Text Processing on the Web

Week 7
Question Answering

The material for these slides are borrowed heavily from the precursor of this course taught by Tat-Seng Chua as well as slides from the accompanying recommended texts Baldi et al. and Manning et al.

Recap: Passage Retrieval and External Resources

- Tuning the performance of IR systems using
  - Query expansion
  - External resources
- Passage Retrieval
  - Can use simple document methods
  - Are a good platform for trying more substantial processing
  - Emphasizing precision; relegate document retrieval to high recall

Three-week Outline

Last Time

- External Resources
  - Thesaurii
  - Wikipedia
  - Domain specific Sites
- Query Expansion
  - Query logs to suggest
- Ranking
  - Density Based
  - Dependency Based

Today

- What is Question Answering?
  - TREC
  - Def, List, Factoid
  - Closed vs. Open Domain
- Question Analysis
  - Question Typologies
- Structural use of terms in QA

Question Answering

- Open Domain
  - Find answers to natural language questions by searching and locating answers in a free (or semi structured) text collection
  - Typically non-interactive
  - A focus of TREC
- Closed Domain
  - QA in a closed domain (e.g., intranet of company)
  - Might simply do routing (classify to closest FAQ)
  - May use ontological knowledge

Text REtrieval Conference

- Annual bakeoff competition of IR systems
- Helped to do large scale testing in a rigorous way
  + Standard corpus and standard answers
  + Technology transfer and visibility
  – All systems start to look the same after a while; less room for innovation

- Structured like our HWs with query relevance assessed by participants or paid volunteers

TREC Tracks

http://trec.nist.gov/tracks.html

- Question Answering
- Blog
- Enterprise – a bit like closed domain
- Genomics – data, but also documentation

Past tracks

- Terabyte – over large datasets
- Novelty – finding interesting new results
- Cross Language – retrieving documents in other languages
- Interactive – user in the loop
- Video – not just text anymore

Question Types in the QA Track

- Factoid – exact answer to a factual question
  - How long is the coastline of England?
- List – listing of items to answer a question
  - Which countries import rice?
- Definition – give a NL definition to a topic
  - Who is Aaron Copeland?

- Topic-Based – Culmination of all three on a particular topic

Example Topic Questions

In 2004 TREC switched to topic style questions

- Topic: Hale Bopp Comet
  - FACTOID: When was the comet discovered
  - FACTOID: How often does it approach the earth?
  - LIST: In what countries was the comet visible on its last return?
  - OTHER: (other relevant info not explicitly asked)

- Topic: James Dean
  - FACTOID: When was James Dean born
  - FACTOID: When did James Dean die
  - FACTOID: How did James Dean die?
  - LIST: What movies did he appear in?
  - FACTOID: Which was the first movie that he was in?
  - OTHER

How are definition questions related to these question types?

Answering Questions

In TREC QA, answers to factoids need to be exact

- Q: Which river is the longest river in the US?
- ~~A: At 2,348 miles the Mississippi River is the longest river in the U.S.~~

- A: Mississippi
- A: Mississippi River

Evaluating QA

Get the correct answer: precision

Get succinct answers: recall

$$P = \frac{\text{\# of matching words}}{\text{\# of words in answer key}}$$

$$R = \frac{\text{\# of matching words}}{\text{\# of words in system response}}$$

