Question answering[1]

Victor Kitov

v.v.kitov@yandex.ru

[1] With materials used from "Speech and Language Processing", D. Jurafsky and J. H. Martin.

Types of tasks

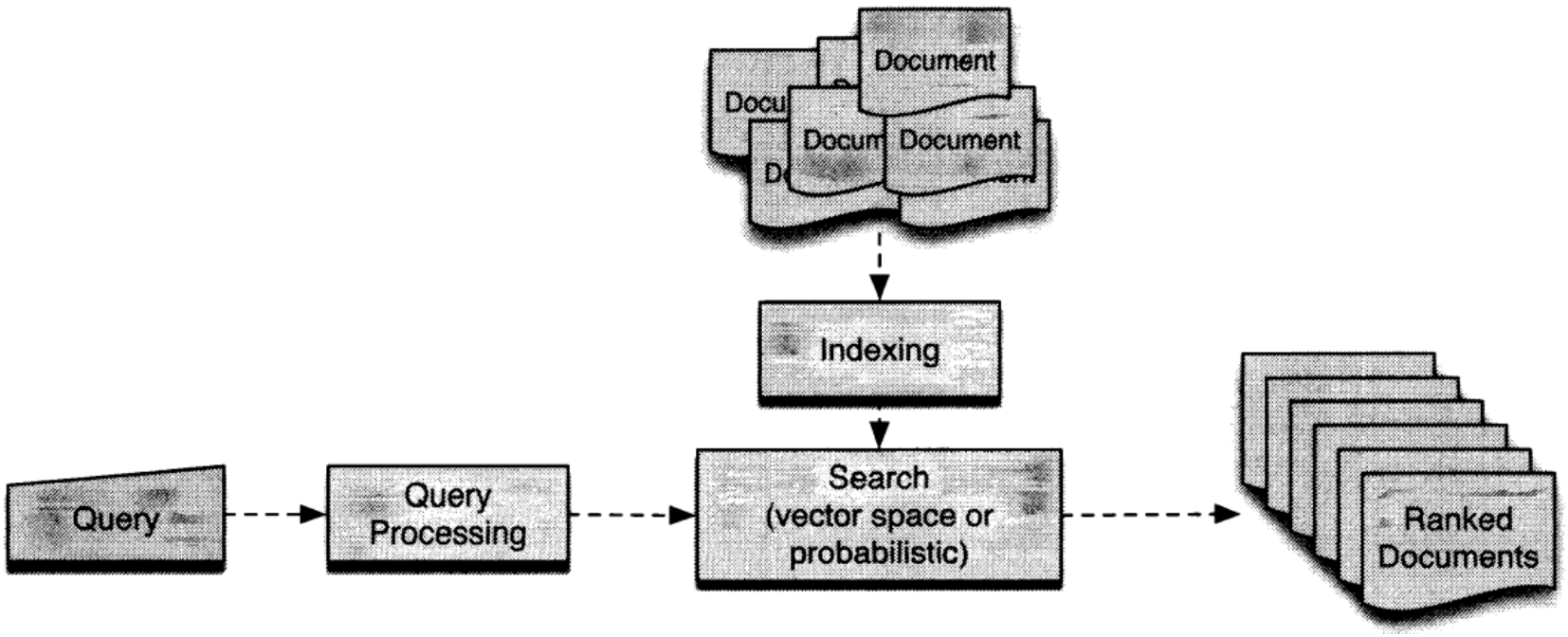
- What is expected as an answer:
  - entire document (information retrieval)
  - exact phrase (factoid question answering)
  - summary (automatic summarization)
- Question answering
  - factoid question answering
  - descriptions, overview questions
- Automatic summarization
  - single document
  - multiple documents
  - query-based summarization

Table of Contents

Information retrieval definitions

- Query
  - keywords
  - exact question
- Documents
  - documents themselves
  - paragraphs
  - sentences
  - entire websites
  - news clusters

Architecture of information retrieval system



Document representations

- Bag of words assumption
  - [I see what I eat] <=> [I eat what I see]
- Terms get represented with
  - indicator that term appeared in document, TF, TF-IDF
- May use other representation: LSI, topic modelling, doc2vec.

