IS2140 Information Storage and Retrieval

Unit 2: Docs Processing

Daqing He
School of Computing and Information
University of Pittsburgh

September 10, 2018

Reading System

- The link: pawscomp2.sis.pitt.edu/ereader/login
- Username: pittID
  - *such as dah44*
- Password: ir+PeoplesoftID,
  - *Assume my peoplesoftID is 4413658, the password will be ir4413658*
  - *But if you have taken DB course, your password will be db4413658*
- Finish all the readings in each week, and complete all the quizs associated with the readings.
- Use the same deadline as before

More on Muddiest Points

- Publish in blogs
  - *Post your blog URL in the courseweb's discussion board*
- The question is about the content discussed in the classroom in the week
  - *For example, unit 2 muddiest point is due this coming Saturday, and is about the content discussed today, not about unit 3 topic that will be discussed next week*

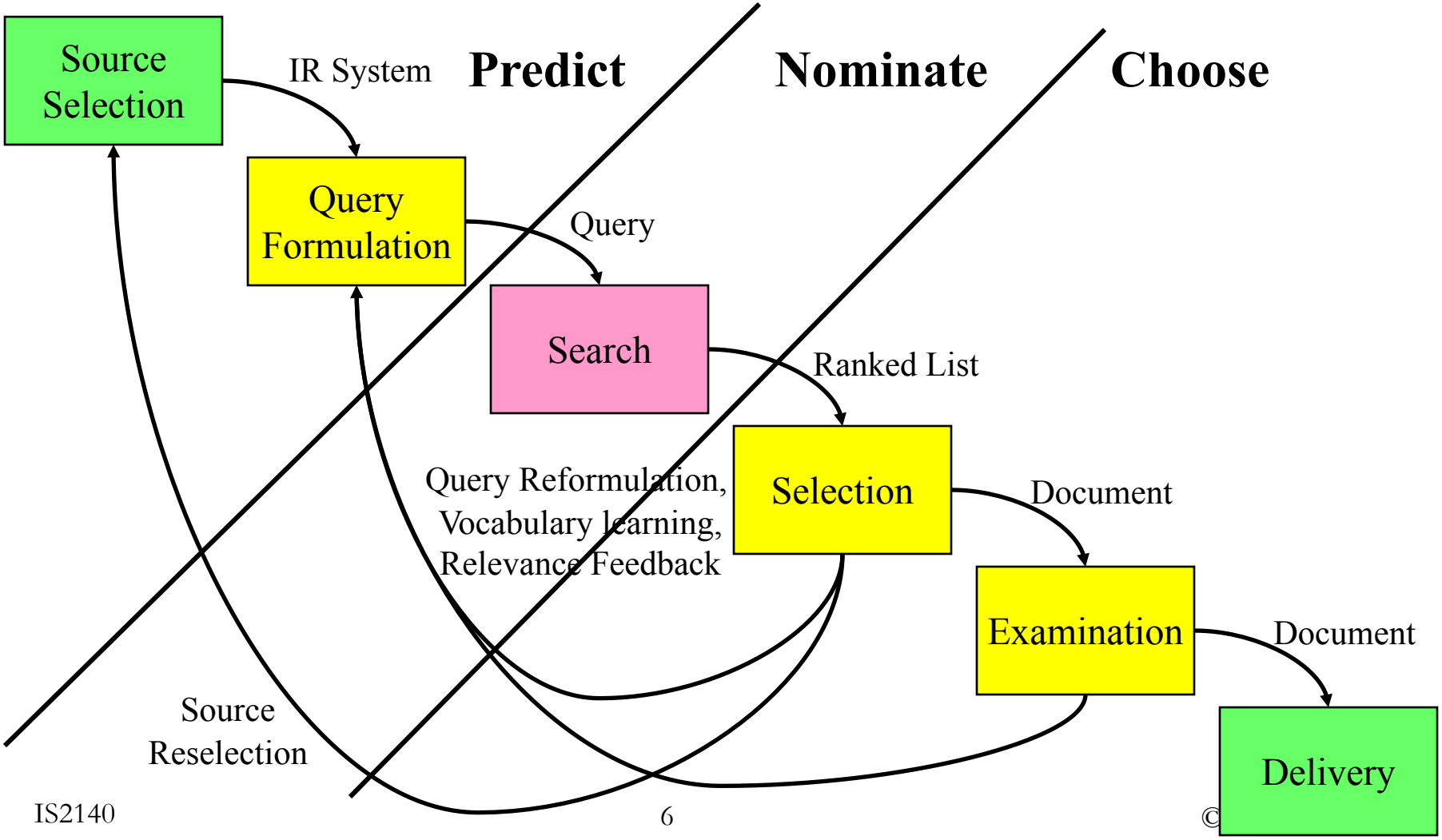
Muddiest Points

- Retrieval
  - *Does the Information Retrieval need the step to formulate the unstructured data into structured data and save it into database?*
  - *How do search engines like Google quickly match queries from users with a big amount of data in a short period of time? Do they need to save some common results beforehand?*
  - *How could we connect the desktop to the web pages? What interface we need to go through?*

Muddiest Points

- Information Retrieval Process
  - *The point is about the lecture slide on page 31. Since I believe this graph briefly summarizes the information retrieval process, I still have several points which are ambiguous to me. The first one is that is there any meaning behind the color of different steps in the process? The other is: there are three phases in this graph (predict, nominate, choose), so do they have different emphases on those phases and why we need to differentiate them from each other.*
  - *We talked about the new form of information retrieval which is that the system proactively learns what the users would need based on past activities and provides relative information. However, without the users initiating the information exchange process, is it still a "retrieval"?*

Information Retrieval Cycle



Muddiest Points

- Retrieval
  - *In our lecture slides (P36), it is said that recoverability is downplayed by IR.Why IR does not care about the recoverability?*
  - *Unstructured data including text documents as well as images and videos.We could use tf-idf or some other models to rank text documents, but how to retrieve images or videos, especially those without a certain title or description.*

Muddiest Points

- Information Needs
  - *Stable needs such as new data mining algorithms can be categorized as long-term information need, but why does looking for information about green energy projects belong to a short-term need. Is green energy projects not a kind of stable need, like an existing knowledge?*
  - *How to do the information retrieval when the query is ambiguous and incomplete? Does the system need to take the users' background, browser history or some other factors into account?*

