Lecture 3: N-Gram Language Models (I)

Instructor: Stephanie Schoch

CSCI 680: Natural Language Processing
Department of Computer Science, College of William & Mary
August 31, 2026

WILLIAM & MARY

Many slides are adapted from Jurafsky & Martin (3rd ed.)

Lecture Outline

- Project Discussion
- N-Gram Language Models

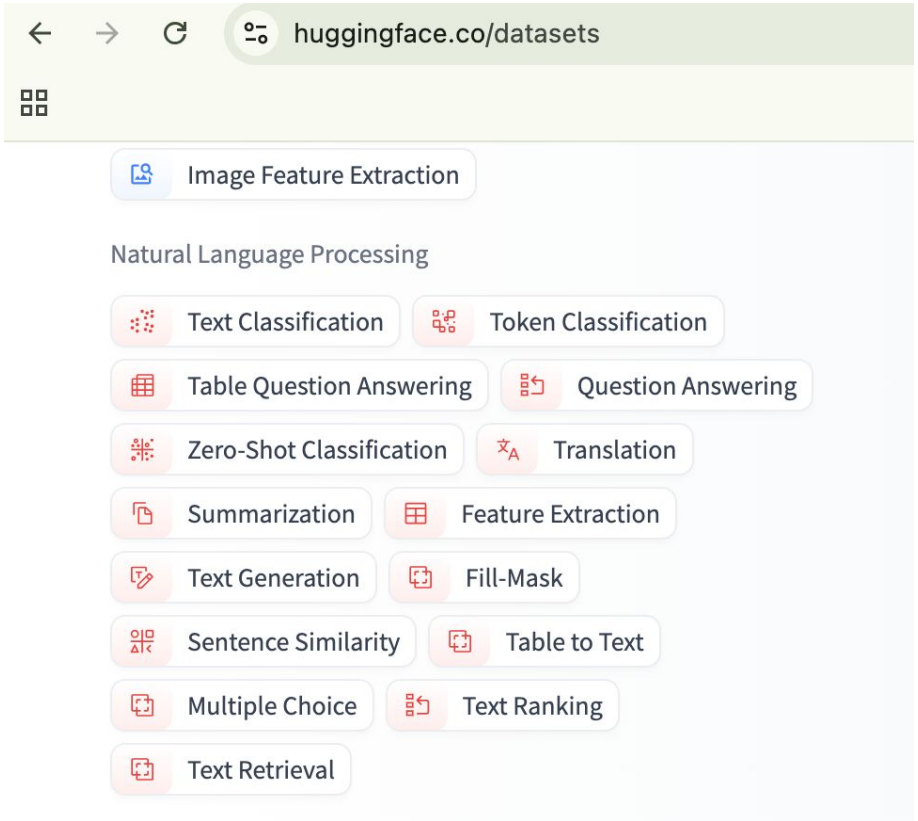
Project: Considerations

- Clearly define the research problem
- Constrain the scope so it is achievable in a semester
- Identify what data is available for your type of task
- Consider compute resources

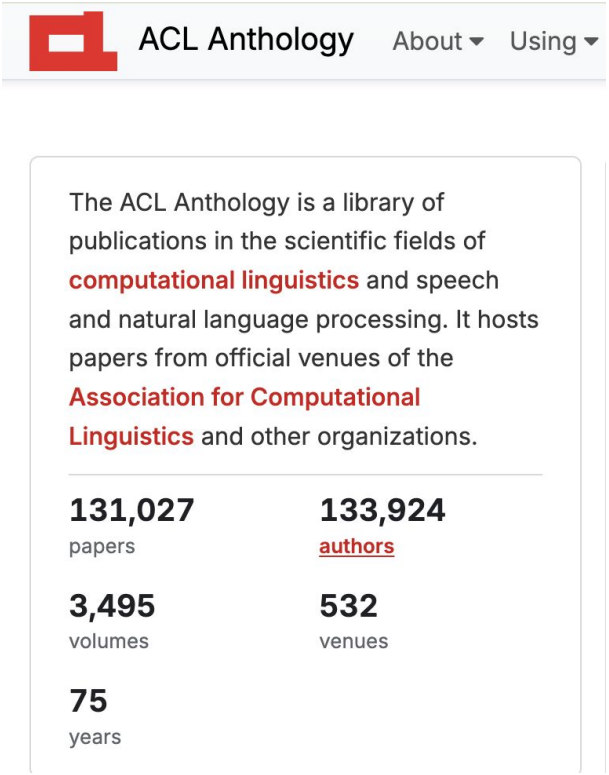
Project: Where To Get Started?

