Mini-QA

Team Name: SEALS
Swapnil Sagar
Suganya Sivakumar
Akhauri Prateek Shekhar

Approaches to Question Answering

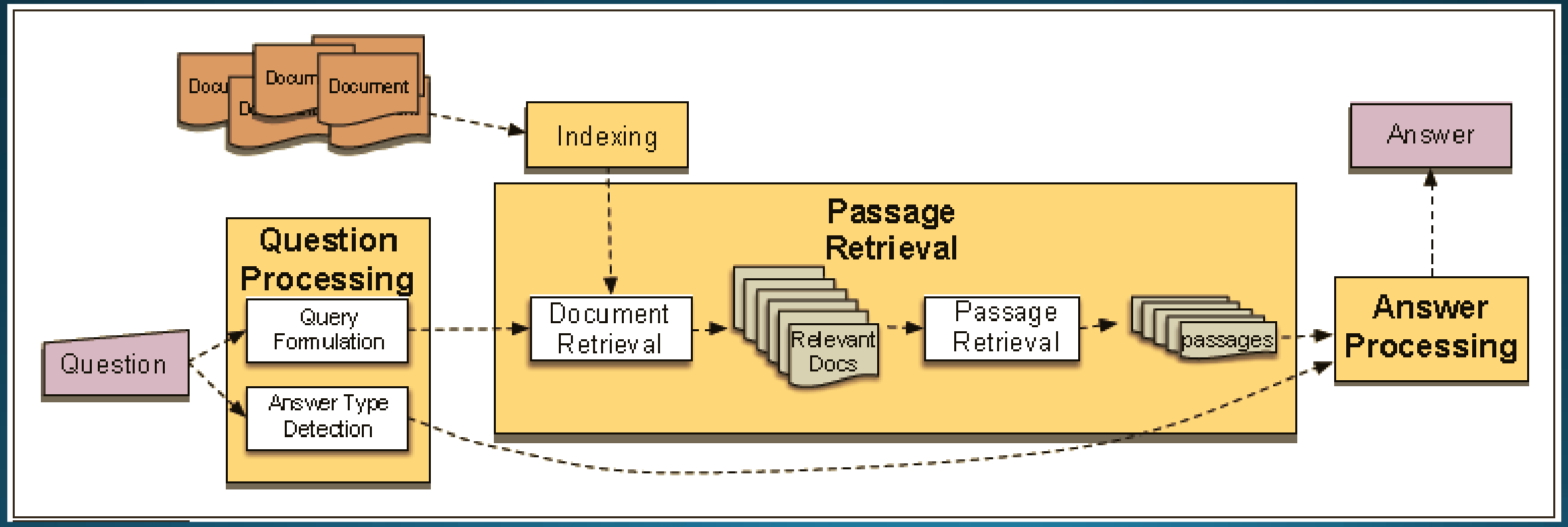
- Knowledge based approach - Apple SIRI
- Information Retrieval based approach - Google
- Hybrid approach – IBM Watson

Knowledge based Question Answering

- Rule based approach using context free grammar
- A **context-free grammar**(CFG) is a set of recursive rewriting rules (or productions) used to match patterns of strings.
- Uses start symbol, terminal and non terminal symbols and a set of productions
- QA is restricted to the grammar
- Requires number of handwritten rules
- Time consuming
- Requires specific/ structured queries to the database

Information Retrieval based Question Answering

- IR based QA uses the indexed documents to search for significant keywords
- Can work on huge datasets
- Very fast and scalable



