

NLP – Unit 4 (Information Retrieval using NLP) – END-SEM PYQ Answers

MAY-JUNE 2023

Q3a) Describe the concept of Information Retrieval system in Natural Language Processing. [4 Marks]

Information Retrieval (IR) in NLP

Information Retrieval is the science of finding material — usually documents — that satisfies an information need from within large collections. The information need is expressed as a query (typically a few keywords or a natural language question), and the system returns a ranked list of relevant documents.

Core components of an IR system:

- Document Collection: The corpus of text to be searched (e.g., the web, legal databases, email archives).
- Indexer: Processes each document at ingestion time, tokenises it, removes stop words, applies stemming/lemmatisation, and builds an inverted index mapping each term to the list of documents it appears in.
- Query Processor: Takes the user's query, applies the same preprocessing, and transforms it into a form compatible with the index.
- Retrieval Model: Computes a relevance score between the query and each candidate document (e.g., TF-IDF cosine similarity, BM25, language model probability).
- Ranking Engine: Sorts documents by relevance score and returns the top-K results.
- Evaluation Module: Uses metrics like Precision, Recall, and Mean Average Precision (MAP) to assess system quality.

Role of NLP in IR:

- Tokenisation and normalisation ensure terms are indexed consistently.
- Stemming/lemmatisation reduces 'running', 'runs', 'ran' to the same root, improving recall.
- Named Entity Recognition links queries about 'Sachin Tendulkar' to documents mentioning 'Tendulkar'.
- Query expansion uses synonyms (from WordNet) to broaden the search.
- Neural/semantic retrieval (BERT-based dense retrieval) moves beyond keyword matching to meaning-level matching.

Note: IR and NLP are deeply intertwined. Modern search engines use transformer models to understand query intent, not just match keywords.

Q3b) What is Named Entity Recognition (NER)? Describe the various metrics used for evaluation. [8 Marks]

Named Entity Recognition (NER)

NER is the NLP task of identifying and classifying named entities (real-world objects with proper names) in raw text into predefined categories such as person names, organisations, locations, dates, quantities, monetary values, and more.

Example:

'Narendra Modi visited New Delhi on 15 August 2024.'

- Narendra Modi → PERSON
- New Delhi → LOCATION (GPE — Geo-Political Entity)
- 15 August 2024 → DATE

NER System Building Process:

- Rule-based approach: Hand-crafted patterns and gazetteers (lists of known entities). High precision but low recall; brittle.
- Feature-based ML (CRF): Extract features (word shape, POS tag, surrounding words, prefix/suffix) and train a Conditional Random Field to label each token with an IOB tag: B-PER (beginning of a person), I-PER (inside), O (outside).
- Deep learning (BiLSTM-CRF): A bidirectional LSTM captures context from both directions; a CRF layer enforces valid label sequences.
- Transformer-based (BERT fine-tuned): The current state of the art. BERT's contextual embeddings are fed into a token-classification head.

IOB Tagging Scheme:

- B-TAG: Beginning of an entity of type TAG.
- I-TAG: Inside (continuation of) an entity of type TAG.
- O: Outside — not part of any entity.

Evaluation Metrics:

- Precision (P): Of all entities the system predicted, what fraction were correct? $P = TP / (TP + FP)$.
- Recall (R): Of all true entities in the text, what fraction did the system find? $R = TP / (TP + FN)$.
- F1-Score: The harmonic mean of Precision and Recall. $F1 = 2PR / (P + R)$. This is the most commonly used single metric for NER.
- Entity-level vs Token-level evaluation: Entity-level requires an entire entity span and its type to match exactly; token-level is more lenient.
- Micro-averaged F1: Aggregates TP, FP, FN across all entity types before computing F1. Dominated by frequent types.
- Macro-averaged F1: Computes F1 per entity type, then averages. Treats all types equally.

Note: The CoNLL-2003 benchmark (news wire, IOB tagging for PER, ORG, LOC, MISC) is the standard for English NER evaluation. State-of-the-art F1 scores exceed 93% on this benchmark.

Q3c) What is Cross-Lingual Information Retrieval and how is it used in NLP? Provide an example. [6 Marks]

Cross-Lingual Information Retrieval (CLIR)

Cross-Lingual Information Retrieval is the task of retrieving relevant documents written in a language different from the language of the query. It bridges language barriers so users can query in their native language and retrieve documents from multilingual corpora.

