Web-based Factoid Question Answering (including a sketch of Information Retrieval)

Slides adapted from Dan Jurafsky, Jim Martin and Ed Hovy

Today

- Web-based Question Answering
- Information Retrieval (briefly)

II. Question-Answering

- The notion of getting computers to give reasonable answers to questions has been around for quite awhile
- Three kinds of systems
  1) Finding answers in text collections
  2) Interfaces to relational databases
  3) Mixed initiative 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 Question Answering

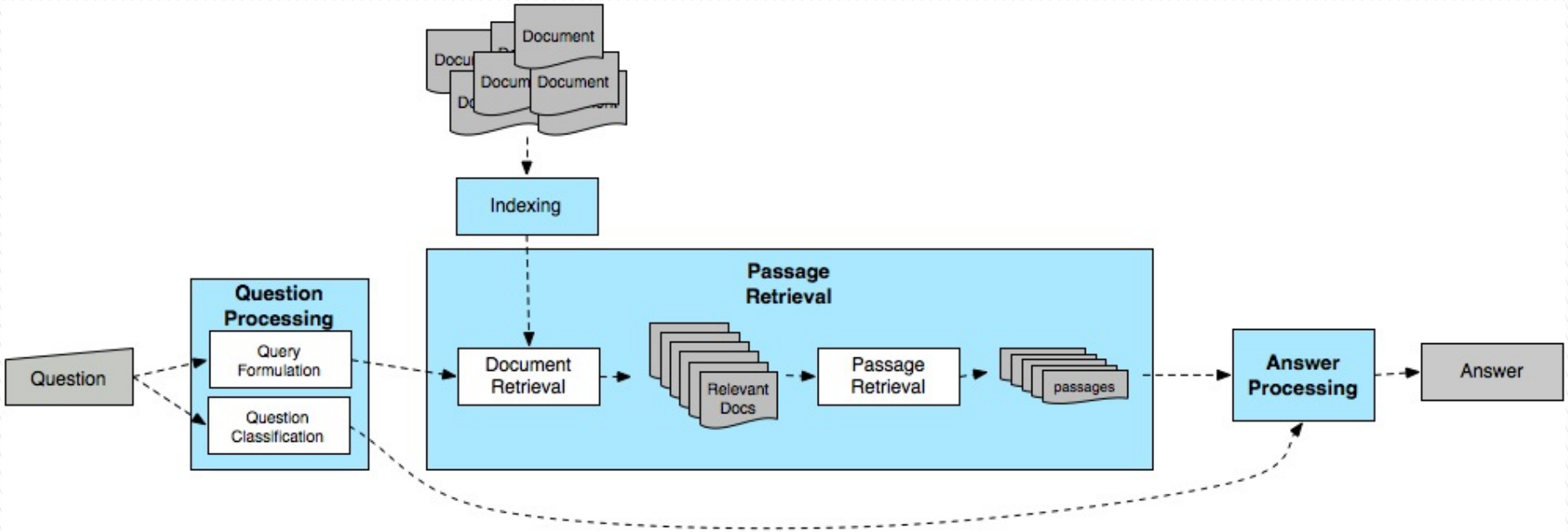
- Today
  - Introduction to Factoid QA
  - A typical full-fledged factoid QA system
  - A simpler alternative from MSR

- TREC: A Conference where many simultaneous evaluations are carried out
  - IR
  - QA

