ACADEMIC YEAR 2026/27

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

Text Representation

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

Text Representation

Introduction

Text Representation – What is Text Representation?

- Definition. Text representation expresses natural language text in a computational form that a model can process, compare, or learn from.
  - Computational forms mainly means numerical structures such as vectors, matrices, or sequences of vectors.
- Represented units. We can represent individual characters, tokens, words, sentences, passages, or entire documents. Which textual unit we will represent, depends on the task and the model we will use to perform it.
  - For instance, language generation tasks require us to work on token level representations.
  - Doing sentiment analysis requires us to work on sentence or passage level.

Text Representation Techniques – How can we represent texts?

- Sparse lexical representations. Represent text through term presence or frequency, using binary vectors, word counts, or TF-IDF weights. [Today's Lesson]
- Latent representations. Capture patterns in term co-occurrence through methods such as Latent Semantic Analysis and topic models. [Next Lessons in PART I & Additional Readings]
- Static word embeddings. Assign each word a fixed dense vector across contexts, using methods such as Word2Vec, GloVe, and fastText. [Next Lessons in PART I]
- Contextual embeddings. Produce token vectors that depend on the surrounding text, using models such as ELMo and BERT. [Next Lessons in PART I]
- Sentence and document embeddings*. Represent an entire sentence, passage, or document with a vector for tasks such as clustering, and retrieval. [Part II of the Courses]

Text Representation Techniques – Why different techniques?

- Sparse (Lexical) representations. *"The dog chased the cat."* and *"The cat chased the dog."*
  - as we will soon see, these sentences will produce identical sparse vectors because they contain the same terms with the same frequencies.
- Latent representations. *"Car engine repair" and "automobile motor maintenance"*
  - share no common words but describe related topics. Latent methods use co-occurrence patterns across a collection to represent these relationships.
- Static word embeddings. *"She deposited money in the bank."* and *"She sat beside the river bank."*
  - bank receives the same word vector despite its different meanings.
- Contextual embeddings. *"She deposited money in the bank."* and *"She sat beside the river bank."*
  - bank receives different token vectors because its representation depends on the surrounding text.



Questions?

Comments?

Text Representation

Sparse Lexical Vector Representations and Bag of Words

Sparse and Lexical Representations – Our Example Document Collection

- Document Collection. We will use short texts. D1–D3 are about Sport and D4–D6 about Technology.
  - The documents contain terms. Unique terms can be used to create our vocabulary.

Doc	Original text
D1	News: the team celebrates a goal, another goal, and a match win.
D2	News: the team prepares for a match; the team trains.
D3	News: the team announces a goal update.
D4	News: software analyzes data and produces a model from data.
D5	News: software processes data with new software.
D6	News: the model receives a data update.

Sparse and Lexical Representations– Building a Vocabulary

- Text preprocessing. Lowercase the text and remove punctuation, stemming, lemmatization, among other.
  - Yes, when we work with sparse representations, we usually do some of the above transformations.
- Vocabulary V. Collect all the distinct terms appearing in our documents. Repeated occurrences contribute only one vocabulary entry. For this exercise/lecture, retain only eight selected terms from the collection.
- Vector dimensions. Assign one position to each term. Our vocabulary contains eight terms |V|=8, so each document vector has eight dimensions.
  - We call these representations lexical because their dimensions correspond to terms or other lexical units (tokens).
- Shared order. Use the same term order for every document so corresponding positions describe the same feature.

Building a Shared Vocabulary – The processed collection

Doc	Retained tokens	Length
D1	news team goal goal match	5
D2	news team team match	4
D3	news team goal update	4
D4	news software data data model	5
D5	news software software data	4
D6	news model data update	4

Binary Representation – Does the Term Occur?

Doc	Retained tokens
D1	news team goal goal match
D2	news team team match
D3	news team goal update
D4	news software data data model
D5	news software software data
D6	news model data update

- Each row represents a document. Each column represents the same vocabulary term.
- Our Vocabulary contains the terms: news team match goal software data model update.
- Notice the number of zeros per document vector. This is why we call them sparse representations.

D/V	news	team	match	goal	software	data	model	update
D1	1	1	1	1	0	0	0	0
D2	1	1	1	0	0	0	0	0
D3	1	1	0	1	0	0	0	1
D4	1	0	0	0	1	1	1	0
D5	1	0	0	0	1	1	0	0
D6	1	0	0	0	0	1	1	1

Binary Representation – Limitations

- Term frequency. A term occurring once receives the same value as a term occurring many times.
- Term importance. All present terms receive 1, regardless of their usefulness for distinguishing document content.
- Word order. The representation ignores order and grammatical relationships, so sentences with different meanings can have identical vectors. This is why we call them Bag of Words (BoW) representations.
  - Binary term vectors ignore word order and grammatical relationships.
    - For example, *"The cat chased the dog."* and *"The dog chased the cat."* produce identical vectors, although they describe different animals performing the action.

