ACADEMIC YEAR 2026/27

Text Mining & Search [DS] / Information Retrieval [CS]

Language Modeling

TEXT MINING & SEARCH

Prof. Marco Viviani

University of Milano-Bicocca

Department of Informatics, Systems and Communication

INFORMATION RETRIEVAL

Dr. Georgios Peikos

University of Milano-Bicocca

Department of Informatics, Systems and Communication

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Ikr3 Lab — information and knowledge representation, retrieval, and reasoning LAB

In the previous episodes …

In the previous episodes – Text Mining and Search

- We work with collections of natural language documents.
- Text Mining aims to discover useful information or patterns from text.
- Search aims to identify relevant documents (information items/pieces of information) for an information need.
- A common workflow requires us to process and represent text before applying computational methods.

In the previous episodes – Text Processing

- Raw text must be transformed into units that computational systems can use.
  - The unit of analysis can be a character, token, sentence, chunk, or document.
- Tokenization defines the units that subsequent methods operate on.
- Normalization, stemming, and lemmatization can reduce variation, but they can also remove useful information.
- Processing choices depend on the language, task, context, and model.

In the previous episodes – Text Representation

- Text must be expressed in a computational form, such as vectors.
- We introduced sparse lexical representations, where vector dimensions correspond to vocabulary terms.
- We represented documents using binary values, term counts, TF, and TF-IDF.
- TF-IDF combines information about term frequency within a document and term occurrence across the document collection.

Recap – Following One Document Through Processing

Document collection. We use the same three documents throughout the example.

D1 The news team wins a match against another team.
D2 The news team wins again.
D3 The news software has a software update.

Processing. Lowercase → remove punctuation → tokenize → remove stopwords.
For this example, the stopword set is {the, a, against, another, again, has}.

Document	Retained tokens
D1	news team wins match team
D2	news team wins
D3	news software software update

Recap – Building the Vocabulary and Representing D1

Vocabulary. Collect every distinct retained term across D1, D2, and D3.

V = [match, news, software, team, update, wins]

D1 news team wins match team

Each vector uses the same vocabulary order.

Binary rule. If a vocabulary term appears in a document, then the document's vector has 1 in that position, otherwise 0.

Representation of D1	match	news	software	team	update	wins
Binary	Is "match" in D1? 1	Is "news" in D1? 1	Is "software" in D1? 0	1	0	1
Count	How many times "match" is in D1? 1	1	0	2	0	1

Counts preserve repetition. Binary values preserve presence.

Recap – Calculating Normalized Term Frequency

Counts in D1. [1, 1, 0, 2, 0, 1], using the shared vocabulary order.

Maximum count. $\max_{t' \in V} c(t', D1) = c(\text{team}, D1) = 2$.

$TF(t, D1) = c(t, D1) / \max_{t' \in V} c(t', D1)$; we divide by the frequency of the most frequent term in D1.

Term	Count	Normalized TF calculation
match	1	1 / 2 = 0.500
news	1	1 / 2 = 0.500
software	0	0 / 2 = 0.000
team	2	2 / 2 = 1.000
update	0	0 / 2 = 0.000
wins	1	1 / 2 = 0.500

TF vector. [0.500, 0.500, 0, 1.000, 0, 0.500]

Recap – Calculating Inverse Document Frequency

Collection size. N = 3 documents.

Counting rule. For each term we count in how many documents it exists. We record only 1 presences per document. For example, if a terms appears 10 times in D1, 0 in D2, and 0 in D3, its DF = 1.

$IDF(t) = \log_{10}(N / df(t))$

Term	Documents containing the term	DF	IDF calculation
match	D1	1	$\log_{10}(3 / 1) = 0.477$
news	D1, D2, D3	3	$\log_{10}(3 / 3) = 0.000$
software	D3	1	$\log_{10}(3 / 1) = 0.477$
team	D1, D2	2	$\log_{10}(3 / 2) = 0.176$
update	D3	1	$\log_{10}(3 / 1) = 0.477$
wins	D1, D2	2	$\log_{10}(3 / 2) = 0.176$

News occurs in all three documents, so $IDF(news) = \log_{10}(3 / 3) = 0$.