Document-query matching

- Similarity between document and query: cosine most popular
  - is invariant to scaling of objects (documents)
- Use classifier
  - (document, query) -> match each other?
- Search results should be not only highly relevant, but complete
  - account for synonyms, query reformulations
  - search results should augment each other, not duplicate

Query preprocessing

- Stemming - increases breadth of search but may return false positives
  - good: processing, processed -> process.
  - bad: stocks (акции) and stockings (женские чулки) -> stock
- False positives decrease both precision and recall
  - because relevant documents move down the ranked list of search results
- stop-words removal
  - without uninformative words search becomes more directed
  - search index decreases a lot, because stop-words appear in many documents
  - what to do with query: «to be or not to be»?

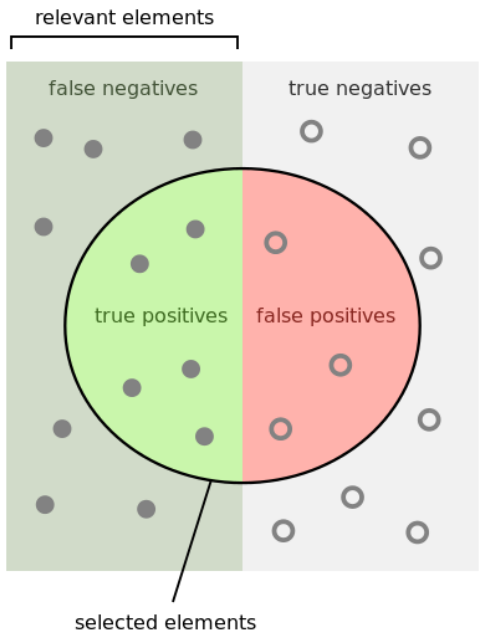
Synonyms and holonyms

- Synonyms - different form but same meaning.
- Holonyms - same form but different meaning
  - e.g. bank (bank of river, saving in a bank)
- Polysemes - same form but different meaning (though from the same origin)
  - e.g. book
    - a bound collection of pages (I read a book)
    - to make an action or event a matter of record (book a hotel room)

Synonyms and holonyms in IR

- Presence of synonyms can reduce precision and recall
  - examples?
  - We can augment queries with synonyms
- Presence of holonyms / polysemes can reduce precision and recall
  - examples?
  - We can perform semantic disambiguation.

Search evaluation



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

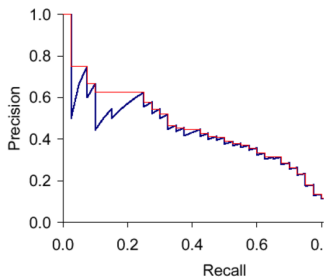
Top K precision and recall

- Precision and recall are not rank oriented
- But we may calculate them for top returned K documents

Rank	Judgment	Precision_{Rank}	Recall_{Rank}
1	R	1.0	.11
2	N	.50	.11
3	R	.66	.22
4	N	.50	.22
5	R	.60	.33
6	R	.66	.44
7	N	.57	.44
8	R	.63	.55
9	N	.55	.55
10	N	.50	.55

Precision(recall)

- $K \uparrow \Rightarrow Precision(K) \downarrow, Recall(K) \uparrow$
- Plot $Precision(Recall)$. Higher curves are more preferable



- To smooth precision-recall curve plot InterpolatedPrecision(Recall)

$$InterpolatedPrecision(Recall) = \max_{r \geq Recall} Precision(r)$$

- Systems that are higher in precision towards the left may favor precision over recall, while systems that are more geared towards recall will be higher at higher levels of recall (to the right).

Mean average precision

- Define R_r - the set of relevant documents at or above r

$$MAP = \frac{1}{|R_r|} \sum_{d \in R_r} precision(r)$$

- MAP averages precisions at positions where relevant documents were retrieved.
- MAP is precision oriented measure at the expense of lower recall.
- TREC (Text REtrieval Conference) every year holds competitions between information retrieval tasks

Query expansion - Relevance feedback

- Relevance feedback (Rocchio, 1971)
  1. user enters query
  2. IR system returns result
  3. user is asked to mark relevant / non relevant documents
     - denote them with R and S respectively
  4. query is expanded, using

$$q_{k+1} = q_k + \frac{\beta}{|R|}\sum_{i\in R} d_i - \frac{\gamma}{|N|}\sum_{i\in N} d_i$$

     - $\beta > 0$, $\gamma > 0$ - parameters.
- We can extract relevant / non-relevant documents from top / bottom of the search query
  - user is not engaged
  - overfitting!

