# ECE448/CS440 Artificial Intelligence

CS 440 Artificial Intelligence

https://github.com/illinois-cs-coursework

https://courses.grainger.illinois.edu/cs440/fa2025/readings.html

> Important
>
> https://courses.grainger.illinois.edu/cs440/fa2025/lectures/probability-review.html

## Introduction

> https://courses.grainger.illinois.edu/cs440/fa2025/lectures/intro.html

### Historical and other trivia

We've seen a lot of trivia, most of it not worth memorizing. The following items are the exceptions. Be able to explain (very briefly) what they are and (approximately) what time period they come from.

- **McCulloch and Pitts**
    - **Time Period:** 1940s
    - **Contribution:** They introduced the *first mathematical model of a neural network*. Their work was foundational, proposing that networks of simple computational units (neurons) could perform complex logical operations. These were *theoretical models on paper*, as the hardware to implement them didn't exist yet.
- **Fred Jelinek**

    - **Time Period:** 1980s – 1990s
    - **Contribution:** A key figure in *speech recognition*. He pioneered the use of statistical models, specifically *n-gram language models* and Hidden Markov Models (HMMs), which dramatically improved the accuracy and utility of speech recognition systems.

- **Pantel and Lin (SpamCop)**
    - **Time Period:** Late 1990s
    - **Contribution:** They were pioneers in using *Naive Bayes classifiers for spam detection*. Their work showed that this statistical approach was highly effective for classifying emails, forming the basis of many modern spam filters.
        - 朴素贝叶斯垃圾邮件分类器

- **Boulis and Ostendorf**
    - **Time Period:** Mid 2000s
    - **Contribution:** They conducted research comparing the performance of Naive Bayes versus Support Vector Machine (SVM) classifiers for gender classification based on transcribed telephone conversations.
    - A Quantitative Analysis of Lexical Differences Between Genders in Telephone Conversations, ACL 2005

- **The Plato System**
    - **Time Period:** Started in the 1960s
    - **Contribution:** An early and influential *computer-assisted instruction system* developed at the University of Illinois. It was a precursor to modern *e-learning platforms and online communities*.

- **The Golem of Prague**
    - **Time Period:** 16th-century Jewish folklore
    - **Contribution:** An early myth or story related to artificial intelligence. It tells of an artificial humanoid creature created from clay to protect the Jewish community. It represents an ancient human desire to create

intelligent, autonomous beings.

## Probability

> https://courses.grainger.illinois.edu/cs440/fa2025/lectures/probability-review.html

**Random variables, axioms of probability:**

- A **random variable** is a variable whose value is a numerical outcome of a random phenomenon.
- The **axioms of probability** (**Kolmogorov's axioms of probability** 柯尔莫哥洛夫概率公理) are fundamental rules:
    1. The probability of any event is non-negative.
    2. The probability of the entire sample space (a certain event) is 1.
    3. The probability of the union of mutually exclusive events is the sum of their individual probabilities.

$$
\begin{aligned}
0 \leq P(A) \\
P(True) = 1 \\
P(A|B) = P(A) + P(B), \text{if A and B are mutually exclusive events}
\end{aligned} \tag{1}
$$

- **Joint, marginal, conditional probability:**
    - **Joint Probability** 联合概率 $P(A, B)$: The probability of two events occurring together
    - **Marginal Probability** 边际概率 $P(A)$: The probability of a single event occurring, irrespective of other events. It can be calculated by summing the joint probabilities over all outcomes of the other variable: $P(A) = \sum_B P(A, B)$
    - **Conditional Probability** 条件概率 $P(A \mid B)$: The probability of event A occurring *given* that event B has already occurred. It is calculated as $P(A|B) = \frac{P(A,B)}{P(B)}$

## Modelling Text Data

### Word types vs. word tokens:

- **Tokens:** The total number of words in a document (e.g., "the cat sat on the mat" has 6 tokens). 单词总数
- **Types:** The number of *unique* words in a document (e.g., "the cat sat on the mat" has 5 types: "the", "cat", "sat", "on", "mat"). 词典条目，唯一的单词数

**The Bag of Words model:** A bag-of-words model determines the class of a document based on the *frequency* of occurrence of each word. It ignores the order in which words occur, which ones occur together, etc. So it will miss some details, e.g. the difference between "poisonous" and "not poisonous." 忽略语法甚至词序但保持多样性