Factoid 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

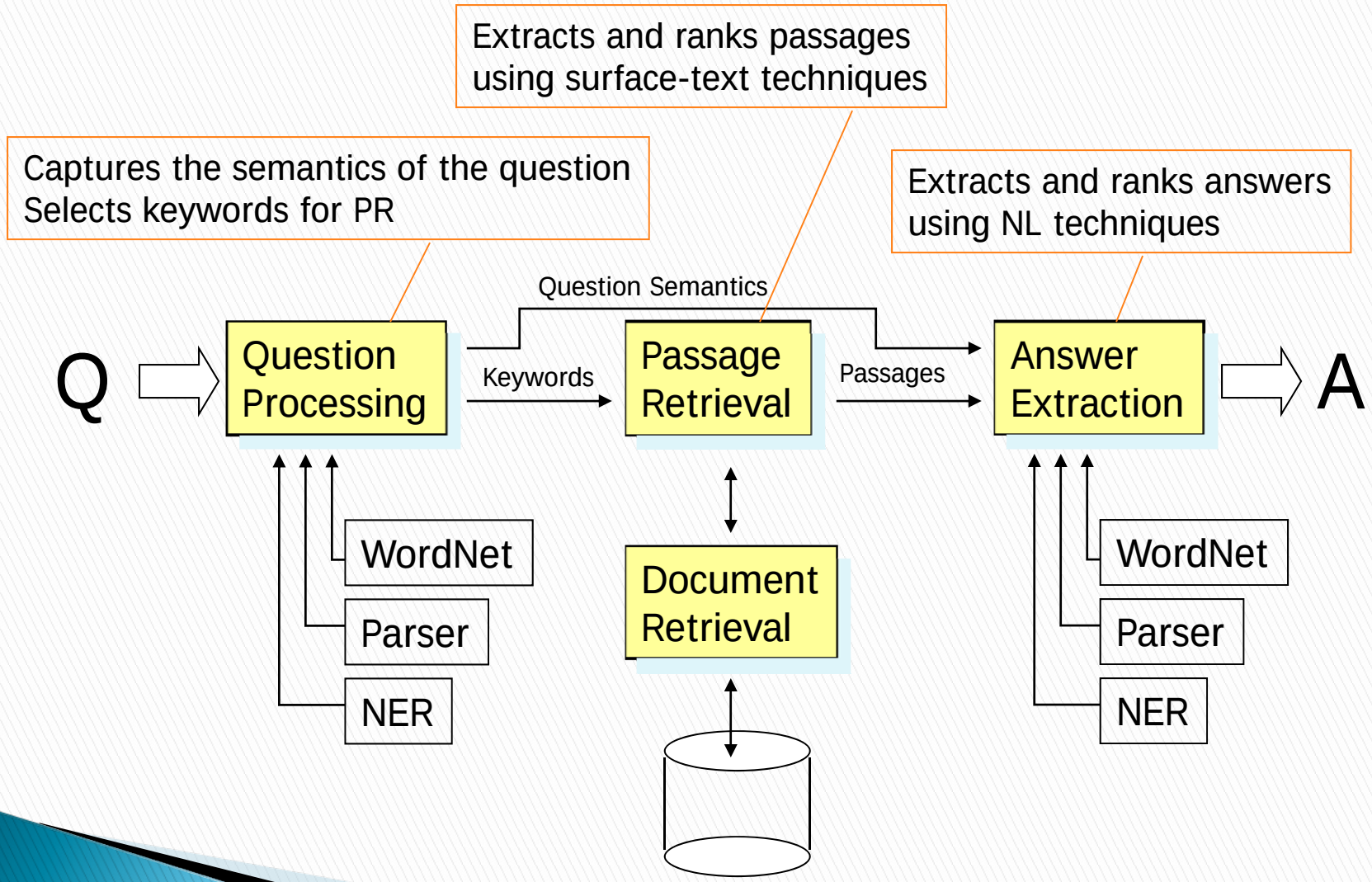
Factoid QA architecture



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



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

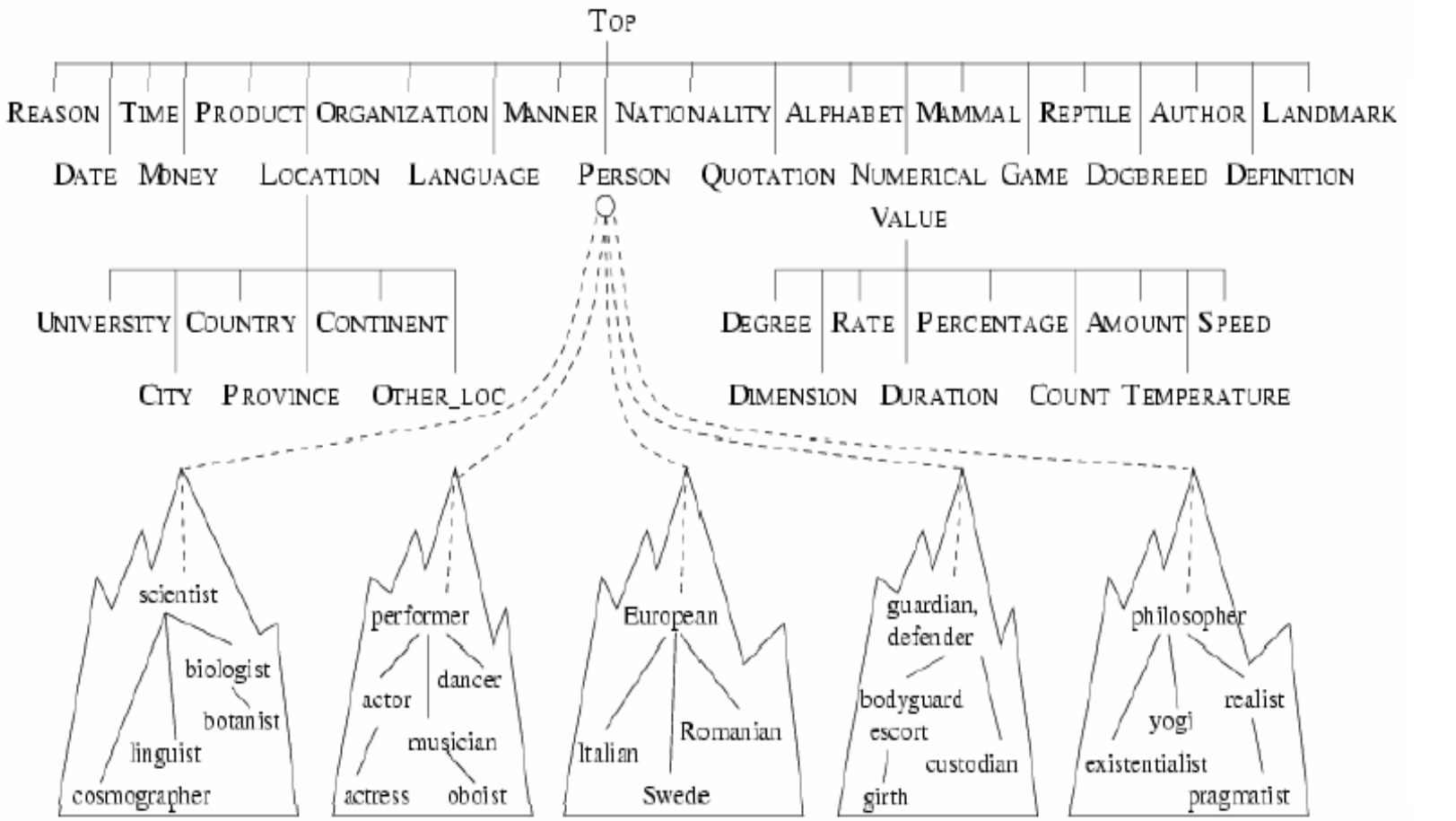
- Factoid questions…
  - Who, where, when, how many…
  - The answers fall into a limited and somewhat predictable set of categories
    - Who questions are going to be answered by…
