19CSE454- INFORMATION RETRIEVAL-CASE STUDY

ENTITY QUERY FEATURE EXPANSION

Yash P CB.EN.U4CSE21270
Arjun P CB.EN.U4CSE21007
Rohith M CB.EN.U4CSE21048
Devanshu V CB.EN.U4CSE21218

AIM

The primary aim of this project is to improve the accuracy and relevance of search results in an information retrieval system by implementing the Entity Query Feature Expansion model. This technique improves query understanding by linking entities within the query to a structured knowledge base and expanding the query with additional related features.

OBJECTIVES

- Entity Linking: Identify and link entities in user queries to corresponding entries in a knowledge base.
- Query Expansion: Expand the original query with related entities, types, and categories derived from the knowledge base to improve retrieval performance.
- Document Retrieval: Retrieve and rank documents that are most relevant to the expanded query.
- Evaluation: Measure the effectiveness of the EQFE method compared to baseline retrieval models using standard IR metrics like Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG).

DATASETS USED

1. TREC Robust04 Dataset

- A collection of news articles from the 1990s, commonly used in IR research.

2. ClueWeb09/ClueWeb12 Datasets

- Large-scale web document collections used for evaluating web-scale IR systems.

3.Wikipedia Dumps/DBpedia

- Knowledge base for entity linking and query expansion.

METHODOLOGY - Implementation using in Python

Entity Recognition and Linking

- Entity Recognition: Use spaCy to identify named entities (e.g., people, places, organizations) within the queries and documents.
- Entity Linking: Map recognized entities to external knowledge bases such as Wikipedia or DBpedia using APIs, enriching the entities with additional context and related attributes.

Query Expansion

- Feature Extraction: Extract entity-related features (e.g., types, categories, and attributes) from linked entities.
- Query Expansion: Expand the original query by incorporating these entity features, leading to a more contextually enriched query.

Document Ranking

- Vectorization: Use TF-IDF or BM25 to convert documents and expanded queries into vector representations.
- Similarity Calculation: Calculate the similarity between the expanded query vectors and document vectors.
- Ranking: Rank documents based on their similarity scores, prioritizing those that are more relevant to the expanded queries

EXAMPLE SCENARIO

When you search for "Andrés Iniesta," a traditional search engine might expand the query with related terms like "Barcelona," "midfielder," or "World Cup."

EXAMPLE SCENARIO

Using EQFE, the search engine identifies "Andrés Iniesta" as a specific entity and links it to a knowledge base with structured information.

Entity Identified:

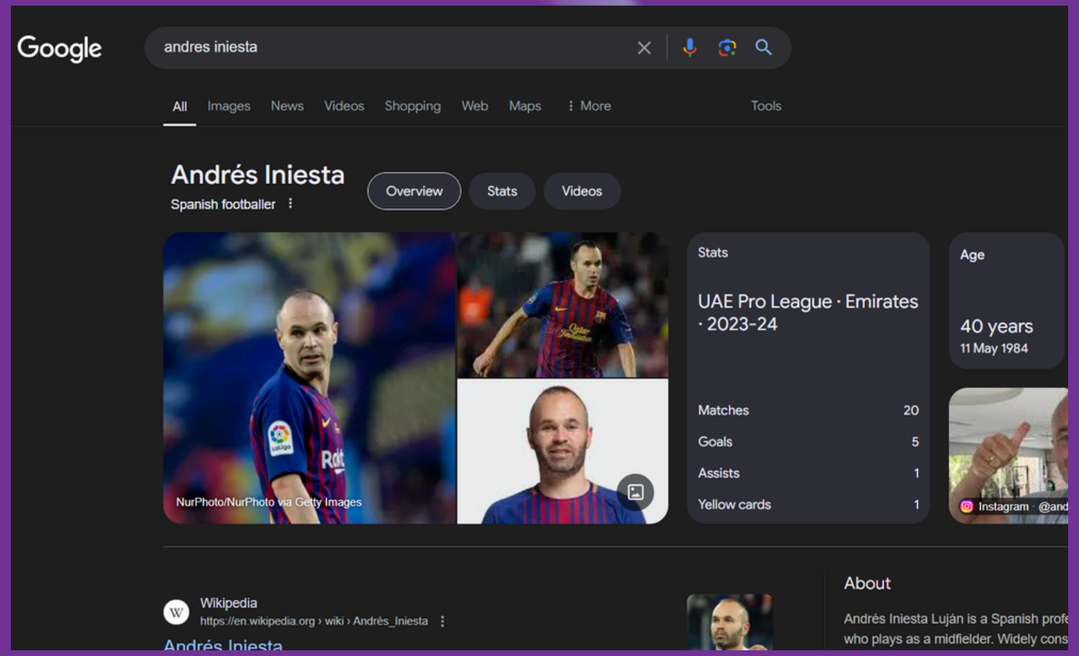
- Entity Name: Andrés Iniesta
- Entity Type: Footballer