Query expansion - synonyms

- We can add synonym terms to the query
- Synonyms
  - taken from thesaurus
  - we can cluster words (by clustering **columns** of design matrix) and add to existing words other words from their cluster
    - offline clustering - once for whole collection
    - online clustering - based on returned documents, related to the query

Table of Contents

Factoid question answering

Examples of factoid questions

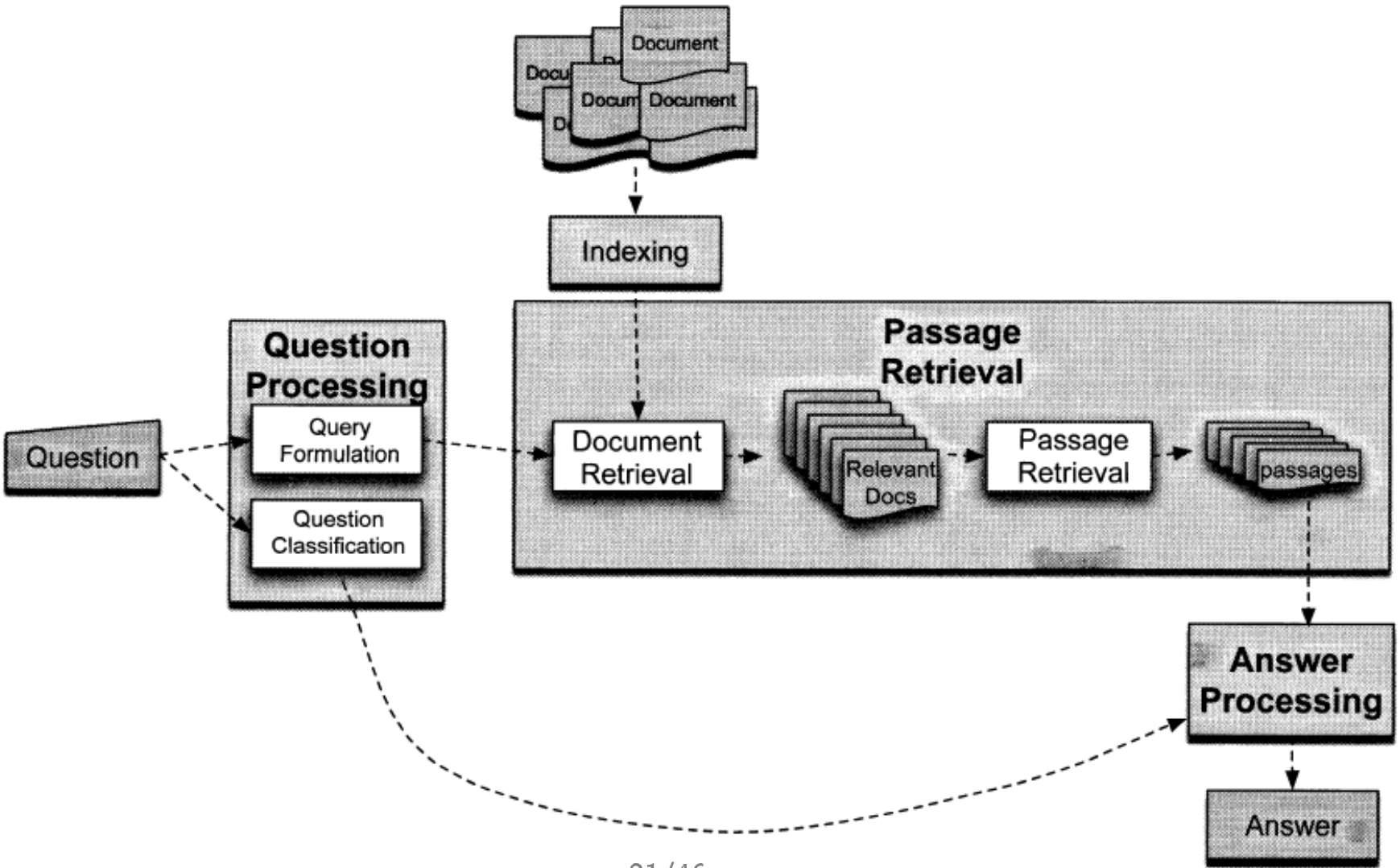
Question	Answer
Where is the Louvre Museum located?	in Paris, France
What's the abbreviation for limited partnership?	L.P.
What are the names of Odin's ravens?	Huginn and Muninn
What currency is used in China?	the yuan
What kind of nuts are used in marzipan?	almonds
What instrument does Max Roach play?	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 question answering

Solution: apply named entity recognition & relation extraction
Problem: answer to the question may be formulated in other terms

- User Question: What company *sells the most* greeting cards?
- Potential Document Answer: Hallmark remains *the largest maker* of greeting cards.