Muddiest Points

- Information Retrieval
  - *Sometimes the document amount that store behind the system is very huge. If we search every query in the whole data, it will waste a lot of time. If we select sources before query, how can we do that?*
  - *Nowadays IR includes modeling search process, web search, text classification / clustering, system architecture, user interfaces, etc. If I want to look for certain things on Google, how can it decide which information I want, just base on key words or there is other solutions to solve it base on information on the internet are unstructured?*
  - *Please explain the difference between Data retrieval and Information retrieval.*

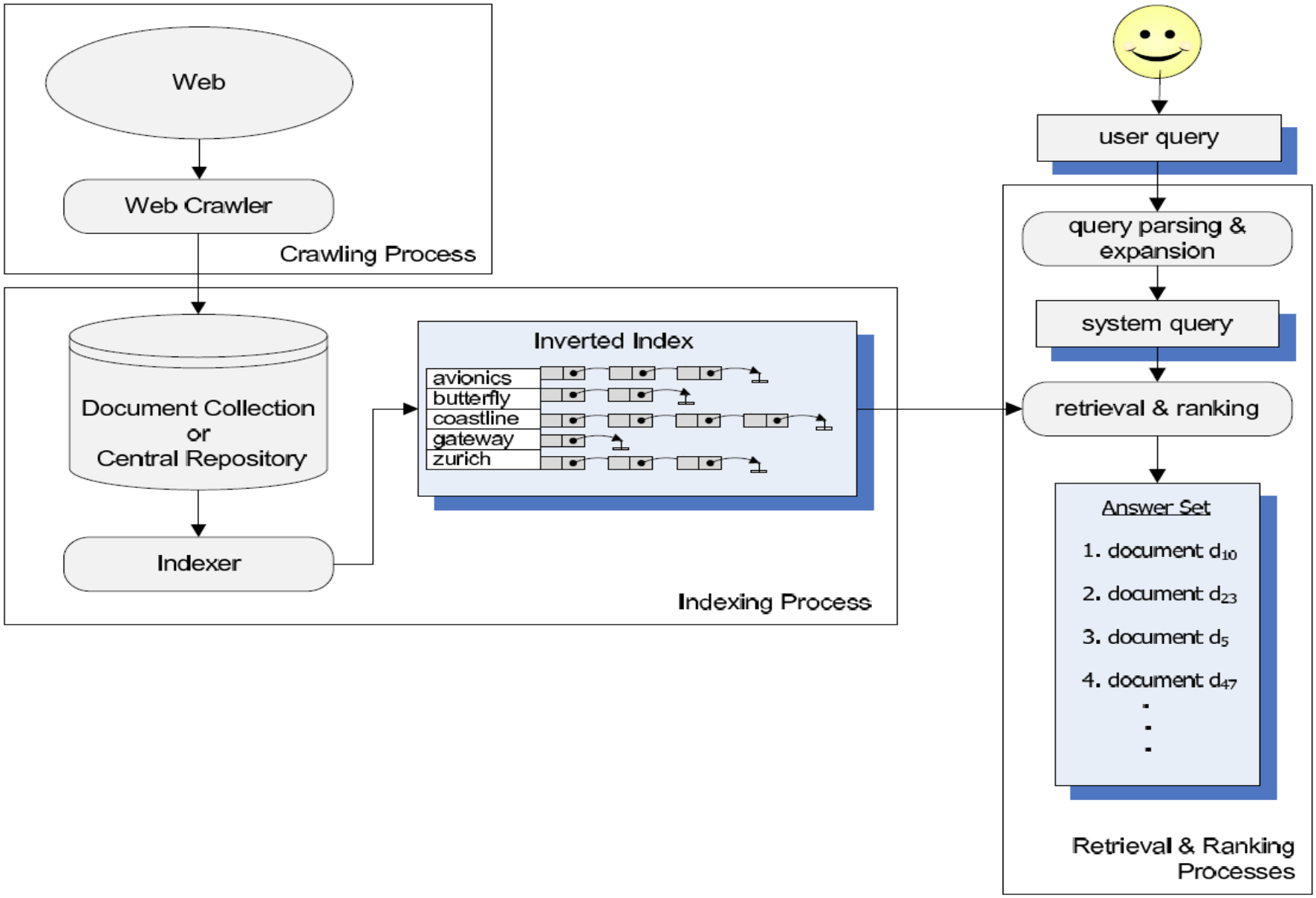
Agenda

- Preprocessing Documents and Queries
  - *Bag of Word Representation*
  - *Document Preprocessing*
  - *Query processing (will be discussed next week)*
- Term Project Introduction

Class Goals

- After this class, you should be able to
  - *know the basic idea of bag of word representation*
  - *Perform basic preprocess steps for documents and queries*
  - *Familiar with the requirements of term project so can start preparing it*

