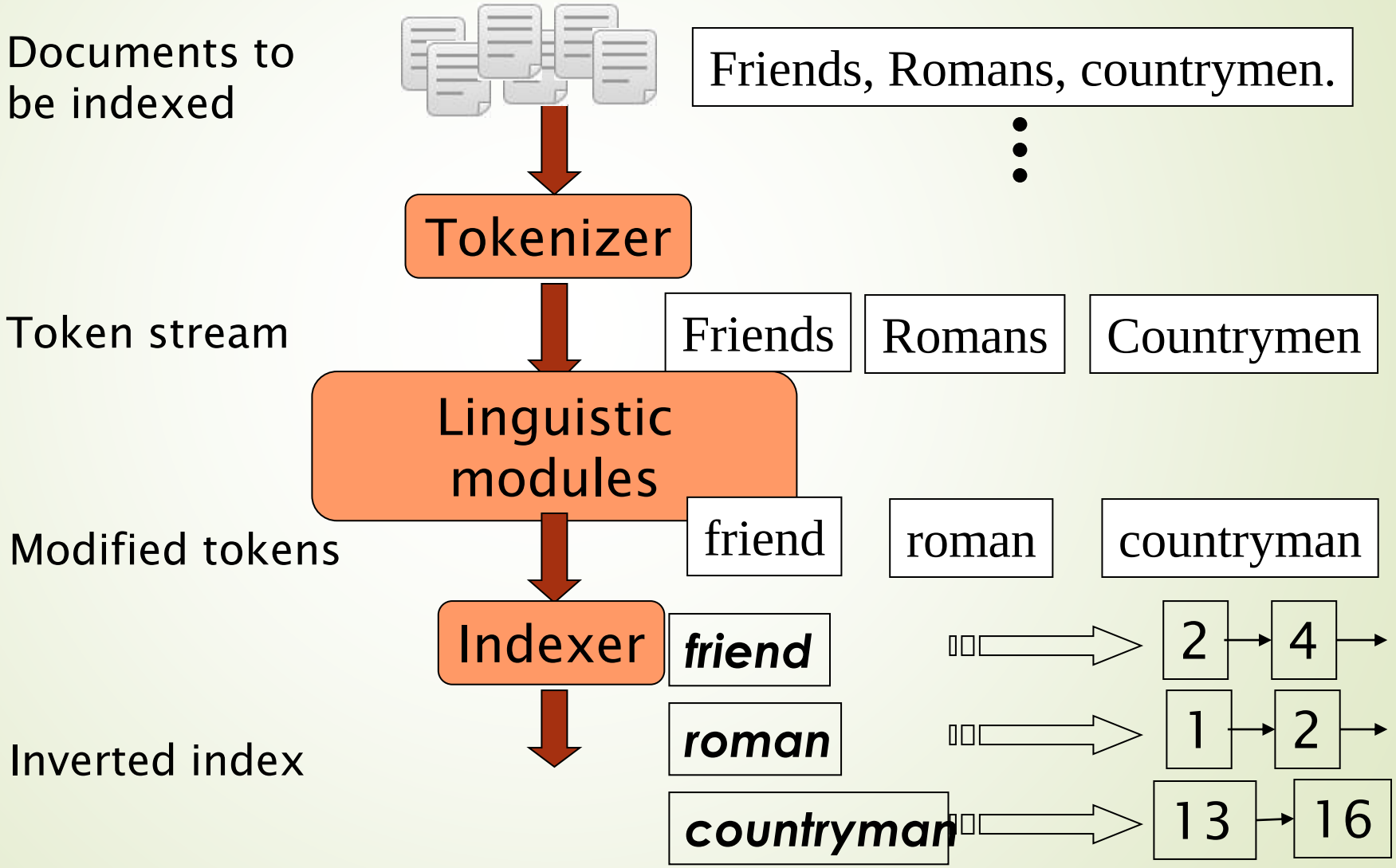
Lect 5: Term Vocabulary

Dr. Subrat Kumar Nayak
Associate Professor
Dept. of CSE, ITER, SOADU

Recall the basic indexing pipeline



Recall the basic indexing pipeline

- Tokenization
  - Cut character sequence into word tokens
    - Deal with ***"John's"***, ***a state-of-the-art solution***
- Normalization
  - Map text and query term to same form
    - You want ***U.S.A.*** and ***USA*** to match
- Stemming
  - We may wish different forms of a root to match
    - ***authorize***, ***authorization***
- Stop words
  - We may omit very common words (or not)
    - ***the, a, to, of***

Parsing a document

- What format is it in?
  - pdf/word/excel/html?
- What language is it in?
- What character set is in use?
  - (CP1252, UTF-8, …)

Each of these is a classification problem, which we will study later in the course.

But these tasks are often done heuristically …

Complications: Format/language

- Documents being indexed can include docs from many different languages
  - A single index may contain terms from many languages.
- Sometimes a document or its components can contain multiple languages/formats
  - French email with a German pdf attachment.
  - French email quote clauses from an English-language contract
- There are commercial and open source libraries that can handle a lot of this stuff

Complications: What is a document?

We return from our query “documents” but there are often interesting questions of grain size:

What is a unit document?

- A file?
- An email? (Perhaps one of many in a single mbox file)
  - What about an email with 5 attachments?
- A group of files (e.g., PPT or LaTeX split over HTML pages)

Tokenization

- **Tokenization** is the task of chopping it up into pieces, called *tokens*, perhaps at the same time.
- throwing away certain characters, such as punctuation.
- Input: "***Friends, Romans and Countrymen***"
- Output: Tokens
  - ***Friends***
  - ***Romans***
  - ***Countrymen***
- A **token** is an instance of a sequence of characters
- Each such token is now a candidate for an index entry, after further processing
  - Described below
- But what are valid tokens to emit?

