CS 489: Introduction to Natural Language Processing

Lecture 1: Introduction & Fundamentals

Instructor: Freda Shi

September 10, 2026

Logistics

Instructor and Websites

- **Instructor:** Freda Shi
- **Office Hours:** Friday 2:00PM–3:00PM, DC 2522
- **Class time:** TuTh 11:30AM–12:50PM
- **Location:** MC 4040

- **Course Website:** `cs.uwaterloo.ca/~fhs/teaching/wat-cs489-f26/syllabus.html`
- **Discussion Forum:** Piazza
- **LEARN** (for assignments): `learn.uwaterloo.ca/d2l/home/1288989`

Teaching Assistants

- **Xiaoxi Luo** (x25luo@; homepage)
- **Yixuan (Tom) Wang** (tyxw@; homepage)
- **Sheng Yao** (s57yao@)

- **TA Office Hours:** announced before each assignment due date.
- Questions about assignments should go through the discussion forum first.

Books

There will be no official textbook. The following references are recommended:

- Jurafsky and Martin, *Speech and Language Processing*, 3rd edition. web.stanford.edu/~jurafsky/slp3
- Jacob Eisenstein, *Introduction to Natural Language Processing*. cseweb.ucsd.edu/~nnakashole/teaching/eisenstein-nov18.pdf
- Yoav Goldberg, *A Primer on Neural Network Models for Natural Language Processing*. u.cs.biu.ac.il/~yogo/nnlp.pdf

Grade Breakdown

- Four assignments: **35%** total (9% / 9% / 9% / 8%)
- Midterm exam: **25%**
- Final exam: **40%**

Assignment Roadmap

All assignments are individual; most tasks include a Kaggle leaderboard and a graded analysis write-up.

- **A1: Data, Tokenization & Statistical LMs** (end of Week 5).
- **A2: Training Neural Language Models** (end of Week 7).
- **A3: Probing & Interpreting Pretrained Transformers** (end of Week 10).
- **A4: Finetuning, Alignment & Modern LM APIs** (end of Week 12).

Lateness Policy

All assignments are due at **11:59 PM Eastern Time** on the specified due date.

A universal **72-hour grace period** applies to all assignment deadlines:

- Submissions within the grace period are accepted without penalty.
- No additional extensions are granted beyond the grace period.

Extensions are considered only for medical emergency:

- A valid Verification of Illness Form (VIF) must be submitted.
- The VIF must cover the original due date and the entire grace period.
- The extension request must be submitted before the original due date.

GenAI Policy

You may use GenAI tools to support your learning, but these tools cannot be fully trusted and should *never* replace your own understanding.

Verify content they generate, especially numerical details and any write-up suggestions.

You are fully responsible for the accuracy and integrity of all work you submit in this course.

No electronic devices (except a non-programmable calculator) are allowed during the final exam.

Collaboration and Citation Policy

- **Permitted collaboration:** Discussion of assignments with fellow students is encouraged, but all submitted work (code and written solutions) must be completed independently.
- **Citation requirements:** All external sources must be properly cited, including
  - Academic references and literature;
  - Generative AI tools (e.g., ChatGPT) if used beyond grammar check;
  - Any other resources consulted during assignment completion.
- **Student responsibilities:**
  - Verify the accuracy of any AI-generated content.
  - Ensure all submitted work reflects your own understanding.
  - Provide complete and accurate citations for all sources.

Academic Integrity and Intellectual Property

Property of the University of Waterloo:

- Lecture content, spoken and written (and any audio/video recording thereof).
- Lecture handouts, presentations, and other materials prepared for the course.
- Questions or solution sets from any assessments (assignments, quizzes, tests).
- Work protected by copyright (any work authored by the instructor or TA, or used with permission of the copyright owner).

Sharing intellectual property without the intellectual property owner's permission is a violation of intellectual property rights.

Should I take the course?

You will find the course helpful if

- you are curious about the fundamentals and detailed implementations of language models;
- you are looking to apply modern NLP techniques to your own goals;
- you are interested in NLP and computational linguistics research.

Should I take the course?

You will find the course helpful if

- you are curious about the fundamentals and detailed implementations of language models;
- you are looking to apply modern NLP techniques to your own goals;
- you are interested in NLP and computational linguistics research.