Question answering system



Query formulation

- When querying Web or some big collection:
  - Create query of every word
    - omit question word (where, when, etc)
  - To get focused results may query only keywords in noun phrases
- For small collections query expansion is needed
  - add to the query all morphological variants of the content words
  - synonyms expansion (thesaurus, from term clustering)
- Query reformulation
  - apply hand-built rules, e.g.:
    - [when was the laser invented?] -> [the laser was invented]
    - [where is the Valley of the Kings?] -> [the Valley of the Kings is located in]

Question classification

- What was asked for?
  - person name, location, organization, abbreviation, biography?
- Usually hierarchical classification is performed
- Examples:

HUMAN	
description	Who was Confucius?
group	What are the major companies that are part of Dow Jones?
ind	Who was the first Russian astronaut to do a spacewalk?
title	What was Queen Victoria's title regarding India?
LOCATION	
city	What's the oldest capital city in the Americas?
country	What country borders the most others?
mountain	What is the highest peak in Africa?
other	What river runs through Liverpool?
state	What states do not have state income tax?
NUMERIC	
code	What is the telephone number for the University of Colorado?
count	About how many soldiers died in World War II?
date	What is the date of Boxing Day?
distance	How long was Mao's 1930s Long March?
money	How much did a McDonald's hamburger cost in 1963?
order	Where does Shanghai rank among world cities in population?
other	What is the population of Mexico?
period	What was the average life expectancy during the Stone Age?
percent	What fraction of a beaver's life is spent swimming?
speed	What is the speed of the Mississippi River?
temp	How fast must a spacecraft travel to escape Earth's gravity?
size	What is the size of Argentina?
weight	How many pounds are there in a stone?

Answer type classification

- Answer type classification
  - rule based
  - supervised ML
- Main feature - headword of first noun-phrase after «wh...» word:
  - Which **city** in China has the largest number of foreign financial companies.
  - What is the state **flower** of California?
- We can use WordNet synset ID of the headword, its hypernyms and hyponyms as extra features.

Passage retrieval

- Search engine can extract passages for us - snippets!
- Passage may be paragraph or sentence.
- Aim of passage retrieval:
  - drop all irrelevant passages
  - rank other passages by relevance
- Important features:
  - presence of named entity type, requested in the question.
  - presence of answer template for given question
    - [when was the laser invented?] -> [the laser was invented]

Passage retrieval

- Other features for passage retrieval:
  - The number of named entities of the right type in the passage
  - The number of question keywords in the passage
  - The longest exact sequence of question keywords that occurs in the passage
  - The proximity of the keywords from the original query to each other.
  - The N-gram overlap between the passage and the question.
  - The rank of the document from which the passage was extracted

Answer search

- Search for answer pattern, corresponding to the question:
  - [when was the laser invented?] -> [the laser was invented]
  - [where is the Valley of the Kings?] -> [the Valley of the Kings is located in]
- Patterns can be extracted with the following algorithm;
  1. For a given relation between two terms (i.e. person-name→year-of-birth), start with a hand-built list of correct pairs (e.g., "Gandhi:1869", "Mozart:1756", etc).
  2. Query the Web with instances of these pairs (e.g.,"Gandhi"and "1869",etc) and examine the top X returned documents.
  3. Break each document into sentences, and keep only sentences containing both terms (e.g., PERSON - NAME and BIRTH - YEAR).
  4. Extract a regular expression pattern representing the words and punctuation that occur between and around the two terms.
  5. Keep all patterns that are sufficiently high-precision.

Answer search

- More accurate method, than rules, is training a supervised classifier
- Features:

Answer type match: True if the candidate answer contains a phrase with the correct answer type.

Pattern match: The identity of a pattern that matches the candidate answer.

Number of matched question keywords: How many question keywords are contained in the candidate answer.

Keyword distance: The distance between the candidate answer and query keywords (measured in average number of words or as the number of keywords that occur in the same syntactic phrase as the candidate answer).

Novelty factor: True if at least one word in the candidate answer is novel, that is, not in the query.