Recap – Calculating the TF-IDF Vector

$\text{TF-IDF}(t, D1) = \text{TF}(t, D1) \times \text{IDF}(t)$

Use full precision during calculation. Round final weights to three decimals.

Term	TF-IDF calculation	Final weight
match	$(1 / 2) \times \log_{10}(3 / 1)$	0.239
news	$(1 / 2) \times \log_{10}(3 / 3)$	0.000
software	$(0 / 2) \times \log_{10}(3 / 1)$	0.000
team	$(2 / 2) \times \log_{10}(3 / 2)$	0.176
update	$(0 / 2) \times \log_{10}(3 / 1)$	0.000
wins	$(1 / 2) \times \log_{10}(3 / 2)$	0.088

D1 The news team wins a match against another team.

D1 = [0.239, 0, 0, 0.176, 0, 0.088]

Questions?

Comments?

What information is in our Vector?

Bag of Words – Same Terms, Different Sequences

D1 The news team wins a match against another team.

D1′ Team another against match a wins team news the.

Document	After the same preprocessing
D1	news team wins match team
D1′	team match wins team news

Same vocabulary. Same term counts. Same binary and TF-IDF vectors.

Are these two sequences equally plausible English sentences?

Bag of Words – A New Question About Text

Both documents receive the same TF-IDF vector in our original collection:

[0.239, 0, 0, 0.176, 0, 0.088]

What is missing?

Our unigram BoW vectors do not preserve token order or the relationships expressed by that order.

A new question

How likely is a sequence, given patterns observed in language?

Sequence Probability Model

The news team wins a ... | **match** | against | the

Language Modeling

Introduction

Language Modeling – What Is a Language Model?

01. SEQUENCE

Sequence probability

A language model assigns probabilities to sequences of tokens.

02. PREDICTION

Next-token probability

It can estimate the probability of an upcoming token given the preceding tokens.

03. SAMPLING

Generation

These probabilities can be used to choose successive tokens and generate text.

PROBABILITY EVALUATION EXAMPLE

P(“Today is Wednesday”)
High Probability (Natural phrase)

VS

P(“Today Wednesday is”)
Low Probability (Unnatural word order)

Language Modeling – What Is a Language Model?

01. SEQUENCE

Sequence probability

A language model assigns probabilities to sequences of tokens.

02. PREDICTION

Next-token probability

It can estimate the probability of an upcoming token given the preceding tokens.