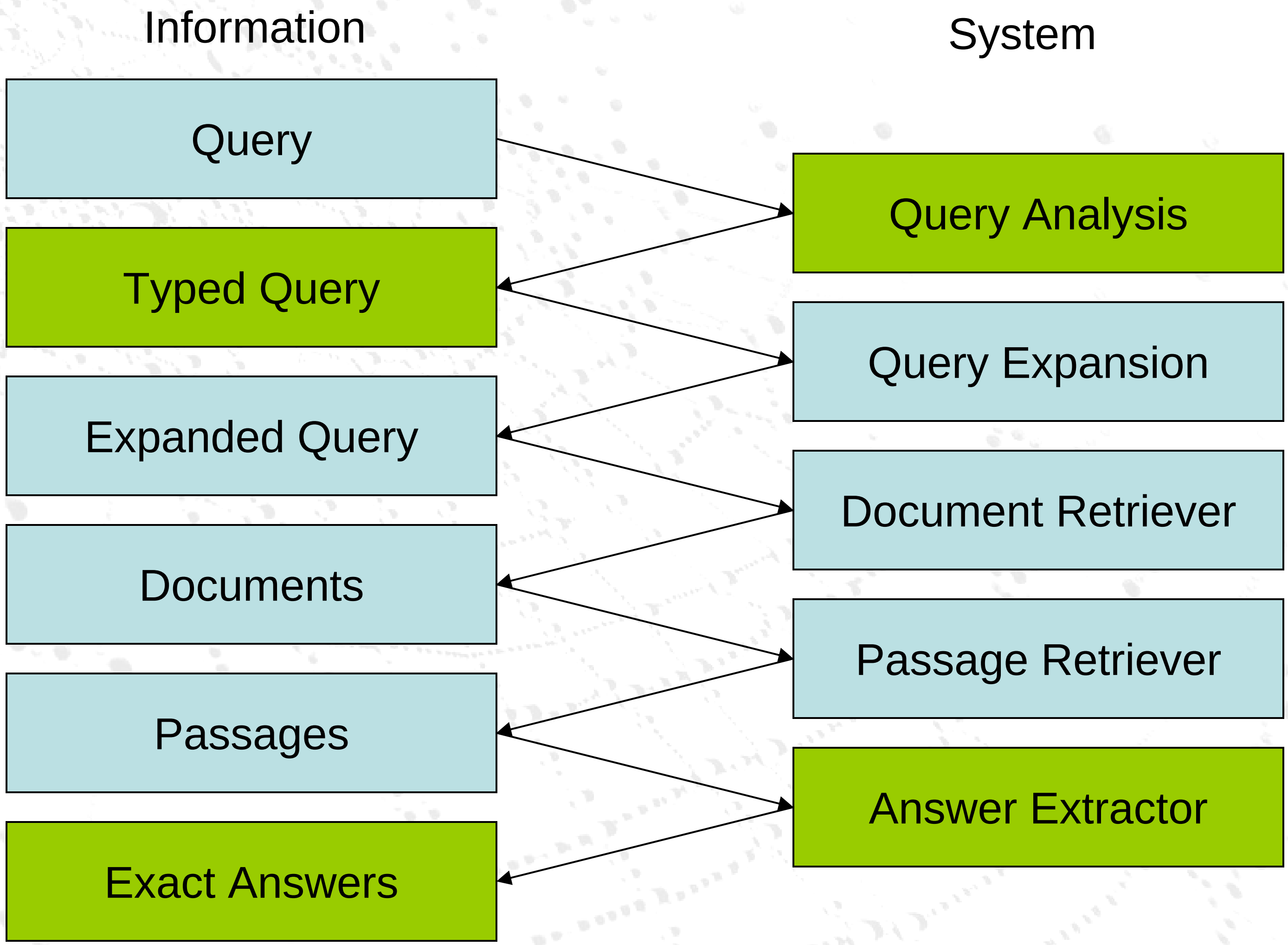
Factoid / List

- DeepRead: only count content words
- Later: TREC 2002 modified this for getting ranking answers