- Start from a topic you are interested in!
  - Software engineering
  - Linguistics
  - Educational applications
  - …
- **Recommended:** Focus on NLP tasks or applications
- Come to office hours if you'd like to chat about your interests or ideas

Project: Where To Get Started?



Project: Where To Get Started?



https://aclanthology.org/

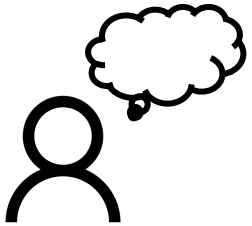
Flagship venues · 9				
ACL Annual Meeting of the Association for Computational Linguistics	2026	2025	2024	2023
AACL Asian Chapter of the Association for Computational Linguistics	2025	2023	2022	2020
CL *Computational Linguistics*	2026	2025	2024	2023
COLING International Conference on Computational Linguistics	2025	2024	2022	2020
EACL European Chapter of the Association for Computational Linguistics	2026	2024	2023	2021
EMNLP Conference on Empirical Methods in Natural Language Processing	2025	2024	2023	2022
Findings Findings of the Association for Computational Linguistics	2026	2025	2024	2023
NAACL Nations of the Americas Chapter of the Association for Computatio...	2025	2024	2022	2021
TACL *Transactions of the Association for Computational Linguistics*	2026	2025	2024	2023

Project: Defining Scope

Project: Example

"I'm interested in LLMs for X"

I wonder
{how well they perform on different types of X tasks,
how well they perform when used to evaluate X tasks, …**}**



Search: "What is it called when LLMs are used to judge outputs?"

Idea: Analyzing LLM-as-Judge performance for LLM-generated X.

Project: A Few Ideas (From ACL Workshops)

Cards Against LLMs: Benchmarking Humor Alignment in Large Language Models

Yousra Fettach, Guillaume Bied, Hannu Toivonen, Tijl De Bie

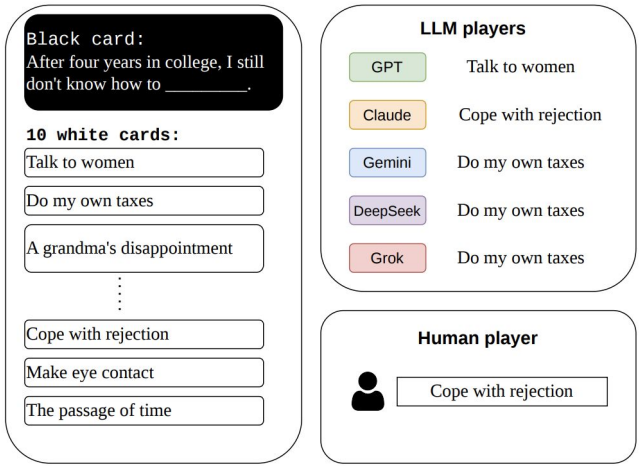


Figure 1: Framework overview. Given a black card prompt and a slate of 10 white card candidates, five frontier LLMs and a human player independently select the card they deem funniest.

Abstract

Humor is one of the most culturally embedded and socially significant dimensions of human communication, yet it remains largely unexplored as a dimension of Large Language Model (LLM) alignment. In this study, five frontier language models play the same *Cards Against Humanity* games (CAH) as human players. The models select the funniest response from a slate of ten candidate cards across 9,894 rounds. While all models exceed the random baseline, alignment with human preference remains modest. More striking is that models agree with each other substantially more often than they agree with humans. We show that this preference is partly explained by systematic position biases and content preferences, raising the question whether LLM humor judgment reflects genuine preference or structural artifacts of inference and alignment.