This course will *not* cover

- basics of Python programming, probability, or algorithms;
- details and low-level implementation of efficient LM systems (Lecture 19 gives a high-level overview);
- GPU programming (except for a few high-level basics).

Prerequisites

The formal prerequisite is sufficient mathematical and programming maturity.

The following background knowledge is strongly recommended:

- Basic knowledge of calculus, linear algebra, and probability.
- Python programming proficiency.
- Fundamentals of algorithms (basic complexity analysis).
- Basic data structures: lists, stacks, queues, trees, graphs.

Overview: NLP

What is Natural Language Processing (NLP)?

A cat is sitting next to a dax.

Q: What is a dax?

What is Natural Language Processing (NLP)?

A cat is sitting next to a dax.

Q: What is a dax?

Goal: understand natural language with computational models.

What is Natural Language Processing (NLP)?

A cat is sitting next to a dax.

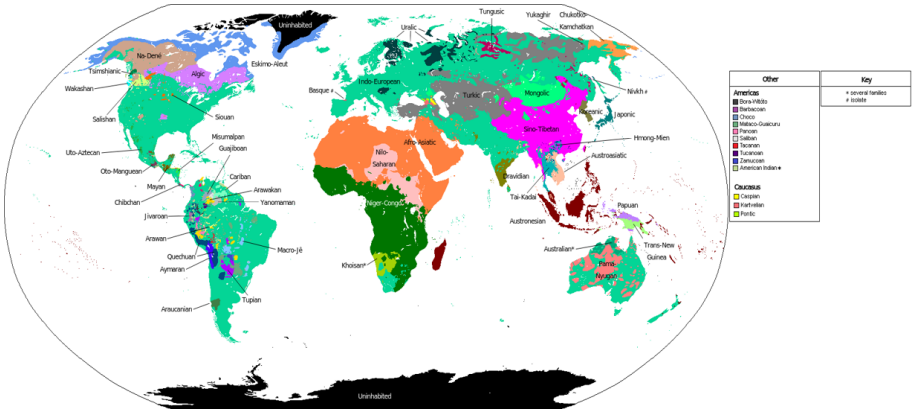
Q: What is a dax?

Goal: understand natural language with computational models.

End systems we want to build:

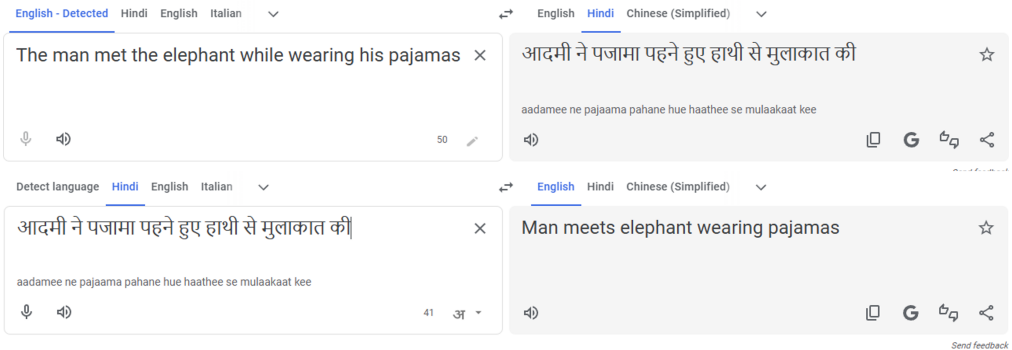
- **Simple:** text classification, grammatical error correction, …
- **Complex:** translation, question answering, speech recognition, …
- **Unknown:** human-level comprehension (is this just NLP?)

The World's Languages



Over **7,000** living languages. NLP coverage is deeply uneven across them.

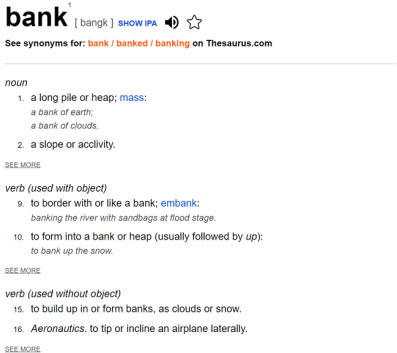
Machine Translation



[Google Translate, 2025-12]

Why is NLP difficult?