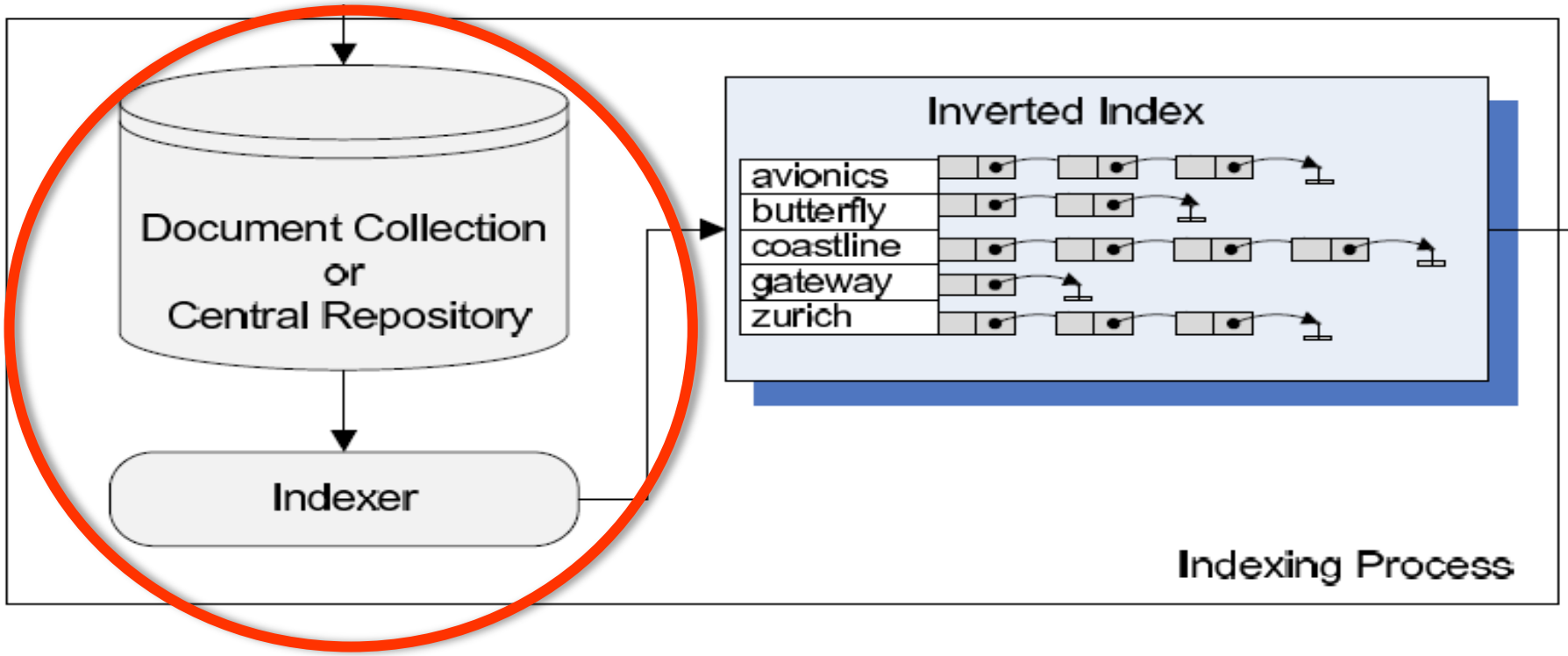
Whole View of System Oriented IR



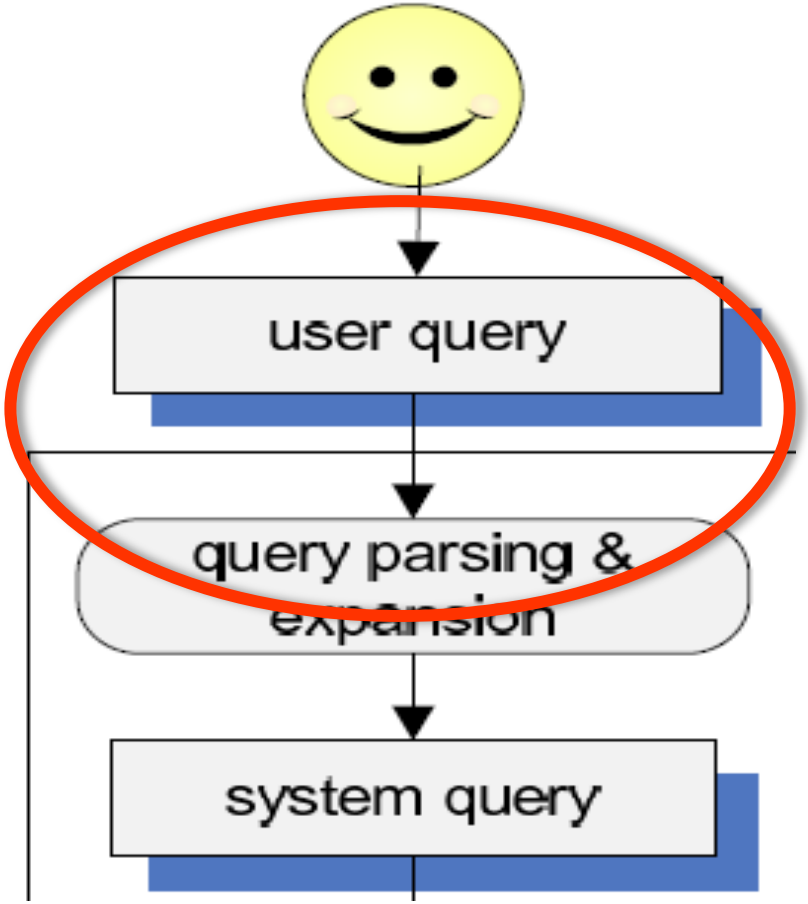
Match on Representations

- Matching documents and queries can be on their original forms
  - *But it would be slow. --- look at searches in MS Outlook*
- Matching between representations of queries and documents
  - *These two presentation should be comparable*
  - *The representation of the documents is called Index*
- Aim: make search faster, more efficient
- **Index**: the file that contains the representation of documents
- **Index terms** are used to build up the index
- **Indexing** is a processing to decide what is used to represent documents