We can do better in preserving more information in our representations!
To preserve repetition, we can replace binary presence values with term counts.

Count-based (Frequency) Representation – How many times a Term Occurs?

Doc	Retained tokens
D1	news team goal goal match
D2	news team team match
D3	news team goal update
D4	news software data data model
D5	news software software data
D6	news model data update

- Each row represents a document. Each column represents the same vocabulary term.
- Our Vocabulary contains the terms: news team match goal software data model update.
- Notice the number of zeros per document vector.
- Counts describe within-document occurrence.

D/V	news	team	match	goal	software	data	model	update
D1	1	1	1	2	0	0	0	0
D2	1	2	1	0	0	0	0	0
D3	1	1	0	1	0	0	0	1
D4	1	0	0	0	1	2	1	0
D5	1	0	0	0	2	1	0	0
D6	1	0	0	0	0	1	1	1

Term Frequency – Alternative Definitions

Let us define **c(t,d)** as the total appearances of **term t in document d**, and **L(d)** as the number of **tokens** after preprocessing, i.e. a document's length.

1. Raw Counts (what we saw)

Measure how often a term occurs. **Longer documents can accumulate larger counts** simply because they contain more text.

$$\mathrm{tf}_{\mathrm{raw_counts}}(t,d) = c(t,d)$$

2. Length-Normalized TF

Measures the **term's share of the document**. This allows us to **compare relative occurrence** across documents of different lengths.

$$\mathrm{tf}_{\mathrm{Length_norm}}(t,d) = \frac{c(t,d)}{L(d)}$$

3. Logarithmic TF

Gives repeated terms more weight, but with **progressively smaller increases**. Four occurrences therefore receive less than twice the weight of two occurrences.

$$\mathrm{tf}_{\log}(t,d) = \begin{cases} 1 + \log_{10}(c(t,d)), & c(t,d) > 0 \\ 0, & c(t,d) = 0 \end{cases}$$

Term Frequency – Alternative Definitions

Let us define **c(t,d)** as the total appearances of **term t in document d**. Let t' be the term **t'** of document **d** that has the highest frequency.

4. Max Term Frequency Normalization

Normalizes raw count by the maximum term frequency in the document. **Introduced to prevent bias toward long documents** and dampen the dominance of highly repetitive terms.

$$\mathrm{tf}_{\max}(t,d) = \frac{c(t,d)}{\max_{t' \in d} c(t',d)}$$

- Reduces the effect of document length because it does not use raw counts directly.
- Limits the dominance of very frequent terms by scaling all counts to the document maximum.
- Makes term frequencies comparable within a document.
- Preserves relative importance because terms that occur more often still receive larger weights.

Exercise – Vocabulary and Document Representations

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

- Build the vocabulary of distinct terms across all three documents. Use alphabetical order.
- Construct a count vector for each document.
- Calculate the length-normalized TF vector for each document. Round to three decimal places.

Raw counts / Raw TF

D/V	match	news	software	team	update	wins
D1	1	1	0	2	0	1
D2	0	1	0	1	0	1
D3	0	1	2	0	1	0

Exercise – Vocabulary and Document Representations

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

- Build the vocabulary of distinct terms across all three documents. Use alphabetical order.
- Construct a count vector for each document.
- Calculate the length-normalized TF vector for each document. Round to three decimal places.

Length-normalized TF

D/V	match	news	software	team	update	wins
D1	0.200	0.200	0	0.400	0	0.200
D2	0	0.333	0	0.333	0	0.333
D3	0	0.250	0.500	0	0.250	0

Exercise – Vocabulary and Document Representations

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

- Build the vocabulary of distinct terms across all three documents. Use alphabetical order.
- Construct a count vector for each document.
- Calculate the length-normalized TF vector for each document. Round to three decimal places.

Logarithmic TF

D/V	match	news	software	team	update	wins
D1	1	1	0	1.301	0	1
D2	0	1	0	1	0	1
D3	0	1	1.301	0	1	0



Questions?

Comments?

Document Frequency – Why do we need it?