Knowledge Base Information:

- Date of Birth: May 11, 1984
- Nationality: Spanish
- Notable Clubs: FC Barcelona, Vissel Kobe
- Positions: Midfielder
- Major Achievements:
  - UEFA Champions League titles with FC Barcelona
  - FIFA World Cup winner with Spain in 2010
  - UEFA European Championship winner with Spain in 2008 and 2012
  - Scored the winning goal in the 2010 World Cup final

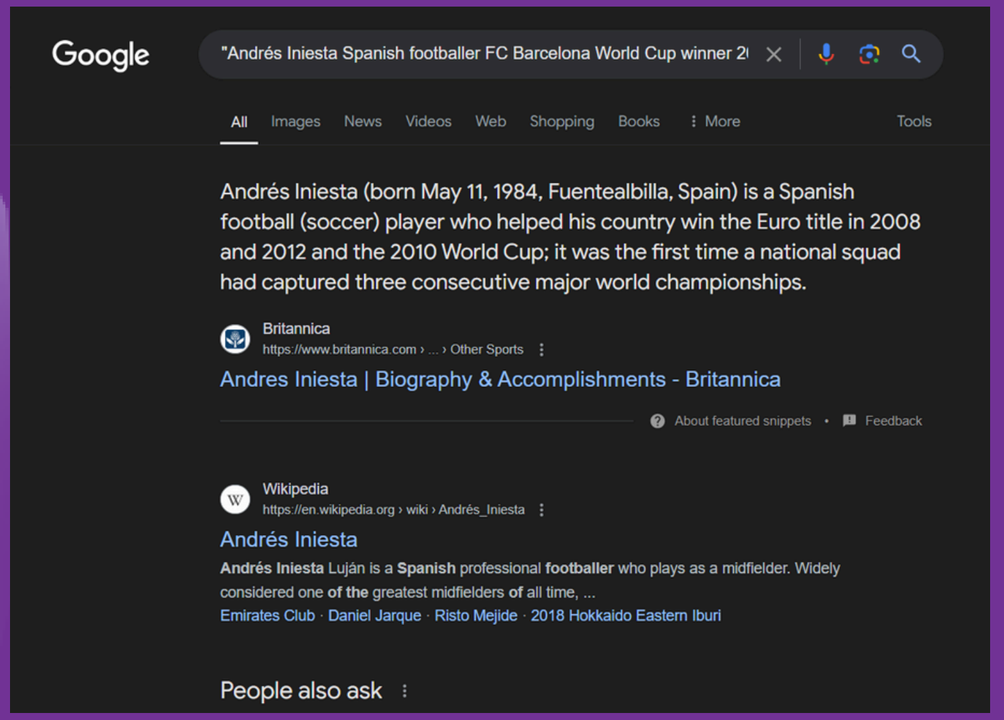
Normal Search Query

"Andrés Iniesta"



Expanded Query

"Andrés Iniesta Spanish footballer FC Barcelona World Cup winner 2010 UEFA Champions League



EVALUATION AND RESULTS

- Metrics: Evaluate the retrieval performance using Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) to assess the effectiveness of the expanded queries.

- Impact Assessment: Analyze the impact of different entity features on document ranking. Determine which features contribute most to improved retrieval performance.

- Visualization of Results: Use Python libraries like matplotlib to create plots that show the performance of the retrieval system, including the ranking quality and feature importance.

CONCLUSION

- Entity-based query expansion significantly improved search result relevance and document ranking.
- Utilized Python libraries for entity recognition, query expansion, and document ranking efficiently.
- Evaluation metrics like MAP and NDCG indicated better retrieval performance compared to baseline methods.
- Analysis revealed the impact of different entity features on ranking quality.
- The method can be further refined with larger knowledge bases and advanced NLP models for even better results.

THANK YOU!