自然语言处理

week-10

凌震华

2024年5月16日

- Question-Answering 问答
- Information Retrieval 信息检索
- Summarization 摘要

Question-Answering

- The notion of getting computers to give reasonable answers to questions

- Three kinds of systems

1) Finding answers in text collections
2) Interfaces to relational databases
3) Mixed initiative (互动式) dialog systems

People *do* ask questions…

Examples from various query logs

Which english translation of the bible is used in official Catholic liturgies?

How tall is the sears tower?

How can i find someone in texas

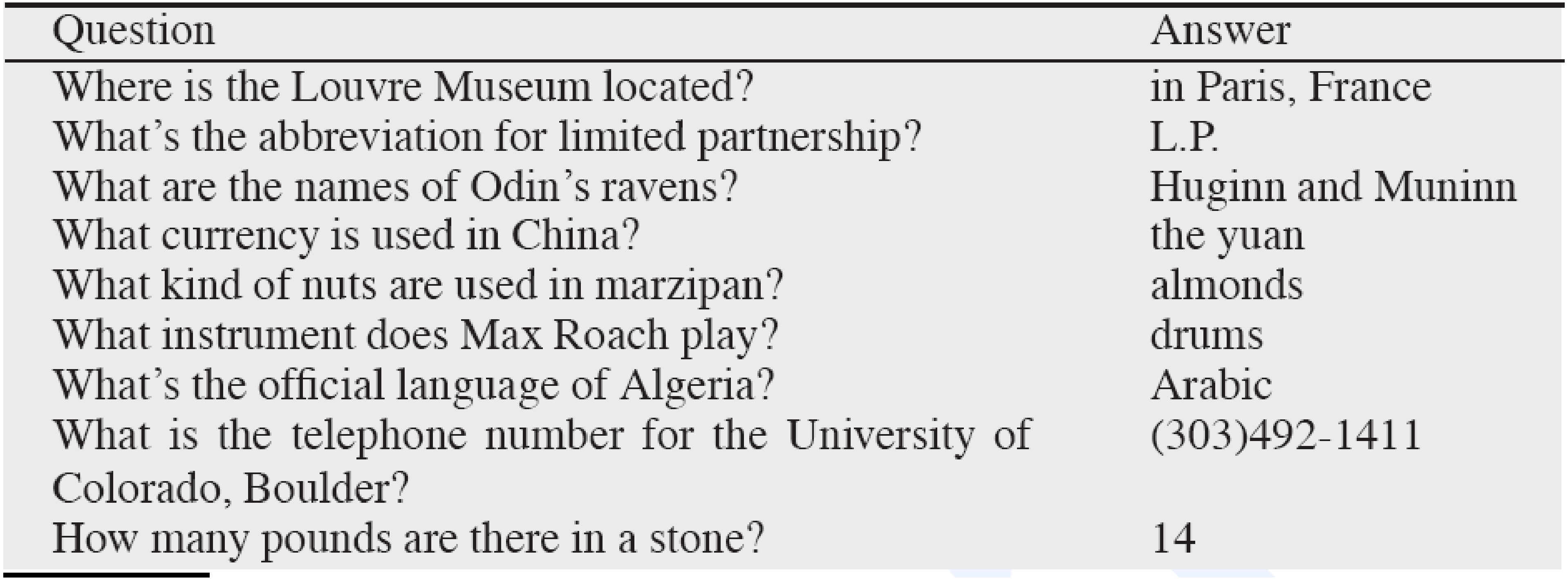
Where can i find information on puritan religion?

What are the 7 wonders of the world

How can i eliminate stress

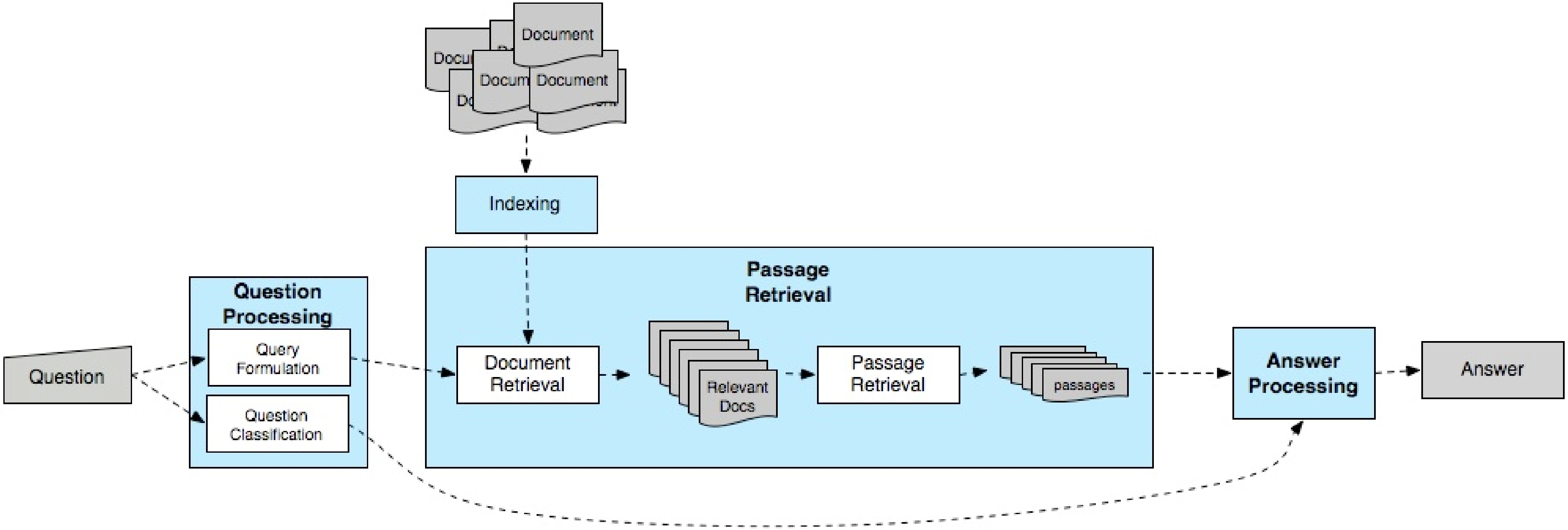
What vacuum cleaner does Consumers Guide recommend

Factoid (事实性) questions



Question	Answer
Where is the Louvre Museum located?	in Paris, France
What’s the abbreviation for limited partnership?	L.P.
What are the names of Odin’s ravens?	Huginn and Muninn
What currency is used in China?	the yuan
What kind of nuts are used in marzipan?	almonds
What instrument does Max Roach play?	drums
What’s the official language of Algeria?	Arabic
What is the telephone number for the University of Colorado, Boulder?	(303)492-1411
How many pounds are there in a stone?	14

