Introduction to Informa l

Lecture 2: Pr g

Recap of the previ ure

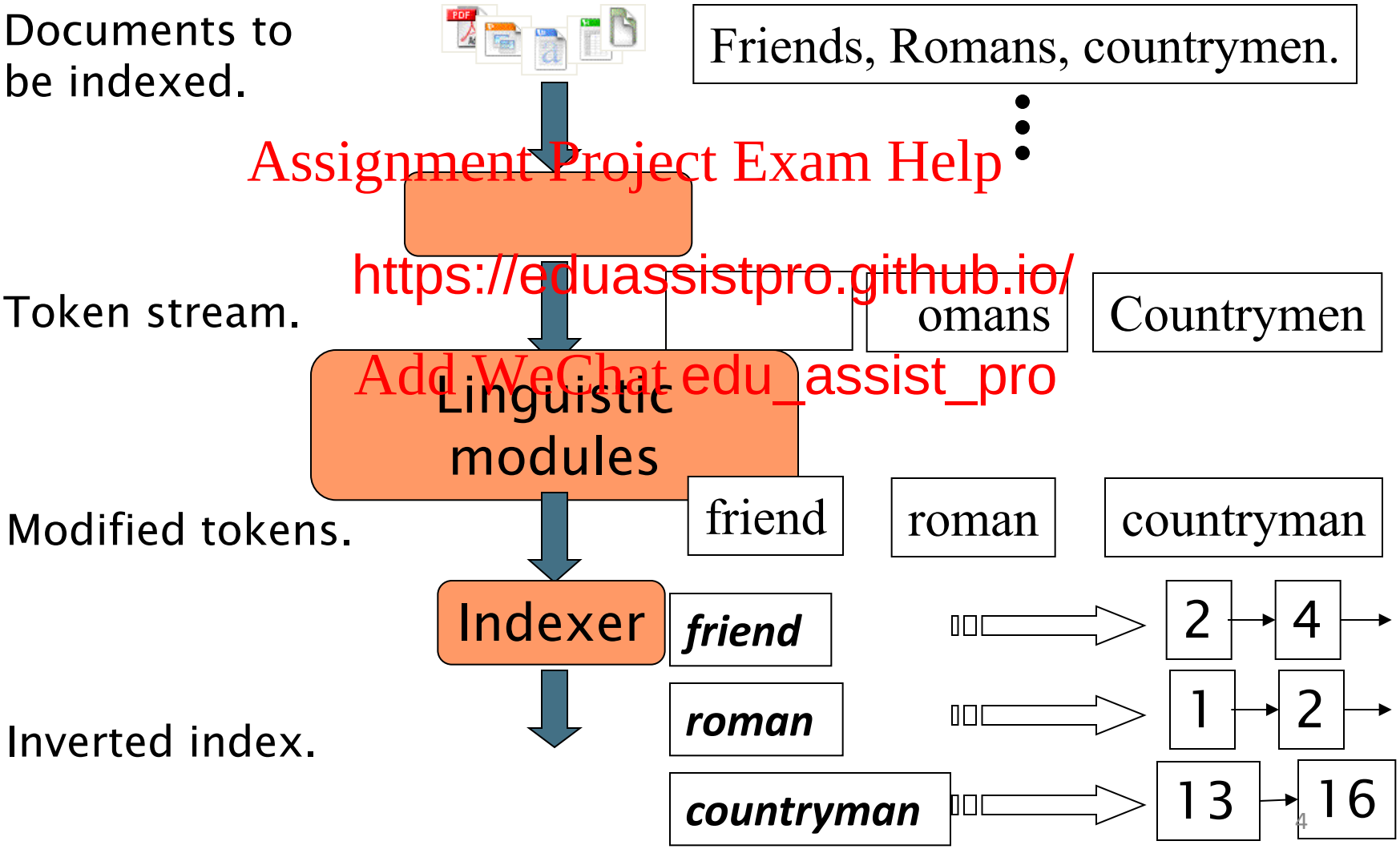
- Basic inverted indexes:
  - Structure: Dictionary and Postings
  - Key step in construction: S
- Boolean query processing
  - Intersection by linear time “merging”
  - Optimizations
  - Positional index

Plan for this lecture

Elaborate basic indexing

- Preprocessing to form the term vocabulary
  - Document
  - Tokenization
  - What *terms* do we put in?