Why CLIR is needed:

- Most content on the web is not in English; monolingual retrieval misses it.
- In multilingual organisations, employees may need to search across documents in multiple languages.

- Patent search, scientific literature retrieval, and intelligence analysis often require cross-lingual access.

Approaches:

- Query Translation: Translate the query from the source language to the target language using a machine translation (MT) system, then run standard monolingual retrieval. Simple, but translation errors propagate.
- Document Translation: Translate all documents in the collection to the query language. High quality but expensive and storage-intensive.
- Bilingual Word Embeddings: Map words from both languages into a shared vector space. No explicit translation needed; retrieval is done by semantic similarity in the shared space.
- Cross-lingual BERT (mBERT, XLM-R): Pre-trained on 100+ languages simultaneously, producing language-agnostic representations for direct cross-lingual matching.

Example:

A Marathi-speaking user types: 'पुण्यातील पूर नियंत्रण' (flood control in Pune). A CLIR system translates this to English as 'flood control in Pune', retrieves relevant English news articles about Pune floods, and either presents them translated back to Marathi or presents the English originals with extracted summaries.

Note: The main challenge in CLIR is that machine translation errors can severely reduce retrieval quality. Bilingual embedding approaches are more robust because they avoid explicit translation.

Q4a) Explain the concept of the Vector Space Model and how it is used in Information Retrieval. [6 Marks]

Vector Space Model (VSM)

The Vector Space Model, introduced by Salton et al. (1975), represents both documents and queries as vectors in a high-dimensional term space. Each dimension corresponds to a unique term in the vocabulary, and the value along that dimension reflects the weight of the term in the document (typically TF-IDF).

Representation:

- Vocabulary: Suppose $V = \{\text{cat, dog, fish}\}$ (3 terms, so 3 dimensions).
- Document $d_1 = \text{'cat fish cat'}$ → term frequency vector: $[2, 0, 1]$. After TF-IDF weighting, rare terms are upweighted.
- Query $q = \text{'cat dog'}$ → vector: $[1, 1, 0]$.

Similarity Computation:

Relevance is measured by cosine similarity between the query vector q and each document vector d :

$$\text{cosine_sim}(q, d) = (q \cdot d) / (|q| \times |d|)$$

Cosine similarity ranges from 0 (orthogonal / unrelated) to 1 (identical direction / very similar). Angle between vectors is more important than magnitude, so normalisation is built-in.

Retrieval Process:

1. Index all documents as TF-IDF vectors.
2. Represent the query as a vector.
3. Compute cosine similarity between query and every document.
4. Rank documents by similarity score and return the top-K.

Strengths:

- Partial matching: documents that share some (but not all) query terms still receive a non-zero score.
- Efficient with inverted index: only non-zero cells matter.
- Well-understood and widely deployed.

Weaknesses:

- Bag-of-words assumption: word order and syntax are ignored.
- No synonymy handling: 'car' and 'automobile' are orthogonal dimensions.
- High dimensionality: vocabulary can be hundreds of thousands of terms.

Note: Modern neural retrieval (bi-encoders, dense passage retrieval) extends VSM by replacing sparse TF-IDF vectors with dense 768-dimensional BERT embeddings, capturing semantic similarity instead of just keyword overlap.

Q4b) Describe entity extraction and relation extraction with the help of examples. [8 Marks]

Entity Extraction

Entity extraction (also called Named Entity Recognition in a broader sense) is the process of identifying and categorising entities — specific real-world things — mentioned in text. It is the first step in building a knowledge graph or populating a database from unstructured text.

Types of entities extracted:

- Named entities: PERSON (Elon Musk), ORGANISATION (Tesla), LOCATION (San Francisco), DATE (January 2024), MONEY (\$1 billion).
- Domain-specific entities: In biomedical text — DISEASE (diabetes), DRUG (metformin), GENE (BRCA1).

Example:

'Apple Inc. announced the iPhone 16 on September 9, 2024, in Cupertino, California.'