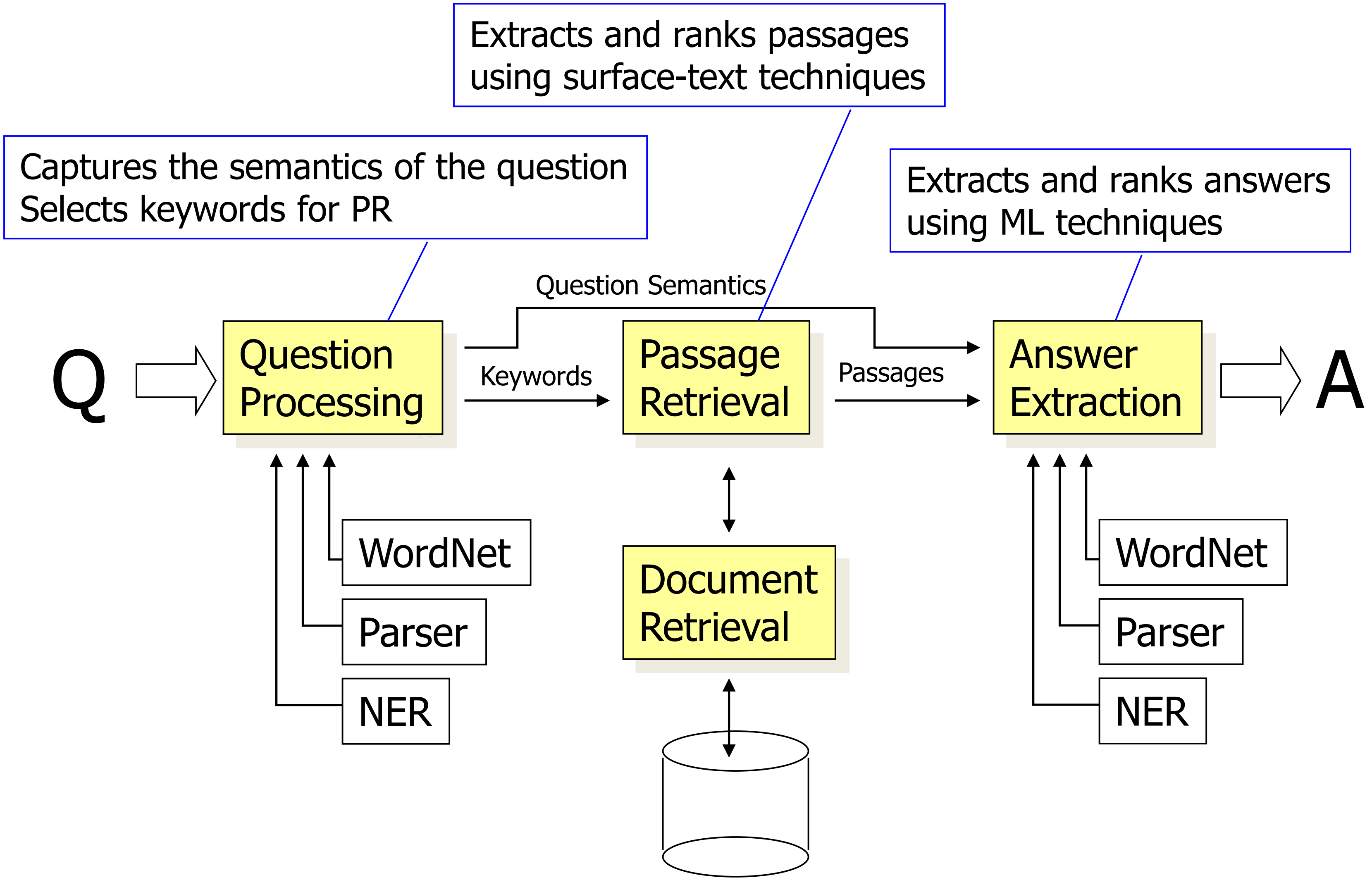
Factoid QA Architecture


Document
Document
Document
Document
Indexing
Passage Retrieval
Question
Question Processing
Query Formulation
Question Classification
Document Retrieval
Relevant Docs
Passage Retrieval
passages
Answer Processing
Answer

UT Dallas Q/A Systems

- This system contains many components used by other systems, but more complex in some ways
- Most work completed in 2001; there have been advances by this group and others since then.
- Next slides based mainly on:
  - Paşca and Harabagiu, *High-Performance Question Answering from Large Text Collections*, SIGIR' 01.
  - Paşca and Harabagiu, *Answer Mining from Online Documents,* ACL' 01.
  - Harabagiu, Paşca, Maiorano: *Experiments with Open-Domain Textual Question Answering.* COLING' 00

QA Block Architecture



Extracts and ranks passages using surface-text techniques

Captures the semantics of the question
Selects keywords for PR

Extracts and ranks answers using ML techniques

Q

Question Processing

Question Semantics

Keywords

Passage Retrieval

Passages

Answer Extraction

A

WordNet

Parser

NER

Document Retrieval

WordNet

Parser

NER

Question Processing

- Two main tasks
  - **Question classification**: Determining the type of the answer
  - **Query formulation**: Extract keywords from the question and formulate a query

Answer Types

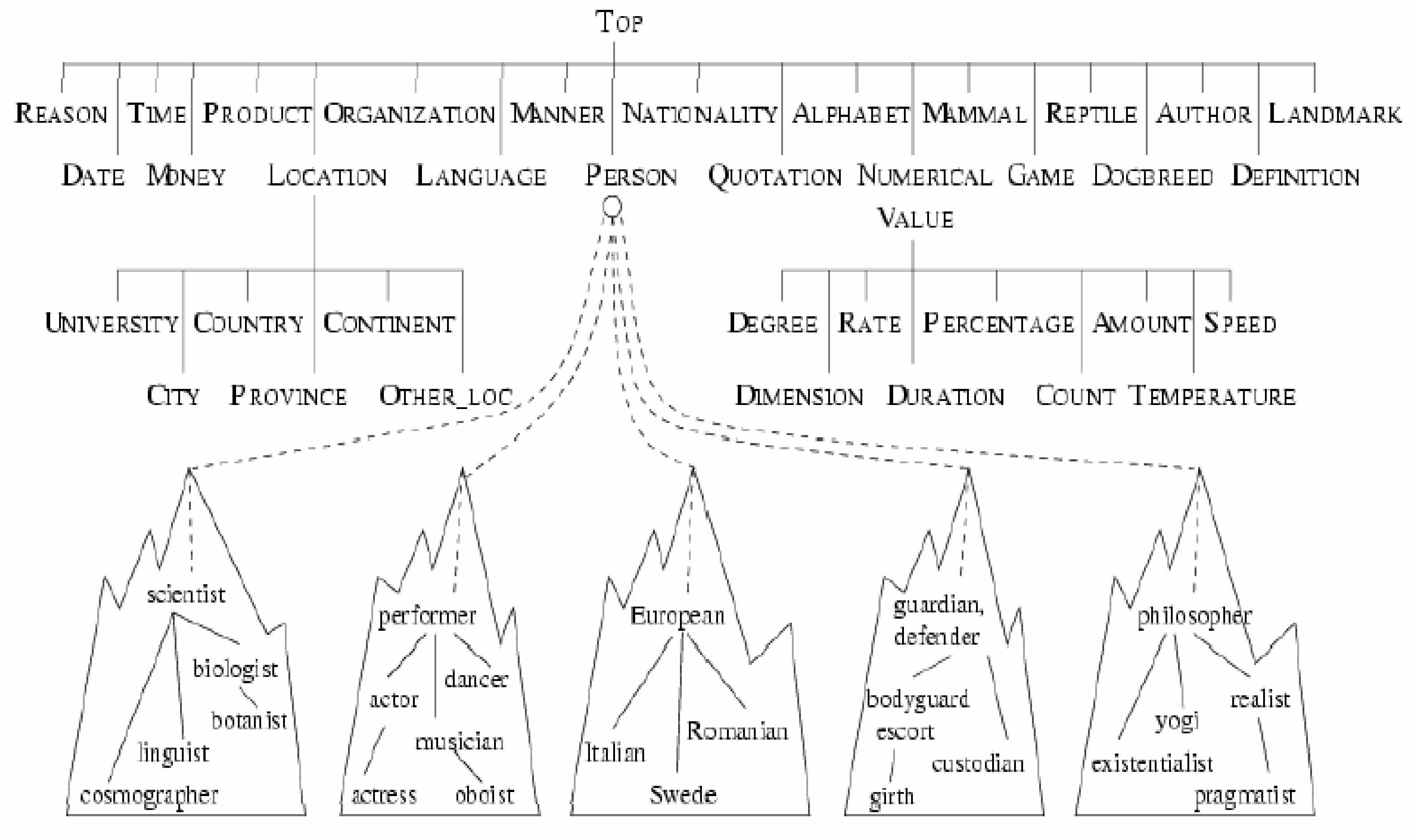
- Factoid questions…
  - Who, where, when, how many…
  - The answers fall into a limited and somewhat predictable set of categories
    - Who questions are going to be answered by…
    - Where questions…
  - Generally, systems select answer types from a set of Named Entities, augmented with other types that are relatively easy to extract

Answer Types

- Of course, it isn't that easy…
  - Who questions can have organizations as answers
    - Who sells the most hybrid cars?
  - Which questions can have people as answers
    - Which president went to war with Mexico?

Answer Type Taxonomy (分类)

- Contains ~9000 concepts reflecting expected answer types
- Merges named entities with the WordNet hierarchy


TOP
REASON
TIME
PRODUCT
ORGANIZATION
MANNER
NATIONALITY
ALPHABET
MAMMAL
REPTILE
AUTHOR
LANDMARK
DATE
MONEY
LOCATION
LANGUAGE
PERSON
QUOTATION
NUMERICAL VALUE
GAME
DOGBREED
DEFINITION
UNIVERSITY
COUNTRY
CONTINENT
CITY
PROVINCE
OTHER_LOC
DEGREE
RATE
PERCENTAGE
AMOUNT
SPEED
DIMENSION
DURATION
COUNT
TEMPERATURE
scientist
biologist
botanist
linguist
cosmographer
performer
dancer
actor
musician
actress
oboist
European
Romanian
Italian
Swede
guardian, defender
bodyguard
escort
custodian
girth
philosopher
realist
yogi
existentialist
pragmatist