Project: A Few Ideas (From ACL Workshops)

Temporal Text Classification with Large Language Models

Nishat Raihan, Marcos Zampieri

Abstract

Languages change over time. Computational models can be trained to recognize such changes enabling them to estimate the publication date of texts. Despite recent advancements in Large Language Models (LLMs), their performance on automatic dating of texts, also known as Temporal Text Classification (TTC), has not been explored. This study provides the first systematic evaluation of leading proprietary (Claude 3.5, GPT-4o, Gemini 1.5) and open-source (LLaMA 3.2, Gemma 2, Mistral, Nemotron 4) LLMs on TTC using three historical corpora, two in English and one in Portuguese. We test zero-shot and few-shot prompting, and fine-tuning settings. Our results indicate that proprietary models perform well, especially with few-shot prompting. They also indicate that fine-tuning substantially improves open-source models but that they still fail to match the performance delivered by proprietary LLMs.

Anthology ID: 2026.nlp4dh-1.10

Volume: Proceedings of the 6th International Conference on Natural Language Processing for the Digital Humanities

Project: A Few Ideas (From ACL Workshops)

Data Contamination in Neural Hieroglyphic Translation: A Reproducibility Study

Ammar Toutou, Abdelrahman Harb, Christine Basta

Abstract

Ancient and endangered languages pose a unique challenge for NLP: their datasets are inherently scarce, difficult to expand, and built from formulaic corpora—making data-quality issues especially consequential yet rarely audited. Motivated by the need to understand what current NMT can realistically achieve for such languages, we investigate hieroglyphic-to-German translation, where a recent study reported 61.5 BLEU using fine-tuned M2M-100. Our reproduction yields only 37.0 BLEU with the released model. Investigating this gap, we find **32% of test targets appear identically in training** (16/50; 50% under 8-gram overlap at 70% threshold). This contamination inflates scores dramatically: contaminated samples achieve up to 83.8 BLEU / 0.924 COMET-22 versus 30.9–39.2 BLEU / 0.622–0.676 COMET-22 on clean samples across five model configurations spanning two architectures. Document-level decontamination reduces contaminated BLEU by only 4.6 points because 8/16 targets persist via other source documents—target-level deduplication is required. We release a decontaminated 34-sample test set and establish corrected baselines (30.9–39.2 BLEU), providing a realistic assessment of NMT capability for this endangered writing system.

PDF

Cite

Search

Fix data

Anthology ID: 2026.nlp4dh-1.6

Volume: Proceedings of the 6th International Conference on Natural Language Processing for the Digital Humanities

Lecture Outline

- Project Discussion
- **N-Gram Language Models**

Probabilistic Language Modeling

Goal: compute the probability of a sentence or sequence of words:

$$P(\mathbf{w}) = P(w_1, w_2, w_3, w_4, w_5, \ldots, w_n)$$

Related Task: probability of an upcoming word:

$$P(w_5 \mid w_1, w_2, w_3, w_4) \quad \text{or} \quad P(w_n \mid w_1, w_2, w_3, w_4, \ldots, w_{n-1})$$

A **language model** assigns probabilities to sequences of words

by computing either $P(\mathbf{w})$ or $P(w_n \mid w_1, w_2, w_3, w_4, \ldots, w_{n-1})$

Probabilistic Language Modeling

Spelling and grammar correction:

$$P(\text{There are two midterms}) > P(\text{Their are two midterms})$$

$$P(\text{Everything has improved}) > P(\text{Everything has improve})$$

Speech recognition systems:

$$P(\text{I will be back soonish}) > P(\text{I will be bassoon dish})$$

How to Compute $P(\mathbf{w})$?

"The water of Walden Pond is so beautifully blue"

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

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

How to compute this joint probability?

$$P(\mathbf{w}) = P(w_1, w_2, w_3, w_4, w_5, \ldots, w_n)$$

Intuition: Let's rely on the **Chain Rule of Probability**

The Chain Rule: Compute Joint Probability of Words in Sentence

$$P(w_{1:n}) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_{1:2}) \cdots P(w_n \mid w_{1:n-1})$$

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

$$P(\text{The water of Walden Pond}) =$$
$$P(\text{The}) \times$$
$$P(\text{water} \mid \text{The}) \times$$
$$P(\text{of} \mid \text{The water}) \times$$
$$P(\text{Walden} \mid \text{The water of}) \times$$
$$P(\text{Pond} \mid \text{The water of Walden})$$

How to Compute These Probabilities?