- Apple Inc. → ORGANISATION
- iPhone 16 → PRODUCT
- September 9, 2024 → DATE
- Cupertino, California → LOCATION

Relation Extraction

Relation extraction identifies semantic relationships between entities mentioned in the same sentence or document. It answers questions like 'Who founded what?', 'Who works for whom?', or 'What drug treats what disease?'

Common relations:

- founded_by(Apple, Steve Jobs)
- located_in(Infosys, Pune)
- treats(metformin, type 2 diabetes)

Example:

'Elon Musk founded SpaceX in 2002 in Hawthorne, California.'

- Entities: Elon Musk (PERSON), SpaceX (ORGANISATION), 2002 (DATE), Hawthorne (LOCATION).
- Relations: founded_by(SpaceX, Elon Musk), founded_in(SpaceX, 2002), located_in(SpaceX, Hawthorne).

Methods for Relation Extraction:

- Pattern-based: Manually define templates like '[PERSON] founded [ORG]' and match them against text.
- Supervised ML: Train a classifier on sentence pairs (entity1, entity2) and predict the relation label.
- Distant supervision: Automatically label training data using an existing knowledge base (e.g., Freebase, Wikidata) and train a model on the noisy labels.
- BERT-based RE: Fine-tune BERT on pairs of entities in context; it achieves state-of-the-art results on benchmarks like TACRED.

Note: Entity extraction and relation extraction together form the backbone of Knowledge Graph construction. The pipeline is: text → entity extraction → relation extraction → knowledge graph triples (subject, predicate, object).

Q4c) What is Coreference Resolution? Give examples. [4 Marks]**Coreference Resolution**

Coreference resolution is the task of finding all expressions in a text that refer to the same real-world entity, and grouping them into coreference chains (also called clusters or antecedent chains). This is essential for understanding who or what is being talked about as pronouns and other referring expressions appear.

Example 1:

'Priya went to the market. She bought vegetables. Later, she cooked dinner.'

- 'Priya', 'She' (first occurrence), 'she' (second occurrence) all refer to the same person → one coreference chain: {Priya, She, she}.

Example 2:

'The committee submitted its report. The document was 200 pages long.'

- 'The committee' and 'its' → coreference chain 1.
- 'its report' and 'The document' → coreference chain 2.

Approaches:

- Rule-based (Hobbs algorithm): Use syntactic heuristics — pronouns typically refer to the most recent compatible noun phrase.
- Mention-pair models: Train a classifier to decide, for every pair of mentions, whether they are coreferent.
- Mention-ranking models: For each mention, rank all candidate antecedents and choose the most likely one.
- End-to-end neural models (Lee et al., 2017): Jointly detect mentions and cluster them using span representations.

Note: Coreference resolution is a prerequisite for tasks like document summarisation, question answering (e.g., 'He said...' — who is He?), and information extraction. The OntoNotes corpus is the standard benchmark.

NOV-DEC 2023

Q3a) Describe the concept of Information Retrieval. Explain the significance of NLP in IR. [4 Marks]

[REPEATED] Q3a of May-June 2023 — Information Retrieval system in NLP

See May-June 2023 Q3a for the complete answer. The additional angle here is the significance of NLP:

- Without NLP, IR is purely keyword matching. NLP adds linguistic understanding: morphological analysis (stemming), semantic analysis (word sense, synonymy via WordNet query expansion), and discourse understanding (coreference for document-level queries).
- Modern semantic search systems use NLP-derived dense embeddings to match 'automobile accident' with documents containing 'car crash'.

Q3b) Explain reference resolution and coreference resolution with example. [8 Marks]

Reference Resolution vs Coreference Resolution

Reference resolution is the broader task of determining what any referring expression (a phrase used to pick out an entity) refers to in the world or in the discourse model. Coreference resolution is a specific subtype.

Types of Reference:

- Pronominal reference: A pronoun refers to a previously mentioned noun. 'Rahul passed the exam. He was happy.' → 'He' refers to 'Rahul'.
- Definite description reference: A definite noun phrase picks out a specific entity. 'The prime minister arrived.' — resolving 'the prime minister' requires knowing the context (which country, which time).
- Coreference: Two or more noun phrases in a text refer to the same real-world entity. This includes pronouns but also full noun phrases: 'Modi', 'the Prime Minister', 'he' may all corefer.
- Bridging reference (associative): 'I walked into the room. The ceiling was low.' — 'the ceiling' is not previously mentioned but is understood as part of 'the room'.