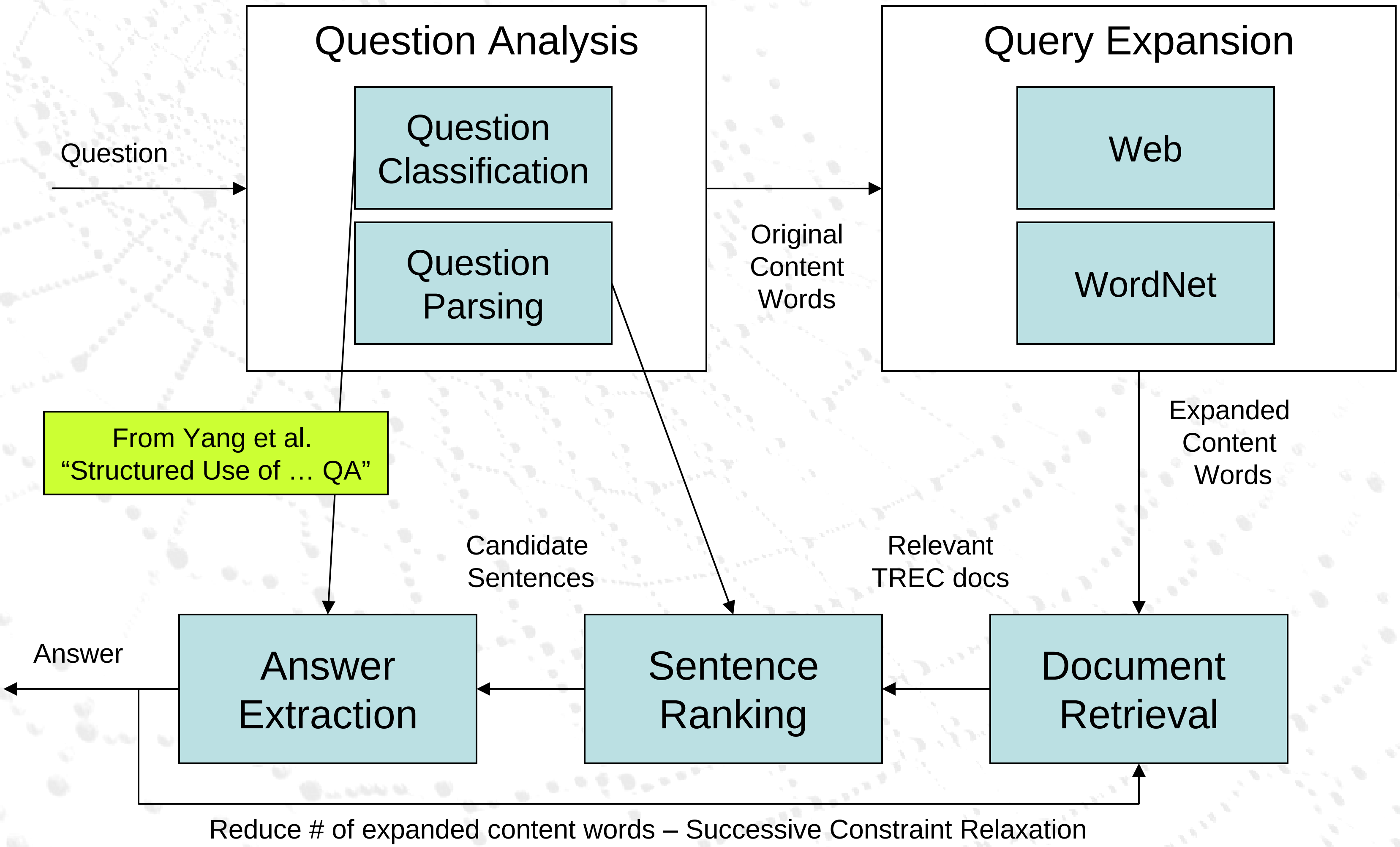
Definition / Other

- Nugget precision/recall/F
  - What's a nugget?
  - Used F_5: recall worth 5x vs. precision (also F_3)
- Vital versus OK nuggets
  - Leads to two different scores

QA Architecture



Typical QA Implementation



Question Classification

- Based on **question focus** and **answer type**
- Divide into main classes
  - HUMan, LOCation, TiME, NUMber, OBJect, DEScription, UNKnown
- Subdivide into detailed classes
  - E.g., under LOC
    LOC_PLANET LOC_TOWN
    LOC_CITY LOC_RIVER
    LOC_CONTINENT LOC_LAKE
    LOC_COUNTRY LOC_MOUNTAIN
    LOC_COUNTY LOC_OCEAN
    LOC_STATE LOC_ISLAND
    LOC_PROVINCE