Document Processing and Indexing



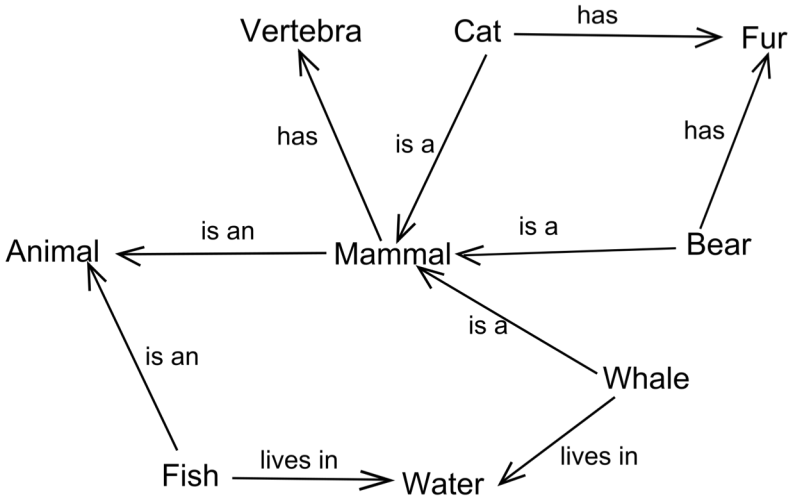
Query Processing



Document Processing

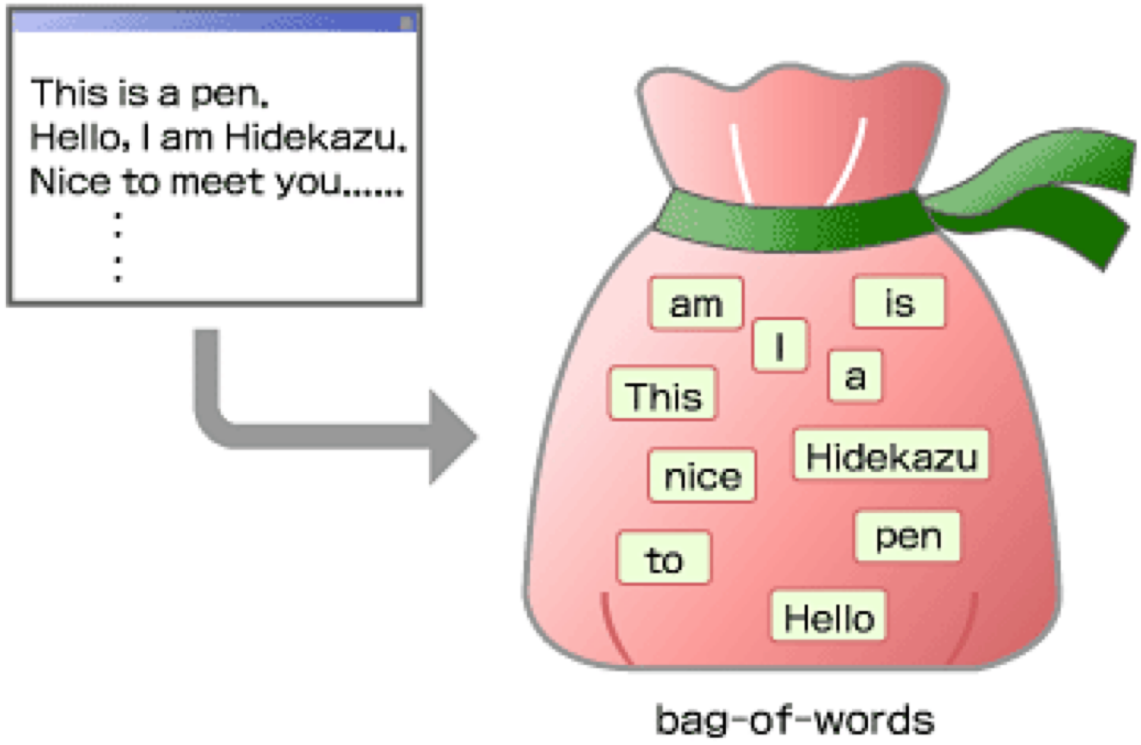
Approaches for Representation - 1

- Ideally, understand the whole content, and represent the understanding in a computer understandable format
  - *Ontology,Wordnet, event frames, knowledge graph etc.*
  - *Natural Language Processing (NLP) is working on it, but still too far from reality*



Approaches for Representation - 2

- In IR, most commonly used are statistical methods based on Bag of word representation
  - *Not aim to fully understand the documents*
  - *Just simple count of terms (words or phrases)*



Bag of Word Representation

- Independence between words
  - *No syntactic relation at all between words*
- Not recognize semantic relationship among words
  - *No synonyms: car is not recognized as being related to automobile,*
  - *No hyponyms, hypernym: apple is not recognized as being related to fruit*
  - *No homonyms: so treat Apple(fruit) as the same as Apple(company)*
- Index Terms are
  - *Word units in documents*
  - *Tokenization: identify the word units for indexing*

Bag of Word Representation –Example

People at San Francisco Apple site say that an apple employee loves to peel apples with a passion of love

Extracted Terms from the Example

- Index terms are text separate by non-letter characters

People
Francisco
say
apple
to
with
of
at
Apple
that
employee
peel
a
love
San
site
an
loves
apples
passion

Extracted Terms from the Example

- Index terms are text separate by non-letter characters

People
Francisco
say
apple
to
with
of

at
Apple
that
employee
peel
a
love

San
site
an
loves
apples
passion

- Possible better index terms

people
site
love

San Francisco
say
peel

apple
employee
passion

Methods of Indexing

- Manual Indexing
  - *People select index terms for a controlled vocabulary source (MeSH, LCSH, Eric Thesaurus)*
  - *Helped with human understanding of the documents*
    - But usually only based on the title and abstract parts
  - *Example at ERIC: www.eric.ed.gov*
- Automatic Indexing
  - *Machine automatically extract index terms from the documents*
  - *So far no true understanding of the documents*
  - *Often applied to the full content, therefore automatic full text indexing*

Manual vs Automatic Indexing

- Experimental evidence is that retrieval effectiveness using automatic indexing can be at least as effective as manual indexing with controlled vocabularies
  - *Original results were from the Cranfield experiments in the 60s*
  - *Considered counter-intuitive*
  - *Other results since then have supported this conclusion*
  - *Broadly accepted at this point*
- Experiments have also shown that using both manually and automatic indexing improves performance
  - *"combination of evidence"*

Based on Allen 04 slides

Major Steps for Automatic Indexing

- Decide the basics of a document
- Parsing the documents
- Tokenization
- Processing Tokens

Decide the Basics of Documents

- Basics include
  - *What is a document?*
    - A book? A chapter? An article? A section? A paragraph? A file? A part of a file?
  - *the right content and the right boundaries*
- Why content becomes a problem?
  - *http://www.nytimes.com/*
  - *Remove irrelevant parts, based on visual cues, or based on DOM*
  - *Link segmented parts together*
- Why boundaries become a problem?
  - *document collection for assignment 1*
  - *A file contains about 50,000 documents*

Parsing a Document - I

- What format is it in?
  - *pdf/word/excel/html?*
- What language is it in?
  - *Documents being indexed can include docs from many different languages*
    - https://www.manualslib.com/manual/191023/Whirlpool-1185020.html
  - *Sometimes a document or its components can contain multiple languages/formats*
    - http://en.wikipedia.org/wiki/Chinglish
    - or

استقلت الجزائر في سنة 1962 بعد 132 عاما من الاحتلال الفرنسي.

← → ← → ← START

'Algeria achieved its independence in 1962 after 132 years of French occupation.'