Answer Type Detection

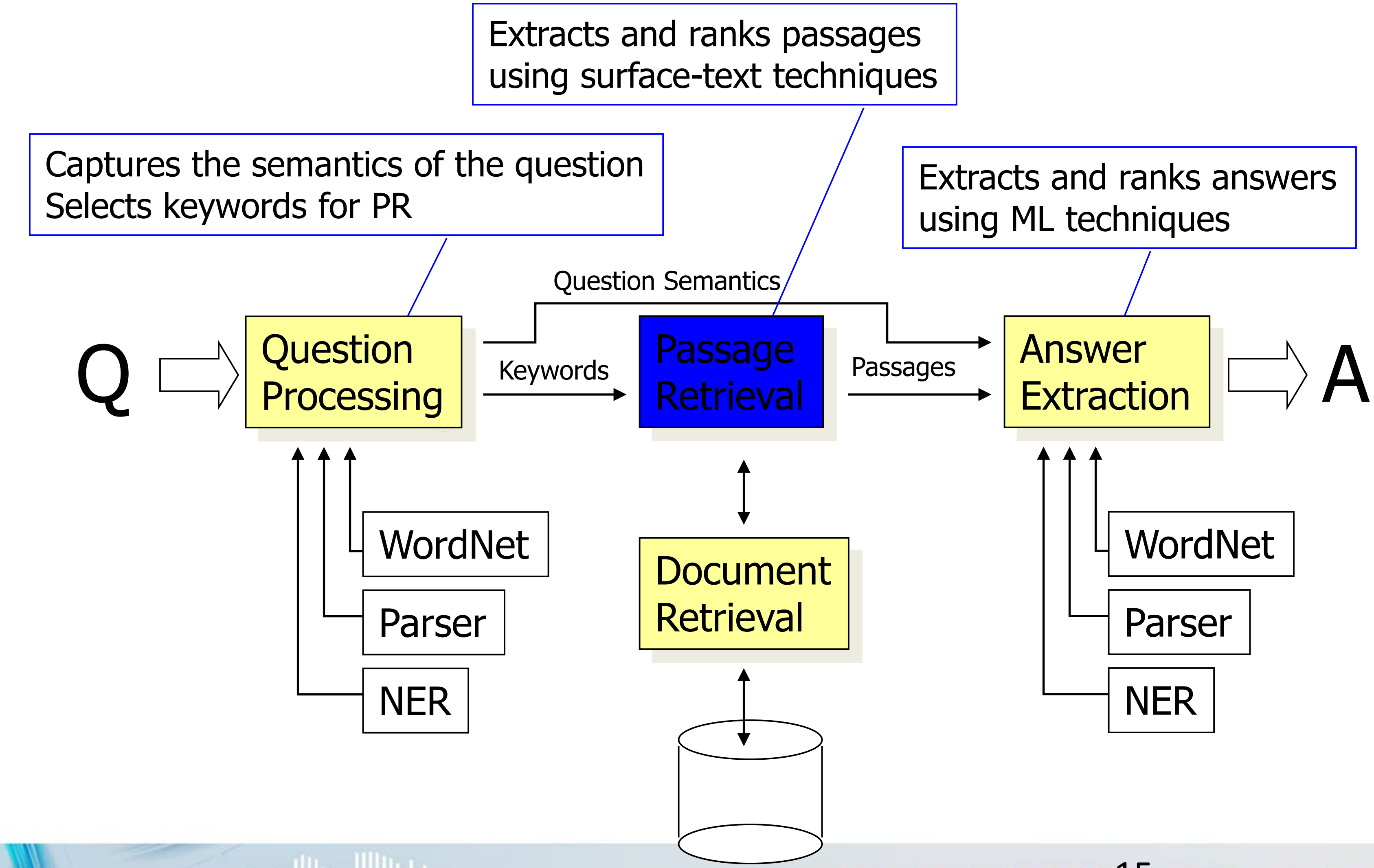
- Most systems use a combination of hand-crafted rules and supervised machine learning to determine the right answer type for a question.
- But how do we use the answer type?

Query Formulation: Lexical Terms Extraction

- Questions approximated by sets of unrelated words (lexical terms)
- Similar to bag-of-word IR models

Question (from TREC QA track)	**Lexical terms**
Q002: What was the monetary value of the Nobel Peace Prize in 1989?	**monetary, value, Nobel, Peace, Prize**
Q003: What does the Peugeot company manufacture?	**Peugeot, company, manufacture**
Q004: How much did Mercury spend on advertising in 1993?	**Mercury, spend, advertising, 1993**
Q005: What is the name of the managing director of Apricot Computer?	**name, managing, director, Apricot, Computer**

Passage Retrieval



Extracts and ranks passages using surface-text techniques

Captures the semantics of the question
Selects keywords for PR

Extracts and ranks answers using ML techniques

Q
Question Processing
Question Semantics
Keywords
Passage Retrieval
Passages
Answer Extraction
A
WordNet
Parser
NER
Document Retrieval
WordNet
Parser
NER

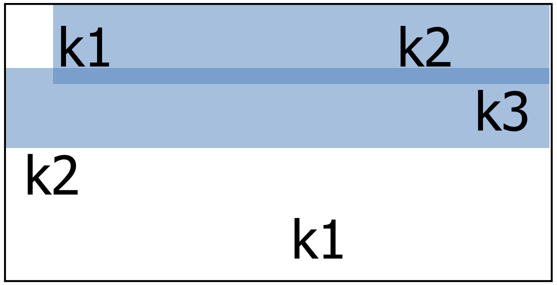
Passage Extraction Loop

- Passage Extraction Component
  - Extracts passages that contain all selected keywords
  - Passage size dynamic
  - Start position dynamic
- Passage quality and keyword adjustment
  - In the first iteration use the first 6 keyword selection heuristics
  - If the number of passages is lower than a threshold ⇒ query is too strict ⇒ drop a keyword
  - If the number of passages is higher than a threshold ⇒ query is too relaxed ⇒ add a keyword

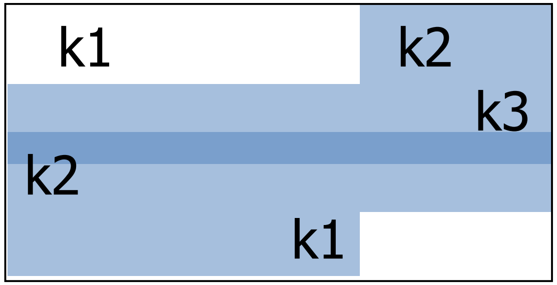
Passage Scoring

- Passages are scored based on keyword windows
  - For example, if a question has a set of keywords: {k1, k2, k3, k4}, and in a passage k1 and k2 are matched twice, k3 is matched once, and k4 is not matched, the following windows are built:

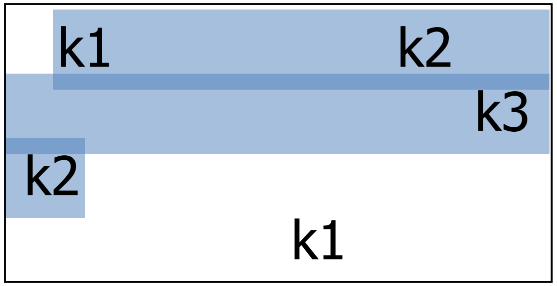
Window 1


k1
k2
k3
k2
k1

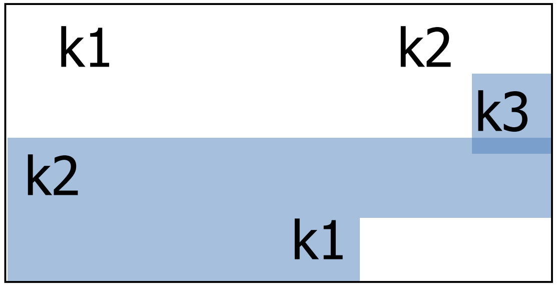
Window 2


k1
k2
k3
k2
k1

Window 3


k1
k2
k3
k2
k1

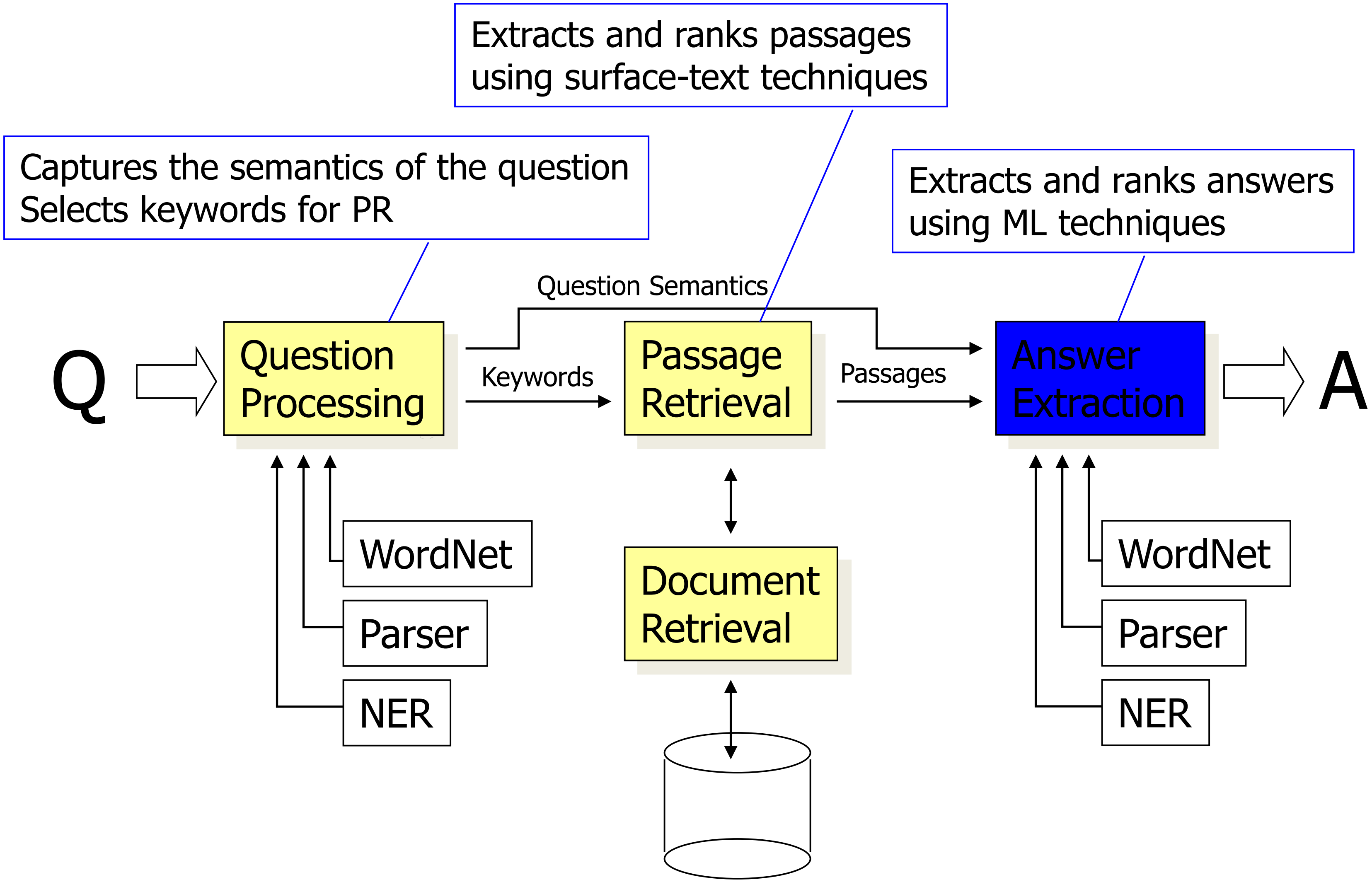
Window 4


k1
k2
k3
k2
k1

Passage Scoring

- Passage ordering is performed using a sort that involves three scores:
  - The number of words from the question that are recognized in the same sequence in the window
  - The number of words that separate the most distant keywords in the window
  - The number of unmatched keywords in the window

Answer Extraction



Extracts and ranks passages using surface-text techniques

Captures the semantics of the question
Selects keywords for PR

Extracts and ranks answers using ML techniques

Q
Question Processing
Keywords
Question Semantics
Passage Retrieval
Passages
Answer Extraction
A
WordNet
Parser
NER
Document Retrieval
WordNet
Parser
NER

Ranking Candidate Answers

Q066: Name the first private citizen to fly in space.

- Answer type: Person
- Text passage:

  "Among them was Christa McAuliffe, the first private citizen to fly in space. Karen Allen, best known for her starring role in "Raiders of the Lost Ark", plays McAuliffe. Brian Kerwin is featured as shuttle pilot Mike_Smith..."

Ranking Candidate Answers

Q066: Name the first private citizen to fly in space.

- Answer type: Person
- Text passage:

  "Among them was Christa McAuliffe, the first private citizen to fly in space. Karen Allen, best known for her starring role in "Raiders of the Lost Ark", plays McAuliffe. Brian Kerwin is featured as shuttle pilot Mike_Smith..."

- Best candidate answer: Christa McAuliffe

Features for Answer Ranking

- Number of question terms matched in the answer passage
- Number of question terms matched in the same phrase as the candidate answer
- Number of question terms matched in the same sentence as the candidate answer
- Flag set to 1 if the candidate answer is followed by a punctuation sign
- Number of question terms matched, separated from the candidate answer by at most three words and one comma
- Number of terms occurring in the same order in the answer passage as in the question
- Average distance from candidate answer to question term matches

SIGIR '01

Where to find the answer?

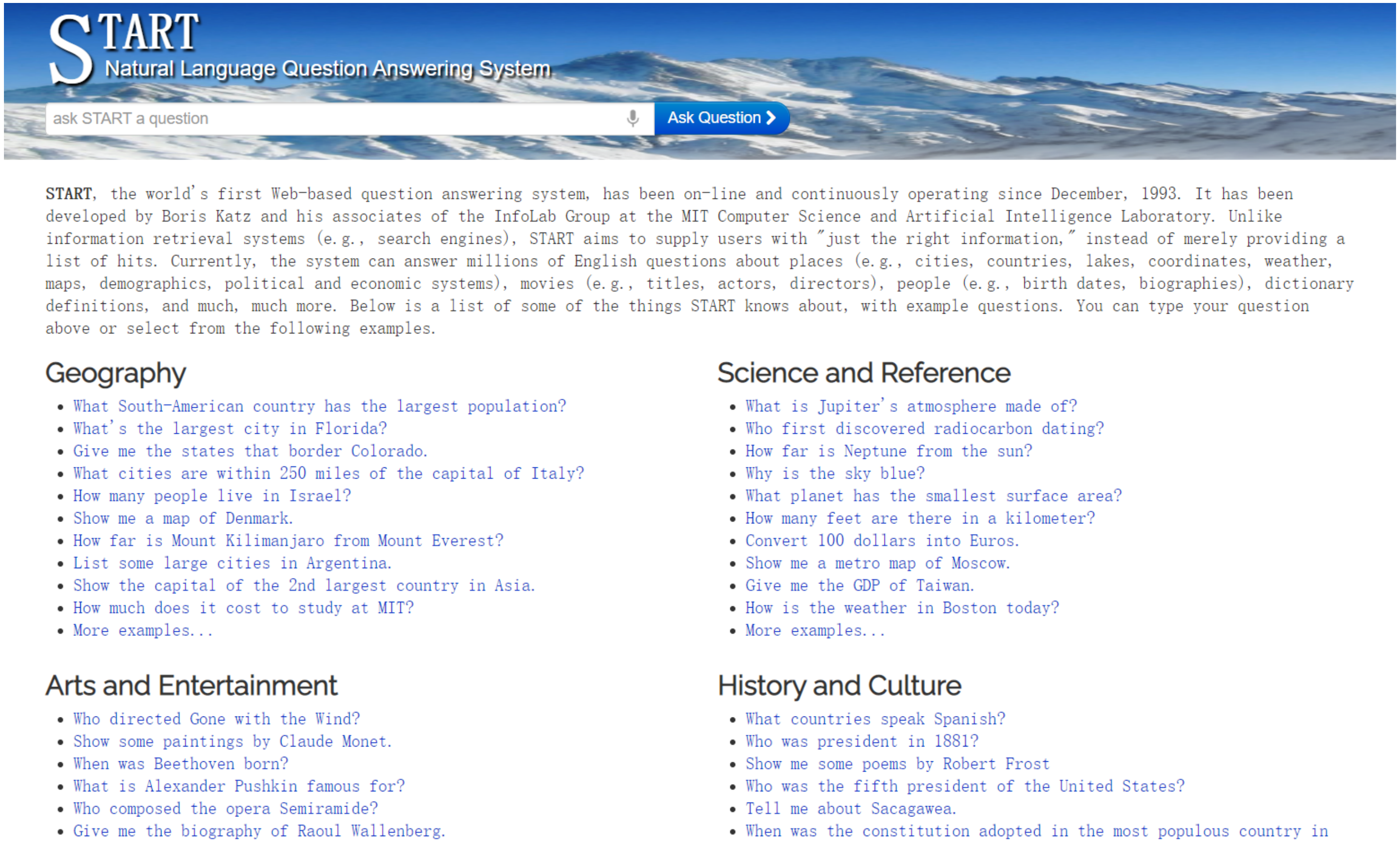
- Wikipedia, WordNet often more reliable

- Wikipedia:
  - Q: What is the Milky Way?
    - Candidate 1: outer regions
    - Candidate 2: the galaxy that contains the Earth

- WordNet
    - Wordnet: Milky Way—the galaxy containing the solar system

An Online QA System

- http://start.csail.mit.edu/index.php



START, the world's first Web-based question answering system, has been on-line and continuously operating since December, 1993. It has been developed by Boris Katz and his associates of the InfoLab Group at the MIT Computer Science and Artificial Intelligence Laboratory. Unlike information retrieval systems (e.g., search engines), START aims to supply users with "just the right information," instead of merely providing a list of hits. Currently, the system can answer millions of English questions about places (e.g., cities, countries, lakes, coordinates, weather, maps, demographics, political and economic systems), movies (e.g., titles, actors, directors), people (e.g., birth dates, biographies), dictionary definitions, and much, much more. Below is a list of some of the things START knows about, with example questions. You can type your question above or select from the following examples.

Geography

- What South-American country has the largest population?
- What's the largest city in Florida?
- Give me the states that border Colorado.
- What cities are within 250 miles of the capital of Italy?
- How many people live in Israel?
- Show me a map of Denmark.
- How far is Mount Kilimanjaro from Mount Everest?
- List some large cities in Argentina.
- Show the capital of the 2nd largest country in Asia.
- How much does it cost to study at MIT?
- More examples...

Arts and Entertainment

- Who directed Gone with the Wind?
- Show some paintings by Claude Monet.
- When was Beethoven born?
- What is Alexander Pushkin famous for?
- Who composed the opera Semiramide?
- Give me the biography of Raoul Wallenberg.