Recall the basic indexing pipeline



Parsing a document

- What format is it in?
  - pdf/word/excel/html?

- What langua
- What charac

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

But these tasks are often done heuristically …

Complications: Form age

- Documents being indexed can include docs from many different languages
  - A single index may have to contain terms of several languages.
- Sometimes a ments can contain multiple languages/
  - French email with a German ent.
- What is a unit document?
  - A file?
  - An email? (Perhaps one of many in an mbox.)
  - An email with 5 attachments?
  - A group of files (PPT or LaTeX as HTML pages)

TOKENS AND TERMS

Tokenization

- Input: "***Friends, Romans and Countrymen***"
- Output: Tokens
  - ***Friends***
  - ***Romans***
  - ***Countrymen***
- A token is an instance of a s f 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

- Issues in tokenization:
  - ***Finland's capital*** →
    Finland? F

  - ***Hewlett-Pa*** ***Packard*** as two tokens?

    - ***state-of-the-art***: break up hyp nce.
    - ***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 3***
- ***(800) 234-2333***
  - Often have embedded space
  - 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**: Lecture 3)
  - Will often index "meta-data" separately
    - Creation date, format, etc.

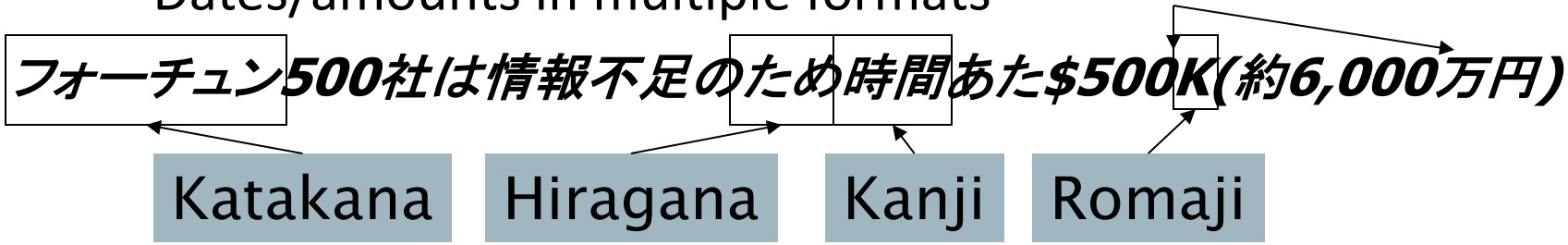
Tokenization: lang ues

- French
  - ***L'ensemble*** → one token or two?
    - ***L*** ? ***L'*** ? ***Le*** ?
    - Want ***l'ense*** ***le***
      - Until at
        - Internationalization!

- German noun compounds are not segmented
  - ***Lebensversicherungsgesellschaftsangestellter***
  - 'life insurance company employee'
  - German retrieval systems benefit greatly from a **compound splitter** module
    - Can give a 15% performance boost for German

Tokenization: lang ues

- 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 nization
- Further complicated in Jap h multiple alphabets intermingled
  - Dates/amounts in multiple formats



End-user can express query entirely in hiragana!

Tokenization: lang ues

- Arabic (or Hebrew) is basically written right to left, but with certain items like numbers written left to right

- Words are se ms within a word form complex

- ← → ← → ← 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

Stop words

- With a stop list, you exclude from the dictionary entirely the commonest words. Intuition:

  - They have little semantic content: *the, a, and, to, be*
  - There are a l r top 30 words

- But the tren his:

  - Good compression techniques (l ns the space for including stopwords in a system is very small
  - Good query optimization techniques (lecture 7) mean you pay little at query time for including stop words.
  - You need them for:
    - Phrase queries: “King of Denmark”
    - Various song titles, etc.: “Let it be”, “To be or not to be”
    - “Relational” queries: “flights to London” vs. “flights from London”