Coreference Resolution Example:

'Sita told Gita that she would come to the party.'

- Ambiguous: does 'she' refer to Sita or Gita? A coreference resolver must use pragmatic and syntactic clues to decide.

Evaluation for coreference: MUC score, B-cubed, and CEAF are standard metrics that compare predicted clusters to gold clusters.

Note: Reference resolution is critical for reading comprehension, dialogue systems, and summarisation — any task where understanding pronouns matters.

Q3c) What is Cross-Lingual Information Retrieval and how is it used in NLP? Provide an example. [6 Marks]

[REPEATED] Q3c of May-June 2023 — Cross-Lingual Information Retrieval

See May-June 2023 Q3c for the complete answer.

Q4a) Explain the concept of the Vector Space Model and how it is used in Information Retrieval. [6 Marks]

[REPEATED] Q4a of May-June 2023 — Vector Space Model

See May-June 2023 Q4a for the complete answer.

Q4b) Describe entity extraction and relation extraction with the help of examples. [8 Marks]

[REPEATED] Q4b of May-June 2023 — Entity Extraction and Relation Extraction

See May-June 2023 Q4b for the complete answer.

Q4c) What is Named Entity Recognition (NER)? Describe the various metrics used for evaluation. [4 Marks]

[REPEATED] Q3b of May-June 2023 — NER and Evaluation Metrics

See May-June 2023 Q3b for the full NER explanation and evaluation metrics (Precision, Recall, F1, Micro/Macro-averaged).

MAY-JUNE 2024

Q3a) Describe the Vector Space Model (VSM) for information retrieval — strengths and weaknesses. [9 Marks]

[REPEATED] Q4a of May-June 2023 — Vector Space Model (extended)

See May-June 2023 Q4a for the core VSM explanation. Additional depth for 9 marks:

Extended: Limitations and Beyond VSM

- Term independence assumption: VSM treats all terms as orthogonal (independent). In reality, 'king' and 'monarch' are related; VSM cannot capture this.
- No word order: 'dog bites man' and 'man bites dog' have identical VSM vectors.
- Vocabulary mismatch problem: if a user queries 'automobile' but a document uses 'car', VSM gives zero similarity even though they are synonymous.
- Solutions: LSA reduces the vector space to capture latent semantic relationships; Word2Vec embeddings cluster similar words; BERT-based bi-encoders produce dense semantic vectors.
- BM25 (Okapi BM25) is the standard practical improvement over TF-IDF-cosine; it applies term frequency saturation (TF doesn't grow linearly) and document length normalisation.

Q3b) Discuss different methods for evaluating NER systems — metrics and result analysis. [9 Marks]

[REPEATED] Q3b of May-June 2023 — NER and Evaluation Metrics (extended)

See May-June 2023 Q3b for Precision, Recall, F1, and IOB tagging. Additional depth:

Extended: Deeper Evaluation Analysis

- Exact match vs partial match: Exact match (strict) requires the full entity span to match; partial match (lenient) gives credit for overlapping spans.

- Per-class F1: Computing F1 separately for PERSON, ORG, LOC, etc. reveals which entity types the system handles poorly (e.g., diseases in a news-trained model).
- Confusion matrix analysis: Reveals common errors — e.g., the system confuses ORG for LOC when 'India' appears (it can be both).
- Error analysis categories: Boundary errors (correct type, wrong span), type errors (correct span, wrong type), missed entities (false negatives), spurious entities (false positives).
- Cross-domain evaluation: A model trained on news wire (CoNLL-2003) degrades sharply on biomedical text; domain adaptation is necessary.

Q4a) Define Cross-Lingual Information Retrieval and discuss challenges. How does MT assist in CLIR? [9 Marks]

[REPEATED] Q3c of May-June 2023 — CLIR (extended)

See May-June 2023 Q3c for the core CLIR explanation. Additional depth on challenges:

Challenges in CLIR

- Translation ambiguity: Many words have multiple translations. 'Bank' can translate to 'banco' (financial) or 'orilla' (riverbank) in Spanish; choosing the wrong one hurts retrieval.
- Out-of-vocabulary (OOV) words: Technical terms, proper nouns, and neologisms may not appear in the MT system's training data.
- Script and encoding issues: Retrieving Arabic documents from a Latin-script query requires proper Unicode handling and transliteration.
- Language-specific morphology: Highly inflected languages (Finnish, Turkish) may have many surface forms for the same root; stemming must be applied carefully.
- Evaluation cost: Building multilingual test collections with relevance judgements in multiple languages is expensive (CLEF and NTCIR are the main evaluation forums).