- **Ambiguity:** one form can have multiple meanings.



bank[1] [bangk] SHOW IPA

See synonyms for: bank / banked / banking on Thesaurus.com

noun

1. a long pile or heap; mass:
   a bank of earth;
   a bank of clouds.
2. a slope or acclivity.

SEE MORE

verb (used with object)

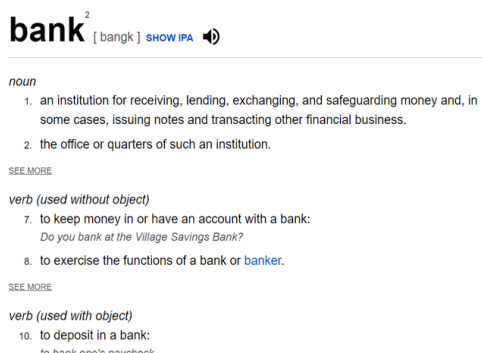
9. to border with or like a bank; embank:
   banking the river with sandbags at flood stage.
10. to form into a bank or heap (usually followed by *up*):
    to bank up the snow.

SEE MORE

verb (used without object)

15. to build up in or form banks, as clouds or snow.
16. *Aeronautics.* to tip or incline an airplane laterally.

SEE MORE



bank[2] [bangk] SHOW IPA

noun

1. an institution for receiving, lending, exchanging, and safeguarding money and, in some cases, issuing notes and transacting other financial business.
2. the office or quarters of such an institution.

SEE MORE

verb (used without object)

7. to keep money in or have an account with a bank:
   Do you bank at the Village Savings Bank?
8. to exercise the functions of a bank or banker.

SEE MORE

verb (used with object)

10. to deposit in a bank:
    to bank one's paycheck.

Why is NLP difficult?

- **Ambiguity:** one form can have multiple meanings.

She saw a cat with a telescope.

Who was with a telescope?

Why is NLP difficult?

- **Ambiguity:** one form can have multiple meanings.

She saw a cat with a telescope.

Who was with a telescope?

Why is NLP difficult?

- **Ambiguity:** one form can have multiple meanings.
- **Variability:** multiple forms can share the same meaning.

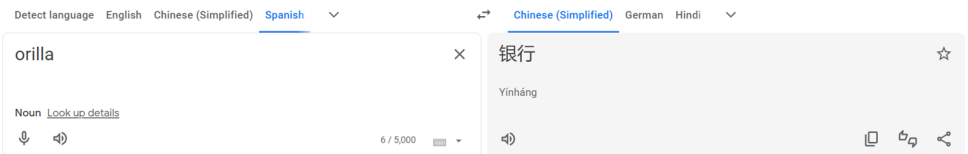
The cat is chasing the mouse.
The mouse is being chased by the cat.

Why is NLP difficult?

- **Ambiguity:** one form can have multiple meanings.
- **Variability:** multiple forms can share the same meaning.
- **Cross-lingual awareness, dialects, and accents.**

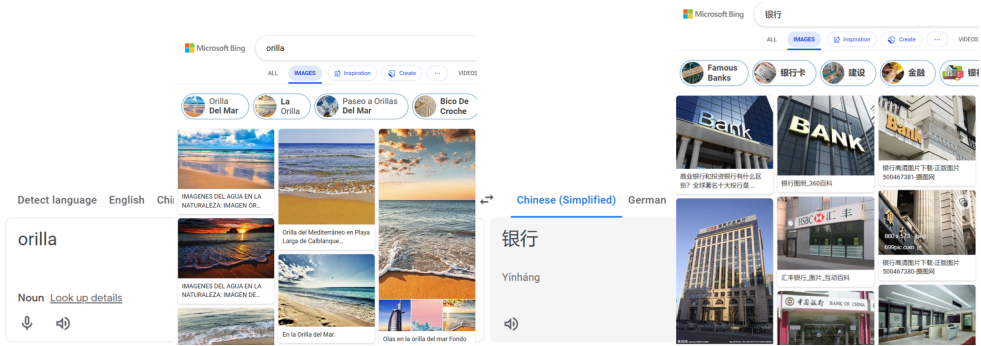
Why is NLP difficult?

- **Ambiguity:** one form can have multiple meanings.
- **Variability:** multiple forms can share the same meaning.
- **Cross-lingual awareness, dialects, and accents.**



Why is NLP difficult?