Parsing a Document - II

- Are sections or fields important?

```
<DOC>
<DOCNO>0000x-05xxxx.00x</DOCNO>
<NAME>John Smith, Jane Smith</NAME>
<MANUALKEYWORD>
        family businesses | family life | Pittsburgh (PA, USA) | food
</MANUALKEYWORD>
<SUMMARY>
        John describes. … born June 8, 1927.
</SUMMARY>
<ASRTEXT2003A>
        were to tell both of them were beautifully together.
</ASRTEXT2003A>
</DOC>
```

Tokenization - 1

- The process of identifying the index terms, called "tokens" or vaguely as "words", "terms"
- Basic approach: all tokens are separated by white spaces

  Happy families are all alike ➔ *happy, families, are, all, alike*

- Be careful about punctuations and accents
  - Apostrophe:
    - Finland's capital ➔ Finland capital
  - Hyphen and Dashes:
    - Hewlett-Packard ➔ Hewlett_Packard or Hewlett, Packard
    - San Francisco-Los Angeles ➔ San_Francisco, Los_Angeles
    - But what about the hold-him-back-and-drag-him-away maneuver, 412-624-2400?
  - Accents: *résumé* ➔ *resume*

Tokenization - 2

- Consider to normalize to one authoritative token, particular locations and common used terms
  - *San Francisco* ➔ *San_Francisco*
  - *White space vs whitespace* ➔ *whitespace*
  - *Lowercase, lower-case, lower case* ➔ *lowercase*

Tokenization - 3

- Using word segmentation to languages without spaces

  天主教教宗若望保祿二世因感冒再度住進醫院。

  ⟶ 天主教 教宗 若望保祿二世 因 感冒 再度 住進 醫院。

- But word segmentation does not guarantee a unique tokenization
  - *下雨天留客天留我不留 can be*
  - *下雨，天留客，天留，我不留！*
    - It's raining. The lord wants the guest to stay. Although the lord wants, I don't
  - *下雨天，留客天，留我不？ 留！*
    - It's a raining day. It is a day to let the guest stay. Will you let me stay? Stay!
  - *新西兰花*
    - 新西兰 花 => New Zealand flowers
    - 新 西兰花 => fresh broccolis

Alternative Method for Indexing without Segmentation

- Common method is called n-gram
  - *Don't segment (you could be wrong!)*
  - *Instead, treat every character n-gram as a term*
- Consider a Chinese sentence with characters: $c_1\ c_2\ c_3\ \ldots\ c_n$
  - *For example character bi-gram*

$$c_1\ c_2\ c_3\ c_4\ c_5\ \ldots\ c_n$$
$$\longrightarrow\ \$\ c_1\ ,\ c_1\ c_2\ ,\ c_2\ c_3,\ c_3\ c_4\ ,\ c_4\ c_5\ \ldots\ c_{n-1}\ c_n,\ c_n\ \$$$

  - *Break up queries the same way*
  - *Works at least as well as trying to segment correctly!*
- So what are the tri-gram of 新西兰花

Tokenization 4: Numbers

- ***Examples***
  - *3/12/91*
  - *Mar. 12, 1991*
  - *55 B.C.*
  - *B-52*
  - *My PGP key is 324a3df234cb23e*
  - *100.2.86.144*
- Generally, don't index as text.
- But more systems treat time, special numbers as tokens
  - *Often mark explicitly as date, part of a name etc.*

Tokenization 5: "of, to, in"

- Not all words are equally significant for representing the meaning of a document
  - *Apple, stories, computing vs of, to, are, for, the*
- Stopwords: words that do not have meanings
  - *Natural candidates: articles, prepositions, and conjunctions*
- Benefits of removing stopwords
  - *Stopwords appear too frequent among documents*
    - Some stopwords appear in 80% of documents
  - *Reduce the size of indexing structure considerably*
    - Reduce about 40% or more
- But need to be careful sometimes
  - *Think about "United States of America", "Library of Congress"*
  - *Think about what happen to "to be or not to be"*

Tokenization 6: Case Folding

- In general, reduce all letters to lower case
  - *Information Retrieval = information retrieval*
- But can have exceptions: upper case (in mid-sentence?)
  - *e.g., General Motors*
- And expect potential confusions
  - *FED vs Fed vs. fed*
    - *FED: FirstFed Financial Group (NYSE)*
    - *Fed: Federal Reserve*
  - *SAIL vs. sail*
    - *Signal Analysis And Interpretation Laboratory*

Tokenization 7: Morphological variation

- Morphological variation

  = *different forms of the same word*

  - *Inflectional morphology: same part of speech*

    break, broke, broken; sing, sang, sung;

    study, studied; invent, invented

    apple, apples; box, boxes; etc.

  - *Derivational morphology: different parts of speech*

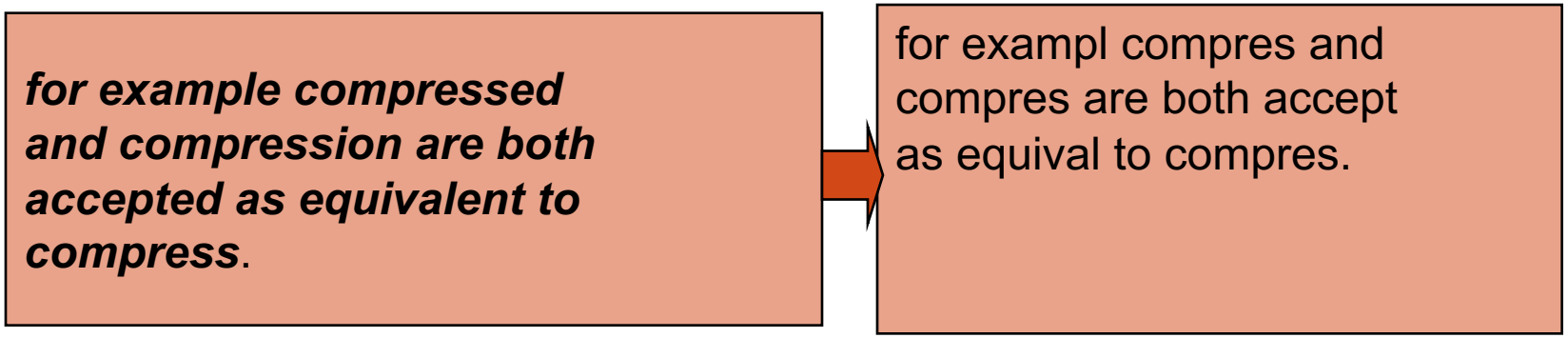
    destroy, destruction; invent, invention, reinvention

Approach 1: Lemmatization

- Reduce inflectional/some derivational variant forms to base form
- The base form is called "lemma" or "dictionary form"
  - *am, are, is* → *be*
  - *car, cars, car's, cars'* → *car*
- *Outcome of Lemmatization*
  - *the boy's cars are in different colors* → *the boy car be in different color*
- *Examples of Lemmatizers:*
  - *http://morphadorner.northwestern.edu/*

Approach 2: Stemming

- Conflate morphological variants to their "roots" , called "stems"
  - *stems might not be a correct word form, such as "stor", "comput"*
  - *language dependent*
  - *e.g.,* ***automate(s), automatic, automation*** *all reduced to* ***automat***.



