Natural Language Processing course

May 27- June17, 2024

Imran Khan, ODU

Note: These slides heavily borrow from the book “Speech and language processing” authored byDaniel Jurafsky and James Martin.

What is Natural Language Processing?

- It's a branch of artificial intelligence and linguistics concerned with the interactions between computers and humans through natural language. NLP enables computers to understand, interpret, and generate human language in a way that is both meaningful and useful.

- It encompasses a range of tasks, including
  - Language translation
  - Sentiment analysis
  - Text summarization
  - Speech recognition and text-to-speech
  - Question answering
  - Information Retrieval
  - Plagiarism check.
  - Chatbot and dialogue system (for extended conversations).

Prerequisites to study NLP

- Proficiency in Python: Numpy and Pytroch
- College-level calculus and Linear Algebra
- Basic probability and statistics
- Basic understanding of Machine Learning

Freely available online resources

Reference text: Dan Jurafsky and James H. Martin. Speech and Language Processing (2024 pre-release)

YouTube videos (Stanford Course): https://www.youtube.com/playlist?list=PLoROMvodv4rMFqRtEuo6SGjY4XbRIVRd4

Hugging face: https://huggingface.co/docs/hub/en/models-libraries

Stanford Course CS 224N: https://web.stanford.edu/class/cs224n/

Andrew Ng course: https://www.youtube.com/watch?v=S7oA5C43Rbc&pp=ygUNbmxwIGFuZHJldyBuZw%3D%3D

https://www.facebook.com/LearnPythonR4Datascience
Please like, follow and share my Facebook page with your friends.

Absolutely Free Natural Language Processing (beginner level) course

Only very limited number of students will be **randomly** selected.
Three lectures (Mon, Wed, Fri) in the first week and subsequently two lectures (Mon, Fri per week.
May 27- June 17

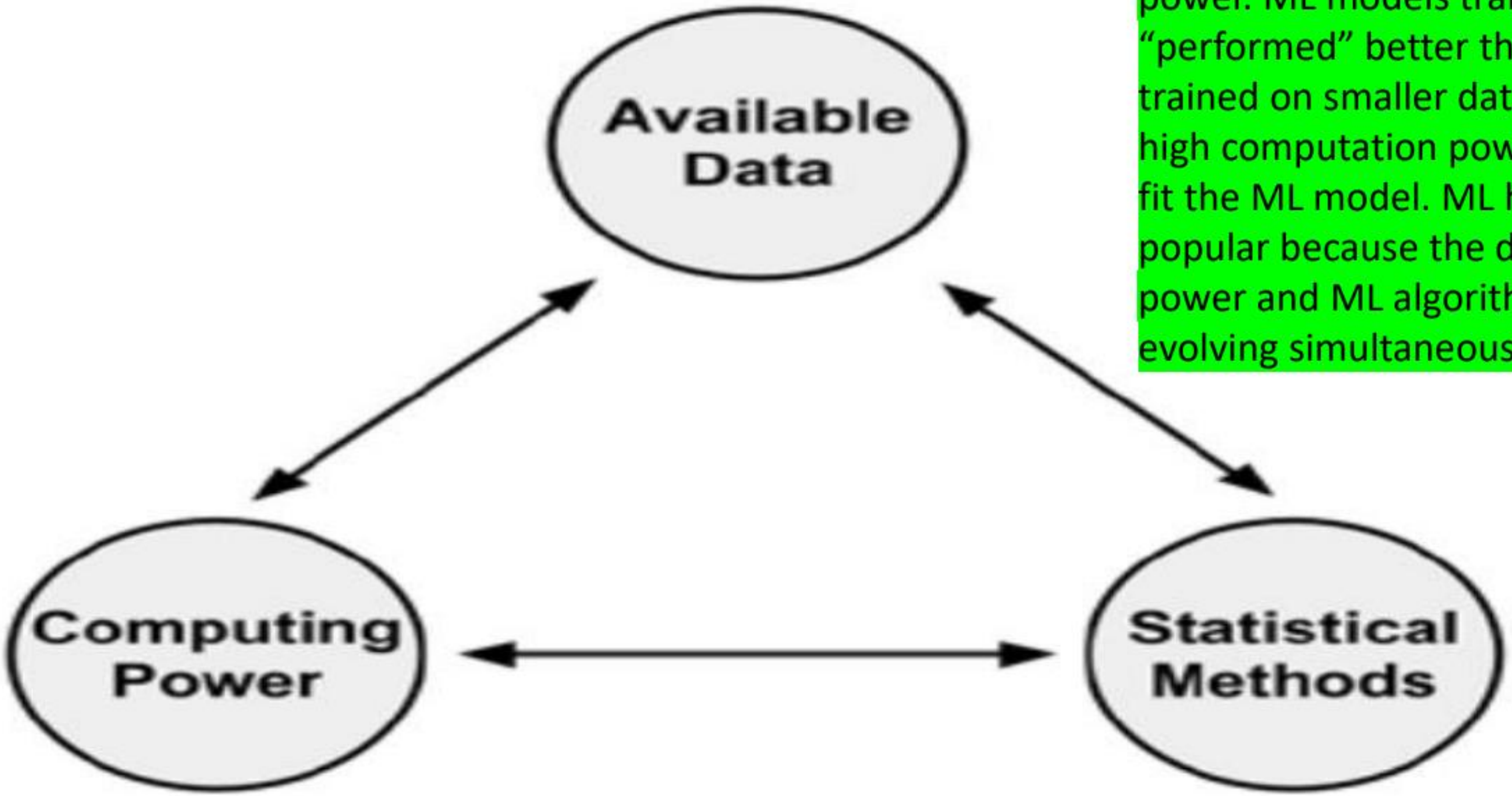
Date	Lecture	Resources
May 27, 2024 (Monday) 09:00 -10:30 AM EST	Basic concepts of probability Basic concepts of Machine Learning N-gram Language Model	**Book Chapter to read (Mandatory)**: Please read **Chapter 3** from the freely available book https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf Please read **Chapter 1** from the book "An introduction to Statistical Learning" freely available on https://www.statlearning.com/ for basic introduction on Machine Learning. Basic review of probability from https://cs229.stanford.edu/section/cs229-prob.pdf
May 29, 2024 (Wednesday) 09:00 -10:30 AM EST	Naive Bayes, Text Classification, and Sentiment	**Book Chapter to read (Mandatory)**: Please read **Chapter 4** from the freely available book https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf
May 31, 2024 (Friday) 09:00 -10:30 AM EST	Logistic Regression	**Book Chapter to read (Mandatory)**: Please read **Chapter 5** from the freely available book https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf
June 03, 2024 (Monday) 09:00 -10:30 AM EST	Vector Semantics and Embeddings	**Online videos to watch:** An introduction to word embeddings (Mandatory) https://www.youtube.com/watch?v=5MaWmXwxFNQ&t=564s Please watch only first 30 minutes of the video. https://www.youtube.com/watch?v=rmVRLeJRkl4&list=PLoROMvodv4rMFqRtEuo6SGjY4XbRIVRd4 **Book Chapter to read (Mandatory)**: Please read Chapter 6 from the freely available book https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf Please read pages (1-7) **(Mandatory)** https://web.stanford.edu/class/cs224n/readings/cs224n_winter2023_lecture1_notes_draft.pdf
June 07, 2024 (Friday) 09:00 -10:30 AM EST	Neural Networks and Neural Language Models	**Book Chapter to read (Mandatory)**: Please read **Chapter 7** from the freely available book https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf
June 10, 2024 (Monday) 09:00 -10:30 AM EST	RNNs and LSTMs	**Online videos to watch:** https://www.youtube.com/watch?v=0LixFSa7yts https://www.youtube.com/watch?v=wzfWHP6SXxY&t=259s https://www.youtube.com/watch?v=ySEx_Bqxvvo **Book Chapter to read (Mandatory)**: Please read **Chapter 9** from the freely available book https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf
June 14, 2024 (Friday) 09:00 -10:30 AM EST	Transformers and Large Language Models	**Online videos to watch (Mandatory)** https://www.youtube.com/watch?v=g2BRIuln4uc https://www.youtube.com/watch?v=eMlx5fFNoYc https://www.youtube.com/watch?v=bCz4OMemCcA https://www.youtube.com/watch?v=SZorAJ4I-sA&t=5s https://www.youtube.com/watch?v=LWMzyfvuehA&t=947s **Book Chapter to read (Mandatory)**: Please read **Chapter 10** from the freely available book https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf Annotated transformer (Not required to read. It is just the implementation of Transformer) https://nlp.seas.harvard.edu/annotated-transformer/
July 17, 2024 (Monday) 09:00 -10:30 AM EST (this lecture might be rescheduled)	Fine-Tuning and Masked Language Models	**Online videos to watch (Mandatory)** https://www.youtube.com/watch?v=DGfCRXuNA2w https://www.youtube.com/watch?v=knTc-NQSjKA https://www.youtube.com/watch?v=90mGPxR2GgY&t=420s (Bert explanation) **Book Chapter to read (Mandatory)**: Please read Chapter 11 from the freely available book https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