- **Ambiguity:** one form can have multiple meanings.
- **Variability:** multiple forms can share the same meaning.
- **Cross-lingual awareness, dialects, and accents.**



Why is NLP difficult?

- **Ambiguity:** one form can have multiple meanings.
- **Variability:** multiple forms can share the same meaning.
- **Cross-lingual awareness, dialects, and accents.**
- **Underlying meanings:** politeness, humor, irony.

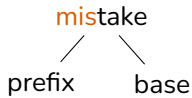
Q: Do you have iced tea?
A1: No.
A2: We do have iced coffee.

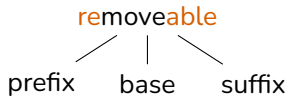
Course Roadmap

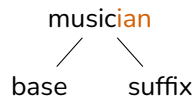
- **Part I: Foundations** (Weeks 1-4)
  What is NLP, language models, probability/information theory, tokenization, embeddings, supervised learning, statistical language models.
- **Part II: The Core Engine** (Weeks 4-6)
  Neural networks (RNNs, LSTMs) and Transformers from scratch; neural and masked LMs; pretraining at scale; syntax and CFGs.
- **Part III: Understanding and Improving Models** (Weeks 7-10)
  Probing and interpretability; evaluation and benchmarking; decoding and in-context learning; finetuning and alignment; reasoning; efficiency.
- **Part IV: The Frontier** (Weeks 10-12)
  Safety, bias, and societal impact; RAG and agents; multimodality and grounding.

Roadmap: Levels of Language

- **Morphology:** What words/subwords are we dealing with?

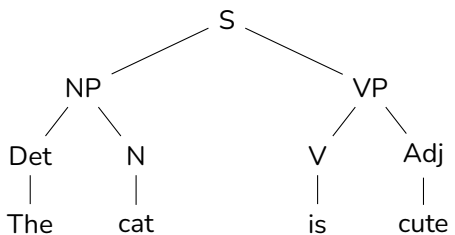






Roadmap: Levels of Language

- **Morphology:** What words/subwords are we dealing with?
- **Syntax:** What phrases are we dealing with?



Roadmap: Levels of Language

- **Morphology:** What words/subwords are we dealing with?
- **Syntax:** What phrases are we dealing with?
- **Semantics:** What is the literal meaning?

What excellent weather!

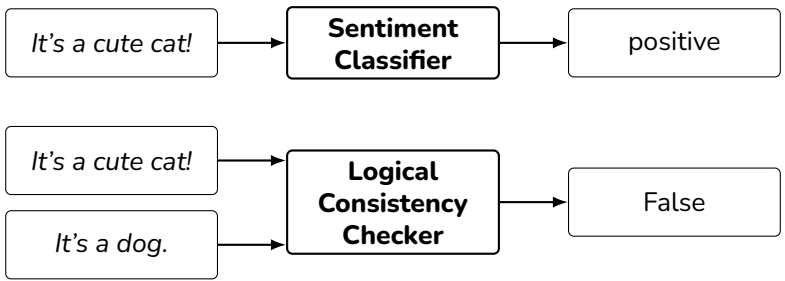
Roadmap: Levels of Language

- **Morphology:** What words/subwords are we dealing with?
- **Syntax:** What phrases are we dealing with?
- **Semantics:** What is the literal meaning?
- **Pragmatics:** What is the underlying meaning?

What excellent weather!

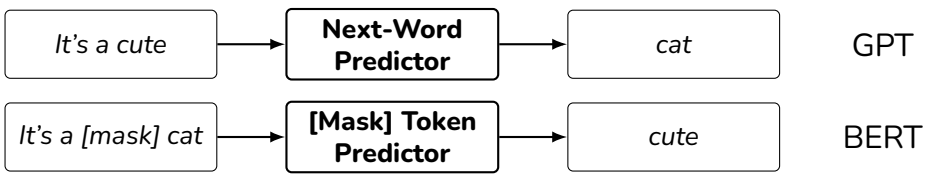
Roadmap: Modeling Approaches

- **Classification**



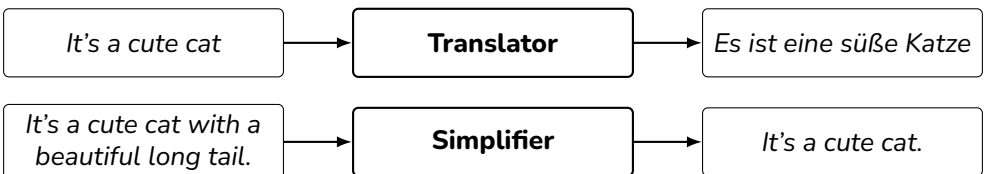
Roadmap: Modeling Approaches

- Classification
- **Language modeling**



Roadmap: Modeling Approaches

- Classification
- Language modeling
- **Sequence-to-sequence modeling**



Brief History of NLP

1960s-1990s: Classical NLP

- Linguistic theories
- Manually-defined rules
- Small-scale datasets and experiments