- Need to ensure accuracy is good. Fortunately, it is very high (> 90%, at least for certain classes)

Question Parsing

- Aim: To extract essential terms in question
  - To extract **answer target** and **content words**
- E.g., for question "What mythical Scottish town appears for one day every 100 years?"
  - $\underline{q}_0$ = (mythical, Scottish, town, appears, one, day, 100, years)
  - Answer target = LOC_TOWN
  - Basic Noun Phrases $\underline{n}$= "mythical Scottish town"
  - Sub-heading words $\underline{h}$= "town"
  - Quotation words $\underline{u}$
    - Present in some questions with titles of works, e.g., What was the original name of the song "The Star Spangled Banner"?

Query Expansion

- Aims to: bring in context to bridge gap between the query and documents
  - Many terms used in queries do not appear in documents or are phrased differently
- For automated Open Domain QA:
  - Make use of external resources to find context
  - Use the Web to extract highly correlated terms with query terms using MI or other co-occurrence metrics
  - Use WordNet to find terms that are lexically related to query using the structure and synsets $\underline{S}_q$ **and gloss $\underline{G}_q$**
  - Combine these three sources to get final ranked list of terms $\underline{I}_q$
  - Use $\underline{I}_q$ for query expansion $q_1 = q_0 + \{$ top m terms from $\underline{I}_q\}$
  - Linear query expansion by varying m to adjust precision of query

Document Retrieval

- Use Lucene to perform Boolean Retrieval
- Retrieve top n = 50 documents using conjunctive syntax
  - If q_1 does not return sufficient documents, remove some extra terms and repeat search
  - This is the successive constraint relaxation, which we will use again later

Passage Retrieval

- Identify sentences from top n = 50 documents
- Density based match to find relevant sentences
- Recall query processing got us $\underline{q}_0$, $\underline{q}_1$, $\underline{n}$, $\underline{h}$, $\underline{u}$

  Score all sentences in top n S_j are scored by

  +1 if match quotation words $\underline{u}$

  +1 if match noun phrases $\underline{n}$

  +1 if match sub heading words $\underline{h}$

  +[0-1] % of terms overlapping between q_0 and S_j

  +[0-1] % of terms matching expanded ($q_1 - q_0$) query terms

  \+ based on other criteria (dependency relation score)
- Select top k sentences based on scores of S_j

Answer Extraction

- Perform NE tagging from top k sentences
  - NE tagging task reviewed later in the course
- For each sentence, extract string matching question target

"What mythical Scottish town appears for one day every 100 years?"

Answer target: LOC_TOWN

Top ranked sentence "Isolated in the rugged heartland of the <LOC_MOUNTAIN Green Mountains>, <LOC_MOUNTAIN Plymouth Notch> has been called <LOC_STATE Vermont>'s <LOC_TOWN Brigadoon>, after the imaginary <LOC_COUNTRY Scottish> village that appears and vanishes in the mists."

The extracted answer is **Brigadoon**

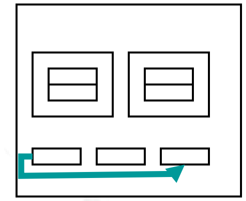
"Who is Tom Cruise married to?"

Answer target: HUM_**BASIC**

Top ranked Candidate Sentence: "Actor <HUM_PERSON Tom Cruise> and his wife <HUM_PERSON Nicole Kidman> accepted "substantial" libel damages on <TME_DATE Thursday> from a <LOC_COUNTRY British> newspaper that reported he was gay and that their marriage was a sham to cover it up."

The extracted answer is **Nicole Kidman**

If no answers… try again



- Perform successive constrain relaxation (SCR):
  - We reduce the number of expanded query terms in q_1 and repeat the document/sentence retrieval and answer extraction
  - Try up to m=5 iterations (afterwards, conclude really no answer *nil*)

- This strategy increases recall
- Question: What does it do to precision?