Friendly introduction to Machine Learning

Popular applications of Machine Learning

- Netflix’s movie recommendation system
- YouTube video recommendation system
- Amazon product recommendation system
- Face recognition system/speech recognition system
- Driverless cars
- Detecting fraudulent credit card transactions
- Google map suggesting you the shortest path (based on distance and traffic flows)
- Email spam filters
- Google search results (for keywords query on google)
- Handwriting recognition
- Sales prediction (inventory restocking)
- Predictive maintenance systems

Why has the ML become so popular recently?



ML needs big data and high compute power. ML models trained on big data “performed” better than the ones trained on smaller datasets. We need high computation power to train and fit the ML model. ML has become so popular because the data, computing power and ML algorithms are rapidly evolving simultaneously.

What is machine learning?

- the use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyze and draw inferences from patterns in data.

- Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behavior.

Types of Machine Learning algorithms

Two types of ML algorithms

1.Supervised ML algorithms

2.Unsupervised ML algorithms

Supervised ML algorithms: Supervised ML is used to construct predictive model. "A predictive model is used for tasks that involve, as the name implies, the prediction of one value (dependent variable) using other values (independent variables) in the dataset. The learning algorithm attempts to discover and model the relationship among the target feature (the feature being predicted) and the other features (independent variables)" by Brett Lantz

Unsupervised ML algorithms: There is NO TARGET FEATURE to predict in the unsupervised ML algorithm. Clustering is an example of unsupervised ML algorithm in which we may try to divide the datasets into mutually exclusive homogenous groups.

For natural language processing, self-supervision is the dominant ML paradigm.
What is self-supervision? Why do we need self-supervision in NLP?

ML jargon alert

Feature: Variable is called feature in ML

Example (or instance): Observation is called example or instance.

Target: Target feature or simply feature is the dependent variable.

Classification problem: Supervised ML problem when the target feature (dependent variable) is **categorical.** ML classification problem could be binary class label or multiclass label.

Regression problem: Supervised ML problem when the target feature (dependent variable) is **continuous.** Suppose, if we are trying to predict the price of house given the size of house and availability of school. It is a regression problem.

Two types of supervised learning

Regression problem: Supervised ML problem when the target feature (dependent variable) is **continuous.**

Regression examples: **1.** Predicting the price of house given the size of house and availability of schools. **2**. Predicting the sales of Walmart products in a given store for next 30 days.

Classification problem: Supervised ML problem when the target feature (dependent variable) is **categorical.**

Classification examples: 1. Whether the tumor is benign or malignant (binary classification problem). 2. Face recognition is classification (multi-class classification) problem in which a photo is predicted to one of many photos.

Give some examples of classification learning problems in NLP

Supervised or unsupervised

Whether the credit card transaction is fraudulent or not?