### Bigrams, ngrams:

- **N-grams** are contiguous sequences of *n* items (e.g., words, letters) from a given sample of text.
    - **N-gram** 是来自给定文本样本的 *n* 个项目（例如单词、字母）的连续序列。
- A **bigram** is a specific n-gram where n=2 (a two-word sequence). For example, in "the cat sat", the bigrams are "the cat" and "cat sat".
    - 特定的 n-gram，其中 n=2（即两个单词的序列）。例如，在“the cat sat”中，二元语法是“the cat”和“cat sat”。

### Data cleaning:

- **Tokenization:** The process of splitting a stream of text into words, phrases, symbols, or other meaningful elements called tokens.
    - **标记化：** 将文本流拆分为单词、短语、符号或其他有意义的元素（称为标记）的过程。定义单词得到 a clean string of words
    - divide at whitespace 在空白处划分
    - normalize punctuation, html tags, capitalization, etc 规范标点符号、html 标签、大写字母等
    - perhaps use a stemmer to remove word endings 使用词干分析器来删除

单词结尾

- **Stemming 分词:** The process of reducing inflected (or sometimes derived) words to their word stem, base or root form. **Julie Lovins** (1968) created one of the first stemming algorithms, and **Martin Porter** (1980) developed the Porter Stemmer, which is one of the most widely used.
    - **词干提取：** 将词形变化的词简化为词干、基词或词根形式的过程。Julie Lovins 创建了最早的词干提取算法之一， Martin Porter 开发了 Porter 词干提取器，它是目前使用最广泛的算法之一。
- **Making units of useful size:** This involves either breaking long words into smaller pieces (common in languages like German) or grouping characters into words (necessary for languages without spaces, like Chinese).
    - 将长单词分成更小的部分，特别是中文（没有空格）

**Special types of words:**

- **Stop words:** Very common words (e.g., "the", "a", "is") that are often removed before processing because they carry little semantic weight.
    - 非常常见的词：function words, fillers, backchannel
- **Rare words:** Words that appear very infrequently. They can be problematic for statistical models and are sometimes removed or replaced with a generic "UNK" (unknown) token.
    - 生僻词：出现频率极低的词，删除一部分或都用UNK标记（视为一个单独的项目）
- **Hapax legomena:** Words that occur only once in a corpus. They are an extreme case of rare words.
    - 罕见词的极端情况，只出现一次
- **Filler:** Words or sounds used to pause in a conversation (e.g., "um," "uh," "like").
    - 填充词
- **Backchannel:** Signals from a listener that indicate they are paying attention

(e.g., "uh–huh," "yeah," "I see").
    - 听众发出的信号词
- **Function vs. content words:** *Content words* (nouns, main verbs, adjectives) carry the primary meaning. *Function words* (prepositions, articles, conjunctions) provide grammatical structure.
    - **实词** （名词、主要动词、形容词）承载主要含义
    - **功能词** （介词、冠词、连词）提供语法结构

## Testing

### Roles of training, development, test datasets:

- **Training set 训练集:** The data used to train the model and learn its parameters.
- **Development set (or validation set) 验证集:** The data used to tune the model's hyperparameters and make design choices. It helps prevent overfitting to the training set. 防止过度拟合
- **Test set 测试集:** The data held back until the very end to provide an unbiased, final evaluation of the model's performance. 保留到最后的数据，以对模型的性能提供公正的最终评估。

### Evaluation metrics for classification:

| | **Labels from Algorithm** | |
|---|---|---|
| | **Cancer** | **Not Cancer** |
| **Correct = Cancer** | True Positive (TP) | False Negative (FN) |
| **Correct = Not Cancer** | False Positive (FP) | True Negative (TN) |

We can summarize performance using the rates at which errors occur:

- False positive rate = FP/(FP+TN) [how many wrong things are in the negative outputs]
- False negative rate = FN/(TP+FN) [how many wrong things are in the positive outputs]
- Accuracy = (TP+TN)/(TP+TN+FP+FN)
- Error rate = 1-accuracy

| Confusion Matrix 混淆矩阵 | Labels from Algorithm | |
|---|---|---|
| / | happen | Not happen |
| Correct = happen | True Positive (TP) | False Negative (FN) |
| Correct = not happen | False Positive (FP) | True Negative (TN) |

- **False positive rate** = $FP/(FP+TN)$ [how many wrong things are in the negative outputs]
- **False negative rate** = $FN/(TP+FN)$ [how many wrong things are in the positive outputs]
- **Accuracy** = $(TP+TN)/(TP+TN+FP+FN)$
    - The fraction of predictions the model got righ
- **Error rate** = $1-accuracy$
- **precision (p)** = $TP/(TP+FP)$ [how many of our outputs were correct?]
- **recall (r)** = $TP/(TP+FN)$
    - True Positive [how many of the correct answers did we find?]
- **F1** = $2pr/(p+r)$
    - F1 is the harmonic mean of precision and recall. Both recall and precision need to be good to get a high F1 value.
- **Confusion Matrix:** A table that visualizes the performance of a classifier, showing the counts of true positives, true negatives, false positives, and false negatives.

