ACADEMIC YEAR 2026/27

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

Text Processing in Practice - System Design

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

Part 1
From Methods to Applications

From Methods to Applications

Text similarity

TEXT SIMILARITY – WHAT IT IS AND HOW IT WORKS

Text similarity measures resemblance under a chosen representation.

INPUT

Two documents or a query and a document.

SIMPLE APPROACH

1. Apply consistent tokenization and normalization on texts.
2. Represent each text with TF-IDF.
3. Compute cosine similarity between the vectors.

Cosine Mechanics

Compares vector directions using the weights of shared terms (i.e. the vocabulary terms).

Key Applications

Uses include candidate retrieval and duplicate detection.

What does the tf-idf score depend on?

TEXT SIMILARITY – EXAMPLE AND LIMITATION

Identical word counts can express different situations.

TEXT A

“Export fails. Import works.”

TEXT B

“Export works. Import fails.”

The same BoW and TF-IDF vectors give cosine similarity 1.

Word counts lose which operation is associated with “fails”.

A high score can provide a candidate that still needs contextual checks.

From Methods to Applications
Information Extraction

KEYWORDS AND KEYPHRASES – WHAT THEY ARE

Keywords and keyphrases provide short descriptions of a document.

INPUT DOCUMENT

“The city approved a renewable energy project. Solar panels will supply public buildings.”

ILLUSTRATIVE OUTPUT

Keyword candidates: energy, panels

Keyphrase candidates: renewable energy project, solar panels

A keyword is a selected term. A keyphrase contains one or more words.

Uses include document labels and query suggestions, among other.

What information does a phrase preserve that separate words may lose?

KEYPHRASE EXTRACTION – A SIMPLE COMBINED APPROACH

POS patterns generate candidates. TF-IDF helps rank them.

CANDIDATE GENERATION

“public buildings” has an adjective followed by a noun.

“renewable energy project” has an adjective and two nouns.

RANKING

One can create estimate the TF-IDF weights of the words in each phrase, so that can identify those with high scores.



Keep all candidates from the original text.



Rank them based on tf-idf. Are the keyphrase better?

Does a grammatical noun phrase necessarily make a good keyphrase?

PATTERN-BASED EXTRACTION – RULES FOR PREDICTABLE FIELDS

Explicit patterns extract fields with recognizable formats.

INPUT

“Record AB123. Planned opening on 12 May 2027.”

OUTPUT & PATTERN

Record identifier = AB123

Date expression = 12 May 2027

Pattern = `\b[A-Z]{2}[0-9]{3}\b`

Regular expressions match character sequences. Dictionaries match known names.

A format match needs validation. A date alone does not identify its event.

Does extracting the date tell us what happened on that date? Does it give the complete information?

RELATION EXTRACTION – DEFINITION, METHOD, AND EXAMPLE

Relation extraction identifies a labeled connection between mentions.

INPUT

“Anna Rossi founded Acme Labs.”

OUTPUT & PATTERN

OUTPUT = (Anna Rossi, founded, Acme Labs)

SIMPLE APPROACH = Match [PERSON] founded [ORGANIZATION]



Define the relation inventory and the direction of its arguments.



Uses include structured archives and questions about connected entities.

Can we reverse the two arguments without changing the meaning?

From Methods to Applications

Information Linking

Dependency parsing describes grammatical relationships between words.

ACTIVE VOICE PATTERN

“Anna founded Acme.”

Anna is the subject of founded. Acme is its object.

PASSIVE VOICE PATTERN

“Acme was founded by Anna.”

The grammatical structure changes, but Anna is still the founder.

A parser supplies links. Rules interpret those links for the extraction task.



POS gives categories. Dependency parsing adds grammatical links.



A simple baseline can use separate active and passive patterns.

Coreference links expressions that refer to the same entity.

INPUT TEXT

“Acme announced a centre. The company may open it in 2027.”

MENTION LINKS

- **The company** → refers to Acme
- **It** → refers to the centre

The opening record can inherit its organization and facility.

A restricted rule baseline can use compatible mentions and nearby context. It will fail on ambiguous cases.

Multiple plausible referents can make a link uncertain.

Why is sentence-by-sentence processing insufficient here?

From Methods to Applications

Negation and uncertainty

Context distinguishes asserted, denied, and possible information.

ASSERTED STATEMENT

“Acme opened a centre.”

Opening reported as completed.

DENIED (NEGATION)

“Acme did not open a centre.”

Opening denied.

POSSIBLE (UNCERTAINTY)

“Acme may open a centre.”

Possible future opening.

STEP 1: CUE DETECTION

A baseline finds cues such as **not**, **no**, **may**, and **possible**.

STEP 2: TARGET ATTACHMENT

It then determines which statement each cue affects.

Which word changes the status in the third sentence?

A cue affects a particular statement, rather than the whole document.

INPUT STATEMENT

“Acme did not open the Milan centre, but opened one in Turin.”

EXTRACTED OUTPUT

Milan opening: Denied

Turin opening: Reported as completed

CLAUSE SEPARATION

The conjunction separates two clauses with different event status.

KEYWORD SEARCH LIMITATION

A keyword search for **not** can mislabel the second event.