Science and Reference

- What is Jupiter's atmosphere made of?
- Who first discovered radiocarbon dating?
- How far is Neptune from the sun?
- Why is the sky blue?
- What planet has the smallest surface area?
- How many feet are there in a kilometer?
- Convert 100 dollars into Euros.
- Show me a metro map of Moscow.
- Give me the GDP of Taiwan.
- How is the weather in Boston today?
- More examples...

History and Culture

- What countries speak Spanish?
- Who was president in 1881?
- Show me some poems by Robert Frost
- Who was the fifth president of the United States?
- Tell me about Sagagawea.
- When was the constitution adopted in the most populous country in

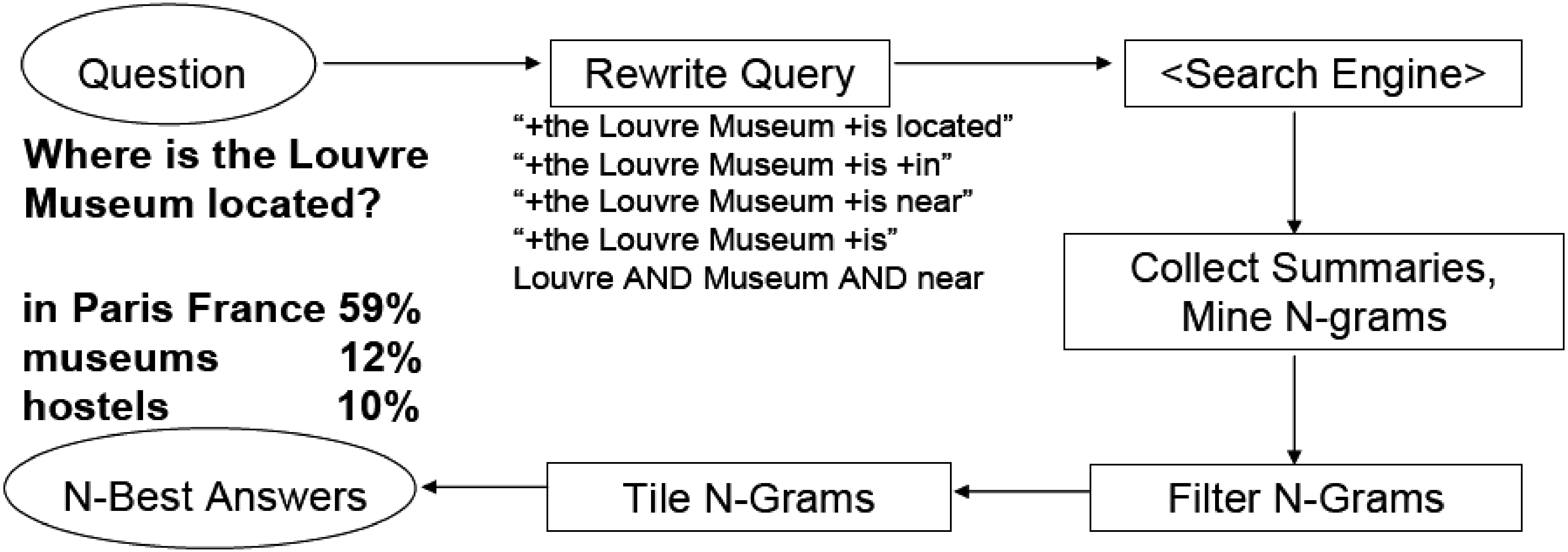
The Web is Different

- On the Web popular factoids are likely to be expressed in many different ways.
- At least a few of which will likely match the way the question was asked.
- So why not just grep (or agrep) the Web using all or pieces of the original question.

AskMSR

- Process the question by…
  - – Simple rewrite rules to rewriting the original question into a statement
    - Involves detecting the answer type
- Get some results
- Extract answers of the right type based on
  - – How often they occur

AskMSR



Question

Rewrite Query

<Search Engine>

Collect Summaries, Mine N-grams

Filter N-Grams

Tile N-Grams

N-Best Answers

Where is the Louvre Museum located?

“+the Louvre Museum +is located”
“+the Louvre Museum +is +in”
“+the Louvre Museum +is near”
“+the Louvre Museum +is”
Louvre AND Museum AND near

in Paris France	**59%**
museums	**12%**
hostels	**10%**

Step 1: Rewrite the questions

- Intuition: The user's question is often syntactically quite close to sentences that contain the answer
  - Where is the Louvre Museum located?
    - The Louvre Museum is located in ***Paris***
  - Who created the character of Scrooge?
    - ***Charles Dickens*** created the character of Scrooge.

Query rewriting

Classify question into seven categories

- **Who** is/was/are/were…?
- **When** is/did/will/are/were …?
- **Where** is/are/were …?

a. Hand-crafted category-specific transformation rules

e.g.: For *where* questions, move 'is' to all possible locations

Look to the right of the query terms for the answer.

"Where is the Louvre Museum located?"

→ "is the Louvre Museum located"

→ "the is Louvre Museum located"

→ "the Louvre is Museum located"

→ "the Louvre Museum is located"

→ "the Louvre Museum located is"

Step 2: Query search engine

- Send all rewrites to a Web search engine
- Retrieve top N answers (100-200)
- For speed, rely just on search engine's "snippets"（摘要), not the full text of the actual document

Step 3: Gathering N-Grams

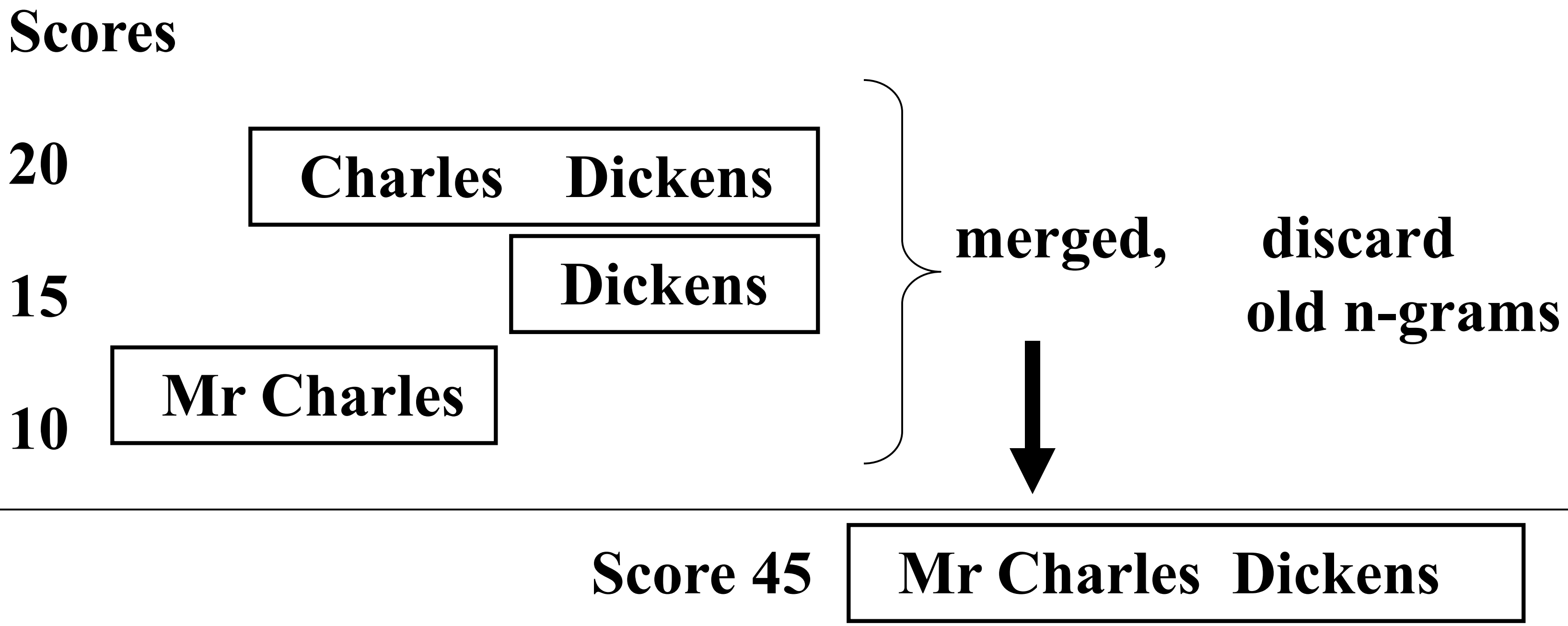
- Enumerate all N-grams (N=1,2,3) in all retrieved snippets
- Weight of an n-gram: occurrence count, each weighted by "reliability" (weight) of rewrite rule that fetched the document
  - Example: "Who created the character of Scrooge?"

Dickens	117
Christmas Carol	78
Charles Dickens	75
Disney	72
Carl Banks	54
A Christmas	41
Christmas Carol	45
Uncle	31

Step 4: Filtering N-Grams

- Each question type is associated with one or more "data-type filters" = regular expressions for answer types
- Boost score of n-grams that match the expected answer type.
- Lower score of n-grams that don't match.
- For example
  - The filter for
    - How many dogs pull a sled in the Iditarod?
  - prefers a number
  - So disprefer candidate n-grams like
    - Dog race, run, Alaskan, dog racing
  - Prefer canddiate n-grams like
    - Pool of 16 dogs

Step 5: Tiling the Answers



Scores

20 Charles Dickens

15 Dickens

10 Mr Charles

merged, discard old n-grams

Score 45 Mr Charles Dickens

Evaluation

- Evaluation of this kind of system is usually based on some kind of TREC-like metric.
- In Q/A the most frequent metric is
  - Mean Reciprocal Rank (平均排序倒数)

    You're allowed to return N answers. Your score is based on 1/Rank of the first right answer.