- Limitation of TF. Term frequency describes repetition within a document. It does not tell us whether the same term also appears in many other documents.
- Common terms. News appears in all six documents, including both sport and technology texts. Its presence does not help us distinguish these topics.
- Collection evidence. Counting document frequency helps us distinguish terms shared by many documents from terms concentrated in fewer documents.
- TF tells us how often a term occurs within a document. DF tells us how widely it occurs across the collection.

Document Frequency – What is it?

- Definition. Document frequency, DF, is the number of documents in a collection that contain a particular term.
- Counting rule. Each document contributes one to the term's DF if it contains that term. Repeated occurrences within the same document do not increase its DF.
- Difference from TF. Term frequency describes how often a term occurs within one document. Document frequency describes how widely that term appears across the collection.

Doc	Retained tokens
D1	news team goal goal match
D2	news team team match
D3	news team goal update

Term	Documents containing the term	DF
news	D1, D2, D3, D4, D5, D6	6
team	D1, D2, D3	2
goal	D1, D3	2

Document Frequency – Why Does Term Distribution in a Document Collection Matter?

- Our goal. We want a model to classify documents (i.e. text classification) as sport or technology using their term vectors.
  - ➢ Text Classification with ML approaches uses this kind of vector representations.
- Shared vocabulary. News appears in all six documents. Its presence alone does not help the model distinguish sport from technology.
- Topic evidence. Team appears in the three sport documents, while data appears in the three technology documents. Their distributions provide evidence for distinguishing the categories.

Term	Sport documents containing it	Technology documents containing it	DF
news	3	3	6
team	3	0	3
data	0	3	3

A new document contains news and team. Which term provides stronger evidence that it belongs to the sport category? Why?

Document Frequency – How is it calculated?

- DF measures commonness. A higher DF means that a term appears in more documents. Using DF directly as a weight would give larger weights to more widely shared terms.
- Desired weighting. Widely shared terms provide little evidence for distinguishing document content.
  - Terms appearing in fewer documents may provide more specific evidence and should therefore receive higher weights. Is it happening? Which words will have the highest document frequency?
- Inverse weighting. **IDF** decreases as document frequency increases. It gives lower weights to widely shared terms and higher weights to less common terms.

Inverse Document Frequency – How is it calculated?

Variables

Let *N* be total documents in collection, and *df(t)* documents containing term *t*.

Inverse Ratio

Dividing *N* by *DF* gives larger values to terms appearing in fewer documents.

Logarithmic Scaling

Compresses range of weights while preserving order. Less common terms get higher weights.

Terms Present Everywhere

If a term appears in every document (*DF* = *N*), its IDF is **zero** because it provides **no specificity**.

IDF Formula

$$\text{idf}(t) = \log_{10}\left(\frac{N}{\text{df}(t)}\right)$$

Higher IDF indicates **rare, highly informative** terms across the document collection.

Exercise – Document Frequency and Inverse Document Frequency

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

- Build the vocabulary of distinct terms across all three documents. Use alphabetical order.
- Calculate the DF and IDF for each term.

Term	Documents containing it	DF	N / DF	IDF
match	D1	1	3	0.477
news	D1, D2, D3	3	1	0
software	D3	1	3	0.477
team	D1, D2	2	1.5	0.176
update	D3	1	3	0.477
wins	D1, D2	2	1.5	0.176



Questions?

Comments?

Term Weighting – Can We Combine Both Types of Evidence?

- Within a document. Repetition may indicate that a term is central to the document's content. TF captures this evidence, but it does not consider how common the term is elsewhere.
- Across the collection. A term appearing in fewer documents may provide more specific evidence about their content. IDF captures this evidence, but it does not consider repetition within a document.
- Combined evidence. We want a weight that reflects both a term's frequency within the document and its specificity across the collection.

Term	Raw TF in D1	IDF	What do we observe?
news	1	0	Appears in every document.
team	2	0.176	Repeats in D1 and appears in two documents.
match	1	0.477	Appears once in D1 and nowhere else.

Term Weighting – How Did Researchers Bring These Ideas Together?

- Word significance, Luhn, 1958. Luhn used word frequency and distribution to estimate the significance of words and sentences for automatic summarization.
- Term specificity, Spärck Jones, 1972. Karen Spärck Jones introduced inverse document frequency as a measure of term specificity. Terms appearing in fewer documents could receive greater weight when retrieving documents.

Can a term's weight reflect both how often it occurs in a document and how specific it is within the collection?