Normalization to t

- We need to "normalize" words in indexed text as well as query words into the same form
  - We want to match ***U.S.A.*** and ***USA***
- Result is terms ed) word type, which is an en ctionary
- We most commonly implicitly de nce classes of terms by, e.g.,
  - deleting periods to form a term
    - ***U.S.A., USA → USA***
  - deleting hyphens to form a term
    - ***anti-discriminatory, antidiscriminatory → antidiscriminatory***

Normalization: Other languages

- Accents: e.g., French ***résumé*** vs. ***resume*.**
- Umlauts: e.g., German: ***Tuebingen*** vs. ***Tübingen***
  - Should be equivalent

- Most importa

  - How are your users like to w eries for these words?

- Even in languages that standardly have accents, users often may not type them
  - Often best to normalize to a de-accented term
    - ***Tuebingen, Tübingen, Tubingen*** ***Tubingen***

Normalization: Other languages

- Normalization of things like date forms
  - ***7月30日 vs. 7/30***
  - ***Japanese use of kana vs. Chinese characters***

- Tokenization and normalization may depend language and so is intertwi nguage detection

Morgen will ich in MIT …

Is this German "mit"?

- Crucial: Need to "normalize" indexed text as well as query terms into the same form

Case folding

- Reduce all letters to lower case
  - exception: upper case in mid-sentence?
    - e.g., ***General Motors***
    - ***Fed*** vs. ***fed***
    - ***SAIL*** vs. ***sail***
  - Often best to lower case everything, since users will use lowercase regardless of 'correct' capitalization…
- Google example:
  - Query ***C.A.T.***
  - #1 result is for "cat" (well, Lolcats) *not* Caterpillar Inc.

Normalization to t

- An alternative to equivalence classing is to do asymmetric expansion

- An example o sеful

  - Enter: ***window*** Search: ***windo***

  - Enter: ***windows*** Search: ***Windo*** ***window***
  - Enter: ***Windows*** Search: ***Windows***
- Potentially more powerful, but less efficient

Thesauri and soun

- Do we handle synonyms and homonyms?
  - E.g., by hand-constructed equivalence classes
    - ***car*** = ***automobile*** ***colour*** = ***color***
  - We can rewr ass terms
    - When the d , index it under ***car-automobile***
  - Or we can expand a query
    - When the query contains ***automobile***, look under ***car*** as well
- What about spelling mistakes?
  - One approach is soundex, which forms equivalence classes of words based on phonetic heuristics
- More in lectures 3 and 9

Lemmatization

- Reduce inflectional/variant forms to base form
- E.g.,
  - *am, are, is* → *be*
  - *car, cars, car*
- *the boy's cars are different* → *the boy car be different color*
- Lemmatization implies doing "proper" reduction to dictionary headword form

Stemming

- Reduce terms to their "roots" before indexing
- "Stemming" suggest crude affix chopping
  - language dependent
  - e.g., ***automa*** ***tion*** all reduced to ***automat***.

for example compressed and compression are both accepted as equivalent to compress.

for exampl compress and compress ar both accept as equival to compress

Other stemmers

- Other stemmers exist, e.g., Lovins stemmer
  - http://www.comp.lancs.ac.uk/computing/research/stemming/general/lovins.htm
  - Single-pass, longest suffix removal (about 250 rules)

- Full morpholo t modest benefits for r

- Do stemming and other no s help?
  - English: very mixed results. Helps recall for some queries but harms precision on others
    - E.g., operative (dentistry) ⇒ oper
  - Definitely useful for Spanish, German, Finnish, …
    - 30% performance gains for Finnish!

Language-specific

- Many of the above features embody transformations that are
  - Language-specific and
  - Often, applic
- These are "pl indexing process
- Both open source and com g-ins are available for handling these

Dictionary entries – II

ensemble.french
時間.japanese
MIT.english
mit.german
guaranteed.english
entries.english
sometimes.english
tokenization.english

These may be grouped by language (or not…).
More on this in ranking/query processing.

Resources for today

- IIR 2
- MG 3.6, 4.3; MIR 7.2
- Porter's stem
  http://www.tarta er/