1980s-2013: Statistical NLP

- Supervised ML with annotated data
- (Mostly) linear models with hand-crafted features
- Linguistic knowledge used in annotation

2012-now: Neural NLP

- Neural networks as primary architecture
- Representations learned from large corpora
- Less hand-crafting of features

2022-now: Large language models

- Pretrain a language model
- Adapt the pretrained LM to various tasks

From Shannon to GPT

- Shannon's prediction-and-entropy-of-English experiments (1948, 1951) framed language modeling as *uncertainty about the next symbol*.
- Statistical MT (IBM models, 1990s) introduced data-driven alignment.
- Word embeddings, Transformers, and the pretraining paradigm drove the current LLM era.

Overview: Language Models

What is a Language Model?

A **language model** assigns a probability to a sequence of tokens:

$$P(\langle w_1, w_2, \ldots, w_T \rangle).$$

It answers: *how likely is this string of text?*

What is a Language Model?

A **language model** assigns a probability to a sequence of tokens:

$$P(\langle w_1, w_2, \ldots, w_T \rangle).$$

It answers: *how likely is this string of text?*

Which sentence is more probable?

(a) *The cat is sleeping on the mat.*

(b) *Sleeping the cat mat the on is.*

What is a Language Model?

A **language model** assigns a probability to a sequence of tokens:

$$P(\langle w_1, w_2, \ldots, w_T \rangle).$$

It answers: *how likely is this string of text?*

Which sentence is more probable?

(a) *The cat is sleeping on the mat.*
(b) *Sleeping the cat mat the on is.*

(a) *Students opened their books.*
(b) *Students opened their windows.*

What is a Language Model?

A **language model** assigns a probability to a sequence of tokens:

$$P(\langle w_1, w_2, \ldots, w_T \rangle).$$

It answers: *how likely is this string of text?*

Which sentence is more probable?

(a) *The cat is sleeping on the mat.*

(b) *Sleeping the cat mat the on is.*

(a) *Students opened their books.*

(b) *Students opened their windows.*

A good LM gives higher probability to sentences that are both *grammatical* and *plausible in context*.

Background: Common Notations of (Discrete) Probability

$P(X = x)$ denotes the probability that random variable X takes on the value x. Given two random variables X, Y:

Joint probability: $P(X = x, Y = y) = P(x, y)$.

Background: Common Notations of (Discrete) Probability

$P(X = x)$ denotes the probability that random variable X takes on the value x.
Given two random variables X, Y:

Joint probability: $P(X = x, Y = y) = P(x, y)$.

Marginal probability: $P(X = x) = P(x) = \sum_y P(x, y)$.
X, Y are independent iff $P(x, y) = P(x)\,P(y)$.

Background: Common Notations of (Discrete) Probability

$P(X = x)$ denotes the probability that random variable X takes on the value x.
Given two random variables X, Y:

Joint probability: $P(X = x, Y = y) = P(x, y)$.

Marginal probability: $P(X = x) = P(x) = \sum_y P(x, y)$.
X, Y are independent iff $P(x, y) = P(x)\,P(y)$.

Conditional probability: $P(X = x \mid Y = y) = P(x \mid y) = \dfrac{P(x, y)}{P(y)}$.
X, Y independent $\iff P(x \mid y) = P(x) \iff P(y \mid x) = P(y)$.

Background: Common Notations of (Discrete) Probability

$P(X = x)$ denotes the probability that random variable X takes on the value x.
Given two random variables X, Y:

Joint probability: $P(X = x, Y = y) = P(x, y)$.

Marginal probability: $P(X = x) = P(x) = \sum_y P(x, y)$.
X, Y are independent iff $P(x, y) = P(x)\,P(y)$.