Further Readings

- The Voorhees paper gives an overview of the TREC QA tasks.
- Read the Hirschman et al. article to get an understanding of a complete, early QA system
- Then read the Yang et al. paper to see how it works in a more modern system.
- (Supplemental) the Moldovan and Novischi article describes a more principled way to use WordNet (definitions + ontology data) to relate to synonymous words together

Leveraging structure in QA

Two parts

Looking at the structure of terms to boost QA

- Structured Queries
  (from Yang et al. 2002)
  - In query expansion
  - No linguistic knowledge
  - Distance between terms

- Soft Patterns
  (from Cui et al. 2004)
  - For filtering in passage retrieval
  - Part of speech tagging
  - Order between terms

Could be applied elsewhere (e.g., NLP, IE, IR), not just QA

Structured Queries

Goal: know which terms in (expanded) query belong together

Such semantic groups should correspond to set of elements in a QA event

Given any two distinct terms t_i and t_j, we compute 3 correlations

- Lexical:
  - Use WordNet (gloss and hierarchy)
  - Give bonus if t_i and t_j related (e.g., in same synset)
- Co-occurrence
  - Find mutual information between t_i and t_j
  - Give bonus if $MI(t_i,t_j)$ greater than average
- Distance:
  - Density based methods (i.e., find how close in the snippets or documents t_i and t_j occur)
  - Give bonus proportional to reciprocal of avg. distance between t_i and t_j

Note: Co-occurrence and distance correlations **overlap** (not independent)