- Different to lemmatization
  - *Stemming works most on derivational variants,*
  - *Lemmatization works most on inflectional variants*

Porter's algorithm

- Commonest algorithm for stemming English
  - *Code available at http://tartarus.org/~martin/PorterStemmer/*
  - *Another code source http://snowball.tartarus.org/*
- Conventions + 5 phases of reductions
  - *phases applied sequentially each phase consists of a set of commands*
  - *sample convention: Of the rules in a compound command, select the one that applies to the longest suffix.*
- Typical rules in Porter
  - *sses* → *ss*
  - *ies* → *i*
  - *ational* → *ate*
  - *tional* → *tion*

Other stemmers

- Other stemmers exist, e.g., Lovins stemmer
  - *http://www.comp.lancs.ac.uk/computing/research/stemming/general/lovins.htm*
  - *http://sourceforge.net/projects/stemmers/* *a java version*
- Single-pass, longest suffix removal (about 250 rules)
- Motivated by Linguistics as well as IR

Results of Stemming

Sample text: Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Lovins stemmer: such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

Porter stemmer: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Questions about Stemming

- What could be the effect of stemming?
  1. *Would stemming avoid missing relevant documents?*
  2. *Would stemming help to return only relevant documents?*
  3. *Would stemming reduce or increase the number of tokens extracted from a document?*
  4. *If stemming is applied to processing the documents, should it be applied to the query processing?*

Some Problems with Stemming

- Sometimes the conflation can be too aggressive
  1. *computers, computer, computing => the same root "comput"*
  2. *marketing, markets => the same root "market"*
  3. *universities, universe => the same root "univers"*
  4. *ignore, ignorant => the same root "ignor"*
- Missing some useful conflations
  - *E.g., "European"/ "Europe", "matrices"/ "matrix" etc.*
- Produce stems that are not words, so could be difficult for user to understand
  - *With Porter, "iter" and "gener" are the stems. What are the original words?*

Does Stemming Work?

- Generally, yes! (in English)
  - *Helps more for longer queries*
  - *Lots of work done in this area*

Donna Harman (1991) How Effective is Suffixing? Journal of the American Society for Information Science, 42(1):7-15.

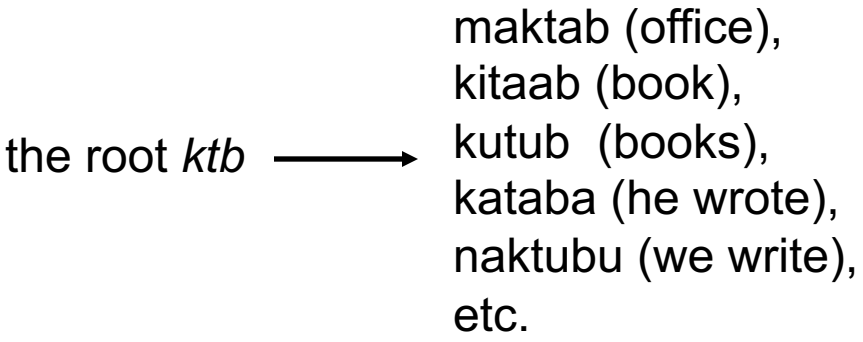
Robert Krovetz. (1993) Viewing Morphology as an Inference Process. Proceedings of SIGIR 1993.

David A. Hull. (1996) Stemming Algorithms: A Case Study for Detailed Evaluation. Journal of the American Society for Information Science, 47(1):70-84.

And others…

Stemming in Other Languages

- Arabic makes frequent use of infixes



- What's the most effective stemming strategy in Arabic? Open research question…

Idea: Words = wrong indexing unit!

- Synonymy

  = *different words, same meaning*

  {dog, canine, doggy, puppy, etc.} → concept of *dog*

- Polysemy

  = *same word, different meanings*

  Bank: financial institution or side of a river?
  Crane: bird or construction equipment?

- It'd be nice if we could index concepts!
  - *Word sense: a coherent cluster in semantic space*
  - *Indexing word senses achieves the effect of conceptual indexing*

Indexing Word Senses

- How does indexing word senses solve the synonym/polysemy problem?

  {dog, canine, doggy, puppy, etc.} → concept 112986

  I deposited my check in the bank. bank → concept 76529
  I saw the sailboat from the bank. bank → concept 53107

- Okay, so where do we get the word senses?
  - *WordNet: a lexical database for English*

    http://wordnet.princeton.edu/
  - *Automatically find "clusters" of words that describe the same concepts*
    - How?
  - *Other methods also have been tried…*

Word Sense Disambiguation

- Given a word in context, automatically determine the sense (concept)
  - *This is the Word Sense Disambiguation (WSD) problem*
- Context is the key:
  - *For each ambiguous word, note the surrounding words*

    bank {river, sailboat, water, etc.} → side of a river
    bank {check, money, account, etc.} → financial institution

  - *"Learn" a classifier from a collection of examples*
  - *Use the classifier to determine the senses of words in the documents*

Does it work?

- Nope!

  Ellen M. Voorhees. (1993) Using WordNet to Disambiguate Word Senses for Text Retrieval. Proceedings of SIGIR 1993.

  Mark Sanderson. (1994) Word-Sense Disambiguation and Information Retrieval. Proceedings of SIGIR 1994

  And others…

- Examples of limited success….

  Hinrich Schütze and Jan O. Pedersen. (1995) Information Retrieval Based on Word Senses. Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval.

  Rada Mihalcea and Dan Moldovan. (2000) Semantic Indexing Using WordNet Senses. Proceedings of ACL 2000 Workshop on Recent Advances in NLP and IR.