Information Retrieval based Question Answering

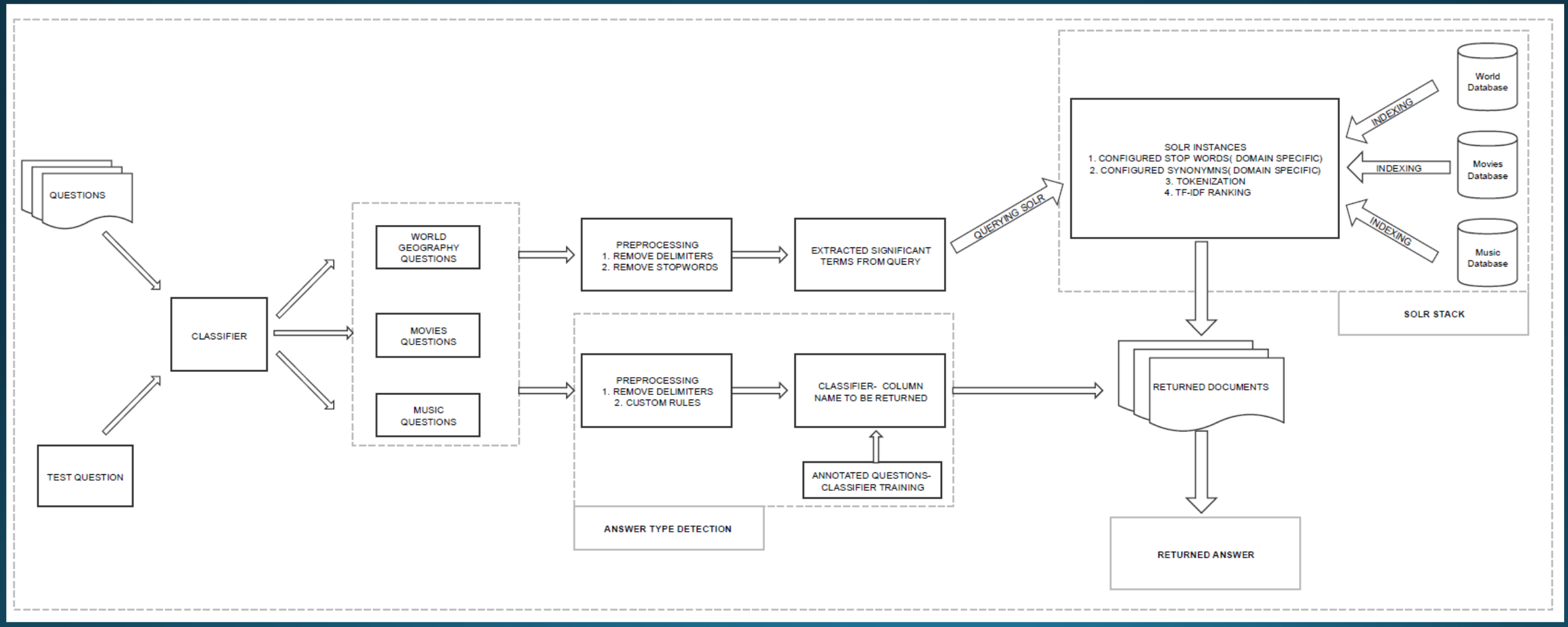


Figure: Mini QA Stack : CS 421 – Natural Language Processing

Apache Solr - Introduction

- Enterprise Search platform built on top of Apache Lucene which is high-performance, full-featured text search engine library
- Open source platform written in Java.
- Highly reliable, scalable, fault tolerant.
- Created by Yonik Seeley for CNET Networks in 2004.
- In January 2006, CNET Networks donated it to the Apache Software Foundation.
- http://lucene.apache.org/solr

Apache Solr - Features

- Exposes REST APIs for interaction
- Full Text Search
- Faceted Search
- More items like this (Recommendation)/ Related searches
- Spell Suggest/Auto-Complete
- Custom document ranking/ordering (TF-IDF Scores)
- And a lot more…

Apache Solr – Basic Concepts

- Indexing:

  Process where a search engine collects, parses and stores data to facilitate fast and accurate Information Retrieval.