## Naive Bayes

> https://courses.grainger.illinois.edu/cs440/fa2025/lectures/bayes.html

### Basic definitions and mathematical model

- P(A | C) is the probability of A in a context where C is true
    - Definition of conditional probability: $P(A|C) = \frac{P(A,C)}{P(C)}$
        - $P(A) = \sum P(A|Z)p(Z,\theta)$
    - $P(A,C) = P(A) \times P(C|A) = P(C,A) = P(C) \times P(A|C)$
    - **Bayes Rule**: $P(C|A) = \frac{P(A|C) \times P(C)}{P(A)}$
        - $P(cause|evidence) = \frac{P(evidence|cause)P(cause)}{P(evidence)}$
        - posterior likelihood prior normalization
- **Likelihood:** $P(evidence \mid cause)$ 概率
- **Prior:** $P(cause)$ 先验
- **Posterior:** $P(cause \mid evidence)$ 后验
- **argmax operator:** Returns the input value that maximizes a function. In classification, we use it to find the class with the highest posterior probability.
    - 表示返回使函数最大化的输入值的符号，用来找到后验概率最高的类
- **Independence vs. Conditional Independence:** Naive Bayes makes a "naive" assumption of *conditional independence* of features: features are independent of each other *given the class*. This is a stronger assumption than simple independence.
    - **Independence** 独立性
        - Two events A and B are independent **iff** $P(A,B) = P(A) \times P(B)$
    - **Conditional Independence** 条件独立性
        - Definition： $P(A,B|C) = P(A|C) \times P(B|C)$, 等价于 $P(A|B) = P(A), P(B|A) = P(B)$
    - 独立性很少成立；条件独立性是在特定的上下文中，两个变量是否独立，近似合理。
- **MAP vs. ML estimate**
    - **Maximum Likelihood (ML)** chooses the parameters that maximize the likelihood of the data. 概率最大化

    - **Maximum a Posteriori (MAP)** incorporates a prior probability, choosing parameters that maximize the posterior probability. The prior acts as a regularizer. 后验概率最大化

- **Combining evidence:** Under the conditional independence assumption, the likelihood of all evidence is simply the product of the likelihoods of each individual piece of evidence:
    - $P(evidence_1, \ldots, evidencen \mid cause) = \prod_i P(evidence_i \mid cause)$.
- **Model size:** Naive Bayes dramatically reduces the number of parameters needed compared to a full joint distribution table, making it computationally feasible and less prone to overfitting on small datasets.

## Applying Naive Bayes to text classification

- **Estimation equations:** You estimate the prior probability of a class by its frequency in the training data, and the likelihood of a word given a class by its frequency within documents of that class.
    - 估计方程，通过训练数据中某个类别的频率来估计该类别的先验概率
- **Avoiding underflow:** Since you are multiplying many small probabilities, the result can become too small for a computer to store (underflow). To fix this, you work with the sum of log probabilities instead:
$log(A \cdot B) = log(A) + log(B)$.
    - 对数防止乘以太小的数字影响准确性（计算精度），将朴素贝叶斯算法最大化
- **Avoiding overfitting (Smoothing)** 平滑处理
    - **Why it's important:** If a word never appears in the training data for a certain class, its probability will be zero, causing the entire posterior probability for that class to become zero. 过度拟合，0会破坏朴素贝叶斯算法
    - **Laplace smoothing:** Adds a small constant (usually 1) to every count, ensuring no probability is ever zero.

- 拉普拉斯平滑：

  $$V = \text{number of word TYPES seen in training data}$$
  $$P(\text{UNK} \mid C) = \frac{\alpha}{n+\alpha(V+1)}$$
  $$P(W \mid C) = \frac{\text{count}(W)+\alpha}{n+\alpha(V+1)}$$

  - $n$ = number of words in our Class C training data
    $count(W)$ = number of times W appeared in Class C training data

    $\alpha$: a constant positive number
    $V$ = number of word **types** seen in training data