From Methods to Applications

Event extraction

EVENT EXTRACTION – DEFINITION & WORKFLOW

Event extraction identifies actions and their associated information.

INPUT STATEMENT

“Acme opened a centre in Milan on 12 May 2027.”

WORKFLOW STEPS

1. Find the action trigger **opened**.
2. Identify the organization and location mentions.
3. Connect the facility, date, and status to that action.

EVENT SCHEMA

An event schema defines the required fields and argument roles.

PRACTICAL USES

Uses include event monitoring, timelines, and structured filtering.

Why should the event schema be defined before annotation?

EVENT EXTRACTION – EXAMPLE OUTPUT & LIMITATION

A useful event record retains status and source evidence.

EXTRACTED EVENT RECORD

Organization: Acme

Action: opening

Facility: centre

Location: Milan

Time: 12 May 2027

Status: reported as completed

Evidence: original sentence and document identifier

STATUS VARIATION

“Acme may open…” would produce a possible status instead.

SYSTEM BOUNDARY

An extraction system represents source claims. It does not verify world facts.

What changes when opened becomes may open?

Part 2
From Applications to Systems

From Applications to Systems

System Design

System Design Connects Methods to a Need

System design chooses components, the data they exchange, and when they run to meet a user's need. We compare processing cost, errors, and result quality.

EXAMPLE JOB LISTING

"Senior NLP Engineer. Remote within Germany. Five years of experience required."

Would you recommend this advertisement to a graduate in Milan seeking a junior NLP job?

User Profile & Mismatch Analysis

For a graduate seeking junior work in Milan, this listing is **UNSUITABLE**

NLP and remote match the topic, but experience level and location conditions fail.

System Objective

A useful job tracker must go beyond basic keyword matching:

1. Finds relevant jobs based on domain topics
2. Rigorously checks **ELIGIBILITY** constraints
3. Present the information to the user

The Requirements for a Suitable Result

The job seeker wants currently open junior data science or NLP roles at those companies, in Milan or remote from Italy.

Representation of the Information need
“Open junior data science or NLP jobs at Acme, Nova or Orion. Milan, or remote work available from Italy.”

System’s Mandatory Conditions

- Chosen company, junior suitability, acceptable location, and open status.
- **A failed condition excludes a job.**
- Missing eligibility needs explicit user checking.

Ranking Preference

Among eligible jobs, favour responsibilities most similar to the user’s interests.

Optimizes relevance score after filtering out ineligible candidates.

Do you agree with this split between mandatory conditions and ranking preferences?

Result Presentation: Each result shows title, company, location, source link, and evidence for the match.

Job advertisement data collection, metadata and processing

DATA COLLECTION

Where can one collect available job advertisements?

- Is it possible to get this information?
- Are there any permissions required?

TEXT REPRESENTATION

Create normalized tokens and TF-IDF vectors separately.

- Keep the original wording exact.

"This role is not remote" changes meaning if we remove "not".

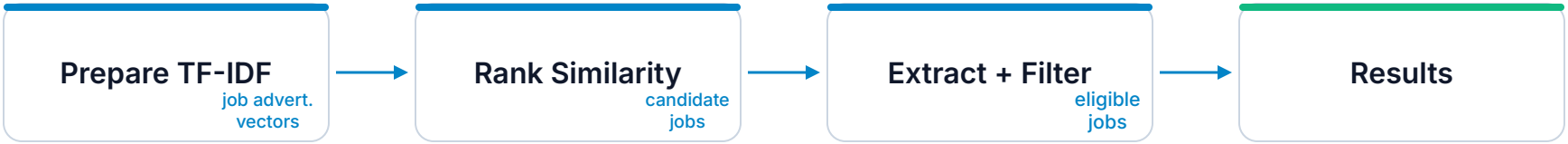
SOURCE RECORD

Keep the original advertisement, source URL, company, collection time, and known availability.

- Some fields come from metadata.
- Others need extraction from text.

System 1: Search First, Then Inspect Candidates

INPUT: COLLECTED ADVERTS, The query we saw before: Open junior data science or NLP jobs at Acme, Nova or Orion. Milan, or remote work available from Italy.



BEFORE A QUERY

Preprocess the collection, build TF-IDF vectors, and prepare an index that links terms to documents.

DURING A SEARCH

Encode the query in the same TF-IDF space, rank by cosine similarity, then extract fields and check mandatory conditions in the top 100 candidates.

What about time, cost, effectiveness? Is there a bottleneck here?

This limits extraction work. Suitable jobs outside the candidates remain missed.

Only the retrieved candidates receive full field extraction and verification.

System 1 Results: Candidates and eligibility decisions

A • RECOMMEND

Acme: Junior Data Scientist, Milan. Graduates welcome.

Recommend: company, junior suitability, and location match.

B • EXCLUDE

Nova: Senior NLP Engineer, remote within Germany. Five years required.

Exclude: experience and remote eligibility fail our requirements, despite matching NLP vocabulary.

C • CHECK

Orion: Junior NLP Developer, remote. Country eligibility unstated.

Check: remote work from Italy remains unconfirmed.