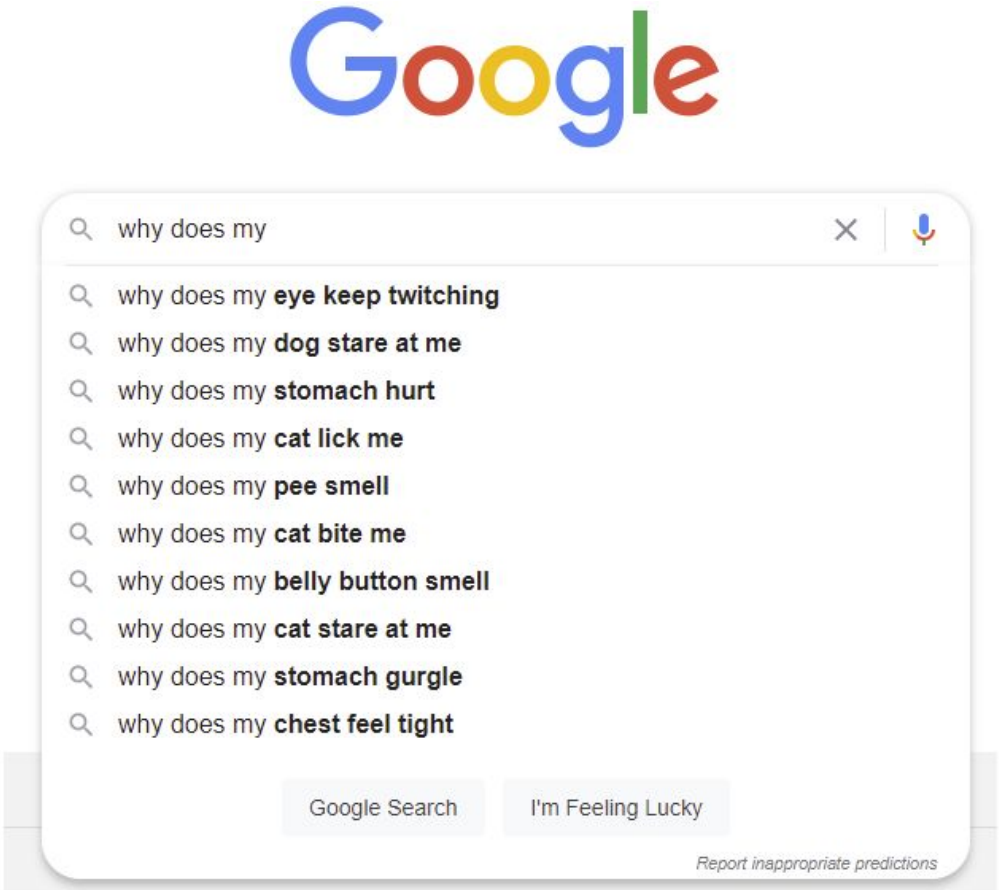
03. SAMPLING

Generation

These probabilities can be used to choose successive tokens and generate text.

Sequence Probability Model

The news team wins a ... | **match** | against | the


Google
why does my
why does my eye keep twitching
why does my dog stare at me
why does my stomach hurt
why does my cat lick me
why does my pee smell
why does my cat bite me
why does my belly button smell
why does my cat stare at me
why does my stomach gurgle
why does my chest feel tight
Google Search
I'm Feeling Lucky
Report inappropriate predictions

Language Modeling – A definition

Language modeling is the task of assigning a probability to sentences in a language. […] Besides assigning a probability to each sequence of words, the language models also assign a probability for the likelihood of a given word (or a sequence of words) to follow a sequence of words.

Source: Page 105, Neural Network Methods in Natural Language Processing, 2017.

Language Modeling – Why Do We Need It?

Application	Candidate sequences	Language evidence
Machine translation	high winds tonight / large winds tonight	Prefer a natural word combination.
Spelling correction	about fifteen minutes from / about fifteen minuets from	Use surrounding words to resolve an error.
Speech recognition	I saw a van / eyes awe of an	Help score alternative transcriptions.

Researchers needed probabilities to compare possible linguistic outputs.

Language Modeling – Why Is It Useful for Mining and Search?

01

Text Categorization

A news article contains “baseball” three times and “game” once. Which topic model better explains these words, sport or politics?

02

Information Retrieval

Given a document about sport, how likely is a query under that document’s language model?

03

Other Applications

Language models also contribute to summarization, question answering, and text generation.

We use statistical evidence from text to predict, compare, and make decisions.

Questions?

Comments?

Language Modeling

How do we estimate text probabilities?

Language Modeling – Notation and the Prediction Question

Notation	Meaning
$w_1, \ldots, w_n$	An ordered sequence of n tokens.
$P(w_1, \ldots, w_n)$	Probability of the token sequence.
$P(w_n \mid w_1, \ldots, w_{n-1})$	Probability of the next token, given its preceding context.
c(h, w)	Number of times context h is immediately followed by w.

Example P(the | its water is so transparent that)

How to compute $P(w_n|w_1,w_2 \ldots w_{n-1})$?

- Let us start by computing $P(w_n|w_1,w_2 \ldots w_{n-1})$, the probability of a word w_n given a sequence of words.
- For example: P(the|its water is so transparent that)
- Relative frequency counts: given a very large corpus, count the number of times we see *its water is so transparent that*, and count the number of times it is followed by *the*.

$$P(\text{the}|\text{its water is so transparent that}) = \frac{C(\text{its water is so transparent that the})}{C(\text{its water is so transparent that})}$$

Too many possible sentences!

How to compute P(W)?

- We want to estimate P(W), the probability of an entire word sequence (i.e. sentence) W.
  - For example, W is *"its water is so transparent"*.
  - This sentence contains five words.
  - P(W) is their joint probability in this order.

- We could estimate P(W) by counting occurrences in a corpus.
  - Count how often "its water is so transparent" occurs.
  - Divide this count by the total number of observed five-word sequences.
  - This gives the proportion of observed five-word sequences that match W exactly.

- That seems rather a lot to estimate!!!

- We want to estimate a sentence probability without requiring many repeated observations of that exact sentence.
  - On the next slide, we introduce the probability rule that makes this decomposition possible.