Why Disambiguation Hurts

- Bag-of-words techniques already disambiguate
  - *Context for each term is established in the query*
    - E.g., "apple's latest stock price" vs "apple's sale price at walmart's"
- WSD is hard!
  - *Many words are ambiguous: bank, apple, interest, stock, etc.*
  - *Clues are often far away*
    - People like apple because it is cool to have apple as part of their lives. It is such delicious fruit. Vs People like apple because it is cool to have apple as part of their lives. Just think about iphone, ipad, and iwatch.
  - *Granularity of senses is often domain/application specific*
- WSD tries to improve precision
  - *But incorrect sense assignments would hurt recall*

Indexing Phrases

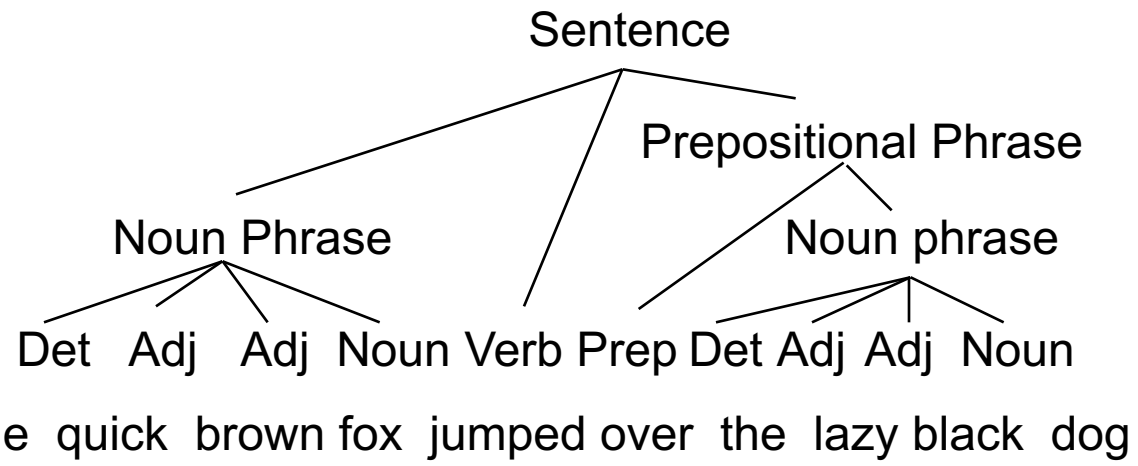
- Why use phrases?
  - *"white house" is more precise than "white AND house", "white, house"*
- Two types of phrases
  - *Those that make sense, e.g., "school bus", "hot dog"*
  - *Those that don't, e.g., bigrams in Chinese*
- Treat multi-word tokens as index terms
- Three sources of evidence for identifying phrases
  - *Dictionary lookup*
  - *Linguistic analysis*
  - *Statistical analysis (e.g., co-occurrence)*

Phrase Recognition: Dictionary Lookup

- Compile a term list that includes phrases
  - *Technical terminology can be very helpful*
- Index any phrase that occurs in the list
- Most effective in a limited domain
  - *Otherwise hard to capture most useful phrases*

Phrase Recognition: Syntactic Parsing

- Parsing = automatically assign structure to a sentence



- "Walk" the tree and extract phrases
- Very slow, common in NLP, but not in IR community
- Stanford Parser
  - *http://nlp.stanford.edu/software/lex-parser.shtml*

Syntactic Variations

- What does linguistic analysis buy?
  - *Coordinations*

    lung and breast cancer →
    lung cancer, breast cancer

  - *Substitutions*

    inflammatory sinonasal disease →
    inflammatory disease, sinonasal disease

  - *Permutations*

    addition of calcium → calcium addition

Phrase Recognition: POS Tagging

- Assign part of speech (POS) tags
  - *With a probabilistic or rule based POS tagger*
  - *Example: The/DT white/JJ house/NN announced/VBD yesterday/NN …*
- Match phrases by POS patterns
  - *Example: NN+ (house), JJ NN+ (white house)*
- Reasonable fast, but slower than statistical recognition
- Used often in the IR community
- Stanford POS Tagger
  - *http://nlp.stanford.edu/downloads/tagger.shtml*

Phrase Recognition: Statistical Analysis

- Consider all word bigrams
  - *"The white house announced yesterday …"* -> *"the white"*, *"white house"*, *"house announced"*, *"announced yesterday"*, …
- Automatically discover phrases based on co-occurrence probabilities
  - *How much more do words x and y co-occur than if they were independent?*
  - ***Using Pointwise mutual information (PMI)***:

$$\mathrm{PMI}(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

  - *The words bigrams that give highest PMI values can be considered as phrases*
- Use this method to automatically learn a phrase dictionary
- Very fast, often used in IR community

How are Phrases Used

- Precoordinate
  - *Recognize phrase during indexing*
    - Insert phrase into index, e.g., "white_house"
    - Possibly also insert constituents into index, e.g., "white", "house"
  - *Replace query constituents with phrase*
    - E.g. "white_house"
- Postcoordinate
  - *Don't recognize phrases at indexing time*
  - *Specify phrase at query time*
    - Insert a query operator e.g., #3(white house)
  - *Query operator retrieves and merges constituent inverted lists*

Does Phrasal Indexing Work?

- In a very large corpora, yes…
  - *E.g. the Web …*
  - *High precision effects are desirable*
  - *Loss of recall is not a problem, why?*
- In smaller corpora it isn't clear
  - *High precision effects are still desirable*
  - *But loss of recall is more a problem, why?*
  - *Mixed results in most studies*
    - 5% improvement?