How Machine Translation assists CLIR:

- Query-side MT: The most common approach. The query is translated to the target language and standard IR proceeds. Errors affect only one short query, not the entire corpus.
- Document-side MT: Every document is translated to the query language at index time. Higher quality retrieval but massive computational cost.
- Pivot-based MT: For low-resource language pairs (e.g., Swahili → Kannada), translate through English as a pivot.

Q4b) Explain entity extraction in NLP. How does it differ from NER? Real-world applications. [9 Marks]

[REPEATED] Q4b of May-June 2023 — Entity Extraction and Relation Extraction

See May-June 2023 Q4b for entity extraction details. Key additional point on the distinction:

Entity Extraction vs Named Entity Recognition — The Distinction

NER is specifically about named entities — things with proper names (persons, places, organisations). Entity extraction is a broader term that includes NER but also covers:

- Temporal expressions: 'last Tuesday', 'three days ago', 'Q3 2024' (not proper names but entities).
- Numerical expressions: prices (\$499), percentages (15%), quantities (3 kg).
- Domain-specific concepts: In medical NLP, symptoms, procedures, and medications are entities but not 'named entities' in the classical sense.

- Events: 'the 2016 floods in Chennai' — the flood event itself is an entity.

Real-world applications:

- Finance: Extract company names, financial metrics, and events from earnings reports to populate financial knowledge graphs.
- Healthcare: Extract drug names, dosages, adverse events from clinical notes for pharmacovigilance.
- Legal: Extract party names, dates, statutes, and case references from court documents for e-discovery.
- News intelligence: Extract events, people, and locations from news streams for situational awareness dashboards.

NOV-DEC 2025

[REPEATED] Q3a) Explain VSM in IR — document/query representation, relevance computation. [9 Marks] -> See: 2023/2024 Unit 4 document, May-Jun 2023 Q4a (6 marks) and May-Jun 2024 Q3a (9 marks — extended version with BM25 and neural retrieval). Use the 9-mark extended answer.

Q3b) Compare and contrast Entity Extraction, Relation Extraction, and Coreference Resolution. How do they together build knowledge from unstructured text? Illustrate with examples. [8 Marks]

The Knowledge Extraction Pipeline

Entity extraction, relation extraction, and coreference resolution are three complementary NLP tasks that together transform raw unstructured text into structured, queryable knowledge. Think of them as three successive layers of understanding: who/what is mentioned (entity extraction), how are they connected (relation extraction), and are different mentions referring to the same real-world thing (coreference resolution).

Entity Extraction

Entity extraction identifies and classifies spans of text that refer to real-world things. These include named entities such as PERSON, ORGANISATION, LOCATION, DATE, and MONEY, but also domain-specific concepts like drugs, genes, and legal clauses. The output is a set of labelled spans. Example sentence: 'Tata Motors announced a merger with Jaguar Land Rover on 26 March 2023 for \$2.3 billion.'

- Tata Motors -> ORGANISATION
- Jaguar Land Rover -> ORGANISATION
- 26 March 2023 -> DATE
- \$2.3 billion -> MONEY

Relation Extraction

Relation extraction goes one level deeper: given a pair of entities that have already been identified, it classifies the semantic relationship between them. The output is a set of triples (subject, predicate, object) that can directly populate a knowledge graph.

- (Tata Motors, acquired, Jaguar Land Rover)
- (acquisition, date, 26 March 2023)

- (acquisition, amount, \$2.3 billion)

Without entity extraction, relation extraction has nothing to work with. The two tasks are therefore almost always pipelined together, and increasingly handled jointly by a single neural model.