How to compute P(W) practically

PROBLEM / EXAMPLE

For example, how to compute this joint probability of a token sequence:

P(its, water, is, so, transparent, that)

SOLUTION CONCEPT

Intuition: let’s rely on the Chain Rule of Probability

Reminder: The Chain Rule

RECALL: CONDITIONAL PROBABILITY

$p(B|A) = P(A,B) / P(A)$ ➔ Rewriting: $P(A,B) = P(A)\,P(B|A)$

MORE VARIABLES

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

THE CHAIN RULE IN GENERAL

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

Chain Rule for Sentence Probability

GENERAL FORMULA

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_1 w_2 \ldots w_{i-1})$$

EXAMPLE APPLICATION

P(“its water is so transparent”) =

P(its) × P(water|its) × P(is|its water) × P(so|its water is) × P(transparent|its water is so)

TAKEAWAY & LIMITATION

We can compute joint probability by multiplying conditional probabilities, but calculating full histories is impractical. Next: we approximate the “history”!

Questions?

Comments?

Markov Assumption

BIGRAM ASSUMPTION (1 PRECEDING WORD)

$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{that})$

TRIGRAM ASSUMPTION (2 PRECEDING WORDS)

$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{transparent that})$

KEY CONCEPT

Rather than looking at the entire history, we approximate history using only a fixed window of preceding words.

- A bigram uses 1 preceding word. A trigram uses 2 preceding words.



Andrei Markov

(1856–1922)

Russian mathematician known for his work on stochastic processes.

Approximating the History

Approximate the full context using only the most recent tokens.

History used	Conditional probability
Full history	P(the | its water is so transparent that)
Previous 1 token	P(the | that)
Previous 2 tokens	P(the | transparent that)

This is a Markov assumption about the context needed for prediction.

N-gram Models – A 4-gram Worked Example

as the proctor started the clock, the students opened their _____

A 4-gram model uses only the preceding three tokens students opened their.

Observed sequence	Corpus Sequence Frequency Count
students opened their	1000
students opened their books	400
students opened their exams	100

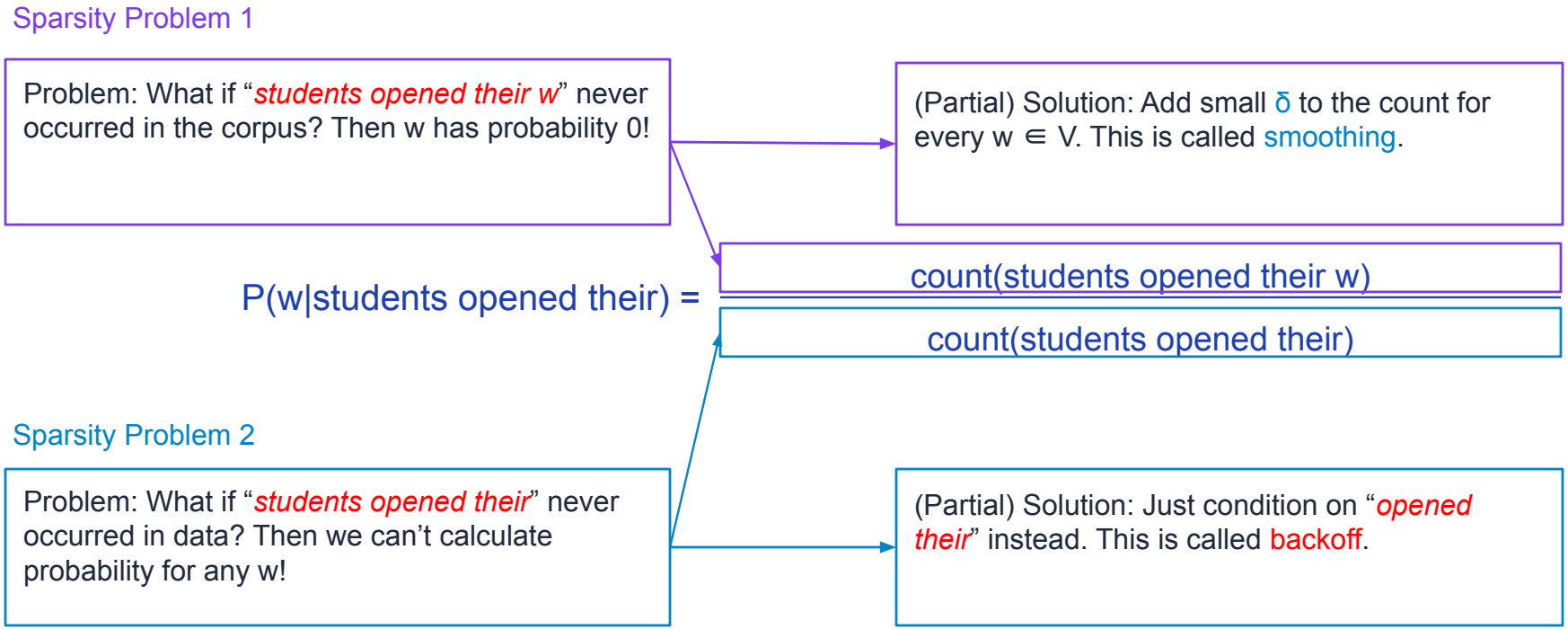
P(books | students opened their) = 400 / 1000 = 0.4