Sparse Lexical BoW Representations

Term Weighting

TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF – Combining Term Frequency and Term Specificity

- Definition. TF-IDF assigns a weight to each term in a document by combining its term frequency with its inverse document frequency.
- Document evidence. TF measures how often the term occurs within the document. Repeated occurrences can indicate that the term is central to its content.
- Collection evidence. IDF reduces the weight of widely shared terms and gives higher weights to terms appearing in fewer documents.

$$w(t,d) = \mathrm{tf}(t,d) \times \mathrm{idf}(t)$$

$$\text{tf-idf}(t,d) = \underbrace{\frac{c(t,d)}{\max_{t' \in d} c(t',d)}}_{\text{Max-normalized TF}} \cdot \underbrace{\log_{10}\left(\frac{N}{\mathrm{df}(t)}\right)}_{\text{IDF}}$$

TF-IDF – How does the weight increase?

$$\text{tf-idf}(t,d) = \underbrace{\frac{c(t,d)}{\max_{t' \in d} c(t',d)}}_{\text{Max-normalized TF}} \cdot \underbrace{\log_{10}\left(\frac{N}{\text{df}(t)}\right)}_{\text{IDF}}$$

- Variations of tf-idf. Several variants exist, using different definitions of TF, IDF, and normalization. They share the idea of combining within-document frequency with collection specificity.
- Within-document frequency. The weight increases when the term's frequency increases relative to the most frequent term in the document, provided IDF is positive.
- Collection rarity. The weight increases when the term appears in fewer documents, provided its TF is positive.

Exercise – Constructing TF-IDF Document Vectors

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

- Calculate the Normalized TF and IDF for each term in the vocabulary.
- Calculate TF-IDF. Use full precision and round final weights to three decimals.
- What is the tf-idf representation of D1? What is the tf-idf representation of vocabulary term match, in this collection?
- Interpret the result. Explain why news receives zero weight and why software receives a higher weight than update in D3.

D / V	match	news	software	team	update	wins
D1						
D2						
D3						

Exercise – Constructing TF-IDF Document Vectors

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

- Calculate the Normalized TF and IDF for each term in the vocabulary.
- Calculate TF-IDF. Use full precision and round final weights to three decimals.
- What is the tf-idf representation of D1? What is the tf-idf representation of vocabulary term match, in this collection?
- Interpret the result. Explain why news receives zero weight and why software receives a higher weight than update in D3.

D / V	match	news	software	team	update	wins
D1	0.500	0.500	0	1.000	0	0.500
D2	0	1.000	0	1.000	0	1.000
D3	0	0.500	1.000	0	0.500	0

The highest term counts are 2 in D1, 1 in D2, and 2 in D3.

Exercise – Constructing TF-IDF Document Vectors

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

- Calculate the Normalized TF and IDF for each term in the vocabulary.
- Calculate TF-IDF. Use full precision and round final weights to three decimals.
- What is the tf-idf representation of D1? What is the tf-idf representation of vocabulary term match, in this collection?
- Interpret the result. Explain why news receives zero weight and why software receives a higher weight than update in D3.

Term	match	news	software	team	update	wins
DF	1	3	1	2	1	2
IDF	0.477	0	0.477	0.176	0.477	0.176

Exercise – Constructing TF-IDF Document Vectors

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

- Calculate the Normalized TF and IDF for each term in the vocabulary.
- Calculate TF-IDF. Use full precision and round final weights to three decimals.
- What is the tf-idf representation of D1? What is the tf-idf representation of vocabulary term match, in this collection?
- Interpret the result. Explain why news receives zero weight and why software receives a higher weight than update in D3.

D / V	match	news	software	team	update	wins
D1	0.239	0	0	0.176	0	0.088
D2	0	0	0	0.176	0	0.176
D3	0	0	0.477	0	0.239	0

Exercise – Constructing TF-IDF Document Vectors

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

- Calculate the Normalized TF and IDF for each term in the vocabulary.
- Calculate TF-IDF. Use full precision and round final weights to three decimals.
- What is the tf-idf representation of D1? What is the tf-idf representation of vocabulary term match, in this collection?
- Interpret the result. Explain why news receives zero weight and why software receives a higher weight than update in D3.

Using vocabulary order [match, news, software, team, update, wins], D1 is represented by its row.

$$\mathbf{v}_{D1} = [0.239,\ 0,\ 0,\ 0.176,\ 0,\ 0.088]$$

Using document order **[D1, D2, D3]**, *match* is represented across the collection by its column.

$$\mathbf{v}_{\text{match}} = [0.239,\ 0,\ 0]$$

Exercise – Constructing TF-IDF Document Vectors

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

- Calculate the Normalized TF and IDF for each term in the vocabulary.
- Calculate TF-IDF. Use full precision and round final weights to three decimals.
- What is the tf-idf representation of D1? What is the tf-idf representation of vocabulary term match, in this collection?
- Interpret the result. Explain why news receives zero weight and why software receives a higher weight than update in D3.