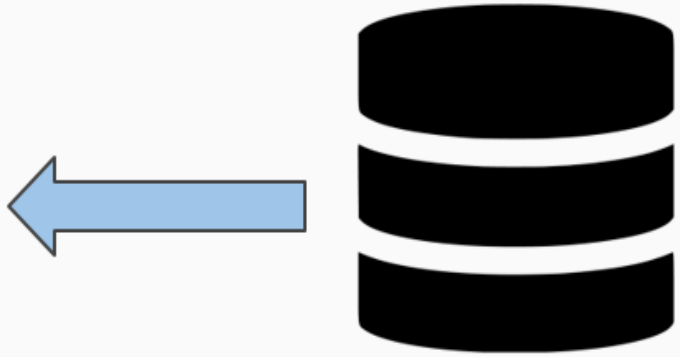
- Inverted Index:

  Search engines store data in the form of inverted indexes for fast querying

- Document:
  - Basic unit of a search index.
  - Composed of multiple fields where each field has its own
  - Field Type like (‘text’, ‘string’, ‘int’ etc.)
  - Set of Attributes like indexed, stored which determine the field’s behavior in the index.

Apache Solr – Basic Concepts

Id	Title	Source
1	The bright blue butterfly hangs on the breeze	A
2	Its best to forget the great sky and to retire from every wind	B
3	Under blue sky, in bright sunlight, one need not search around	A



Solr Index

id: 1
title: The bright blue butterfly hangs on the breeze
source:A

Document 1

Id: 2
Title: It's best to forget the great sky and to retire from every wind
source:B

Document 2

Id: 3
Title: Under blue sky, in bright sunlight, one need not search around
source:A