    - Where questions…
  - Generally, systems select answer types from a set of Named Entities, augmented with other types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

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



Answer Type Detection

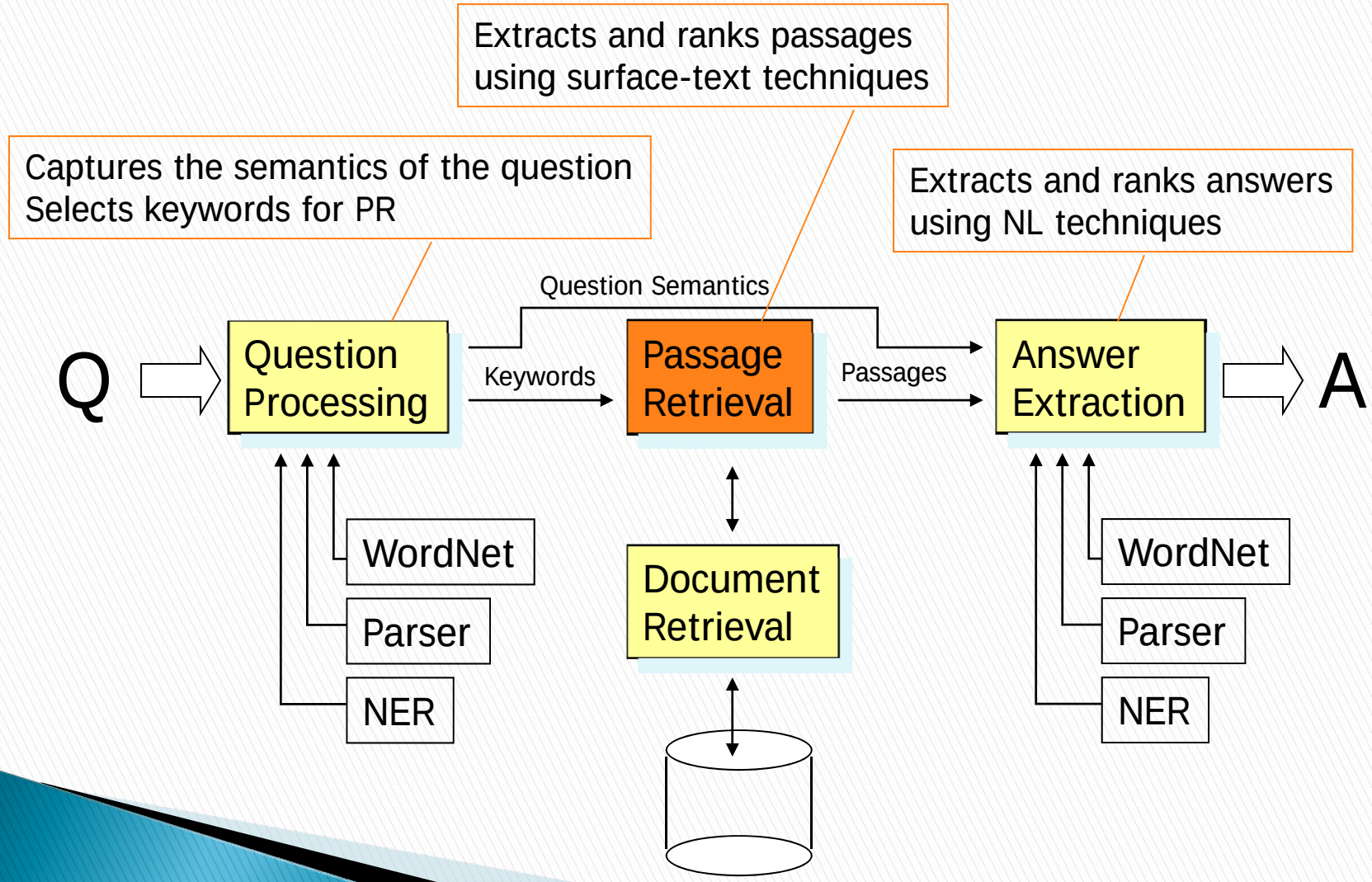
- Most systems use a combination of hand-crafted rules and supervised machine learning to determine the right answer type for a question.
- But how do we use the answer type?