Conditional probability: $P(X = x \mid Y = y) = P(x \mid y) = \dfrac{P(x, y)}{P(y)}$.
X, Y independent $\iff P(x \mid y) = P(x) \iff P(y \mid x) = P(y)$.

Expectation: $\mathbb{E}[X] = \mathbb{E}_{x \sim P}[x] = \sum_x x\,P(x)$.

Background: Chain Rule of Probability

For any random variables $X_1, X_2, \ldots, X_n$:

$$\begin{aligned} P(x_1, x_2, \ldots, x_n) &= P(x_1)\, P(x_2 \mid x_1)\, P(x_3 \mid x_1, x_2) \cdots P(x_n \mid x_1, \ldots, x_{n-1}) \\ &= \prod_{i=1}^{n} P(x_i \mid x_{<i}). \end{aligned}$$

No assumptions are made: the chain rule follows directly from the definition of conditional probability.

Background: Chain Rule of Probability

For any random variables $X_1, X_2, \ldots, X_n$:

$$\begin{aligned} P(x_1, x_2, \ldots, x_n) &= P(x_1)\, P(x_2 \mid x_1)\, P(x_3 \mid x_1, x_2) \cdots P(x_n \mid x_1, \ldots, x_{n-1}) \\ &= \prod_{i=1}^{n} P(x_i \mid x_{<i}). \end{aligned}$$

No assumptions are made: the chain rule follows directly from the definition of conditional probability.

Consequence for language modeling:

$$P(w_1, \ldots, w_T) = \prod_{t=1}^{T} P(w_t \mid w_{<t}).$$

Different LM families (n-gram, RNN, Transformer) are different *approximations* of $P(w_t \mid w_{<t})$.

Language Modeling with Next-Token Predictors

By the **chain rule** of probability,

$$P(w_1, \ldots, w_T, \texttt{<eos>}) = \left(\prod_{t=1}^{T} P(w_t \mid w_1, \ldots, w_{t-1}) \right) P(\texttt{<eos>} \mid w_1, \ldots, w_T).$$

Language Modeling with Next-Token Predictors

By the **chain rule** of probability,

$$P\left(w_1, \ldots, w_T, \texttt{<eos>}\right) \;=\; \left(\prod_{t=1}^{T} P(w_t \mid w_1, \ldots, w_{t-1})\right) P\left(\texttt{<eos>} \mid w_1, \ldots, w_T\right).$$

So modeling $P(w_1, \ldots, w_T)$ reduces to modeling the conditional

$$P(w_t \mid w_{<t}).$$

Language Modeling with Next-Token Predictors

By the **chain rule** of probability,

$$P(w_1, \ldots, w_T, \texttt{<eos>}) = \left(\prod_{t=1}^{T} P(w_t \mid w_1, \ldots, w_{t-1}) \right) P(\texttt{<eos>} \mid w_1, \ldots, w_T).$$

So modeling $P(w_1, \ldots, w_T)$ reduces to modeling the conditional

$$P(w_t \mid w_{<t}).$$

This *autoregressive* factorization is what GPT-style models implement.

From Prediction to Generation

An LM does not only *score* text. Repeatedly **sampling** from $P(w_t \mid w_{<t})$ and feeding each draw back in *generates* text.

Repeat until <eos>.

From Prediction to Generation

An LM does not only *score* text. Repeatedly **sampling** from $P(w_t \mid w_{<t})$ and feeding each draw back in *generates* text.

Repeat until <eos>.

How we sample (effectively and efficiently) is a *decoding* topic we return to later.

More Background

Bayes' Theorem

$$P(y \mid x) \;=\; \frac{P(x \mid y)\,P(y)}{P(x)} \;=\; \frac{P(x \mid y)\,P(y)}{\sum_{y'} P(x \mid y')\,P(y')}.$$

Bayes' Theorem

$$P(y \mid x) = \frac{P(x \mid y)\,P(y)}{P(x)} = \frac{P(x \mid y)\,P(y)}{\sum_{y'} P(x \mid y')\,P(y')}.$$

$$\underbrace{P(y \mid x)}_{\text{posterior}} \propto \underbrace{P(x \mid y)}_{\text{likelihood}}\,\underbrace{P(y)}_{\text{prior}}.$$

