning of **bass** depends on the context
- Are we talking about music, or fish?
 - ▶ An electric guitar and **bass** player stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.
 - ▶ And it all started when fishermen decided the striped **bass** in Lake Mead were too skinny.

Disambiguation

- The task of disambiguation is to determine which of the senses of an ambiguous word is invoked in a particular use of the word.
- This is done by looking at the context of the word's use.

Entity Linking

Iranian POW negotiator holds talks with Iraqi ministers

The head of [Iran's prisoner of war](#) commission met with two [Iraqi](#) Cabinet ministers Saturday in a bid to glean information about thousands of Iranian POWs allegedly in Iraq, the official Iraqi News Agency reported.

Iraqi Foreign Minister [Mohammed Saeed al-Sahaf](#) told Abdullah al-Najafi that the two states needed to ``speed up the closure of what remains from the POW and Missing-In-Action file," INA said.

The issue of POWs and missing persons remains a stumbling block to normalizing relations between the two neighbors.

Iraq has long maintained that it has released all Iranian prisoners captured in the [1980-88 Iran-Iraq War](#). The countries accuse each other of hiding POWs and preventing visits by the [International Committee of the Red Cross](#) to prisoner camps.

The ICRC representative in [Baghdad](#), Manuel Bessler, told The [Associated Press](#) that his organization has had difficulty visiting POWs on both sides on a regular basis.

In April, Iran released 5,584 since [1990](#).

More than 1 million people w

Baghdad

Baghdad is the capital of Iraq and of Baghdad Governorate. With a metropolitan area estimated at a population of 7,000,000, it is the largest city in Iraq. It is the second-largest city in the Arab world (after Cairo) and the second-largest city in southwest Asia (after Tehran).

[open in wikipedia](#)

fied as civil law detainees in the largest exchange

Sentence Similarity

id	qid1	qid2	question1	question2	is_duplicate
447	895	896	What are natural numbers?	What is a least natural number?	0
1518	3037	3038	Which pizzas are the most popularly ordered pizzas on Domino's menu?	How many calories does a Dominos pizza have?	0
3272	6542	6543	How do you start a bakery?	How can one start a bakery business?	1
3362	6722	6723	Should I learn python or Java first?	If I had to choose between learning Java and Python, what should I choose to learn first?	1

Quora Question Pairs

Quora Question Pairs (QQP) dataset consists of over 400,000 question pairs, and each question pair is annotated with a binary value indicating whether the two questions are paraphrase of each other.

Question Answering

The Norman dynasty had a major political, cultural and military impact on medieval Europe and even the Near East. The Normans were famed for their martial spirit and eventually for their Christian piety, becoming exponents of the Catholic orthodoxy into which they assimilated. They adopted the Gallo-Romance language of the Frankish land they settled, their dialect becoming known as Norman, Normaund or Norman French, an important literary language. The Duchy of Normandy, which they formed by treaty with the French crown, was a great fief of medieval France, and under Richard I of Normandy was forged into a cohesive and formidable principality in feudal tenure. The Normans are noted both for their culture, such as their unique Romanesque architecture and musical traditions, and for their significant military accomplishments and innovations. Norman adventurers founded the Kingdom of Sicily under Roger II after conquering southern Italy on the Saracens and Byzantines, and an expedition on behalf of their duke, William the Conqueror, led to the Norman conquest of England at the Battle of Hastings in 1066. Norman cultural and military influence spread from these new European centres to the Crusader states of the Near East, where their prince Bohemond I founded the Principality of Antioch in the Levant, to Scotland and Wales in Great Britain, to Ireland, and to the coasts of north Africa and the Canary Islands.

Ground Truth Answers: William the Conqueror William the Conqueror William the Conqueror

Who ruled the duchy of Normandy

Ground Truth Answers: Richard I Richard I Richard I

What religion were the Normans

Ground Truth Answers: Catholic Catholic orthodoxy Catholic

What type of major impact did the Norman dynasty have on modern Europe?

Ground Truth Answers: <No Answer>

Who was famed for their Christian spirit?

Ground Truth Answers: <No Answer>

Who assimilated the Roman language?

Ground Truth Answers: <No Answer>

SQuAD2.0

The Stanford Question Answering Dataset

More Example Tasks, Benchmarks

2892 dataset results for Texts X

Search for datasets 🔍

☰ grid Best match ▼

Filter by Modality (clear)

Texts X

Images 3021

Videos 949

Audio 453

Filter by Task

Question Answering 295

Language Modelling 121

Text Generation 115

Text Classification 111

GLUE (General Language Understanding Evaluation benchmark)

General Language Understanding Evaluation (GLUE) benchmark is a collection of nine natural language understanding tasks, including single-sentence tasks CoLA and SST-2, similarity...

3,019 PAPERS • 24 BENCHMARKS

SST (Stanford Sentiment Treebank)

The Stanford Sentiment Treebank is a corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language. The corpus is bas...

2,223 PAPERS • 10 BENCHMARKS

SQuAD (Stanford Question Answering Dataset)

The Stanford Question Answering Dataset (SQuAD) is a collection of question-answer pairs derived from Wikipedia articles. In SQuAD, the correct answers of questions can be any se...

2,077 PAPERS • 13 BENCHMARKS

MultiNLI (Multi-Genre Natural Language Inference)

The Multi-Genre Natural Language Inference (MultiNLI) dataset has 433K sentence pairs. Its size and mode of collection are modeled closely like SNLI. MultiNLI offers ten distinct genre...

1,762 PAPERS • 3 BENCHMARKS

IMDb Movie Reviews

The IMDb Movie Reviews dataset is a binary sentiment analysis dataset consisting of 50,000 reviews from the Internet Movie Database (IMDb) labeled as positive or negative. The datas...

1,716 PAPERS • 11 BENCHMARKS

REFERENCES

- Daniel Jurafsky and James H. Martin. 2024. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. Online manuscript released August 20, 2024.



THANK YOU