Extracted fields need context

SOURCE ADVERTISEMENT TEXT

“Junior NLP developer. Remote from Italy. Python required. SQL preferred.”

STRUCTURED EXTRACTS

seniority = **junior**

remote country = **Italy**

required skill = **Python**

preferred skill = **SQL**

CONTEXT & RULES DISCOVERY

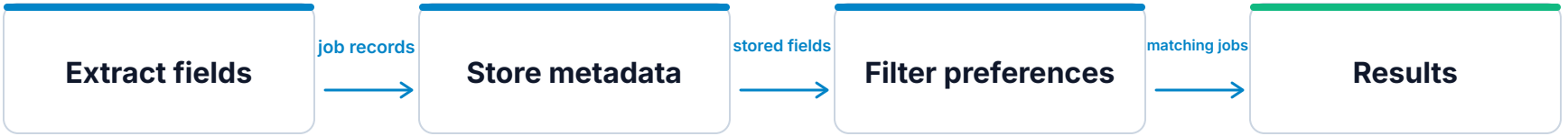
- **NER** identifies Italy as a location.
- A **skill dictionary** matches Python and SQL.
- **Context rules** connect mentions to their specific roles and separate required from preferred skills.

Finding Italy alone does not establish remote eligibility.

Interpret the phrase and keep the supporting source text.

System 2: Extract first, then filter metadata

INPUT: ALL COLLECTED ADVERTISEMENTS



OFFLINE

Use NER, dictionaries, and rules on every advertisement as it arrives or changes. Store the extracted records.

DURING THE SESSION (i.e. WE PUSH INFORMATION)

Filter stored fields by company, location, and junior eligibility. No TF-IDF query is required. Similarity can optionally rank the matches.

The cost is collection-wide extraction.

Field errors can exclude suitable jobs.

Extraction work across repeated searches

ASSUMPTIONS & BASELINE

Assume 1,000,000 advertisements and an estimate of 100 candidates per search. Search-first re-extracts candidates.

System 2: EXTRACT FIRST

1,000,000 once

One-time information extraction for the collection.

System 1: SEARCH FIRST

100 × searches

Re-extracts candidates per query. Extractions scale linearly with search volume. Can anyone think of something better?

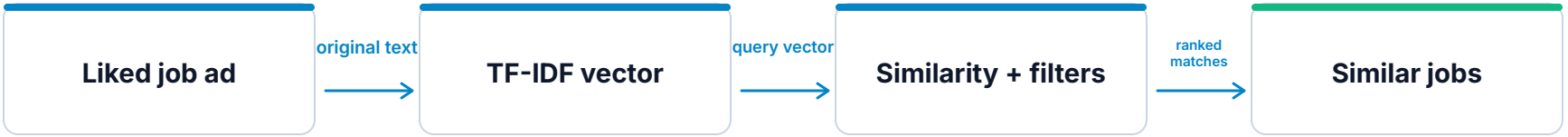
WORKLOAD COMPARISON OVER TIME

After 10 searches: 100 × 10 = 1,000 extractions.
After 10,000 searches: 100 × 10,000 = 1,000,000 extractions.

Equal extraction counts do not mean equal total costs. Because we will probably extracted data from the same ads.

System 3: A liked advertisement becomes the query

INPUT: ONE SELECTED ADVERTISEMENT



QUERY SELECTION & VECTOR MATCHING

The user selects an interesting job instead of writing a query. Compare its TF-IDF vector with collection vectors in the same representation space.

VOCABULARY & ELIGIBILITY FILTERS

Similar vocabulary finds related roles. Liking the responsibilities does not imply accepting the location. Keep explicit eligibility conditions.

Remove the selected advertisement and duplicate vacancies from the results.

Failures point to different components

RETRIEVAL FAILURE

A useful job is absent from the top 100

Improve the query or inspect more candidates. Measure recall and added work.

EXTRACTION FAILURE

Germany-only remote becomes worldwide

Improve context rules and test them on labelled examples.

STALE AVAILABILITY

A closed vacancy remains recommended

Recheck the source and remove or invalidate closed records.

A larger candidate set can recover jobs while adding irrelevant results and work.

Precision and recall of the complete result

10 jobs recommended • 6 recommendations suitable • 8 suitable jobs in total

PRECISION

60% (6 / 10)

Suitable recommendations divided by all recommendations.

4 recommendations fail the requirements.

RECALL

75% (6 / 8)

Suitable recommendations divided by all suitable jobs.

2 suitable jobs were missed.

Four recommendations fail requirements. Two suitable jobs were missed.

Do we keep or change the system? Which component we change?

A design choice with an explicit tradeoff

Requirements and workload guide the choice

EXTRACT IN ADVANCE

For many repeated company and location searches, extracting fields in advance is a reasonable start: searches reuse them.

Filter mandatory conditions, then optionally rank eligible jobs using a written query or liked advertisement.

SEARCH FIRST

Search first is useful when only a small candidate subset needs extraction.

The choice depends on workload, field quality, and refresh needs.

Four questions for any text system

01

What must pass?

02

When does work happen?

03

How will we test the result?

04

What happens if we have 10x or 100x the input load?

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