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



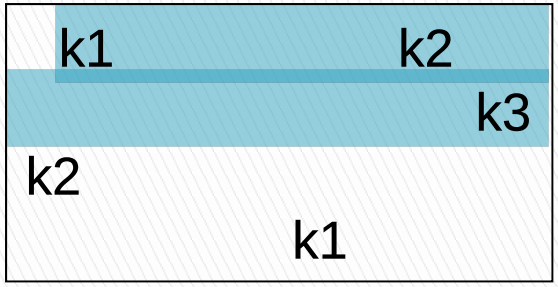
Passage Extraction Loop

- 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 $\Rightarrow$ query is too strict $\Rightarrow$ drop a keyword
  - If the number of passages is higher than a threshold $\Rightarrow$ query is too relaxed $\Rightarrow$ add a keyword

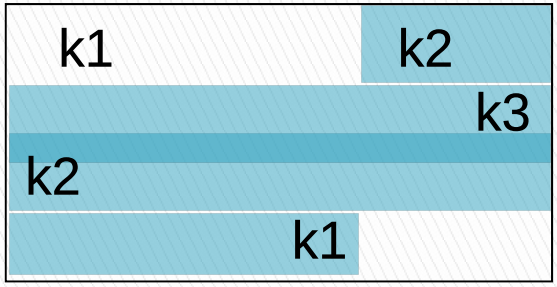
Passage Scoring

- 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:

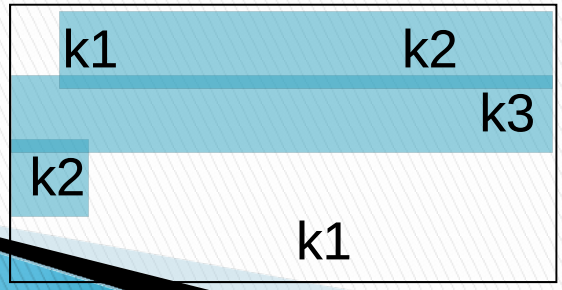
Window 1



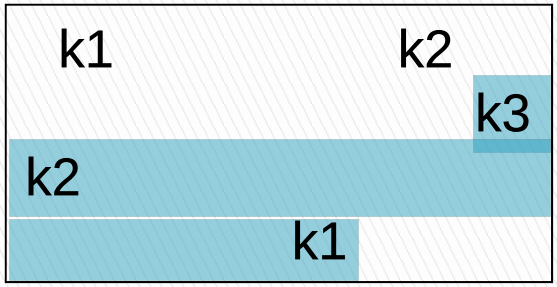
Window 2



Window 3



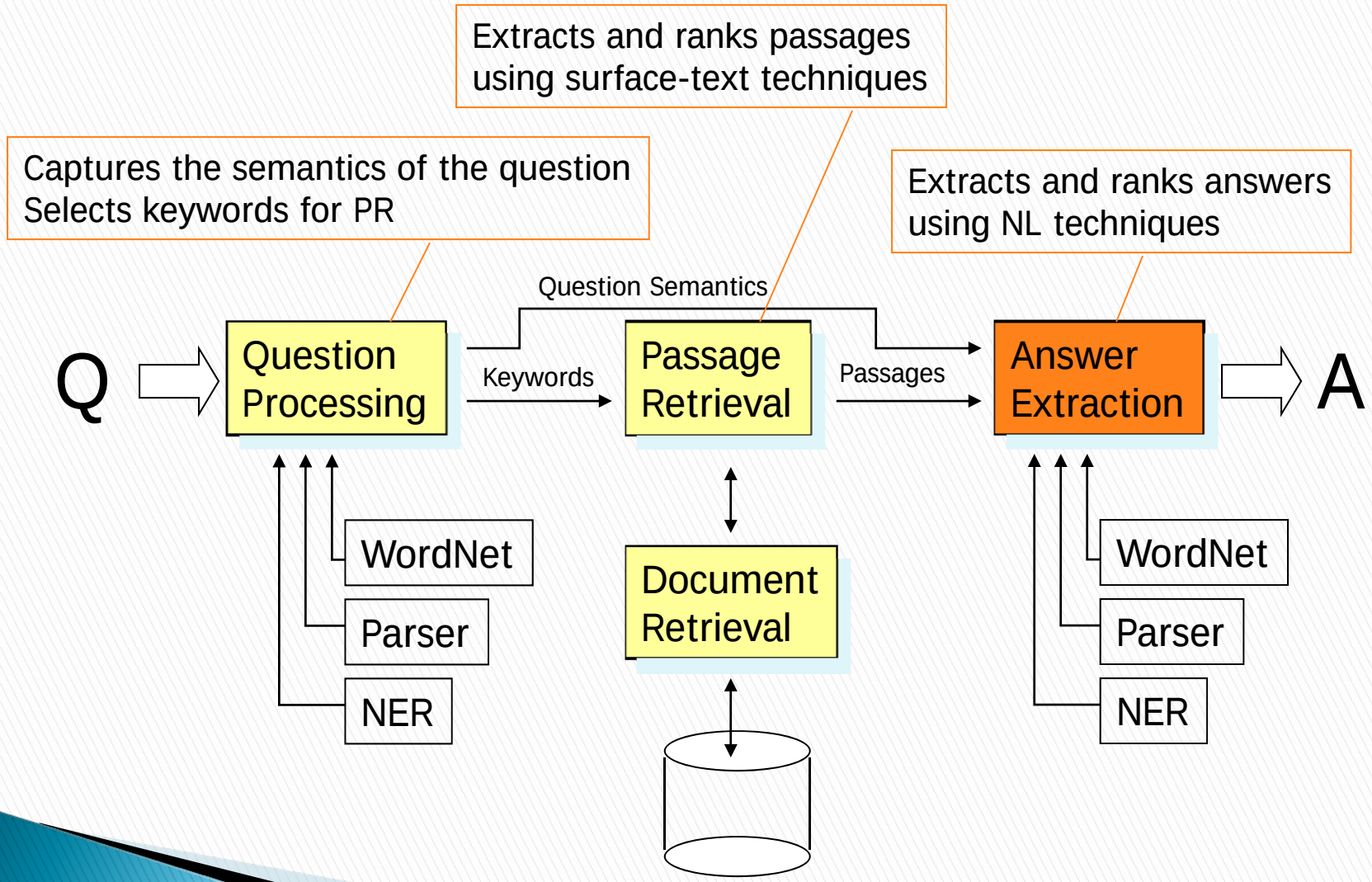
Window 4



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



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

Other Methods? Other Questions?

- When was Barack Obama born?
- Where was George Bush born?
- What college did John McCain attend?
- When did John F Kennedy die?

How does IE figure in?

Some examples

- Q: What is the population of Venezuela?
  - Patterns (with Precision score):
    - 0.60 <NAME> ' s <C-QUANTITY> population
    - 0.37 of <NAME> ' s <C-QUANTITY> people
    - 0.33 <C-QUANTITY> people in <NAME>
    - 0.28 <NAME> has <C-QUANTITY> people
- 3.2 Q: What is the population of New York?
  - S1. The mayor is held in high regards by the 8 million New Yorkers.
  - S2. The mayor is held in high regards by the two New Yorkers.

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

Is the Web Different?