Count and divide?

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

No: Too many possible sentences!
We'll never see enough data to estimate these.

Markov Assumption

Simplifying Assumption:

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

$$\approx \quad P(\texttt{blue}|\texttt{beautifully})$$

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

Andrei Markov

Bigram Markov Assumption

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

instead of

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

We approximate each component of this product.

General N-Gram Model For Each Component

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

Simple Case: Unigram Model

$$P(w_1, w_2, \ldots, w_n) \approx \prod_i P(w_i)$$

(the probability of a word only depends on itself)

Some automatically generated sentences from a Shakespeare unigram model:

```
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
```

Bigram Model

$$P(w_i \mid w_1, w_2, \ldots, w_{i-1}) \approx P(w_i \mid w_{i-1})$$

(the probability of a word is conditioned on the previous word)

Some automatically generated sentences from a Shakespeare bigram model:

```
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.
```

Trigram Model

$$P(w_i \mid w_1, w_2, \ldots, w_{i-1}) \approx P(w_i \mid w_{i-2}, w_{i-1})$$

(the probability of a word is conditioned on the previous two words)

Some automatically generated sentences from a Shakespeare trigram model:

```
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
```

Problems with N-Gram Language Models

Can extend this to 4-grams, 5-grams… **But:**

1) N-grams can’t model long-range dependencies in language

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

2) N-grams don't do well at modeling new sequences (even if they have similar meanings to sequences they've seen)

I **love** this **movie** ≈ I **adore** this **film**

$$P(\text{love} \mid \text{I}) \quad P(\text{movie} \mid \text{this})$$
$$P(\text{adore} \mid \text{I}) \quad P(\text{film} \mid \text{this})$$

Estimating Bigram Probabilities

The Maximum Likelihood Estimate

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

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

Simple Example

Corpus

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

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

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

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

Larger Example: Berkeley Restaurant Project (BRP) Sentences

Example Sentences

- can you tell me about any good cantonese restaurants close by
- mid priced thai food is what i'm looking for
- tell me about chez panisse
- can you give me a listing of the kinds of food that are available
- i'm looking for a good place to eat breakfast
- when is caffe venezia open during the day

Total: 9222 sentences

Raw Bigram Counts

Out of 9222 sentences

Next Word

History

	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

Bigram Probabilities

Normalize by Unigrams:

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

Results:

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

w_i

w_{i-1}	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

What Kinds of Knowledge Are Represented?

$$P(\text{to} \mid \text{want}) = .66$$
$$P(\text{want} \mid \text{spend}) = 0$$
$$P(\text{of} \mid \text{to}) = 0$$

$$P(\text{dinner} \mid \text{lunch or}) = .83$$
$$P(\text{dinner} \mid \text{for}) \sim P(\text{lunch} \mid \text{for})$$

$$P(\text{chinese} \mid \text{want}) > P(\text{english} \mid \text{want})$$

Knowledge of Grammar

("want" is followed by infinitive "to")
("of" is not a verb)

Knowledge of Meaning

The words "dinner" and "lunch" are semantically related.

Knowledge about the World

Chinese food is popular.

Bigram Estimates of Sentence Probabilities

$$
\begin{aligned}
P(&\texttt{<s>}\ \text{I want english food}\ \texttt{</s>}) \\
&= P(\text{I} \mid \texttt{<s>}) \\
&\quad \times P(\text{want} \mid \text{I}) \\
&\quad \times P(\text{english} \mid \text{want}) \\
&\quad \times P(\text{food} \mid \text{english}) \\
&\quad \times P(\texttt{</s>} \mid \text{food}) \\
&= .000031
\end{aligned}
$$

Low…

Dealing with Scale in Large N-Grams: Underflow

Probabilities are stored and computed in log format, i.e. **log probabilities**

- Avoids underflow from multiplying many small numbers

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

- If we need probabilities, we can do on exp at the end

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

TODOs:

- Continue brainstorming for your project.

Next Lecture: **N-Gram Language Models (II)**