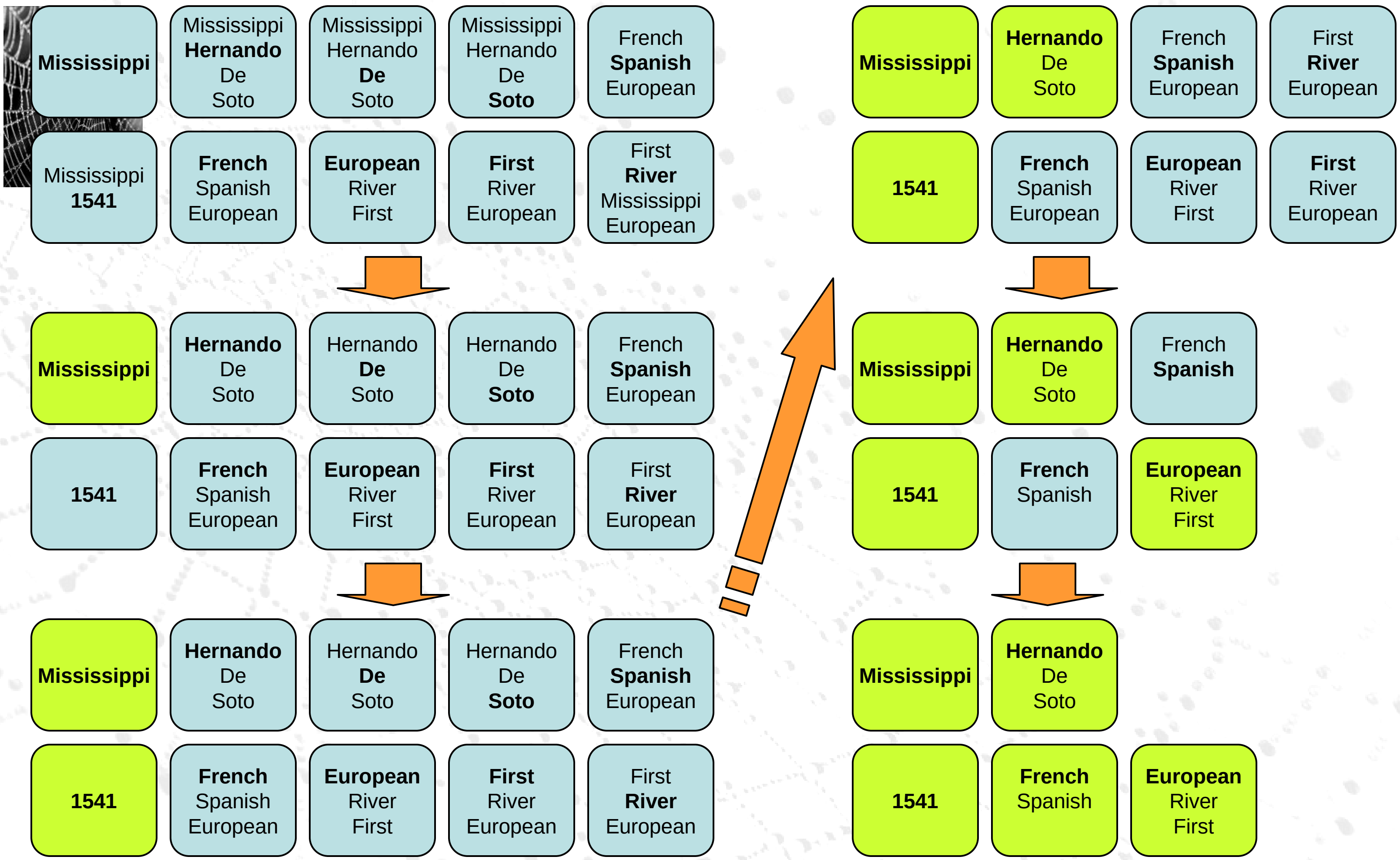
Calculating semantic groups

- Initially, soft cluster terms in (expanded) query by their distance
  - Use fixed thresholds in some number of initial clusters G.
  - Each cluster in G represented by highest IDF word, called main(G_i)
- Then iterate to obtain hard clusters
  - Select cluster G_s to be added to final cluster set E based on having highest weighted main(G_i)
  - Remove any overlapping words in other G_k that overlap with words in G_s
- For each final cluster in E, decide its **group cohesiveness** based on the correlations.
  - If cluster is tight, use an AND syntax (words are part of the same concept)
  - If cluster is loose, use an OR syntax to connect words (words are synonymous)

Example

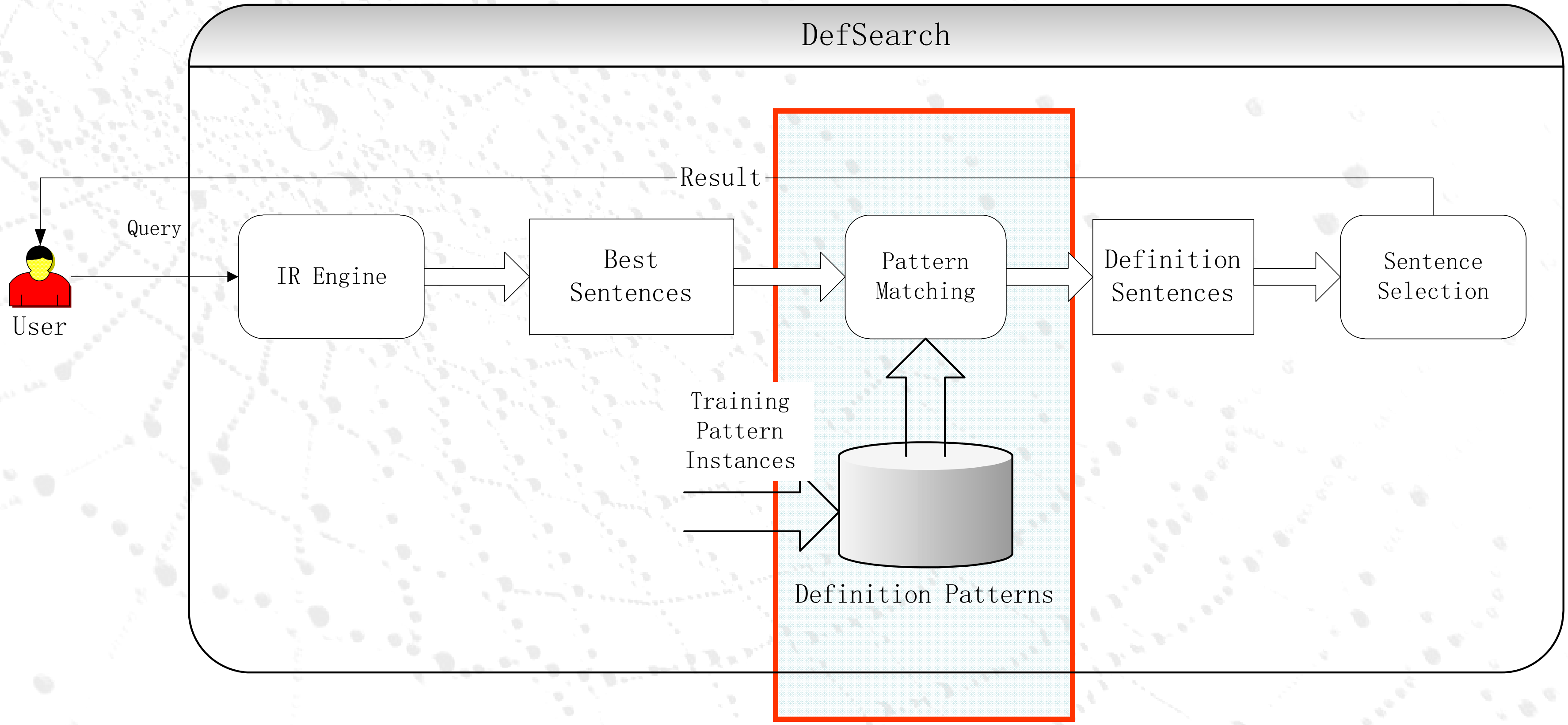
- For question: “What Spanish explorer discovered the Mississippi River?”
- Expanded Query= {Mississippi, Hernando, Soto, De, Spanish, 1541, French, European, First, river}
- Initial clusters (main word **bolded**)