Apposition features: True if the candidate answer is an appositive to a phrase containing many question terms. Can be approximated by the number of question terms separated from the candidate answer through at most three words and one comma (Pasca, 2003).

Punctuation location: True if the candidate answer is immediately followed by a comma, period, quotation marks, semicolon, or exclamation mark.

Sequences of question terms: The length of the longest sequence of question terms that occurs in the candidate answer.

Evaluation of question answering

- Question answering can return:
  - exact answer
  - passage with answer highlighted
  - ranked list of answers
- Evaluation of ranked list of answers - mean reciprocal rank (MRR):
  - suppose for question i correct answer is located at $rank_i$
  - quality of the QA system on $i = 1, ...N$ questions:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$$

Table of Contents

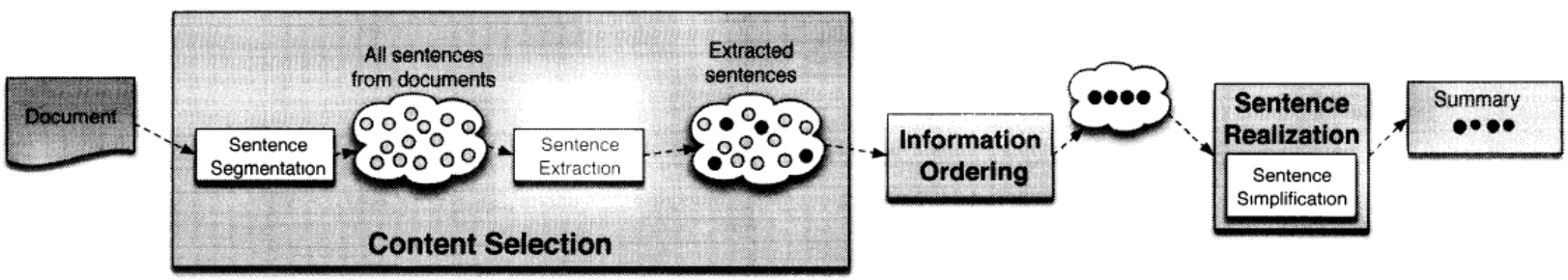
Applications

- **outlines** of any document
- **abstracts** of a scientific article
- **headlines** of a news article
- **snippets** summarizing a web page on a search engine results page
- **action items** or other summaries of a (spoken) business meeting
- **summaries** of email threads
- **compressed sentences** for producing simplified or compressed text
- **answers to complex questions**, constructed by summarizing multiple documents

Automatic summarization

- Summarization approaches categorization:
  - single document / multiple document
  - general / query focused
  - return extracts / abstracts
- Extract - subset of original sentences, abstract - new sentences
  - extracting extract is much easier

Single document summarization system:



Summarization

- Summarization of single document:

1. content selection - choose sentences from the document
2. information ordering - usually sentences are ordered in original order
3. sentence realization - post-processing
    - remove redundant clauses
    - pronoun substitution with original entities
    - substitution of short reference with extended reference
        - he, Bush -> George Bush

Unsupervised content selection

- Approach 1: by words saliency
  - Score of the word:
    - TF*IDF (it should be popular in document but not in the collection)
    - test statistic, testing H_0 : p(w|document)=p(w|collection)
  - Score of the sentence - average score of its words.
- Approach 2: cluster sentences and return centroid sentences for each cluster.

Unsupervised content selection

- Approach 3: by sentence centrality
  - sentence score = average cosine similarity between sentence and other sentences in the document
    - intuition - we rate sentences higher, that are more representative for the document
- Approach 4: by graph sentence centrality (LexRank)
  - construct graph:
    - edges - sentences $s_1, s_2, ...$
    - connection between s_i and s_j <=> $cosine\_sim(s_1, s_2) \geq threshold$
    - weight of connection between s_i and s_j $=cosine\_sim(s_1, s_2)$
  - score of sentence s=PageRank(s)
  - compared to approach 2 - extracts more diverse sentences.

Supervised sentence selection

Supervised sentence selection

1. Obtain training set of (document, extract) pairs with assessors.
2. Denote x - sentence description, $y = \mathbb{I}[x \text{ appears in extract}]$
3. Build classifier on (x, y) pairs

Training set generation from true abstracts

If training extracts are not available - build them from abstracts:

1. take scientific articles with their abstracts
2. build correspondence between sentences in abstracts and sentences in full document

- Example:
  - Human summary: This paper identifies the desirable features of an ideal multisensor gas monitor and lists the different models currently available.
  - Original document sentence: The present part lists the desirable features and the different models of portable, multisensor gas monitors currently available.