Customer segmentation

A football team will win or lose

A person will live past the age of 100

An applicant will default on a loan

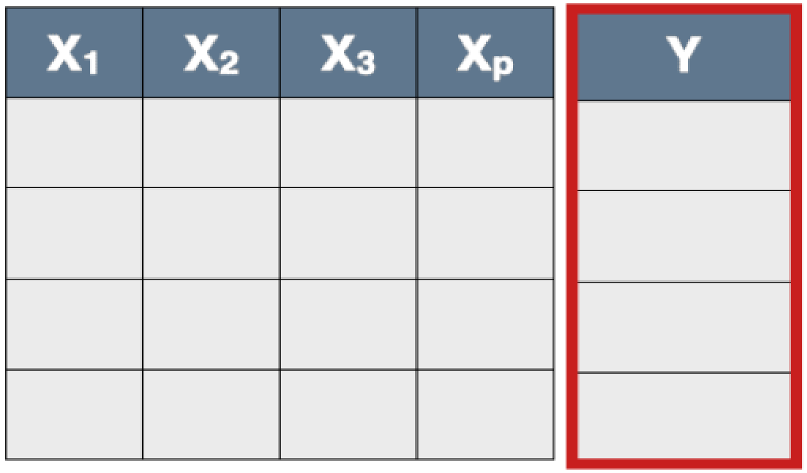
An earthquake will strike next year

Whether the email is spam or not

Is the picture of cat, dog or human?

Supervised vs unsupervised

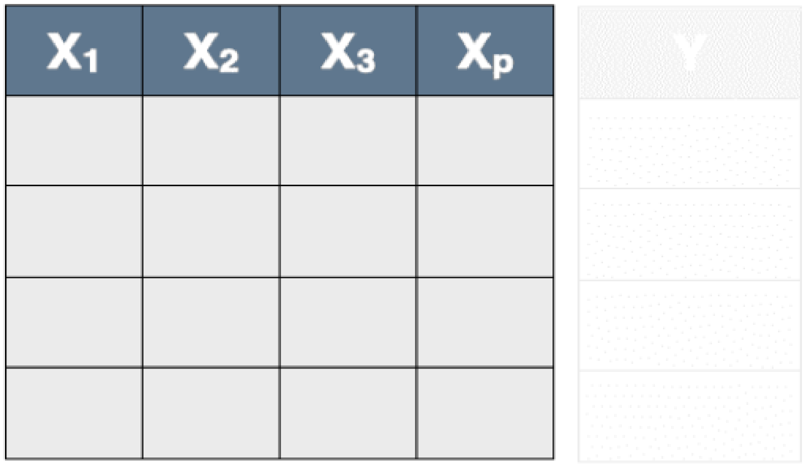
Supervised Learning



X_1	X_2	X_3	X_p	Y

Target

Un-Supervised Learning



X_1	X_2	X_3	X_p	Y

No Target

Steps to apply “supervised” ML algorithm to your data

1.**Collecting data**

2.**Exploring and preparing the data**

3.**Training a model on the data:** We “randomly” divide the dataset into two portions i.e., training dataset and testing dataset. Usually, 80 percent of the dataset is training dataset and 20 percent of the remaining dataset is testing dataset. Note that these two datasets are mutually exclusive. The examples (observations) that are included in training dataset should not be included in the test dataset.

4.Evaluating model performance: We evaluate the model performance (accuracy of the learning process) on the **“test dataset” .** Again, we don’t evaluate the model performance on the **“training dataset”.** The goal of the ML model is NOT the high accuracy on the training dataset but high accuracy on the testing dataset (unforeseen dataset). If the model accuracy is low on the test dataset, then we take steps to improve the performance of the model.

5.Improving model performance: We may need more data, or we may need some sort of feature engineering, or we may even need to use different ML model.

Supervised Learning

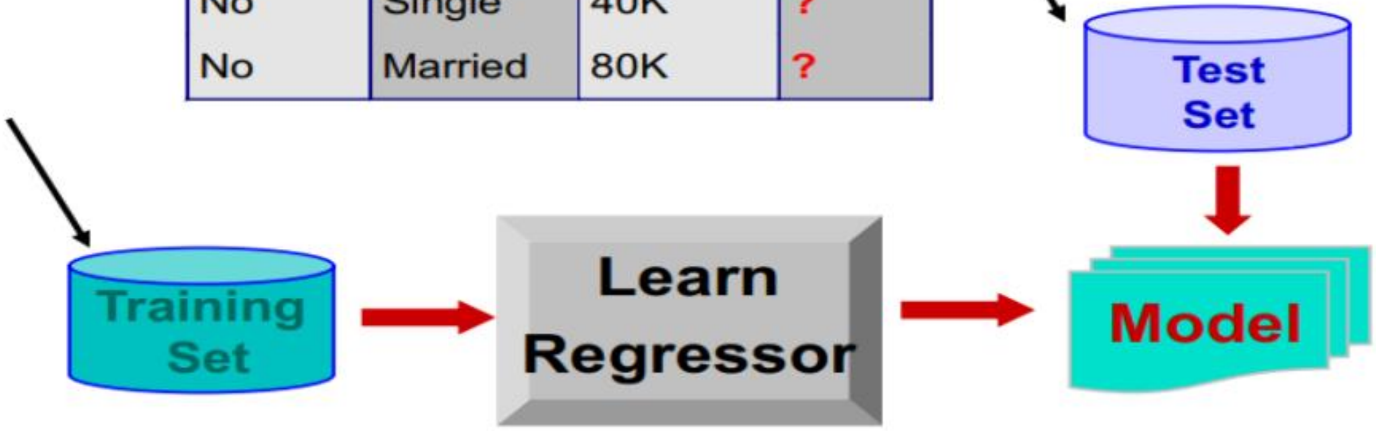
categorical categorical continuous Continuous

Tid	Refund	Marital Status	Taxable Income	Loss
1	Yes	Single	125K	100
2	No	Married	100K	120
3	No	Single	70K	-200
4	Yes	Married	120K	-300
5	No	Divorced	95K	-400
6	No	Married	60K	-500
7	Yes	Divorced	220K	-190
8	No	Single	85K	300
9	No	Married	75K	-240
10	No	Single	90K	90

Past transaction records, label them

Current data, want to use the model to predict

Refund	Marital Status	Taxable Income	Loss
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?



