Information Retrieval

Dra. Mireya Paredes
mireya.paredes@udlap.mx

Inverted Index

1. Collect the documents to be indexed.

2. Tokenize the text.

3. Do linguistic preprocessing of tokens.

4. Index the documents that each term occurs in.

Document decoding and unit

- Convert this byte sequence into a linear sequence of characters.

- Determine the correct encoding.

How would you do that?

machine learning classification

Choosing a document unit

- Email inbox → split into various documents
- Attach zip files → unzip it first

The other way around:

- *Latex* file
- *Powerpoint* file

Index granularity

- Usually for very **LONG** documents

A collection of books

“CHINESE TOYS”

Tokenization per book
First chapter → China
Last chapter→ toys

Indexing granularity

- Using **paragraphs** as mini-documents.
- Using **individual sentences** as mini-documents.
- There is a **precision/recall** trade off.

TOKENIZATION

A **token** is an instance of a sequence of characters that are grouped together as a useful semantic unit for processing.
A **type** is the class of all tokens containing the same character sequence.

A **term** is a type that is included in the IR's system (it is part of the dictionary).

"To sleep perchance to dream" →4 types

What are the correct tokens to use?

“Mr. **O’Neill** thinks that the boys’ stories about Chile’s capital aren’t amusing”.

Neill

Oneill

o’neill

o’neill

o neill?

aren’t

arent

are n’t

arent?

Do the exact tokenization of a document and query words.

Language specific problems

- We need to know the language of a document.
- Language identification based on classifiers.
- Most languages have specific patterns.

Unusual terms

- C++, C#
- Email addresses
- Web URLs
- Numeric IP address
- Package tracking numbers

An option is to omit these terms but it limits what people can search for.

Hyphens

- Co-education
- Hewlett-Packard
- The hold-him-back-and-drag-him-away maneuver

- This is rather complex.

Stop words

- Words of a little value to the search.
- To sort the terms by collection frequency.
- To generate a *stop list*.
- The terms in the *stop list* is not taken into account during *indexing*.

“President of the United States”

Normalization

- There are similar terms with slightly differences.

“USA” and “U.S.A.”

antidiscriminatory and *anti-discriminatory*

- To create an **equivalence class**.

Normalization: It is the process of canonicalizing tokens so that matches occur despite superficial diferences.

Accents and diacritics

Cliché cliché

Normalizing tokens to remove *diacritics*.

“Tú tienes que estudiar para aprobar los exámenes”.
“En **tu** casa tenemos planeado ver la película este fin de semana”.

Accents and diacritics

Cliché cliché

Normalizing tokens to remove *diacritics*.

“**Tú** tienes que estudiar para aprobar los exámenes”.
“En **tu** casa tenemos planeado ver la película este fin de semana”.

Capitalization/case-folding

Case-folding -> Reducing all letter to lower case.

Exceptions:

Company names
Government organizations
Person names

“General Motors”
“Mireya Paredes”

Any idea to solve this problem?

Stemming and lemmatization

“Organize” “Organizes” “Organizing”

“Democracy” “Democratic”   “Democratization”

Am, are and is → verb to be

Stemming → A crude heuristic process that chops off the ends of words.

Lemmatization→ Aims to remove inflectional endings only and return the base.

Porter´s algorithm

- Stemming algorithm for English.
- It consists of 5 phases applied sequentially.

RULE	**Example**
SSES → SS	caresses → caress
IES → I	ponies → poni
SS → SS	caress → caress
S →	cats→cat

Rules

- Measure a ***word*** to check if it is long enough.

RULE → (m>1) **EMENT**

*Replac**ement*** →*replac*

Cement → cement

HOMEWORK

- To investigate what a **LEMMATIZER** is?
- What is the difference between A **STEMMER** and a **LEMMATIZER**?
- To give three examples of **LEMMATIZING**