    Averaged over all the questions you answer.

Results

- Standard TREC contest test-bed (TREC 2001): 1M documents; 900 questions
  - Technique does ok, not great (would have placed in top 9 of ~30 participants)
    - MRR = 0.507
  - But with access to the Web… They do much better, would have come in second on TREC 2001

Harder Questions

- A more interesting task is one where the answers are fluid (不固定的) and depend on the fusion of material from disparate (不同的) texts over time.
  - Who is Condoleezza Rice?
  - Who is Stephen Harper?
  - Why did San Francisco have to hand-count ballots in the last election?

Harder Questions: Query-Based Summarization

- Much of the work has focused on getting the answer from multiple documents
  - Do web search and use snippets (each as a document)
  - Do question-answering from many documents and merge together the answers you get from multiple sources
    - Like multi-document summarization – you want new information and want to avoid redundant information
  - Use "templates" for each type of question – e.g., definition, biography, medicine
    - Use information extraction techniques to find answer.

□Question-Answering 问答

□Information Retrieval 信息检索

□Summarization 摘要

Information Retrieval

- Basic assumption: meanings of documents can be captured by analyzing (counting) the words that occur in them.

- This is known as the bag of words 词袋 approach.

Inverted Index 倒排索引

- The fundamental operation we need is the ability to map from words to documents in a collection that contain those words
- An inverted index is just a list of words along with the document IDs of the documents that contain them
  - `Dog: 1,2,8,100,119,210,400`
  - `Dog: 1:4,7:11,13:15,17`

Stop Lists and Stemming

- IR systems use them

- Stop List
  - List of frequent largely content-free words that are not stored in the index (of, the, a, etc)
  - The primary benefit is in the reduction of the size of the inverted index

- Stemming
  - Are dog and dogs separate entries or are they collapsed to dog?

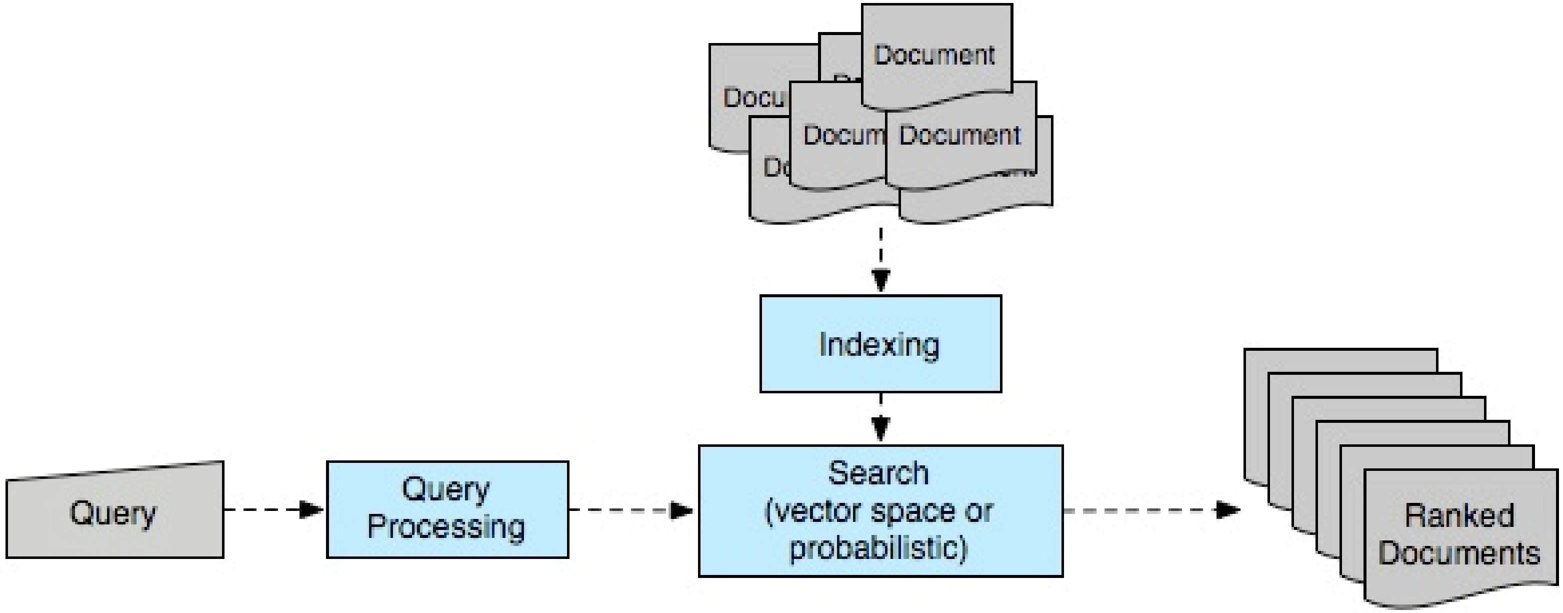
Phrases

- Google et al allow users to perform phrasal searches "big red dog".
  - Hint: they don't grep the collection
  - Add locational information to the index
    - `dog: 1{104}, 2{10}, etc`
    - `red: 1{103},…`
    - `big: 1{102},…`
  - Phrasal searches can operate incrementally by piecing the phrases together.

Ranked Retrieval

- The inverted index is just the start
- Given a query we want to know how relevant all the documents in the collection are to that query

Ad hoc retrieval 特定型检索


Document
Document
Docu
Docum
Dc
Indexing
Query
Query Processing
Search (vector space or probabilistic)
Ranked Documents

Vector Space Model 向量空间模型

- In the vector space model, both documents and queries are represented as vectors of numbers.
- The numbers are derived from the words that occur in the collection

Representation

- Start with bit vectors $\overrightarrow{d_j} = (t_1, t_2, t_3, \ldots t_N)$

- This says that there are N word types in the collection and that the representation of a document consists of a 1 for each corresponding word type that occurs in the document.
- We can compare two docs or a query and a doc by summing the bits they have in common

$$sim(\vec{q}_k, \vec{d}_j) = \sum_{i=1}^{N} t_{i,k} \times t_{i,j}$$

Term Weighting

- Bit vector idea treats all terms that occur in the query and the document equally.

- Its better to give the more important terms greater weight.
    - Why?
    - How would we decide what is more important?

Term Weighting

- Two measures are used
  - Local weight
    - How important is this term to the meaning of this document
    - Usually based on the frequency of the term in the document
  - Global weight
    - How well does this term discriminate among the documents in the collection
    - The more documents a term occurs in the less important it is; The fewer the better.

Term Weights

- Local weights
  - Generally, some function of the frequency of terms in documents is used
- Global weights
  - The standard technique is known as inverse document frequency

$$idf_i = \log\left(\frac{N}{n_i}\right)$$

N= number of documents; ni = number of documents with term i

TF-IDF Weighting

- To get the weight for a term in a document, multiply the term's frequency derived weight by its inverse document frequency.