- In TREC (and most commercial applications), retrieval is performed against a smallish closed collection of texts.
- The diversity/creativity in how people express themselves necessitates all that work to bring the question and the answer texts together.
- But…

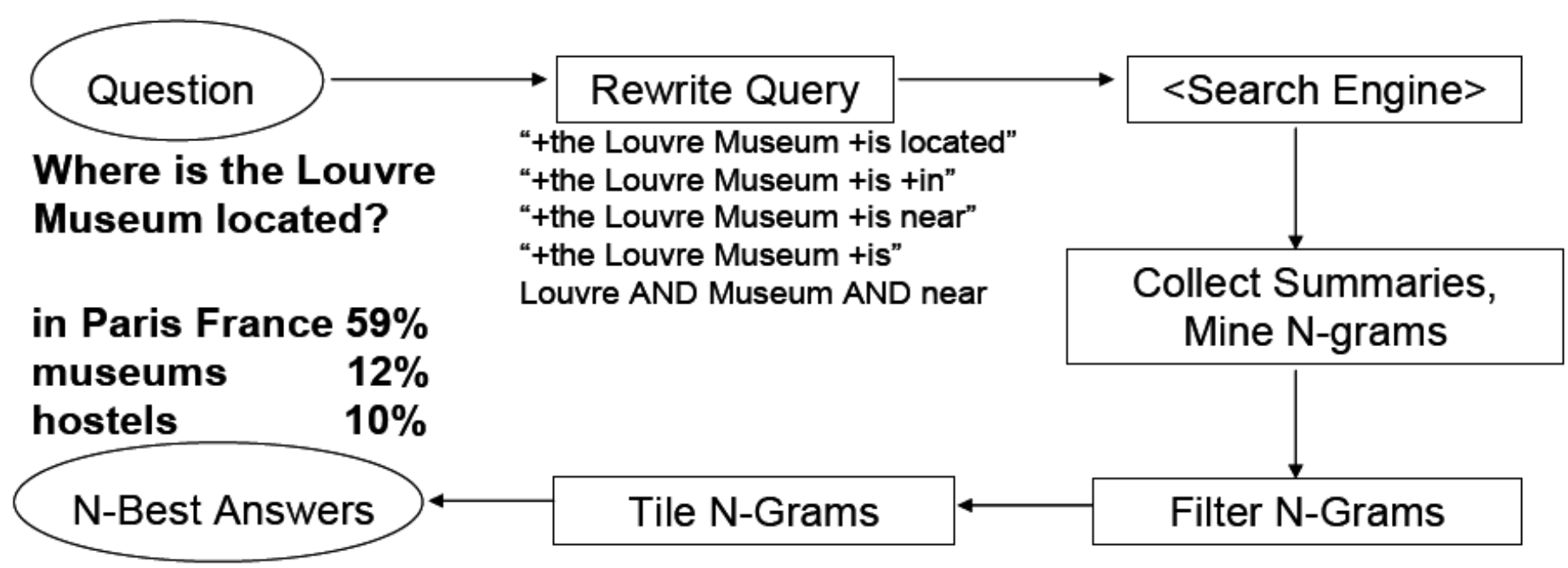
The Web is Different

- On the Web popular factoids are likely to be expressed in a gazzilion 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
Where is the Louvre Museum located?
in Paris France 59%
museums 12%
hostels 10%
N-Best Answers
Rewrite Query
"+the Louvre Museum +is located"
"+the Louvre Museum +is +in"
"+the Louvre Museum +is near"
"+the Louvre Museum +is"
Louvre AND Museum AND near
<Search Engine>
Collect Summaries, Mine N-grams
Filter N-Grams
Tile N-Grams

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", not the full text of the actual document