Coreference Resolution

Even a long document about Tata Motors will not write 'Tata Motors' in every sentence. After the first mention, the text will use 'the company', 'it', 'the automaker', and so on. Coreference resolution groups all these mentions into the same coreference chain, ensuring that downstream systems know they all refer to the same entity.

Example continuation: 'The company confirmed the deal through its spokesperson. It plans to keep all UK manufacturing plants open.'

- 'The company', 'its', 'It' -> all corefer to Tata Motors.

Without coreference resolution, relation extraction would fail to link 'It plans to keep plants open' back to Tata Motors, producing a dangling, unattributed fact.

Comparison Table

| Aspect | Entity Extraction | Relation Extraction | Coreference Resolution |
|-------------------|-------------------------------|---|------------------------------------|
| Input | Raw text | Text + entity spans | Full document |
| Output | (span, entity type) pairs | (e1, relation, e2) triples | Clusters of coreferent mentions |
| Question answered | Who/what is mentioned? | How are entities related? | Are two mentions the same entity? |
| Dependency | None (first step) | Needs entity extraction | Needs entity extraction |
| Key challenge | Boundary detection, ambiguity | Sparse labelled data for rare relations | Long-range pronoun resolution |
| Standard model | BiLSTM-CRF, BERT-NER | BERT fine-tuned on TACRED | End-to-end span model (Lee et al.) |

Together these three tasks form the backbone of Knowledge Graph construction: entity extraction identifies nodes, relation extraction creates labelled edges, and coreference resolution merges duplicate nodes that represent the same real-world entity.

Note: The novel angle in this question compared to earlier papers is the compare-and-contrast framing. Examiners want you to show you understand the dependencies — entity extraction feeds relation extraction, and coreference resolution is what allows both to work correctly across a multi-sentence document, not just sentence by sentence.

[REPEATED] Q4a) Describe NER system building process using supervised learning approach.

[9 Marks] -> See: 2023/2024 Unit 4 document, May-Jun 2023 Q3b (8 marks) — the supervised learning detail (feature engineering, CRF, IOB tagging, BiLSTM-CRF pipeline) is fully covered there. For 9 marks, add: data annotation pipeline (Prodigy/Label Studio), train/dev/test splits, hyperparameter tuning, and error analysis cycle.

[REPEATED] Q4b) What is CLIR? Discuss challenges and evaluate different CLIR approaches.

[8 Marks] -> See: 2023/2024 Unit 4 document, May-Jun 2023 Q3c (6 marks) and May-Jun 2024 Q4a (9 marks — extended with challenge categories and MT-assisted CLIR). Use the 9-mark version.

MAY-JUN 2025

[REPEATED] Q3a) Explain reference resolution and coreference resolution in NLP. How do they help understand entity relationships? Provide examples. [8 Marks] -> See: 2023/2024 Unit 4 document, Nov-Dec 2023 Q3b (8 marks) — full treatment of both reference resolution types and coreference chains with worked examples.

[REPEATED] Q3b) Explain CLIR. Discuss challenges and techniques/models to address them. [8 Marks] -> See: 2023/2024 Unit 4 document, May-Jun 2023 Q3c (6 marks) and May-Jun 2024 Q4a (9 marks). Use the 9-mark extended answer from 2024.

Q3c) What is Information Retrieval, and how is it used in NLP? [2 Marks]

Information Retrieval — 2-Mark Summary

Information Retrieval (IR) is the task of finding documents from a large collection that are relevant to a user's information need, expressed as a query. In NLP, IR is used as a foundation for question answering (retrieve relevant passages before extracting the answer), conversational agents (retrieve knowledge base entries to inform responses), and search engines (rank web pages by relevance to a query). NLP techniques such as tokenisation, stemming, and semantic embeddings are applied at the retrieval stage to improve matching quality beyond simple keyword overlap.

Q4a) Explain the process of Entity Extraction in Information Retrieval. Discuss techniques and algorithms used. [8 Marks]

[REPEATED] Core entity extraction concepts -> See: 2023/2024 Unit 4 document, May-Jun 2023 Q4b (8 marks) — covers definition, types of entities, NER pipeline, and relation extraction.