Document 3

Figure: Document Structure

Apache Solr – Basic Concepts

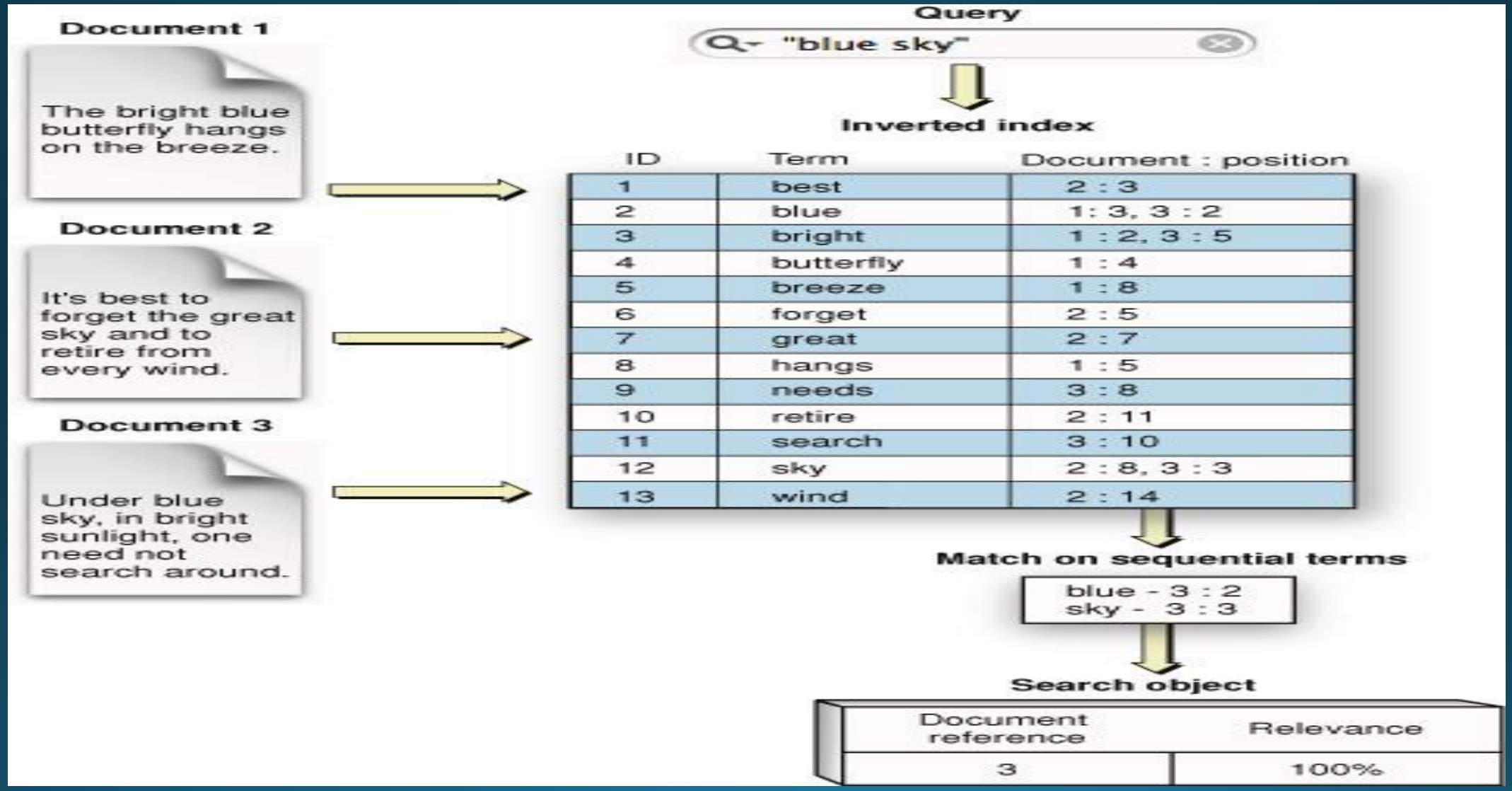


Figure: Inverted Index

IR based Question Answering

- Candidate Documents Retrieval
  - Query preprocessing to remove delimiters, stop words
  - Extracted significant terms from input query
  - Configured Stop-words and Synonyms for each domain in Apache Solr
  - Configured Searchable fields in Apache Solr for each domain.
    - For example: Domain Movies

      Searchable Fields: Movie Name, Actor Name, Director Name, Release Year and so on….
  - Tokenization, Filtering and Parsing of searchable fields.
  - Edismax Query Parser used to fetched the results
  - Set minimum match criteria to 100% to prevent any noisy documents from being retrieved.
  - Boosting to rank documents to get the best result

IR based Factoid Question Answering

- Candidate Documents Retrieval
  - Example Question: "Did Neeson star in Schindler's List?"
  1. Query preprocessing and Stop words removal returns significant terms:
     ['Neeson','star','Schindler','List']
  2. Query Solr using these terms with minimum match criteria set to 100%; i.e. a document should be returned only when all the terms are present in that document.
  3. Solr removes the stop word 'star' which has been pre-configured in its corpora.
  4. Chunk Permutation using Edismax Query does the following:

TERMS	MOVIE NAME	ACTOR NAME	DIRECTOR NAME
Neeson	✗	✓	✗
Schindler	✓	✗	✗
List	✓	✗	✗

IR based Factoid Question Answering

- Yes/No questions:
  - Handwritten set of regular expressions.
  - Return true if Solr returns a document, false otherwise.
- Answer Type Detection
  - Query pre-processing to remove delimiters
  - Head words/ Significant Terms extraction
  - Tried Part of speech tagging using Python's NLTK library
  - Tried Named Entity Recognition using Python's NLTK library
  - Supervised classification for answer type detection worked better
    - Created a questions database and annotated the answer types
    - Used Python's Sklearn library for classifiers.
  - Hand written set of rules

Hurdles and Challenges Faced

- Writing Grammar was a tedious task especially for geography database, as it contains A lot of unrelated tables
- More views so a lot more grammar
- Solr: Extracting Column Name from the question
- Creating training data set to train classifiers to predict column name
- Classifier suffered accuracy loss
- Hand written custom rules on top of classifier to handle edge cases.

Conclusions

- Each approach Knowledge based, IR Factoid based has its own advantages and disadvantages.
- Less noise while using knowledge based and more accuracy
- More noise while using IR Factoid based and relatively less noise.
- IR Factoid based is more scalable and extremely fast.
- IR based QA could be operated on huge databases.
- Doesn't involve writing complicated grammars.
- Set of custom rules could be written which significantly improves the result of IR based QA.

Tools and Libraries Used

- Apache SOLR
- Apache Tomcat server
- SQLite, MySQL
- Supervised Learning Techniques
  - Naïve Bayes' classification for database type detection
  - SVM – Linear for Answer type detection
- Python:
  - NLTK library: query processing, stop words removal
  - Sklearn library: classifiers, TD-IDF Vectorization
  - WX library: for user interface

Thank You

Questions?