Goals: Predict the possible loss in fraud transaction based on customer records

Supervised learning

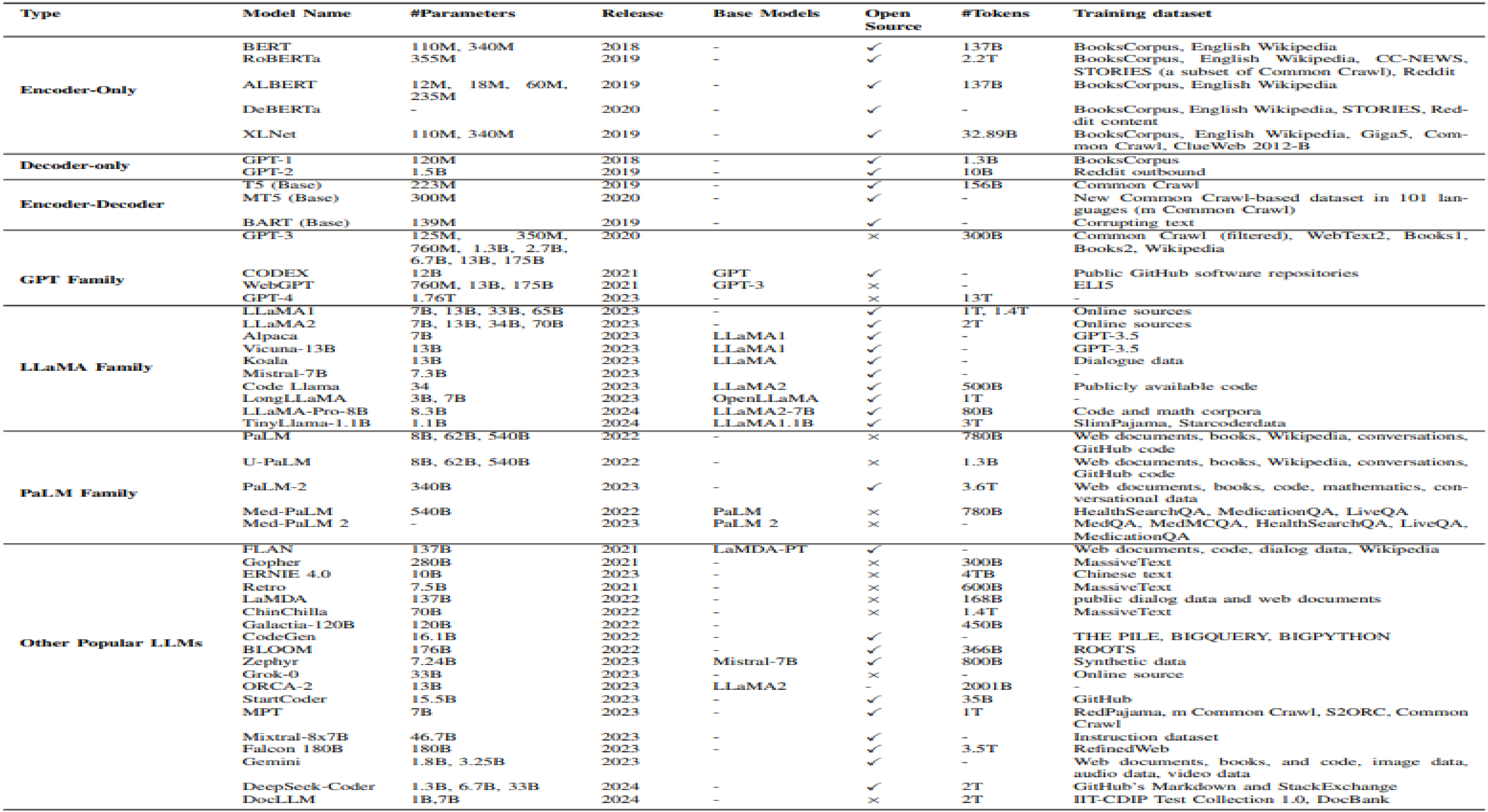
- What do we mean by “learning” in supervised learning? What is the model learning? When we are training the supervised ML model, it is learning the functional form i.e., y = f(x). For example, in linear regression, some linear combination of input features gives you the target feature.

$$\text{Loss(target feature)} = B_0 + B_2\text{*Refund} + B_3\text{*Marital\_status} + B_4\text{*Taxable\_income} + \text{Error}$$

How many parameters (weights and a bias) we have to learn in the above equation?

What does fitting a model on training dataset means? It means learning the coefficients (parameters) of linear regression function. Again, model is learning some function that linearly combines the input features to compute target feature.