Mississippi	Mississippi **Hernando** De Soto	Mississippi Hernando **De** Soto	Mississippi Hernando De **Soto**	French **Spanish** European
Mississippi **1541**	**French** Spanish European	**European** River First	**First** River European	First **River** Mississippi European



- Final Boolean query is: Mississippi & 1541 & (Hernando & De & Soto) & (first | European | River) & (French | Spanish)

Definition QA architecture



How Do Current Systems Identify Definitions?

- Current systems use hand-crafted patterns
  - Appositive
    - e.g. Gunter Blobel , a cellular and molecular biologist,…
  - Copulas
    - e.g. Battery is a kind of electronic device …
  - Predicates (relations)
    - e.g. TB is usually caused by …

Weaknesses of Current Pattern Matching Methods

- Lack of Flexibility – Hard Matching
  - Pattern : <SCH_TERM> , also known as

    ~~*TB , also known as Tuberculosis*~~ *, …*

    ~~*TB (also known as Tuberculosis) …*~~

    mismatch
  - Variations make hard matching fail
  - Introduce **Soft Patterns** with greater flexibility
- Manual labor
  - Introduce unsupervised learning by Group Pseudo-Relevance Feedback (GPRF).

What are Soft Patterns?

- Soft patterns allow partial matching

  TB (also known as Tuberculosis) …

  P((|Slot1) = 0.001, P(also|Slot2) = 0.21, P(known|Slot3) = 0.33, P(as|Slot4) = 0.13

  P(Matching) = 0.23 : still better than non-definition sentences.

- How does it work?
  - Training – accumulating pattern instances in a vector.
    - Derive pattern instances from labeled definition sentences.
  - Matching with a probabilistic model, not regular expressions.
    - Using statistical information from all pattern instances, not generalized rules.
    - Instance-based learning.

Preparing Pattern Instances

*The channel **Iqra** is owned by the Arab Radio and Television company and is the brainc*[illegible]*ire, Saleh Kamel.*

Step 1
POS tagging and noun phrase chunking.

The_DT channel_NN Iqra_NNP is_VBZ owned_VBN by_IN NNP company_NN and_CC is_VBZ the_DT brainchild_NN of_IN NNP.