Back to Similarity

- We were counting bits to get similarity

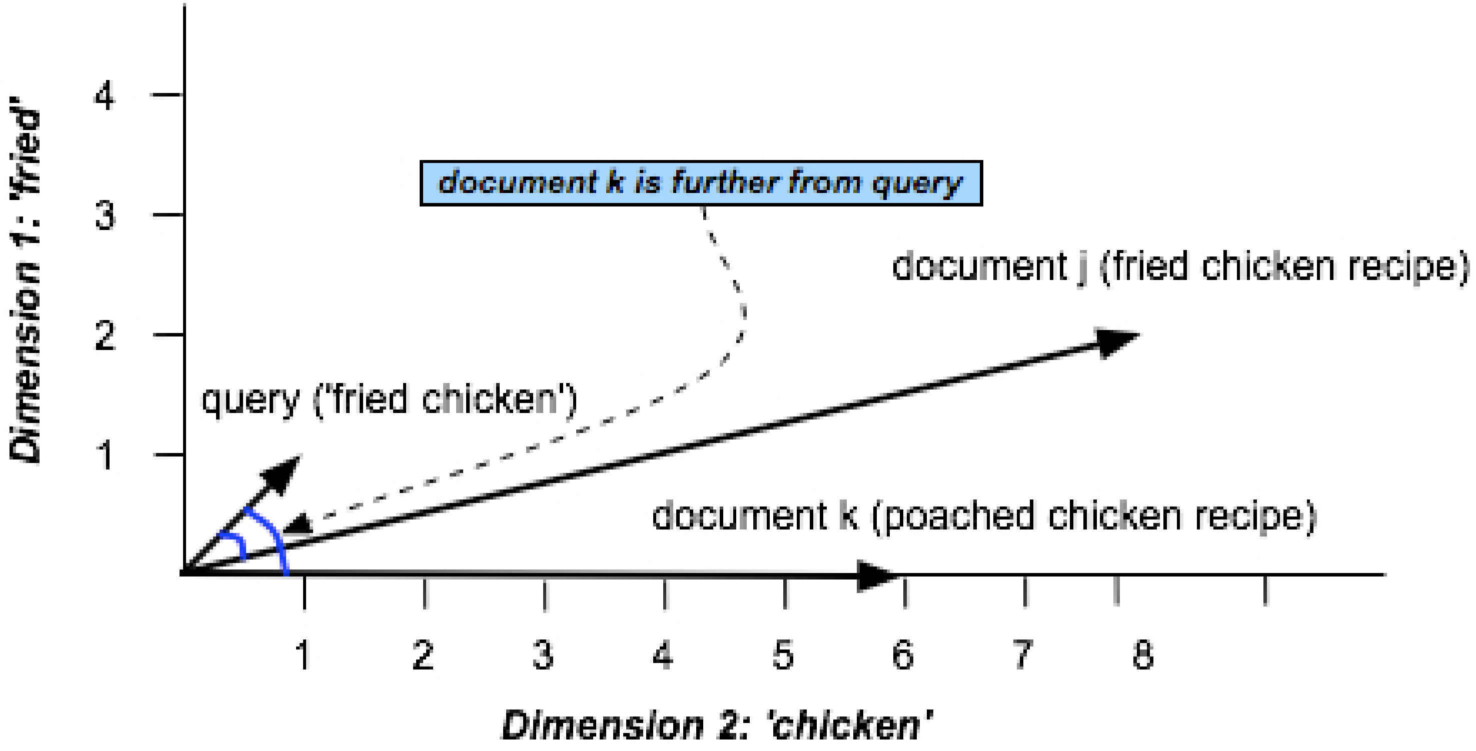
$$sim(\vec{q}_k, \vec{d}_j) = \sum_{i=1}^{N} t_{i,k} \times t_{i,j}$$

- Now we have weights

$$sim(\vec{q}_k, \vec{d}_j) = \sum_{i=1}^{N} w_{i,k} \times w_{i,j}$$

- But that favors long documents over shorter ones

Similarity in Space (Vector Space Model)


document k is further from query
document j (fried chicken recipe)
query ("fried chicken")
document k (poached chicken recipe)
Dimension 1: 'fried'
1
2
3
4
Dimension 2: 'chicken'
1
2
3
4
5
6
7
8

Similarity

- View the document as a vector from the origin to a point in the space, rather than as the point.
- In this view it's the direction the vector is pointing that matters rather than the exact position
- We can capture this by normalizing the comparison to factor out the length of the vectors

Cosine Similarity

$$sim(qk, dj) = \frac{\sum_{i=1}^{N} w_{i,k} \times w_{i,j}}{\sqrt{\sum_{i=1}^{N} w^{2}_{i,k}} \times \sqrt{\sum_{i=1}^{N} w^{2}_{i,j}}}$$

Ad Hoc Retrieval

1. Take a user's query and find all the documents that contain any of the terms in the query
2. Convert the query to a vector using the same weighting scheme that was used to represent the documents
3. Compute the cosine between the query vector and all the candidate documents and sort

□Question-Answering 问答

□Information Retrieval 信息检索

□Summarization 摘要

What is Summarization?

- Data as input (database, software trace, expert system), text summary as output
- Text as input (one or more articles), paragraph summary as output
- Multimedia in input or output
- Summaries must convey maximal information in minimal space

Types of Summaries

- Informative 报道性 vs. Indicative 指示性
  - Replacing a document vs. describing the contents of a document
- Extractive 摘抄 vs. Generative 生成 (abstractive 摘要)
  - Choosing bits of the source vs. generating something new
- Single document vs. Multi Document
- Generic vs. user-focused

Types of Summaries

- Informative vs. Indicative
  - Replacing a document vs. describing the contents of a document
- Extractive vs. Generative
  - Choosing bits of the source vs. generating something new
- Single document vs. Multi Document
- Generic vs user-focused

Single-Document Summarization Stages

1. **Content Selection:** Choose units (sentences?) to extract from the document

2. **Information Ordering:** Choose an order in which to place these sentences in the summary

3. **Sentence Realization:** Clean-up the sentences, e.g., by removing non-essential phrases, by fusing multiple sentences, by fixing problems of coherence

Foundations of Summarization – Luhn; Edmunson

- Text as input

- Single document

- Content selection

- Methods
  - Sentence selection
  - Criteria

Sentence extraction

- Sparck Jones:

‘what you see is what you get’，some of what is on view in the source text is transferred to constitute the summary

Luhn 58

- Summarization as sentence extraction
- Term frequency determines sentence importance
    - TF*IDF
    - Stop word filtering
    - Similar words count as one
    - Cluster of frequent words indicates a good sentence

TF*IDF

- Intuition: Important terms are those that are frequent in this document but not frequent across all documents

Term Weights

- Local weights
  - Generally, some function of the frequency of terms in documents is used
- Global weights
  - The standard technique is known as inverse document frequency

$$idf_i = \log\left(\frac{N}{n_i}\right)$$

TFxIDF Weighting

- To get the weight for a term in a document, multiply the term's frequency derived weight by its inverse document frequency.

TF*IDF

Edmunson 69

Sentence extraction using 4 weighted features:

- Cue words (“In this paper..” , “The worst thing was ..”)
- Title and heading words
- Sentence location
- Frequent key words

Sentence extraction variants

- Lexical Chains
  - Barzilay and Elhadad
  - Silber and McCoy

- Discourse coherence
  - Baldwin

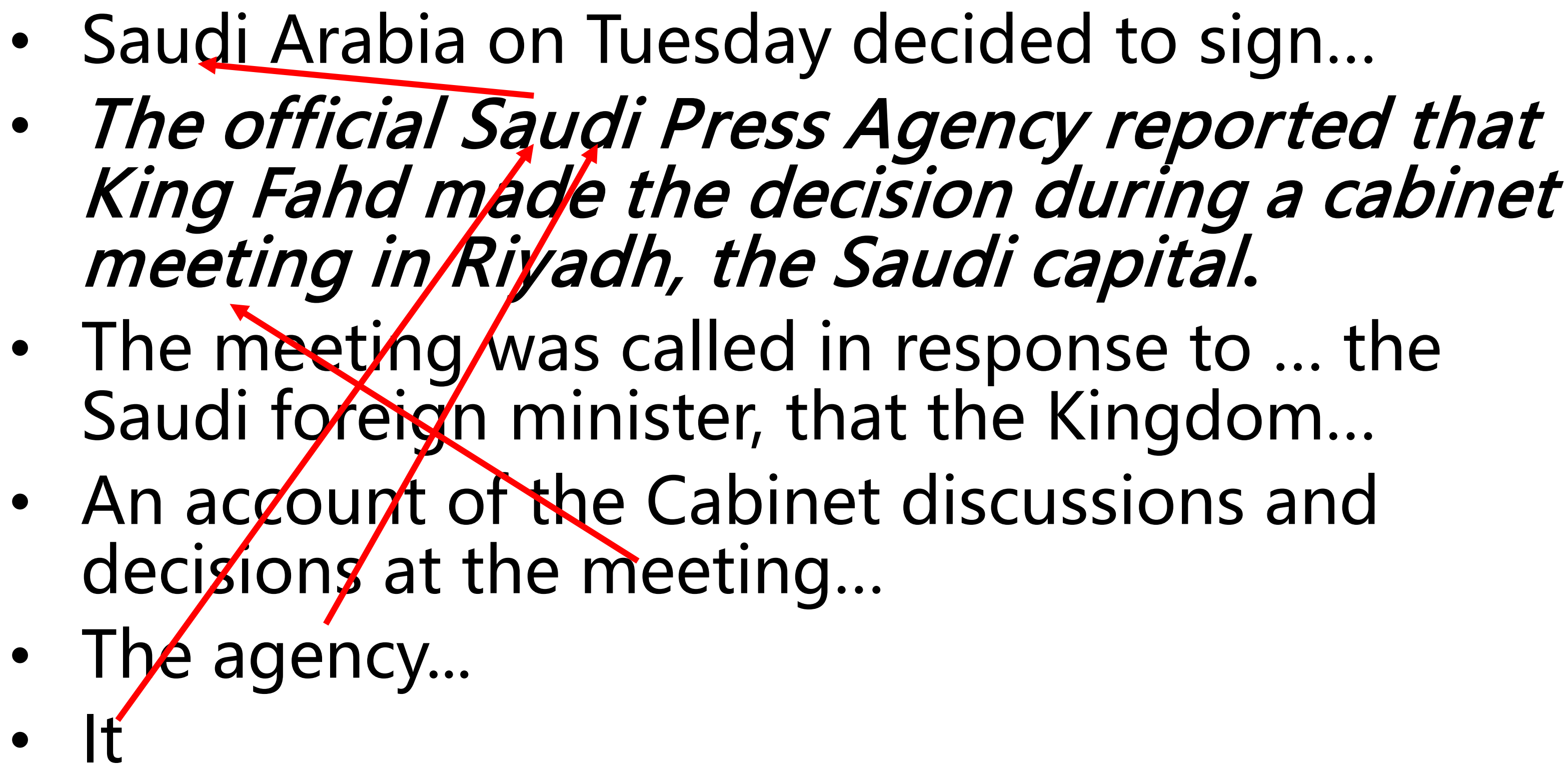
- Topic signatures
  - Lin and Hovy

- Lets see examples in next few slides

Lexical Chains

- “Dr.Kenny has invented an anesthetic machine. This device controls the rate at which an anesthetic is pumped into the blood. “
- “Dr.Kenny has invented an anesthetic machine. The doctor spent two years on this research. “
- Algorithm: Measure strength of a chain by its length and its homogeneity
  - Select the first sentence from each strong chain until length limit reached