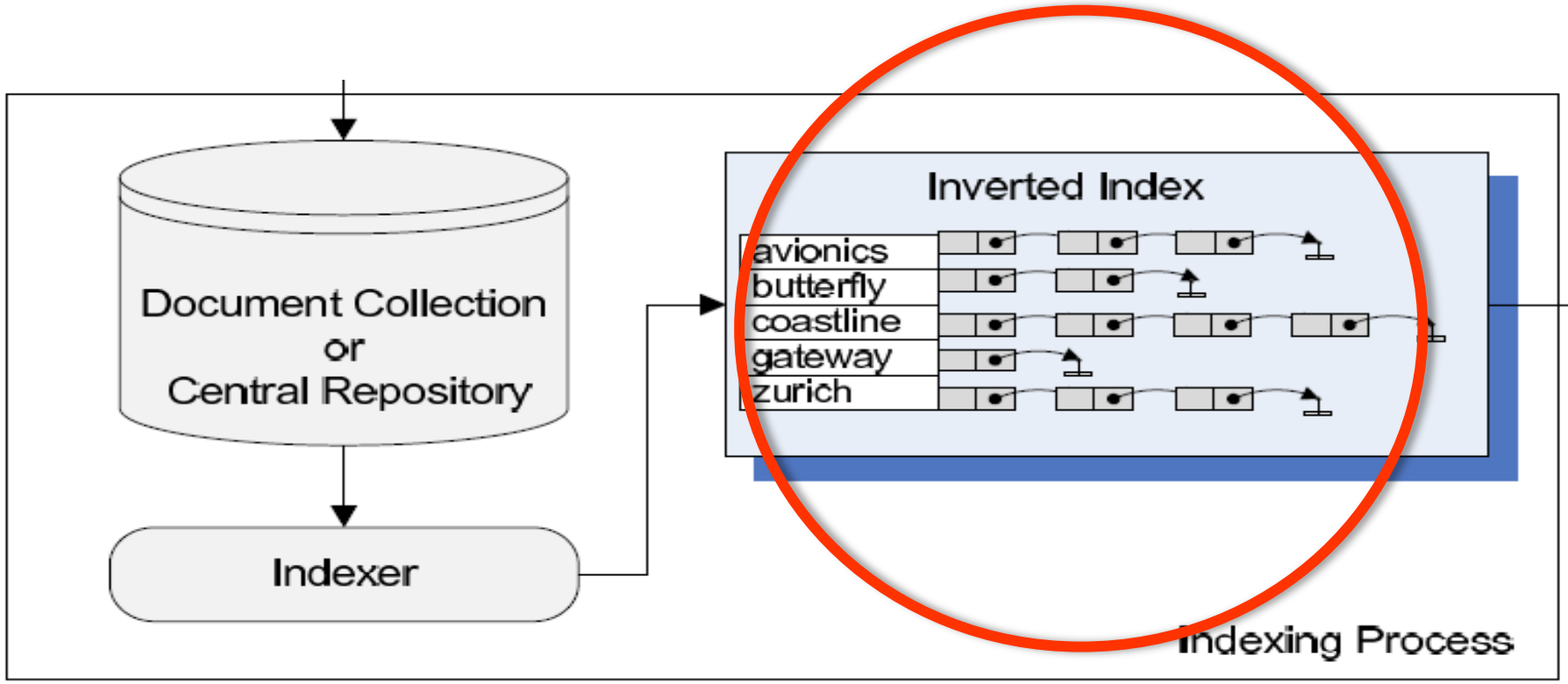
Bag of Word Representation –Example

People at San Francisco Apple site say that an apple employee loves to peel apples with a passion of love

The index terms after tokenization with stopword removing, case folding, stemming and phrase identification

peopl	San_Francisco	appl
site	sai	employe
love	peel	passion

Document Processing and Indexing



Term Project: Introduction

Term Project

- Task
  - *Work in a team of 3 people*
  - *Work on an IR related project, develop and deploy the prototype system*
- In three categories
  - *Group 1:Web Retrieval Enhancements*
    - Various online sites could have better search capabilities
  - *Group 2: Mobile Search*
    - Various online sites can have better mobile access capabilities
  - *Group 3: Search on existing test collections*
    - TREC: http://trec.nist.gov/
    - CLEF: http://clef2017.clef-initiative.eu/index.php?page=Pages/labs_info.php
    - NTCIR: http://research.nii.ac.jp/ntcir/index-en.html

Open Source Resources

- Lucene: Lucene Core, Solr, PyLucene, Elasticsearch
  - *Lucene.apache.org*
- Lemur Toolkits
  - *http://www.lemurproject.org/lemur/*
  - *Toolkits are in C++, interface is in JAVA Swing*
- Indri search engine
  - *http://www.lemurproject.org/indri/*
  - *In C++*
- Terrier
  - *http://terrier.org/*
  - *umIn JAVA*

Online Sources

- IRIS@Pitt group:
  http://crystal.exp.sis.pitt.edu:8080/iris/index.jsp
- LTI@CMU: https://www.lti.cs.cmu.edu/work
- CIIR@UMass: https://ciir.cs.umass.edu/download
- NLP@Stanford: https://nlp.stanford.edu/software/
- TIMAN@UIUC: http://sifaka.cs.uiuc.edu/ir/downloads.html
- TREC: https://trec.nist.gov/data.html

Time Table

- Introduction of term project: Lecture 2
- Team formation/project proposal: Lecture 4
  - *Send email to TA and me before 5pm on Monday of Lecture 4 the latest*
    - List team members: name and email
    - A short intro of your project
- Project initial presentation: Lecture 8
  - *Up to 5 minutes Jing Video for a intermediary project progress*
  - *Post online and save link in the courseweb*
- Final project Presentation: Lecture 14
- Project Demo: Last week

Project Intro Examples

- **Project Title: Enhancement of Spotify's Search Capabilities**
- **Background:** Spotify is a music streaming application providing a vast amount of digital rights managements-protected content. It provides searching by artist name, album, genre, and mode of the user. However, it does not provide searching by lyrics of a song. Hence, the goal of this project is to enhance search capability of Spotify so that users can find songs when they do not remember the name of the artist or song but part of lyrics. To provide this capability, we will use YouTube API. As optional enhancement, retrieving the complete lyrics of songs and biography of artists may be considered after a feasibility analysis.
- **Outcome:** An online web application which provides a user interface to search for song lyrics in the collection of Spotify in addition to existing searching capabilities.

Project Intro Examples

Project title: Stack Overflow Search for android

- **Background:** Stack Overflow is a huge collection of online questions and their answers. There are many searches performed there everyday to look for questions and answers satisfying the queries, either as keywords or a whole natural language question. This project is to explore a nice mobile search interface for retrieving similar questions for a given query on Android platform. You have the freedom to define what is the right Stack Overflow mobile search and presentation on a smartphone screen, and make it fashion, make it right.
- **Outcome:**
  a nice interface for searching and presenting Stack Overflow data on android phone.

Project Intro Examples

Project title: TREC Clinical Decision Support Track

- **Background:** Clinical Decision Support Track retrieves biomedical articles relevant for answering generic clinical questions. The retrieval will use admission notes from MIMIC-III, which describe a patient's chief complaint, relevant medical history, and any other information obtained during the first few hours of a patient's hospital stay, such as lab work. This project will retrieve full-text biomedical articles that answer questions related to several types of clinical information needs for a given EHR note. Retrieved articles will be judged relevant if they provide information of the specified type that is pertinent to the given patient.
- http://www.trec-cds.org/2016.html
- **Outcome:** a search engine with explicit design for handling medical related issues in returning biomedical articles