Step 2
Selective substitution – replace those specific words with more general tags.
Other tokens remain unchanged.

DT$ NN <SEARCH_TERM> BE$ owned by DT$ NNP and BE$ DT$ NN of NNP.

Preparing Pattern Instances – Cont’d

DT$ NN <SCH_TERM> BE$ owned by

Pattern Instance

Step 3

Crop a text window around the tag “<SCH_TERM>” (window size = 3 for each side)

Example Pattern Generation

…… The channel **Iqra** is owned by the …
…… severance packages, known as **golden parachutes**, included ……
A **battery** is a cell which can provide electricity.

DT$ NN <Search_Term> BE$ owned by
known as <Search_Term> , VB
<Search_Term> BE$ DT$

…… <Slot$_{-2}$>	<Slot$_{-1}$>	<Search_Term>	<Slot$_{1}$>	<Slot$_{2}$> ……
NN 0.12	NN 0.11		, 0.40	DT$ 0.2
known 0.09	as 0.20		BE$ 0.2	VB 0.1
DT$ 0.04				owned 0.09

*<Slot$_{-w}$, ……, Slot$_{-2}$, Slot$_{-1}$, SEARCH_TERM , Slot$_{1}$, Slot$_{2}$, …… Slot$_{w}$: **Pa**>*

Matching Soft Patterns

- Test sentences are reduced to a vector *S* using the same strategy.

$$<token_{-w}, \ldots, token_{-1}, SEARCH\_TERM, token_{1}, \ldots, token_{w} : S>$$

- Matching Soft Patterns – similarity between the pattern vector *Pa* and the test vector *S*.
  - Independent slot content similarity.
  - Slot sequence fidelity.

Probabilistic Matching Degree

- Individual slot similarity – independent assumption

$$Pa\_weight_{Slots} = \Pr(S \mid Pa) = \prod_{i=-w}^{w} \Pr(token_i \mid Slot_i)$$

- Sequence fidelity – bigram model

$$\begin{aligned}\Pr(right\_seq) &= \Pr(token_1, token_2 \cdots token_w \mid Pa)\\ &= P(token_1)P(token_2 \mid token_1)\cdots P(token_w \mid token_{w-1})\end{aligned}$$

$$\begin{aligned}Pa\_weight_{Seq} &= (1-\alpha)\cdot\Pr(left\_seq \mid Pa)\\ &+ \alpha\cdot\Pr(right\_seq \mid Pa)\end{aligned}$$

- Combined to get the matching degree

$$Pattern\_weight = \frac{Pa\_weight_{Slots} \times Pa\_weight_{Seq}}{fragment\_length}$$

Unsupervised Labeling of Definition Sentences using GPRF

- Pattern instances obtained from labeled definition sentences.
  - Manual labeling is too expensive.
- Pseudo-relevance Feedback in document retrieval
  - Take the top n ranked documents as relevant.
- Employ Group pseudo-relevance feedback (GPRF)
  - Statistical ranking – centroid based method.
  - Perform PRF over a group of questions (top 10 sentences for each question).
  - Generate soft patterns from all auto-labeled sentences for all questions.

Analysis of GPRF

- Assumption 1 – some definition sentences can be ranked high using statistical method.
  - Word co-occurrence metrics can well model descriptive sentences.
    - Over 33% of top ranked sentences are definitional.
  - Noise introduced in each question’s top list is mitigated by our group strategy
- Assumption 2 – definition patterns are general and can be used across questions.

Summary

- Question Answering as exact answer retrieval
  - Different types of QA
  - Definitional QA as summarization (keep this in mind **next week**)
- Less volume of information allows more intensive statistical NLP to be applied
  - Pre-process: question typing
  - Post-process: answer extraction
  - Successive Constraint Relaxation to expand queried to find less exact answers.
- Use structure
  - Associating terms into groups (keep in mind for clustering later)
  - Soft patterns for capturing context in an unsupervised way using PRF