- News. It appears in all three documents. Its IDF is $\log_{10}(3/3) = 0$, so its TF-IDF weight is zero in every document.
- Software versus update. Both appear only in D3, so they have equal IDF. Software occurs twice, while update occurs once. Their normalized TF values are 1 and 0.5, so software receives twice the TF-IDF weight.



Questions?

Comments?

Text Processing & Representation

Recap Q&A

Text Processing – Q1. Tokens and Vocabulary

Question

Consider the following text:

“Cats chase mice. Cats chase birds.”

Lowercase the text, remove the periods, and split on spaces.

- Write the resulting tokens.
- How many token occurrences are there?
- How many distinct terms are there?

Answer to reveal

Tokens

`[cats, chase, mice, cats, chase, birds]`

Token occurrences

6, including repetitions.

Vocabulary

`{cats, chase, mice, birds}`

containing 4 distinct terms.

Text Processing – Q2. Matching Different Word Forms

Question

A document contains *connect*, *connected*, and *connecting*. A user searches for *connect*.

- Which two processing methods could help match these forms?
- What is the main difference between them?
- Should we always apply these methods?

Answer to reveal

Stemming

Applies rules to word endings. The resulting stem need not be a dictionary word.

Lemmatization

Maps words to their dictionary forms using linguistic information. These verb forms have the lemma `connect`.

Task dependence

These methods can support lexical matching, but they may remove useful grammatical distinctions. They are optional processing choices.

Text Processing – Q3. Can Cleaning Change Meaning?

Question

Consider the following sentences:

“The treatment is effective.”
“The treatment is not effective.”

A pipeline removes *the*, *is*, and *not*.

- What remains in each sentence?
- Can we still distinguish their meanings?
- What should we change in the pipeline?

Answer to reveal

Result

Both sentences become `[treatment, effective]`.

Information loss

Removing *not* eliminates the distinction between the positive and negative statements.

Processing decision

Preserve negation when it matters to the task. There is no universal stopword list suitable for every application.

Text Processing – Q4. Design a Suitable Pipeline

Question

You must retrieve passages from scanned reports and use a pretrained Transformer to answer questions.

A colleague proposes the following approach:

“Split each page into 100-word chunks, lowercase everything, remove punctuation and stopwords, and apply stemming.”

Identify **three changes** you would make. Explain why each change matters.

Answer to reveal

1. Extract and inspect

Apply OCR first. Check text accuracy, reading order, and table structure before creating chunks.

2. Preserve meaning and follow the model

Retain useful information, including negation. Use the model’s associated tokenizer and expected preprocessing.

3. Preserve context

Prefer meaningful sentence or section boundaries where practical. Keep headings and source identifiers, and evaluate whether chunks contain enough context.

Text Representation – Q1. Understanding Vector Positions

Question

Use the shared vocabulary order below:

`[match, news, team]`

Represent the document *“news team team”*.

- Write its binary vector.
- Write its count vector.
- Explain why all documents must use the same vocabulary order.

Answer to reveal

Binary vector

`[0, 1, 1]`

Count vector

`[0, 1, 2]`

Shared dimensions

Each position must correspond to the same term across documents so their vectors can be compared.

Text Representation – Q2. What Does Bag of Words Lose?

Question

Consider the following sentences:

“The cat chased the dog.”
“The dog chased the cat.”

- Will their binary vectors differ?
- Will their count vectors differ?
- Will TF-IDF distinguish them when we use the same collection?

Answer to reveal

1. Binary representation

The vectors are identical because both sentences contain the same terms.

2. Count representation

The vectors are identical because the term counts are also equal.

3. TF-IDF representation

The vectors remain identical. Changing term weights does not recover the word order or identify which animal performed the action.

Text Representation – Q4. Can an Unchanged Document Get a Different Vector?

Question

Add a fourth document to the previous collection.

D4. `"match"`

D1 remains *"news team team wins match"*.

- Does its normalized TF change?
- Does its TF-IDF weight for *team* increase or decrease?
- Does its TF-IDF weight for *match* increase or decrease?

Explain your predictions before calculating.

Answer to reveal

1. Normalized TF

It remains unchanged because D1's term counts have not changed.

2. Team

Its weight increases. It now appears in 2 of 4 documents rather than 2 of 3.

3. Match

Its weight decreases. It now appears in 2 of 4 documents rather than 1 of 3.

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