The new angle here is specifically how entity extraction is used within an Information Retrieval system. This connection deserves its own explanation:

Role of Entity Extraction Inside an IR System

Standard IR matches query terms against document terms using TF-IDF or BM25. This fails when the query is 'Who is the CEO of Infosys?' — the answer 'Salil Parekh' does not lexically overlap with the query. Entity extraction enables entity-aware retrieval:

- At index time: Run NER over all documents and store entity annotations in the index. 'Salil Parekh: PERSON, CEO' gets linked to Infosys documents.
- At query time: Parse the query, identify 'Infosys' as ORGANISATION and 'CEO' as a role. Match against entity-enriched index entries rather than raw text.
- Result: the system retrieves documents where Salil Parekh is mentioned in connection with Infosys, even if the exact phrase 'CEO of Infosys' never appears.

Techniques and Algorithms

- Gazetteers and rule-based matching: Maintain lists of known company names, people, locations. Fast but incomplete — cannot handle newly formed entities.
- CRF (Conditional Random Field) with handcrafted features: Captures sequential dependencies between labels; still a strong baseline. Features: word shape (capitalisation), POS tag, prefix/suffix patterns, surrounding word identity.

- BiLSTM-CRF: A bidirectional LSTM captures context from both directions; a CRF output layer enforces valid IOB label sequences (e.g., an I-ORG cannot follow a B-PER). This was state-of-the-art around 2016-2018.
- BERT fine-tuned for NER (token classification head): Each token's contextual embedding is passed to a linear layer that predicts its IOB label. State-of-the-art; especially strong on entity boundary detection.
- Distant supervision for entity linking: Instead of manually labelling all entities, use a knowledge base (Wikidata, DBpedia) to automatically generate training data by matching known entity names to their occurrences in text.

Note: The distinction between entity extraction and entity linking is important for IR: extraction finds the mention span and assigns a coarse type (PERSON), while linking connects it to a canonical identifier in a knowledge base (e.g., 'Salil Parekh' -> wd:Q25160897). Linking is what enables truly precise entity-based retrieval.

[REPEATED] Q4b) Describe VSM for IR. How does it represent documents/queries? Similarities? Strengths/weaknesses? [8 Marks] -> See: 2023/2024 Unit 4 document, May-Jun 2024 Q3a (9 marks extended version) — includes BM25, neural dense retrieval comparison, and full weakness analysis.

Q4c) What is Named Entity Recognition (NER)? [2 Marks]

Named Entity Recognition (NER) is the NLP task of identifying spans of text that refer to named real-world entities and classifying them into predefined categories such as PERSON, ORGANISATION, LOCATION, DATE, and MONEY. For example, in 'Sundar Pichai leads Google in California', the NER system identifies and labels: Sundar Pichai as PERSON, Google as ORGANISATION, and California as LOCATION. NER is a foundational task used in information extraction, question answering, and knowledge graph construction.

Reference: What is New in These 2025 Unit 4 Questions

Most Unit 4 questions in both 2025 papers are repeats of 2023/2024 content. The genuinely new material is: (1) the compare-and-contrast framing of Entity Extraction, Relation Extraction, and Coreference Resolution as a unified pipeline in Nov-Dec 2025 Q3b — earlier papers asked about these individually; (2) the explicit connection between entity extraction and IR system design in May-Jun 2025 Q4a; and (3) the 2-mark mini-questions on IR and NER which are new in format but trivially answered from the longer treatments in prior papers.

Additional Reference Notes for Unit 4

BM25 — Better than TF-IDF Cosine

BM25 (Okapi Best Match 25) is an improved retrieval function that adds TF saturation (so doubling a term frequency does not double the score) and document length normalisation (shorter documents are preferred over longer ones with the same raw count). It is the default in Elasticsearch and Solr.

$$\text{BM25}(q,d) = \sum \text{IDF}(t) \times [\text{tf}(t,d) \times (k_1 + 1)] / [\text{tf}(t,d) + k_1 \times (1 - b + b \times |d| / \text{avgdl})]$$

Evaluation Metrics Summary

| Metric | Formula | Use Case |
|-----------|--|--------------------------------|
| Precision | $TP / (TP + FP)$ | How accurate are predictions? |
| Recall | $TP / (TP + FN)$ | How complete are predictions? |
| F1 Score | $2PR / (P + R)$ | Single balanced metric for NER |
| MAP | Mean of Average Precision across queries | Ranking quality in IR |