Step 3: Gathering N-Grams

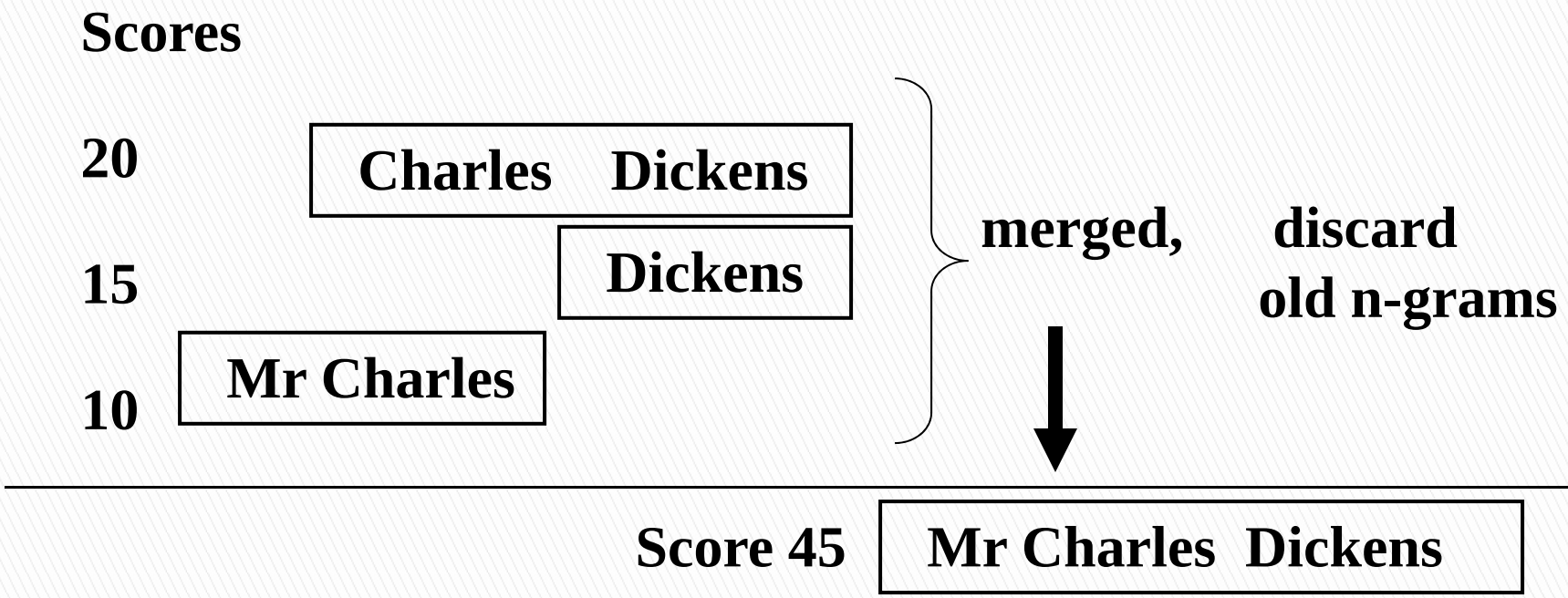
- Enumerate all N-grams (N=1,2,3) in all retrieved snippets
- Weight of an n-gram: occurrence count, each weighted by "reliability" (weight) of rewrite rule that fetched the document
  - Example: "Who created the character of Scrooge?"

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



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

    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
    - Be suspicious of any after the bake-off is over metrics

Which approach is better?

Harder Questions

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

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 approach.

Inverted Index

- 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?

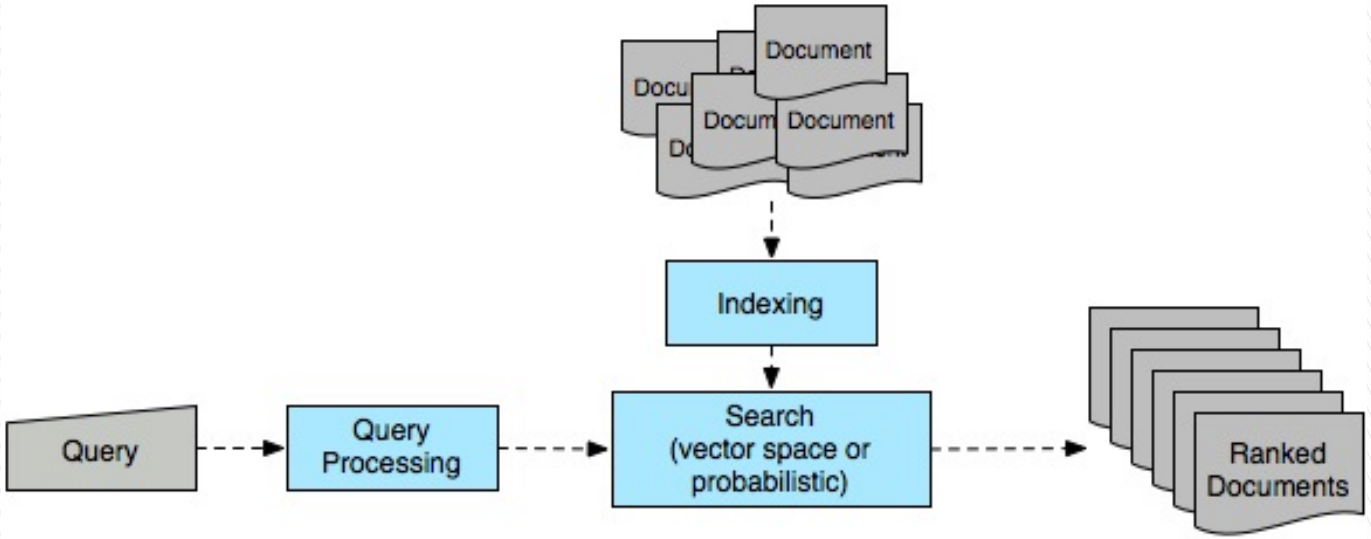
Phrases

- 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



Vector Space Model

- 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 $\vec{d}_j = (t_1, t_2, t_3, \ldots t_N)$
- 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