Bayes' Theorem

$$P(y \mid x) \;=\; \frac{P(x \mid y)\,P(y)}{P(x)} \;=\; \frac{P(x \mid y)\,P(y)}{\sum_{y'} P(x \mid y')\,P(y')}.$$

$$\underbrace{P(y \mid x)}_{\text{posterior}} \;\propto\; \underbrace{P(x \mid y)}_{\text{likelihood}}\,\underbrace{P(y)}_{\text{prior}}.$$

Example: sentiment classification Let x be a review and $y \in \{\text{pos}, \text{neg}\}$ its label. Then

$$P(y \mid x) \;\propto\; P(x \mid y)\,P(y).$$

$P(x \mid y)$ is a *class-conditional* language model; $P(y)$ is the label prior. This generative view is the basis of Naïve Bayes, and is a recurring pattern in NLP.

Entropy

For a discrete random variable $X \sim P$, the **entropy** is

$$H(X) = -\sum_{x} P(x) \log_2 P(x) = \mathbb{E}_{x \sim P}[-\log_2 P(x)].$$

Entropy

For a discrete random variable $X \sim P$, the **entropy** is

$$H(X) = -\sum_x P(x) \log_2 P(x) = \mathbb{E}_{x \sim P}[-\log_2 P(x)].$$

Two interpretations:

- *Average surprise*: how unpredictable a draw from P is.

Entropy

For a discrete random variable $X \sim P$, the **entropy** is

$$H(X) = -\sum_{x} P(x) \log_2 P(x) = \mathbb{E}_{x \sim P}[-\log_2 P(x)].$$

Two interpretations:

- *Average surprise*: how unpredictable a draw from P is.
- *Expected code length*: the minimum average number of bits needed to encode a draw from P (Shannon's source-coding theorem).

Entropy

For a discrete random variable $X \sim P$, the **entropy** is

$$H(X) = -\sum_x P(x) \log_2 P(x) = \mathbb{E}_{x \sim P}[-\log_2 P(x)].$$

Two interpretations:

- *Average surprise*: how unpredictable a draw from P is.
- *Expected code length*: the minimum average number of bits needed to encode a draw from P (Shannon's source-coding theorem).

Bounded by $0 \leq H(X) \leq \log_2 |\mathcal{X}|$; maximum at the uniform distribution.
Practice: why?

Information Content and Optimal Code Lengths

The **information content** (self-information) of outcome x is

$$I(x) = -\log_2 P(x) \quad \text{bits.}$$

- Rare event ($P(x)$ small) $\Rightarrow$ large $I(x)$: more surprising, carries more info.
- Certain event ($P(x) = 1$) $\Rightarrow I(x) = 0$: no information gained.

Information Content and Optimal Code Lengths

The **information content** (self-information) of outcome x is

$$I(x) = -\log_2 P(x) \quad \text{bits.}$$

- Rare event ($P(x)$ small) $\Rightarrow$ large $I(x)$: more surprising, carries more info.
- Certain event ($P(x) = 1$) $\Rightarrow I(x) = 0$: no information gained.

Optimal prefix code: assign a codeword of length $\lceil -\log_2 P(x) \rceil$ bits to symbol x. The expected code length then equals the entropy:

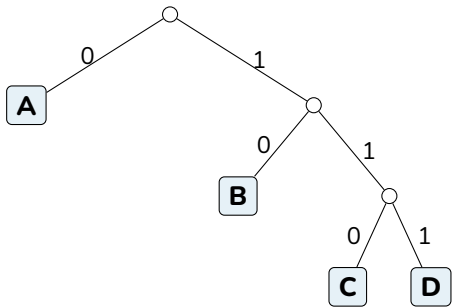
$$\mathbb{E}[\text{code length}] = \sum_x P(x) \cdot (-\log_2 P(x)) = H(X).$$

Huffman Coding: Optimal Encoding

Example: $P(A) = \frac{1}{2},\ P(B) = \frac{1}{4},\ P(C) = P(D) = \frac{1}{8}$

Huffman Coding: Optimal Encoding

Example: $P(A) = \frac{1}{2},\ P(B) = \frac{1}{4},\ P(C) = P(D) = \frac{1}{8}$



Symbol	Code	Bits
A	0	1
B	10	2
C	110	3
D	111	3

Expected length:

$\frac{1}{2}(1) + \frac{1}{4}(2) + \frac{1}{8}(3) + \frac{1}{8}(3) = 1.75$ bits

$H(X) = 1.75$ bits. ✓

Cross-Entropy, KL Divergence, and Perplexity

Cross-entropy of a *true* distribution P and a *model* Q:

$$H(P, Q) \;=\; -\sum_{x} P(x) \log_2 Q(x) \;=\; \mathbb{E}_{x\sim P}[-\log_2 Q(x)]\,.$$

Bits per draw from P when encoded with Q's code.