TABLE I: High-level Overview of Popular Language Models



Type	Model Name	#Parameters	Release	Base Models	Open Source	#Tokens	Training dataset
Encoder-Only	BERT	110M, 340M	2018	-	✓	137B	BooksCorpus, English Wikipedia
	RoBERTa	355M	2019	-	✓	2.2T	BooksCorpus, English Wikipedia, CC-NEWS, STORIES (a subset of Common Crawl), Reddit
	ALBERT	12M, 18M, 60M, 235M	2019	-	✓	137B	BooksCorpus, English Wikipedia
	DeBERTa	-	2020	-	✓	-	BooksCorpus, English Wikipedia, STORIES, Reddit content
	XLNet	110M, 340M	2019	-	✓	32.89B	BooksCorpus, English Wikipedia, Giga5, Common Crawl, ClueWeb 2012-B
Decoder-only	GPT-1	120M	2018	-	✓	1.3B	BooksCorpus
	GPT-2	1.5B	2019	-	✓	10B	Reddit outbound
Encoder-Decoder	T5 (Base)	223M	2019	-	✓	156B	Common Crawl
	MT5 (Base)	300M	2020	-	✓	-	New Common Crawl-based dataset in 101 languages (m Common Crawl)
	BART (Base)	139M	2019	-	✓	-	Corrupting text
GPT Family	GPT-3	125M, 350M, 760M, 1.3B, 2.7B, 6.7B, 13B, 175B	2020		×	300B	Common Crawl (filtered), WebText2, Books1, Books2, Wikipedia
	CODEX	12B	2021	GPT	✓	-	Public GitHub software repositories
	WebGPT	760M, 13B, 175B	2021	GPT-3	×	-	ELI5
	GPT-4	1.76T	2023	-	×	13T	-
LLaMA Family	LLaMA1	7B, 13B, 33B, 65B	2023	-	✓	1T, 1.4T	Online sources
	LLaMA2	7B, 13B, 34B, 70B	2023	-	✓	2T	Online sources
	Alpaca	7B	2023	LLaMA1	✓	-	GPT-3.5
	Vicuna-13B	13B	2023	LLaMA1	✓	-	GPT-3.5
	Koala	13B	2023	LLaMA	✓	-	Dialogue data
	Mistral-7B	7.3B	2023		✓	-	-
	Code Llama	34	2023	LLaMA2	✓	500B	Publicly available code
	LongLLaMA	3B, 7B	2023	OpenLLaMA	✓	1T	-
	LLaMA-Pro-8B	8.3B	2024	LLaMA2-7B	✓	80B	Code and math corpora
	TinyLlama-1.1B	1.1B	2024	LLaMA1.1B	✓	3T	SlimPajama, Starcoderdata
PaLM Family	PaLM	8B, 62B, 540B	2022	-	×	780B	Web documents, books, Wikipedia, conversations, GitHub code
	U-PaLM	8B, 62B, 540B	2022	-	×	1.3B	Web documents, books, Wikipedia, conversations, GitHub code
	PaLM-2	340B	2023	-	✓	3.6T	Web documents, books, code, mathematics, conversational data
	Med-PaLM	540B	2022	PaLM	×	780B	HealthSearchQA, MedicationQA, LiveQA
	Med-PaLM 2	-	2023	PaLM 2	×	-	MedQA, MedMCQA, HealthSearchQA, LiveQA, MedicationQA
Other Popular LLMs	FLAN	137B	2021	LaMDA-PT	✓	-	Web documents, code, dialog data, Wikipedia
	Gopher	280B	2021	-	×	300B	MassiveText
	ERNIE 4.0	10B	2023	-	×	4TB	Chinese text
	Retro	7.5B	2021	-	×	600B	MassiveText
	LaMDA	137B	2022	-	×	168B	public dialog data and web documents
	ChinChilla	70B	2022	-	×	1.4T	MassiveText
	Galactia-120B	120B	2022	-		450B	
	CodeGen	16.1B	2022	-	✓	-	THE PILE, BIGQUERY, BIGPYTHON
	BLOOM	176B	2022	-	✓	366B	ROOTS
	Zephyr	7.24B	2023	Mistral-7B	✓	800B	Synthetic data
	Grok-0	33B	2023	-	×	-	Online source
	ORCA-2	13B	2023	LLaMA2	-	2001B	-
	StartCoder	15.5B	2023	-	✓	35B	GitHub
	MPT	7B	2023	-	✓	1T	RedPajama, m Common Crawl, S2ORC, Common Crawl
	Mixtral-8x7B	46.7B	2023	-	✓	-	Instruction dataset
	Falcon 180B	180B	2023	-	✓	3.5T	RefinedWeb
	Gemini	1.8B, 3.25B	2023		✓	-	Web documents, books, and code, image data, audio data, video data
	DeepSeek-Coder	1.3B, 6.7B, 33B	2024	-	✓	2T	GitHub's Markdown and StackExchange
	DocLLM	1B,7B	2024	-	×	2T	IIT-CDIP Test Collection 1.0, DocBank

Reference: Large Language models: A survey by Shervin et.al

Supervised learning

- What is the goal of "learning" in supervised learning?

The actual goal of learning is to accurately predict the previously unseen examples in the test dataset. High accuracy on the **"training dataset"** is not a good measure of model performance. We care about the model performance on the **"test dataset".**

Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

The same function (y = f(x)) that is "learnt" while fitting the model on training dataset is used to make predictions on the testing dataset.

Performance of the model in supervised learning

- How to measure the accuracy of the learning process?

Accuracy of learning in regression problem:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

Again, the error rate on "training dataset" is not the good indicator for the performance of the model. The real goal is to minimize the error rate on the "test dataset".

Accuracy of learning in classification problem:

Classification errors = $\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i).$

Classification error computes the proportion of class labels predicted inaccurately.

Overfitting versus underfitting

Overfitting: When a given ML method yields a small training error but a large test error, we are said to be overfitting the data.

Overfitting:
Small error on training data but large error on testing data. In terms of accuracy, we can say that high accuracy on training data but low accuracy of prediction on test dataset.

Underfitting: When a given ML method yields a large training error as well as large test error.

Underfitting:
large error on both training data as well as testing data. OR low accuracy on the training data as well as on the test dataset.

Please remember that whether the model is underfitting or overfitting, the test error is almost always greater than the training error

Overfitting versus underfitting

What do you thin what is true functional form that can model the training data in the left picture?