P(exams | students opened their) = 100 / 1000 = 0.1

Questions?

Comments?

Sparsity Problems with n-gram Language Models



Note: Increasing n makes sparsity problems worse. Typically we can't have n bigger than 5.

Recap: language models

How do we build probabilities over sequence of terms?

$$P(t_1,t_2,t_3,t_4) = P(t_1) \times P(t_2|t_1) \times P(t_3|t_1,t_2) \times P(t_4|t_1,t_2,t_3)$$

Unigram language model –simplest ; no conditioning context

$$P(t_1,t_2,t_3,t_4) = P(t_1) \times P(t_2) \times P(t_3) \times P(t_4)$$

Bigram language model – condition on previous term

$$P(t_1,t_2,t_3,t_4) = P(t_1) \times P(t_2|t_1) \times P(t_3|t_2) \times P(t_4|t_3)$$

Questions?

Comments?

N-gram Models – The Original Training Corpus

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

Context	Observed continuations
<s>	I × 2; Sam × 1
I	am × 2; do × 1
am	Sam × 1; </s> × 1
Sam	</s> × 1; I × 1

<s> marks the start. </s> marks the end. Never count across these boundaries.

Exercise – Estimate the Bigram Probabilities

Use the same corpus. Count the context and its continuations.

$$\underbrace{\widehat{P}(w_i \mid w_{i-1})}_{\text{Probability of the next word given the previous word}} = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})} = \frac{\text{Number of times the previous word is followed by the next word}}{\text{Total number of observed continuations of the previous word}}$$

Probability	Your calculation
P(I | <s>)	?
P(Sam | <s>)	?
P(am | I)	?
P(do | I)	?
P(Sam | am)	?
P(</s> | Sam)	?

Context	Observed continuations
<s>	I × 2; Sam × 1
I	am × 2; do × 1
am	Sam × 1; </s> × 1
Sam	</s> × 1; I × 1

Then calculate the complete-sentence probability of “I am Sam”.

Exercise – Bigram Estimates and Sentence Probability

Probability	Relative frequency
P(I | <s>)	2/3 ≈ 0.667
P(Sam | <s>)	1/3 ≈ 0.333
P(am | I)	2/3 ≈ 0.667
P(do | I)	1/3 ≈ 0.333
P(Sam | am)	1/2 = 0.5
P(</s> | Sam)	1/2 = 0.5

P(“I am Sam” and </s>) = (2/3) × (2/3) × (1/2) × (1/2) = 1/9

Questions?

Comments?

Language Modeling – Main Takeaways

Why researchers developed it

Systems needed statistical evidence to compare and predict linguistic sequences.

How it works

Estimate token probabilities, condition on context, and combine them through the chain rule.

What we must handle

Limited data, missing counts, and evaluation on unseen text.

Why it remains relevant

Token prediction underlies generation and the learning of useful text representations.

A likely sequence is not necessarily a factually correct or useful answer.

Questions?

Comments?

Contacts

Dr. Georgios Peikos

georgios.peikos@unimib.it

Academic Year 2026/27

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Ikr3 Lab information and knowledge representation, retrieval, and reasoning LAB