Tokenization

- A ***type*** is the class of all tokens containing the same character sequence. A ***term*** is a (perhaps normalized) type that is included in the IR system's dictionary.
- For example, if the document to be indexed is ***to sleep perchance to dream***, then there are **5 tokens**, but **only 4 types** (since there are 2 instances of *to*). However, if ***to*** is omitted from the index (**as a stop word**), then there will be **only 3 terms**: *sleep*, *perchance*, and *dream*.

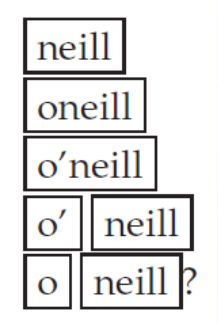
Tokenization

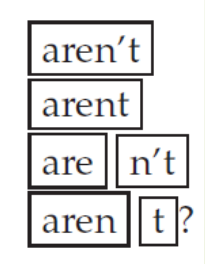
Issues in tokenization:

- The major question of the tokenization phase is **what are the correct tokens to use**? In the previous examples, it looks fairly trivial: you chop on whitespace and throw away punctuation characters.
- But for English there **are a number of tricky cases**. For example, what do you do about **the various uses of the apostrophe** for possession and contractions?

Example: Mr. **O'Neill** thinks that the boys' stories about Chile's capital **aren't** amusing.

For *O'Neill and aren't*, which of the following is the desired tokenization?





Tokenization

- Issues in tokenization:
  - ***Finland's capital*** $\rightarrow$

    Finland AND ***s***? ***Finlands***? ***Finland's***?
  - ***Hewlett-Packard*** $\rightarrow$ ***Hewlett*** and ***Packard*** as two tokens?
    - ***state-of-the-art***: break up hyphenated sequence.
    - ***co-education***
    - ***lowercase***, ***lower-case***, ***lower case*** ?
    - It can be effective to get the user to put in possible hyphens
  - ***San Francisco***: one token or two?
    - How do you decide it is one token?

Numbers

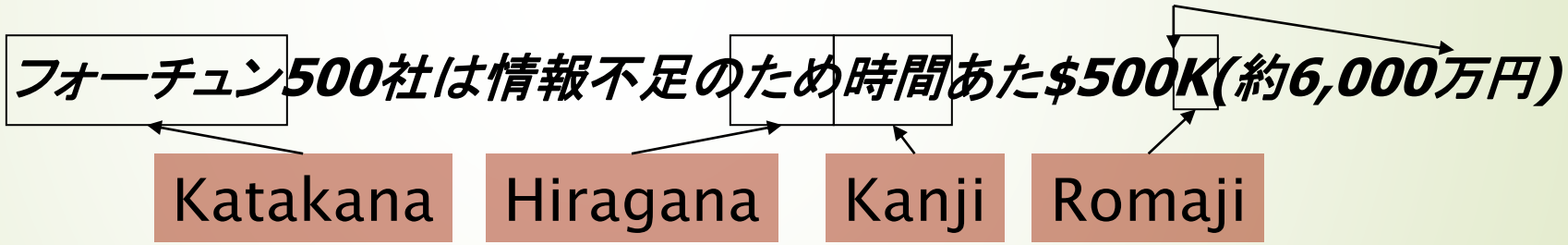
- ***3/20/91*** ***Mar. 12, 1991*** ***20/3/91***
- ***55 B.C.***
- ***B-52***
- ***My PGP key is 324a3df234cb23e***
- ***(800) 234-2333***
  - Often have embedded spaces
  - Older IR systems may not index numbers
    - But often very useful: think about things like looking up error codes/stacktraces on the web
    - (One answer is using n-grams: IIR ch. 3)
  - Will often index “meta-data” separately
    - Creation date, format, etc.

Tokenization: language issues

- These issues of tokenization are **language-specific**. It thus requires the language of the document to be known.
- ***Language identification*** based on classifiers that use short character subsequences as features is highly effective; most languages have distinctive signature patterns.
- French
  - ***L'ensemble*** → one token or two?
    - ***L*** ? ***L'*** ? ***Le*** ?
    - Want ***l'ensemble*** to match with ***un ensemble*** *and to be indexed under* ***ensemble.***
      - Until at least 2003, it didn't on Google
        - Internationalization!
- German noun compounds are not segmented. They write compound nouns without spaces. (*Computerlinguistik* 'computational linguistics')
  - ***Lebensversicherungsgesellschaftsangestellter***
  - 'life insurance company employee'
- German retrieval systems benefit greatly from a **compound splitter** module which is usually implemented by seeing if a word can be subdivided into multiple words that appear in a vocabulary
      - Can give a 15% performance boost for German

Tokenization: language issues

- Chinese and Japanese have no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - Not always guaranteed a unique tokenization
- Further complicated in Japanese, with multiple alphabets intermingled
  - Dates/amounts in multiple formats



End-user can express query entirely in hiragana!

Tokenization: language issues

- Arabic (or Hebrew) is basically written right to left, but with certain items like numbers written left to right
- Words are separated, but letter forms within a word form complex ligatures

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

- ← → ← → ← start
- ‘Algeria achieved its independence in 1962 after 132 years of French occupation.’
- With Unicode, the surface presentation is complex, but the stored form is straightforward