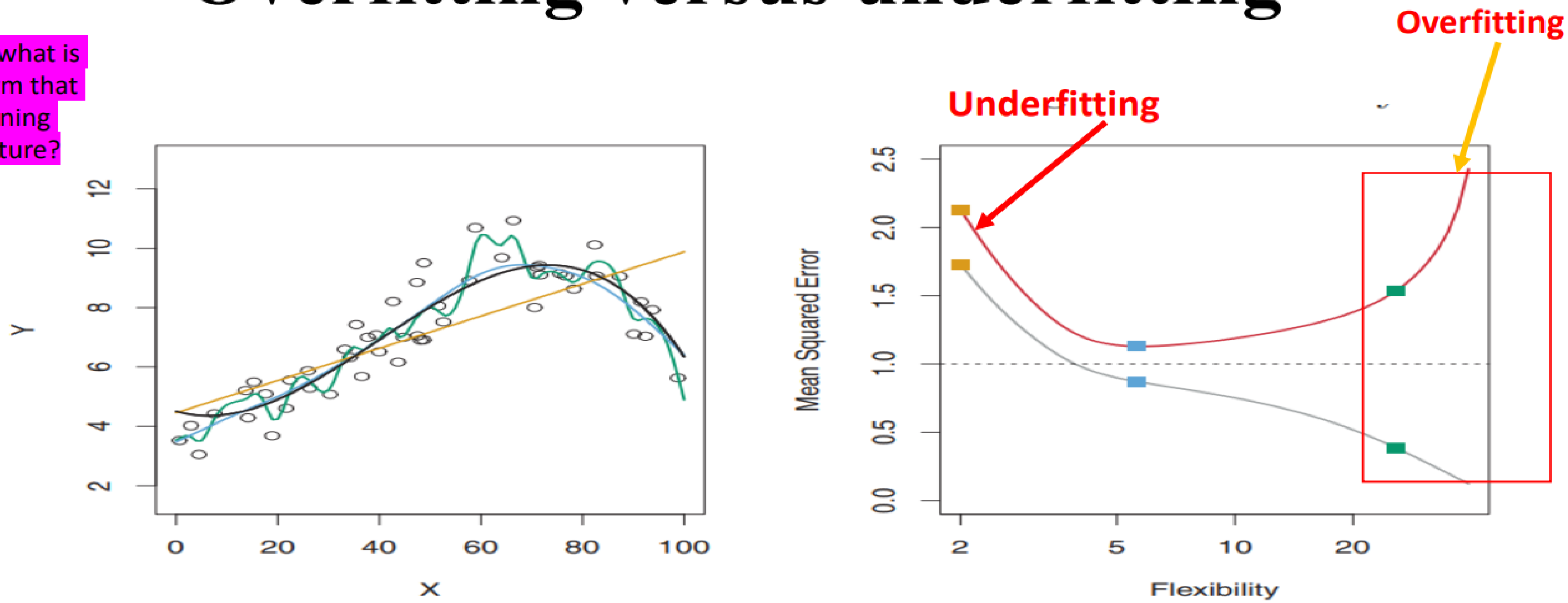


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

From the book: Introduction to statistical learning (Springr)

Why linear regression model is underfitting?
Why the very flexible regression model is overfitting?

Flexibility of the model

- Flexibility means the ability of ML model to fit wide range of patterns in the data. Linear regression model that includes the square terms of features is more flexible than the simple linear regression model (without square terms and higher order polynomials of features). More flexible models require more input data to avoid overfitting. Do you know why?
- We can avoid overfitting or underfitting by choosing the appropriate level of flexibility for the ML model.

Refer to the previous slide to answer the following questions

- How did the training MSE changed with the increase of flexibility of model?

The training MSE monotonically (continuously) decreases with the increase of flexibility of model. But the model starts to fit noise in the data at higher level of flexibility.

- How did the testing MSE changed with the increase of flexibility of ML model?

We got the characteristic U-shaped curve for the test MSE with the increase of flexibility of ML model. Test MSE does not always decrease with the increase in the flexibility of model.

Language Models

Language models

What is a language model?

Predicting a next word given the previous words is a language model. In other words, assigning a probability to a next word given previous words is a language model.

Next word

1. Refrigerator?
2. the?
3. in?
4. Over?

Please turn your homework …

Why would we want to predict upcoming words?

By training the language model to predict next word more accurately, we can better capture the semantic relationship between words, syntactic relationship between words and overall structure of the language.

Simplest language models (n-gram model)

- n-gram language model estimates the probability of next word given the occurrence of n-1 previous words.

- Trigram model is a sequence of three words in which we predict the third word given the previous two words.

- Bigram model is a sequence of two words in which we predict the second word given the previous word.

What's wrong with this idea?

Which one of the two language models (trigram or bigram) will do better job of predicting the next word given previous word(s)?

Problem with estimating probability of a next word

- Suppose the history h is “its water is so transparent that” and we want to know the probability that the next word is “the”

- How to compute the probability P(w/h)?

P(the/its water is so transparent that)

$$\frac{C(\textit{its water is so transparent that the})}{C(\textit{its water is so transparent that})}$$

How to compute these counts?

Build a very large corpus to estimate these counts.

But what if count of sequence is zero in our dataset. Language is creative. New sentences are created.

Trick: Chain rule of probability

- Sequence of n words: (w1, w2 ...wn)

- P(X1 = w1,X2 = w2,X3 = w3,...,Xn = wn)

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

Words never occur independent of each other in any language.

I want to eat English food.

We compute the joint probability of entire sequence by multiplying few conditional probabilities.
Chain rule still don’t help: How to compute the last word given long sequence of preceding words $P(w_n|w_{1:n-1})$

Trick: Chain rule of probability

- Instead of computing the probability of a word given using its *entire* history, we can **approximate** the history by just last few words.

- Bigram model

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

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

Markov assumption:

Probability of a next word only depends on the previous word.

We don't need to look too far in the past to predict the future.

How to compute the probability of bigram?

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

Simple example to compute bigram probabilities

```
<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>
```

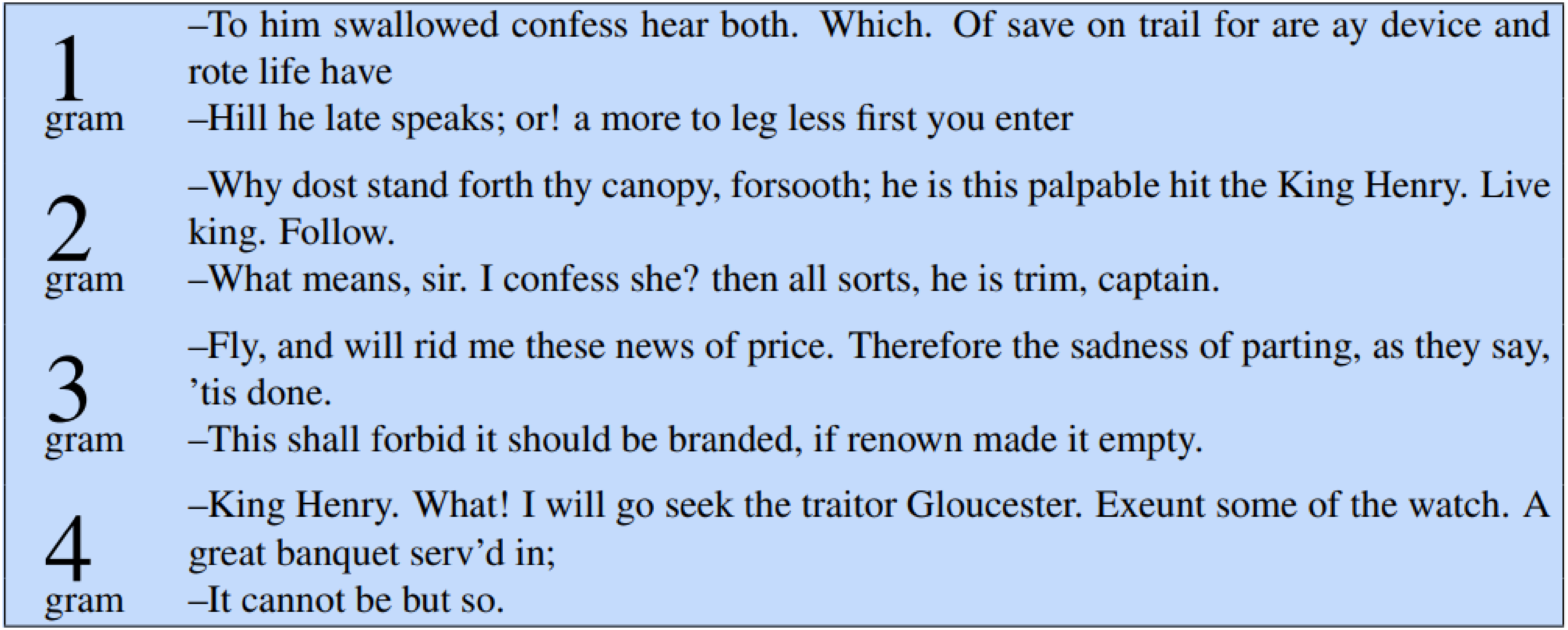
$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