Discourse Coherence



- Saudi Arabia on Tuesday decided to sign…
- *The official Saudi Press Agency reported that King Fahd made the decision during a cabinet meeting in Riyadh, the Saudi capital.*
- The meeting was called in response to … the Saudi foreign minister, that the Kingdom…
- An account of the Cabinet discussions and decisions at the meeting…
- The agency...
- It

Topic Signature Words

- Uses the log ratio test to find words that are highly descriptive of the input
- the log-likelihood ratio test provides a way of setting a threshold to divide all words in the input into either descriptive or not
  - the probability of a word in the input is the same as in the background
  - the word has a different, higher probability, in the input than in the background
- Binomial distribution used to compute the ratio of the two likelihoods
- The sentences containing the highest proportion of topic signatures are extracted.

Summarization as a Noisy Channel Model

- Summary/text pairs
- Machine learning model
- Identify which features help most

Case Study: Julian Kupiec SIGIR 95 Paper Abstract

- To summarize is to reduce in complexity, and hence in length while retaining some of the essential qualities of the original.
- This paper focusses on document extracts, a particular kind of computed document summary.
- Document extracts consisting of roughly 20% of the original can be as informative as the full text of a document, which suggests that even shorter extracts may be useful indicative summaries.
- The trends in our results are in agreement with those of Edmundson who used a subjectively weighted combination of features as opposed to training the feature weights with a corpus.
- We have developed a trainable summarization program that is grounded in a sound statistical framework.

Statistical Classification Framework

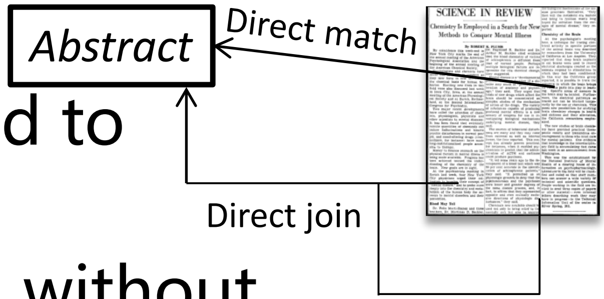
- A training set of documents with hand-selected abstracts
    - Engineering Information Co provides technical article abstracts
    - 188 document/summary pairs
    - From 21 journal articles
    - Average length of summary = 3 sentences
- Bayesian classifier estimates probability of a given sentence appearing in abstract
    - Direct matches (79%)
    - Direct Joins (3%)
    - Incomplete matches (4%)
    - Incomplete joins (5%)
- New extracts generated by ranking document sentences according to this probability

Features

- Sentence length cutoff
- Fixed phrase feature (26 indicator phrases)
- Paragraph feature
  - First 10 paragraphs and last 5
  - Is sentence paragraph-initial, paragraph-final, paragraph medial
- Thematic word feature
  - Most frequent content words in document
- Upper case Word Feature
  - Proper names are important

Sentence Matching

- Using manually created abstracts, match to sentences in orig. document
- **Direct match** - Verbatim 一字不差 or w/o minor modifications
- **Direct join** – 2 or more sentences used to make summary sentence
- **Unmatchable** – suspected fabrication without using sentences in document
- **Incomplete –**
  - Some overlap exists but content is not preserved in summary
  - Summary sentence includes content from original but contains other information that is not covered by a direct join



Evaluation

- Insufficient data for separate test corpus, used cross-validation strategy for evaluation
- Documents from a journal were selected for testing one at a time, all other document summary pairs were used for training
- Results were summed over journals
- Unmatchable/incomplete sentences were excluded from training and testing

Evaluating Performance

- Fraction of manual summary sentences that were reproduced, limited by text excerpting: $^{(451+19)}/_{568} = 83\%$

	# Sentences	Fract of Corpus
Direct Sentence Matches	451	79%
Direct Joins	19	3%
Unmatchable Sentences	50	9%
Incomplete Single Sentences	21	4%
Incomplete Joins	21	4%
Total Manual Summary Sentences	568	

Distribution of Correspondence in Training Corpus

- Sentence produced is correct if:
  - Has direct sentence match & present in manual summary
    – or –
  - Is in manual summary as part of direct join and all other components of join have been produced

Evaluation

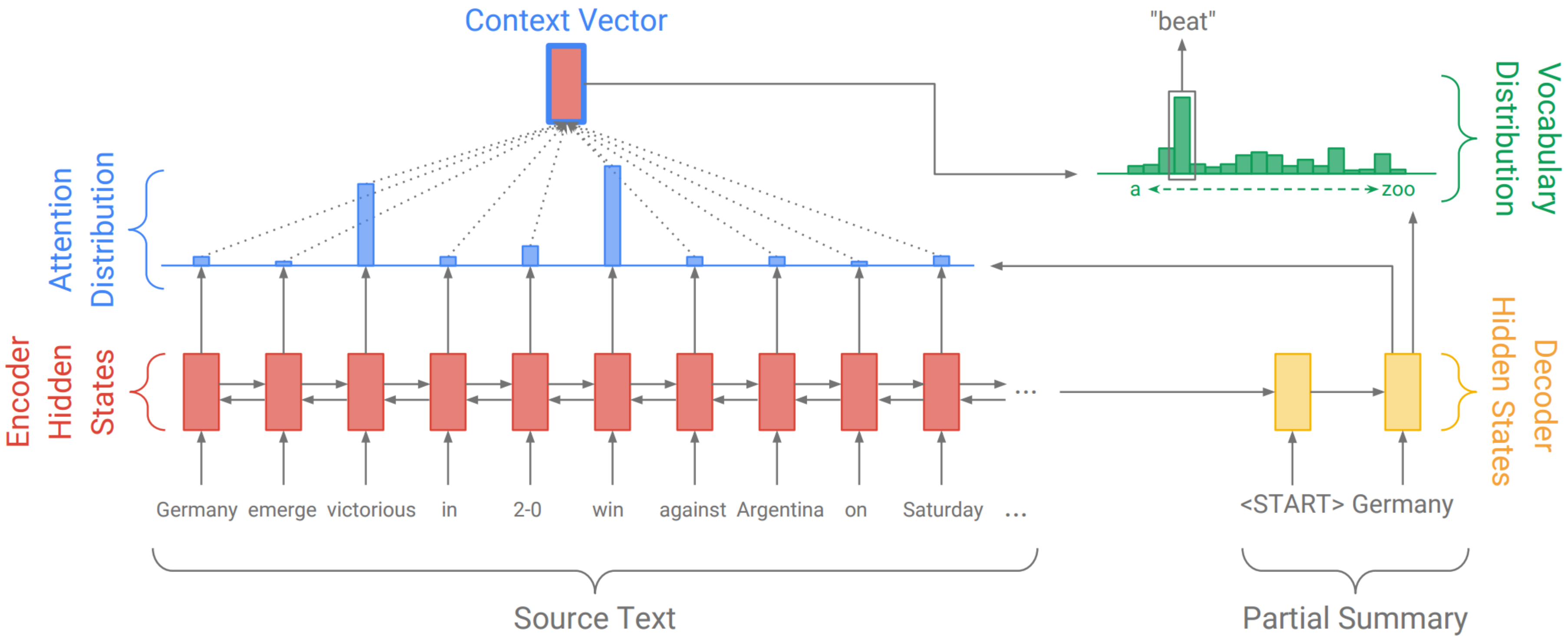
- Precision and recall
- Strict match has 83% upper bound
  - Trained summarizer: 35% correct

- Limit to the fraction of matchable sentences
  - Trained summarizer: 42% correct

- Best feature combination
  - Paragraph, fixed phrase, sentence length
  - Thematic and Uppercase Word give slight decrease in performance

NN-based Generative Summarization

- Attention-based summarization [Rush et al. 2015]


Context Vector
"beat"
Attention Distribution
Vocabulary Distribution
a
zoo
Encoder Hidden States
Decoder Hidden States
Germany
emerge
victorious
in
2-0
win
against
Argentina
on
Saturday
...
<START>
Germany
Source Text
Partial Summary

[From Abigail See]

Automated Evaluation

- ROUGE = Recall-Oriented Understudy for Gisted Evaluation
- Compare machine summary with human reference
- Inspired by BLEU (used in Machine Translation)
- Rouge-N: overlap of N-grams (just recall)
  - Rouge-2 is quite common
- Other variants exist
  - See in wikipedia
- In general, this is a HARD and open problem