- **Deleted estimation 删除估计:** A cross-validation technique used to find optimal smoothing parameters.
  - 对于每个观测到的计数 r，我们将计算一个校正后的计数 Corr(r)。假设 W1,...,Wn是在数据集前半部分出现 r 次的单词。对于此集合中的每个单词 Wk，求出它在数据集后半部分出现的计数 C(Wk) 。我们将这些计数取平均值，得到校正后的计数：
  - $\text{Corr}(r) = \frac{\sum_{k=1}^{n} C(W_k)}{n}$ Corr(r) 预测训练数据中出现 r 次的单词的未来计数。
  - 删除估计已被证明比拉普拉斯平滑更准确

- **N-gram smoothing:** Refers to more advanced techniques (like Good-Turing, Kneser-Ney) used for n-gram models to handle unseen n-grams by redistributing probability mass from seen n-grams. The high-level idea is to "borrow" probability from more frequent events to assign to rare or unseen events.
  - 对于一元模型，我们需要估计 n 个概率，而对于二元模型，我们需要估计 $n^2$ 个概率，但我们仍然拥有相同的 m 个单词作为训练数据
  - Idea 1: If we haven't seen an ngram, guess its probability from the probabilites of its *prefix* (e.g. "the angry") and the *last word* ("armadillo").
  - Idea 2: Guess that an unseen word is more likely in contexts where we've seen many different words.

# Search

State graph representations 状态图

A state graph has the following key parts:

- states (graph nodes) 状态（图节点）
- actions (graph edges, with costs)
- start state 起始状态
- goal states (explicit list, or a goal condition to test) 目标状态（明确列表，或要测试的目标条件）

Basic search outline

Our search will use two main data structures:

- a table of states that have been visited
- the frontier: a queue of states whose outgoing edges still need to be explored

Basic outline for search code

- Loop until you find a goal state or queue is empty
- pop state S off frontier
- follow outgoing edges to find S's neighbors
- for each neighbor X that has not yet been visited, add X to the visited table and to the frontier

Our data structure for storing the frontier determines the type of search

- breadth–first search (BFS): queue
- uniform–cost search (UCS): priority queue using cost so far
- A∗search: priority queue using estimated total cost

BFS

- it returns a path that uses the minimum number of edges

UCS

- 即使边成本发生变化，也能找到最优路径

# Readings

## Quiz 1

### An algorithm for suffix stripping 后缀剥离

The main goal of the paper is to introduce a simple, fast, and effective algorithm for **suffix stripping** (also known as **stemming**). This process removes common endings from words to get to their root form, or "stem."

#### Why is Stemming Important? 🤔

- The primary use is in **Information Retrieval (IR)**, like search engines.
- It groups related words together. For example, **"connect," "connected," "connecting,"** and **"connection"** all get reduced to the single stem **"connect."**
- This **improves search results** because a search for "connecting" will also find documents that only mention "connection."
- It also reduces the total number of unique words a system has to store, making the system smaller and more efficient.

#### How the Algorithm Works ⚙️

The algorithm's cleverness lies in its simplicity. It doesn't use a dictionary. Instead, it uses a set of rules based on the structure of the word itself.

#### The "Measure" of a Word

The core concept is the **measure (m)** of a stem, which roughly corresponds to the number of vowel–consonant sequences it contains.

A word is first broken down into a sequence of vowel groups (V) and consonant groups (C). The form of any word can be represented as:

```
[C](VC)^m[V]
```

- `C`: one or more consonants
- `V`: one or more vowels
- `m`: the **measure**, or how many times the `(VC)` group repeats.

Here are some examples:

- **TR, EE, TREE** →m=0
- **TROUBLE, OATS** →m=1
- **TROUBLES, PRIVATE** →m=2

This measure (m) is used to prevent the algorithm from stripping suffixes from words that are already very short. 10For example, it will remove `-ATE` from "ACTIVATE" (stem `ACTIV` has m>1) but not from "RELATE" (stem `REL` has m=1).

#### The 5 Steps of the Algorithm

The algorithm removes suffixes sequentially in five steps. A word goes through each step in order.

- **Step 1:** Deals with plurals and past participles. It changes suffixes like `-SSES` to `-SS` (e.g., `caresses` → `caress`), `-IES` to `-I` (e.g., `ponies` → `poni`), and removes `-ED` or `-ING` if the stem meets certain conditions.
- **Step 2:** Handles other common suffixes. If the stem's measure (m) is greater than 0, it will change suffixes like `-ATIONAL` to `-ATE` (e.g., `relational` → `relate`) or `-IZATION` to `-IZE` (e.g., `vietnamization` → `vietnamize`).
- **Step 3:** Continues with another set of suffixes for stems where m>0. It changes `-ICAL` to `-IC` (e.g., `electrical` → `electric`) or removes `-NESS` (e.g.,

`goodness` → `good`).