Formula to compute bigram probabilities.

$$P(\text{I}|\text{<s>}) = \tfrac{2}{3} = .67 \qquad P(\text{Sam}|\text{<s>}) = \tfrac{1}{3} = .33 \qquad P(\text{am}|\text{I}) = \tfrac{2}{3} = .67$$

$$P(\text{</s>}|\text{Sam}) = \tfrac{1}{2} = 0.5 \qquad P(\text{Sam}|\text{am}) = \tfrac{1}{2} = .5 \qquad P(\text{do}|\text{I}) = \tfrac{1}{3} = .33$$

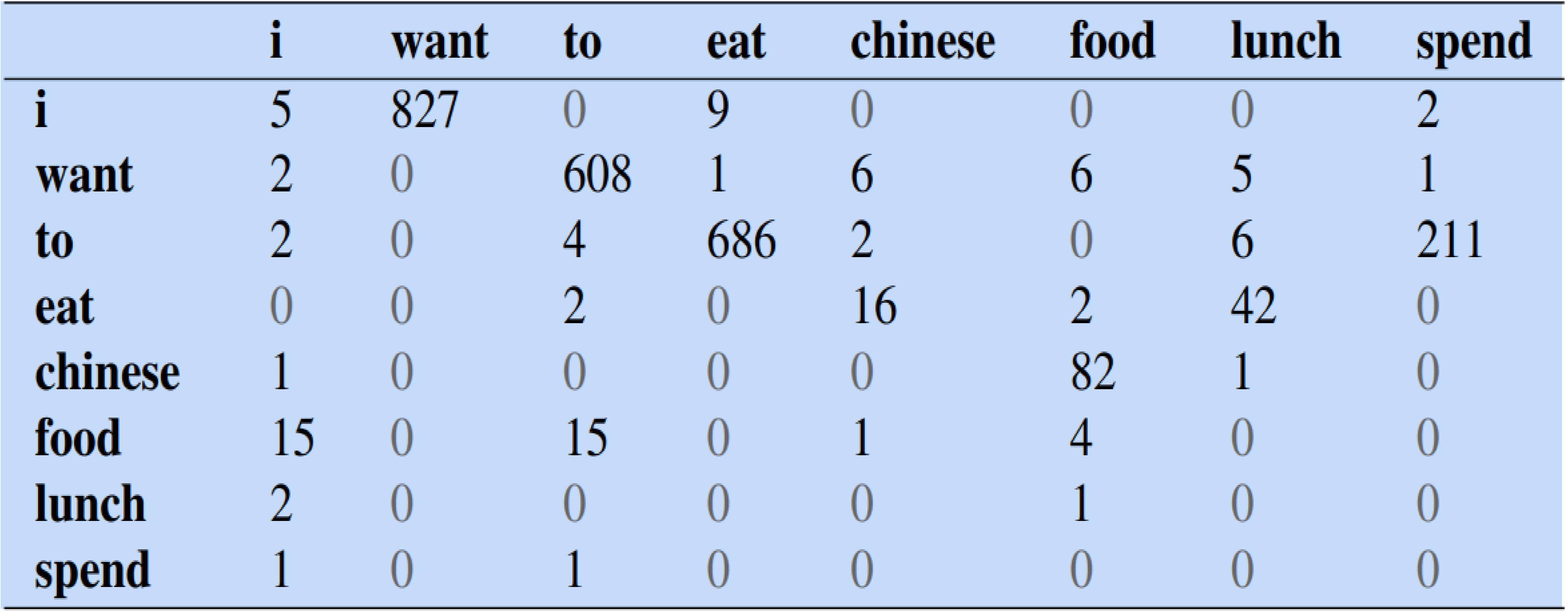
Different n-gram models trained on Shakespear’s words



1 gram	–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
2 gram	–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.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, ’tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv’d in; –It cannot be but so.

Figure 3.4 Eight sentences randomly generated from four n-grams computed from Shakespeare’s works. All characters were mapped to lower-case and punctuation marks were treated as words. Output is hand-corrected for capitalization to improve readability.

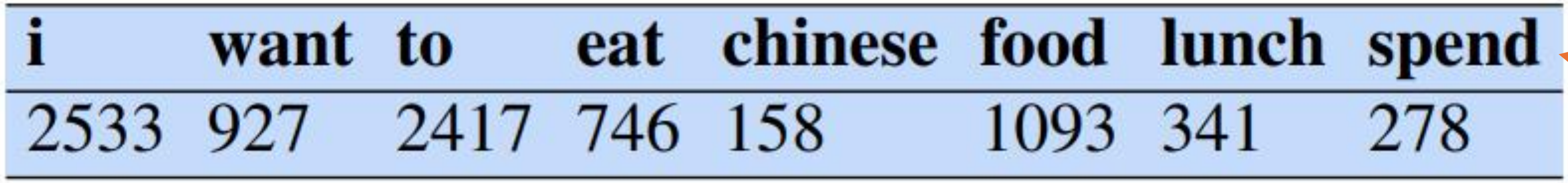
This matrix tries to answer how often these randomly chosen eight words appear next to each other