Abstract sentence s_A corresponds to sentence $\widehat{s}$ if $\widehat{s}$ has

- maximal longest subsequence with s_A
- minimum edit distance with s_A

Features for supervised sentence selection

position	The position of the sentence in the document. For example, Hovy and Lin (1999) found that the single most extract-worthy sentence in most newspaper articles is the title sentence. In the Ziff-Davis corpus they examined, the next most informative was the first sentence of paragraph 2 (P1S1), followed by the first sentence of paragraph 3 (P3S1); thus the list of ordinal sentence positions starting from the most informative was: T1, P2S1, P3S1, P4S1, P1S1, P2S2,... Position, like almost all summarization features, is heavily genre dependent. In *Wall Street Journal articles*, they found the most important information appeared in the following sentences: T1, P1S1, P1S2,...
cue phrases	Sentences containing phrases like *in summary*, *in conclusion*, or *this paper* are more likely to be extract worthy. These cue phrases are very dependent on the genre. For example, in British House of Lords legal summaries, the phrase *it seems to me that* is a useful cue phrase (Hachey and Grover, 2005).
word informativeness	Sentences that contain more terms from the **topic signature**, as described in the previous section, are more extract worthy.
sentence length	Very short sentences are rarely appropriate for extracting. We usually capture this fact by using a binary feature based on a cutoff (true if the sentence has more than, say, five words).
cohesion	Recall from Chapter 21 that a **lexical chain** is a series of related words that occurs throughout a discourse. Sentences that contain more terms from a lexical chain are often extract worthy because they are indicative of a continuing topic (Barzilay and Elhadad, 1997). This kind of cohesion can also be computed by graph-based methods (Mani and Bloedorn, 1999). The PageRank graph-based measures of sentence centrality discussed above can also be viewed as a coherence metric (Erkan and Radev, 2004).

Sentence simplification

- **Original sentence:** ~~When it arrives sometime new year in new TV sets,~~ the V-chip will give parents a ~~new and potentially revolutionary~~ device to block out programs they don't want their children to see.
- **Simplified sentence by humans:** The V-chip will give parents a device to block out programs they don't want their children to see.
- Linguistic rules for simplification:

appositives	Rajam, ~~28, an artist who was living at the time in Philadelphia~~, found the inspiration in the back of city magazines.
attribution clauses	Rebels agreed to talks with government officials, ~~international observers said Tuesday.~~
PPs without named entities	The commercial fishing restrictions in Washington will not be lifted [SBAR unless the salmon population 329 increases [PP ~~to a sustainable number~~]]
initial adverbials	"For example", "On the other hand", "As a matter of fact", "At this point"

- Alternatively we can train a supervised sentence simplifier.

Multi-document summarization

- Complexities of multi-document summarization:
  - sentences are to much similar
    - because different documents can be almost about the same
    - need to penalize similarity!
  - harder to order sentences
    - in single document we could order sentences as they follow in the document
    - we can order sentences in chronological order
    - or using patterns of the summarized event, e.g. earthquake: 1) strength of earthquake 2) location 3) rescue efforts 4) casualties

Multi-document summarization

Sentences are too close:

- We can cluster sentences and return centroid sentences.
- When scoring sentences, we can penalize its similarity with already selected sentences:

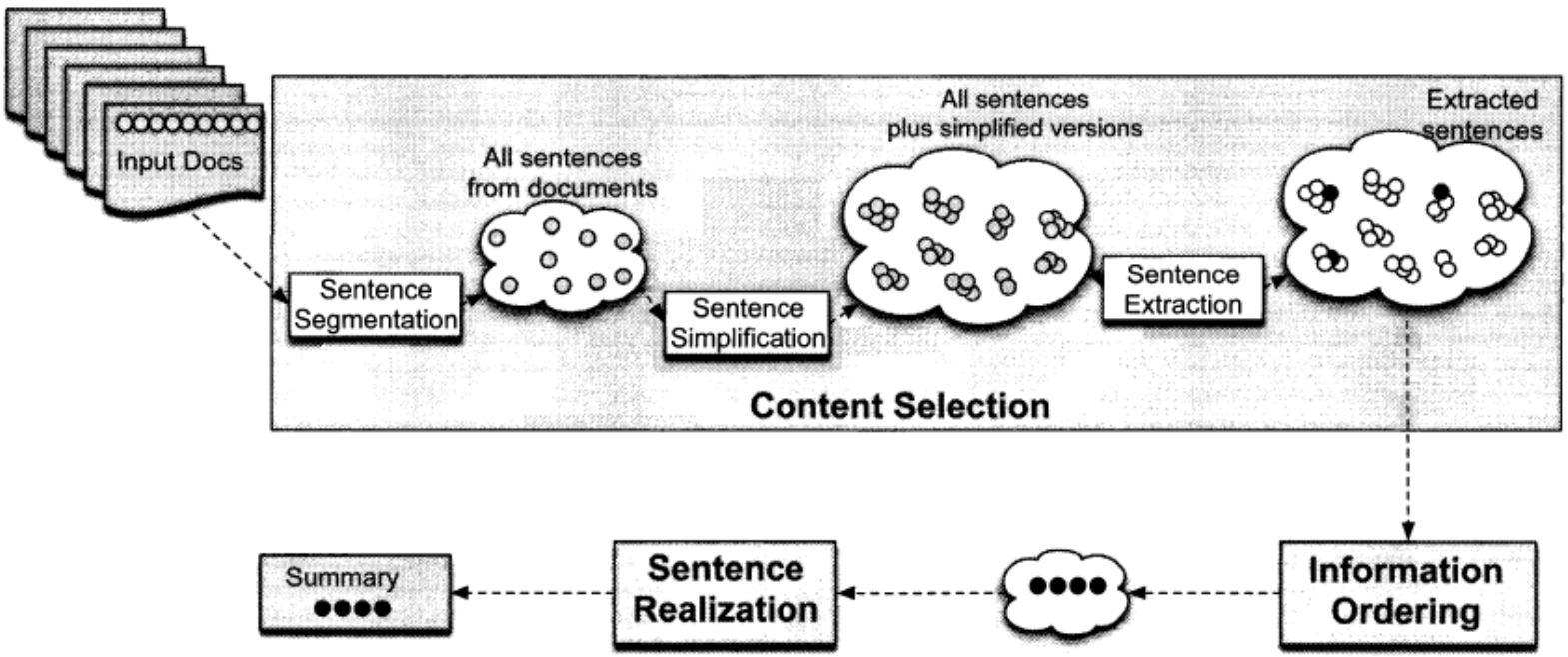
$$regularizer = \lambda \max_{s_i \in summary} sim(s, s_i)$$

In multi-document summarization it is common to join sentence selection and simplification:

- we replace original sentences with all their simplified variants and select sentences from this expanded set:

Multi-document summarization

Multi-document summarization system:



Query-based summarization

- Bottom-up approach: when scoring sentence for selection in summary, add features:
  - sentence contains named entities from query
  - cosine similarity between sentence and query
- Top-down approach:
  1. detect question type (e.g. BIOGRAPHY)
  2. using predefined rules, associate the type structure of answer
     - e.g. BIOGRAPHY -> <NAME> is <WHY FAMOUS>. Born on <BIRTHDATE> in <BIRTHLOCATION>. <EDUCATION>. <DESCRIPTIVE SENTENCE>. <DESCRIPTIVE SENTENCE>.
  3. do information extraction for each part of the answer
  4. combine

Summary evaluation

Evaluation criteria:

- extrinsic (final task based)
  - e.g. measure time needed to answer question by humans, using different summarization algorithms
- intrinsic
  - ROUGE metric (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE score

- Main metric ROUGE2: recall of 2-grams:

$$ROUGE2 = \frac{\sum_{S \in T} \sum_{bigram \in S} count_{match}(bigram)}{\sum_{S \in T} \sum_{bigram \in S} count(bigram)}$$

- $count_{match}(bigram)$ - the number of matched bigrams (that co-occur in algorithm output and human summaries)
- Extensions:
  - ROUGE-N - computes recall of N-grams
  - ROUGE-S counts recall of skip-grams (words that happen in the same order but can have other words in between)
  - ROUGE-L - longest common subsequence between true and generated summary

Comments on evaluation

- Discussion:
  - summary length matters - longer summary-higher recall
  - even recall between human summaries may be small!
- Another approach - humans count how much of the discrete facts we mentioned in the summary among important facts.
- Baselines:
  - random sentences baseline
  - leading (first) N sentences baseline