TFxIDF 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

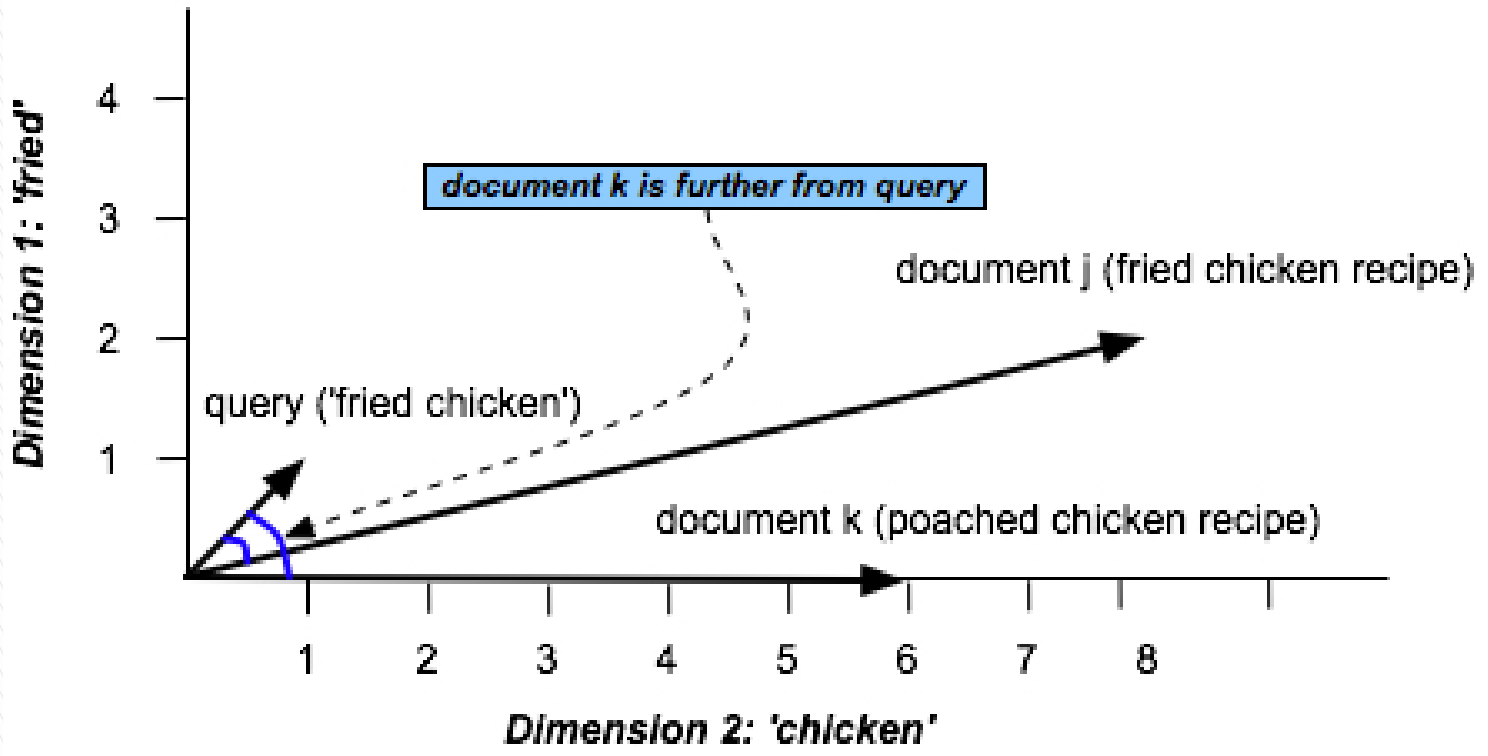
- We were counting bits to get similarity

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

- Now we have weights
- But that favors long documents over shorter ones

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

Similarity in Space (Vector Space Model)



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

Similarity

- The cosine measure

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

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

Summary

- Information Retrieval
- Web-based Question Answering