Cross-Entropy, KL Divergence, and Perplexity

Cross-entropy of a *true* distribution P and a *model* Q:

$$H(P, Q) \;=\; -\sum_{x} P(x) \log_2 Q(x) \;=\; \mathbb{E}_{x \sim P}[-\log_2 Q(x)]\,.$$

Bits per draw from P when encoded with Q's code.

The **Kullback-Leibler divergence** measures the discrepancy between P and Q ($= 0$ iff $P = Q$):

$$\mathrm{KL}(P \,\|\, Q) \;=\; \sum_{x} P(x) \log_2 \frac{P(x)}{Q(x)} \;=\; H(P, Q) - H(P) \;\geq\; 0.$$

Cross-Entropy, KL Divergence, and Perplexity

Cross-entropy of a *true* distribution P and a *model* Q:

$$H(P, Q) \;=\; -\sum_{x} P(x) \log_2 Q(x) \;=\; \mathbb{E}_{x\sim P}[-\log_2 Q(x)]\,.$$

Bits per draw from P when encoded with Q's code.

The **Kullback-Leibler divergence** measures the discrepancy between P and Q ($= 0$ iff $P = Q$):

$$\mathrm{KL}(P \,\|\, Q) \;=\; \sum_{x} P(x) \log_2 \frac{P(x)}{Q(x)} \;=\; H(P, Q) - H(P) \;\geq\; 0.$$

LM training. Maximum likelihood minimizes cross-entropy: $\min_\theta H(P_{\text{data}}, P_\theta)$.

LM evaluation. Perplexity (lower is better):

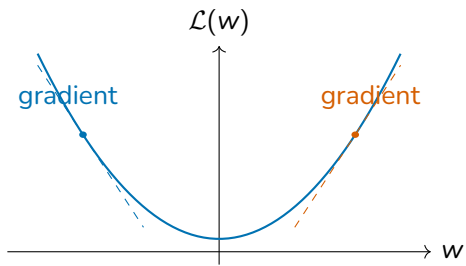
$$\mathrm{PPL}(w_1, \ldots, w_T) \;=\; 2^{-\frac{1}{T}\sum_{t=1}^{T} \log_2 Q(w_t | w_{<t})}.$$

Finding the Optimum of a Function

To minimize a convex function $\mathcal{L}(w; X, y)$, take steps opposite to the gradient:

$$w' \leftarrow w - \eta \nabla_w \mathcal{L}(w; X, y),$$

η: *learning rate* ("step size")

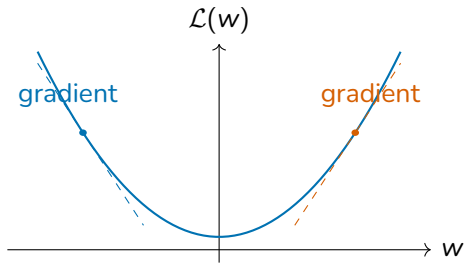


Finding the Optimum of a Function

To minimize a convex function $\mathcal{L}(w; X, y)$, take steps opposite to the gradient:

$$w' \leftarrow w - \eta \nabla_w \mathcal{L}(w; X, y),$$

η: *learning rate* ("step size")



The **gradient** $\nabla_w \mathcal{L}$ is the vector of partial derivatives: $\nabla_w \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial w_1}, \ldots, \frac{\partial \mathcal{L}}{\partial w_d}\right)^\top$; each component $\frac{\partial \mathcal{L}}{\partial w_i}$ measures how $\mathcal{L}$ changes as w_i varies alone.

Next Lecture

Words: Tokenization and Morphology

- What is a word?
- Morphological typology: isolating, agglutinative, fusional.
- Byte-Pair Encoding (BPE) from scratch; WordPiece; Unigram LM tokenizer.
- Cross-lingual considerations: vocabulary sharing, low-resource scripts.