- **Step 4:** Removes a final set of suffixes like `-ANT`, `-ENCE`, `-ER`, and `-IVE`, but only if the stem is long enough (m>1). 16161616This is the step that would turn `GENERALIZE` into `GENERAL`.
- **Step 5:** This is a final cleanup step. It removes a trailing `-E` if the measure is large enough (e.g., `probate` → `probat`) and reduces a double `L` at the end of a word (e.g., `controll` → `control`).

#### Key Takeaways ✅

- **It's Pragmatic, Not Perfect:** The algorithm is designed for IR performance, not perfect linguistics. It will make errors, such as conflating "WAND" and "WANDER," but these errors are acceptable because the overall benefit is positive.
- **Simple is Better:** Despite its simplicity, it performed slightly *better* in tests than a much more complex stemming system.
- **Effective:** In a test with 10,000 words, the algorithm reduced the number of unique stems to 6,370, a reduction of about one-third.

### The Psychological Functions of Function Words

The central argument of the paper is that **function words**—small, common words like pronouns (I, you, we), prepositions (to, for), and articles (a, the)—are powerful indicators of our psychological state, personality, and social dynamics. Because we use these words unconsciously, they provide an unfiltered look into how we think and relate to others.

#### Function Words vs. Content Words

- **Content words** are nouns and regular verbs that carry the primary meaning or topic of what we're saying (e.g., "family," "health," "money").
- **Function words** are the "cement" that holds language together4. They don't have much meaning on their own but show *how* we are expressing ourselves.
- Though there are fewer than 400 function words in English, they account for

over **50% of the words we use** in daily life5.

- We have very little conscious control over or memory of using them, which makes them a great tool for psychological analysis.

### Key Findings from Word Analysis 👨‍🔬

The authors used a computer program called **LIWC (Linguistic Inquiry and Word Count)** to analyze millions of words from blogs, speeches, emails, and experiments7. Here are their most important findings:

1. Depression and Self-Focus

- A higher frequency of **first-person singular pronouns** ("I," "me," "my") is strongly linked to depression and negative emotions8. In fact, pronoun use is a better marker of depression than the use of negative emotion words like "sad" or "angry".
- Poets who committed suicide used "I" at significantly higher rates than non-suicidal poets, suggesting greater self-focus and less social integration.

2. Reactions to Stress

- **Individual Stress:** When facing personal crises (divorce, cancer diagnosis), Mayor Rudy Giuliani's use of "I" words shot up from about 2% to over 7%. This shows that personal distress often leads to an intense focus on the self.
- **Shared Stress:** After the 9/11 attacks, the opposite happened. People's use of "I" words *dropped*, while their use of **first-person plural pronouns** ("we," "us") increased. This suggests that a shared tragedy causes people to focus less on themselves and more on their community and social connections.

3. Honesty and Deception

- When people are **telling the truth**, they tend to use more **first-person singular pronouns** ("I") and more **exclusive words** (e.g., "but," "except," "without").
- This suggests that truthful accounts are more personal and cognitively complex, while lies are simpler and more detached.

4. Social Status

- In a conversation, the person with **higher status** consistently uses **fewer "I" words**.
- For example, in his conversations, President Nixon used "I" far less when speaking to his subordinates John Dean and John Erlichman than they did when speaking to him16. The lower-status person focuses on their own perspective, while the higher-status person focuses on the broader picture.

5. Demographics (Sex and Age)

- **Sex:** Females tend to use "I" words more than males. Males use more articles ("a," "the") and nouns, which is associated with concrete, categorical thinking18. Females use more verbs, which reflects a more relational focus.
- **Age:** As people get older, they use "I" less and "we" more20. They also use more future-tense verbs and fewer past-tense verbs, suggesting their focus shifts over the lifespan.

6. Culture

- Counterintuitively, translated Japanese texts used "I" *more* than American texts. The authors suggest this may be because collectivist values (like harmony and self-criticism) require a high degree of self-focus.
- American texts used more articles ("a," "the"), which supports the idea that Western thought is more categorical.

**Conclusion: Words as Reflections**

The way we use function words is a **reflection** of our underlying psychological state, not a cause of it25. The authors tried to change people's feelings by forcing them to use different pronouns in experiments, but it didn't work26. This means you can't just say "we" more to feel more connected; rather, feeling connected causes you to naturally say "we" more.

In short, these tiny "junk words" are a window into our minds, revealing everything from our emotional state and social status to how honest we're being.