	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

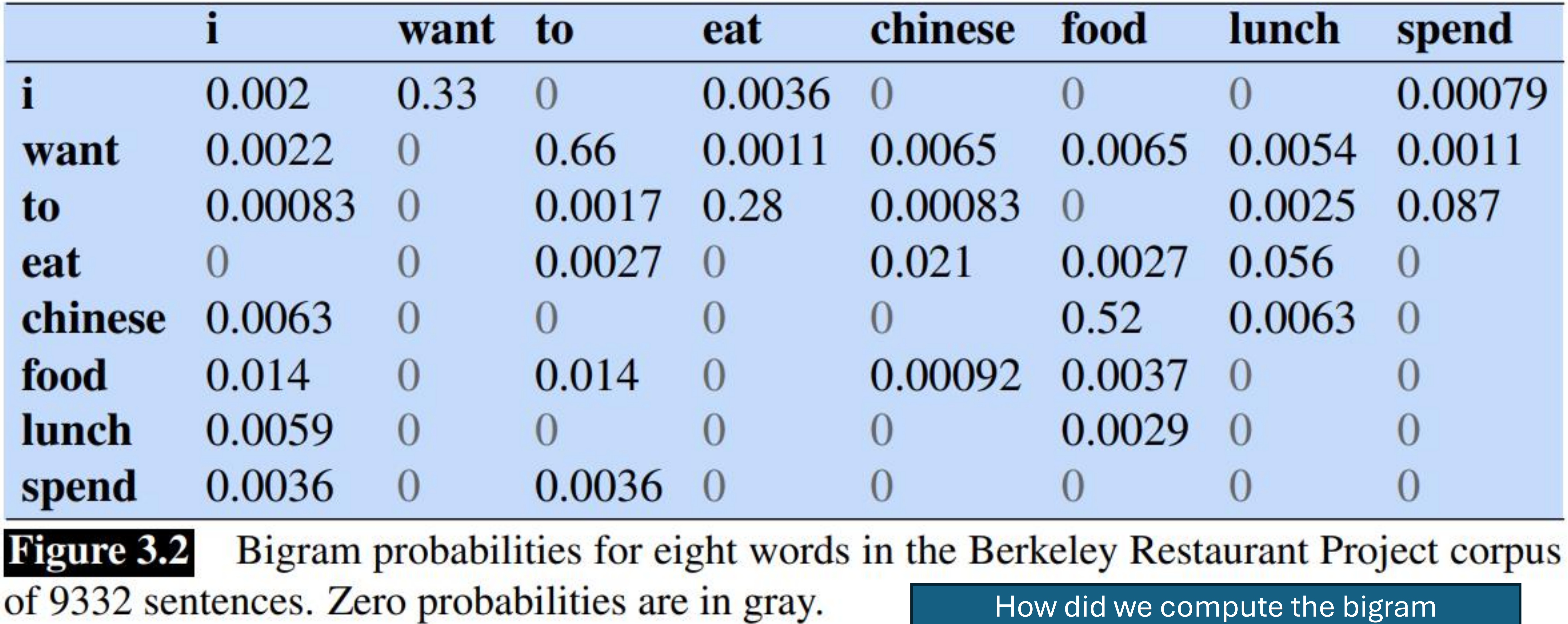
Figure 3.1 Bigram counts for eight of the words (out of $V = 1446$) in the Berkeley Restaurant Project corpus of 9332 sentences. Zero counts are in gray.

It is important to note that most of the words don’t appear next to each other. Therefore, most of the entries are zero. This type of matrix where we have few non-zero entries are known as spare matrix.



i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

Unigram counts



	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Figure 3.2 Bigram probabilities for eight words in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray.

How did we compute the bigram probabilities? Let us try to understand.

$$P(\texttt{i}\,|\,\texttt{<s>}) = 0.25 \qquad P(\texttt{english}\,|\,\texttt{want}) = 0.0011$$

$$P(\texttt{food}\,|\,\texttt{english}) = 0.5 \qquad P(\texttt{</s>}\,|\,\texttt{food}) = 0.68$$

$$\begin{aligned}
P(\texttt{<s> i want english food </s>}) &\\
&= P(\texttt{i}\,|\,\texttt{<s>})P(\texttt{want}\,|\,\texttt{i})P(\texttt{english}\,|\,\texttt{want})\\
&\qquad P(\texttt{food}\,|\,\texttt{english})P(\texttt{</s>}\,|\,\texttt{food})\\
&= .25 \times .33 \times .0011 \times 0.5 \times 0.68\\
&= .000031
\end{aligned}$$

Now let us compute the probability of a sentence “ <s> I want Chinese food <s>”

What kinds of linguistic phenomena are captured in bigram statistics?

- Bigram probabilities encode some facts

a. that we think of as strictly syntactic in nature.

Like the fact that what comes after eat is usually a noun or an adjective, or that what comes after to is usually a verb

b. that even be cultural rather than linguistic

Like the higher probability that people are looking for Chinese versus English food.

How to evaluate the performance of language models?

Shortcoming of extrinsic evaluation
Very expensive

- **Extrinsic evaluation:** The best way to evaluate the performance of the language model is to use it in some NLP application (speech recognition, machine translation etc.) and see how much the NLP application improves. *For example*, we can compare the performance of two language models (bigram vs trigram language model) by embedding it in real-world NLP application to see which language model gives the more accurate results.
- **Intrinsic evaluation:** It measures the quality (or performance) of a language model independent of any application. This means that there is no guarantee that the language model performing high on intrinsic evaluation will also perform well on some real-world NLP applications.

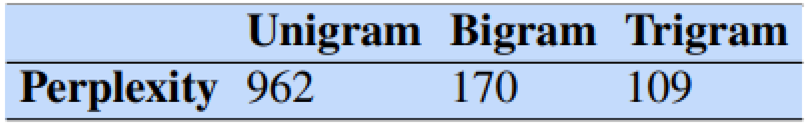
Perplexity is an intrinsic evaluation measure used for evaluating simple n-gram language models and more sophisticated neural language models.

What is perplexity?

It is the inverse probability of the test set normalized by the number of words.

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

- Higher the probability of word sequence, the lower the perplexity. This means lower perplexity score is better.
- In other words, maximizing probability is equivalent to minimizing perplexity



	Unigram	Bigram	Trigram
Perplexity	962	170	109

Which one of the above three language models is the most accurate language model based on perplexity measure?

Smoothing