Text Pre-processing

Dr. Arshin Rezazadeh

CS 4417B/9117/9647

The University of Western Ontario

Stopwords

- Function words (determiners, prepositions) have little meaning on their own (a, an, the, to, or, ...)
- High occurrence frequencies – take up a lot of space in indices
- Treated as stopwords (i.e. removed)
  - reduce index space, improve response time, improve effectiveness
- Can be important in combinations – e.g., “to be or not to be”

Stopwords

- Stopword list can be created from high-frequency words, or can be based on a standard list

- Lists are customized for applications, domains, and even parts of documents
    - e.g., “click” is a good stopword for anchor text

- Best policy is often to index all words in documents, make decisions about which words to use at query time

Aside: Lucene

- Lucene is an open-source text processing library
  - Part of the Apache family, written in Java, other languages are supported. http://lucene.apache.org
  - Used by Wikipedia (and many other projects)

- Typical use: process documents with an Analyzer, then produce an Index.

- StandardAnalyzer – most basic analyzer
  - Tokenize according to Unicode Standard Annex #49
  - Convert to lower case
  - Remove stopwords

Stemming

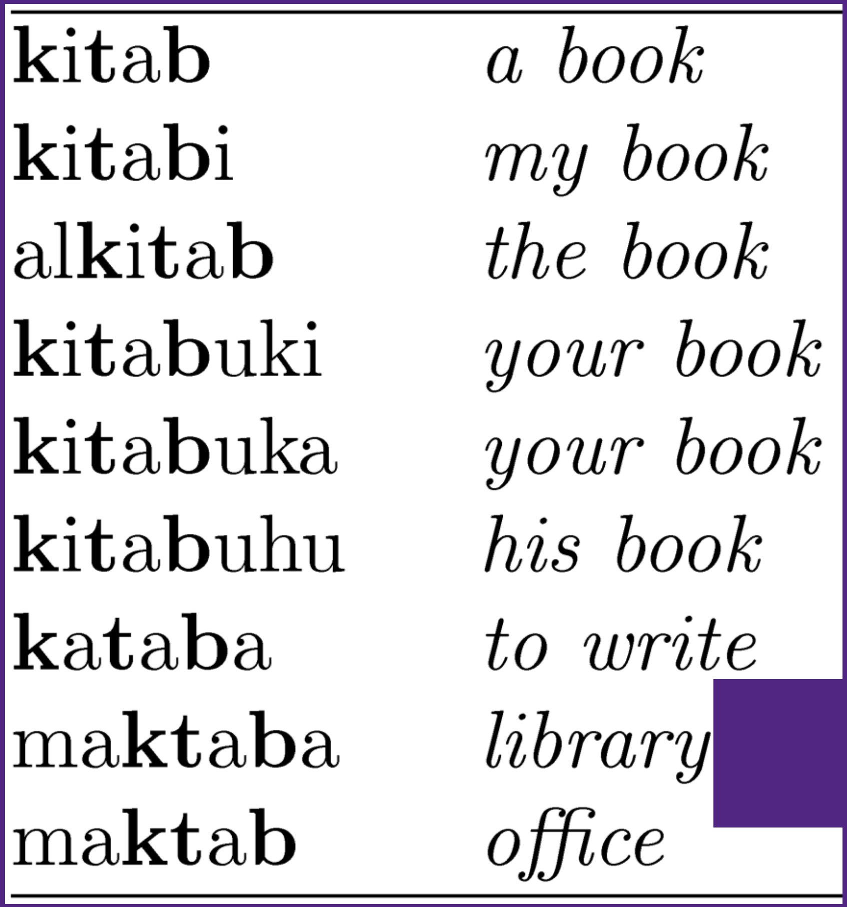
- Many *morphological variations* of words
  - inflectional (plurals, tenses) *book, books*
  - derivational (making verbs nouns etc.) *game, gamer, gaming*
- Different variations often have related meaning
  - "Related enough" depends on application; if we search for *gamer* do we want documents containing *game*?
- Stemmers attempt to simplify morphological variations of words to a common stem
  - usually involves removing suffixes
- Can be done at indexing time and/or as part of query processing

Stemming

- Generally a small but significant effectiveness improvement
- can be crucial for some languages
  - e.g., 5-10% improvement for English, up to 50% in Arabic

Arabic words with the **ktb** root

ك ت ب



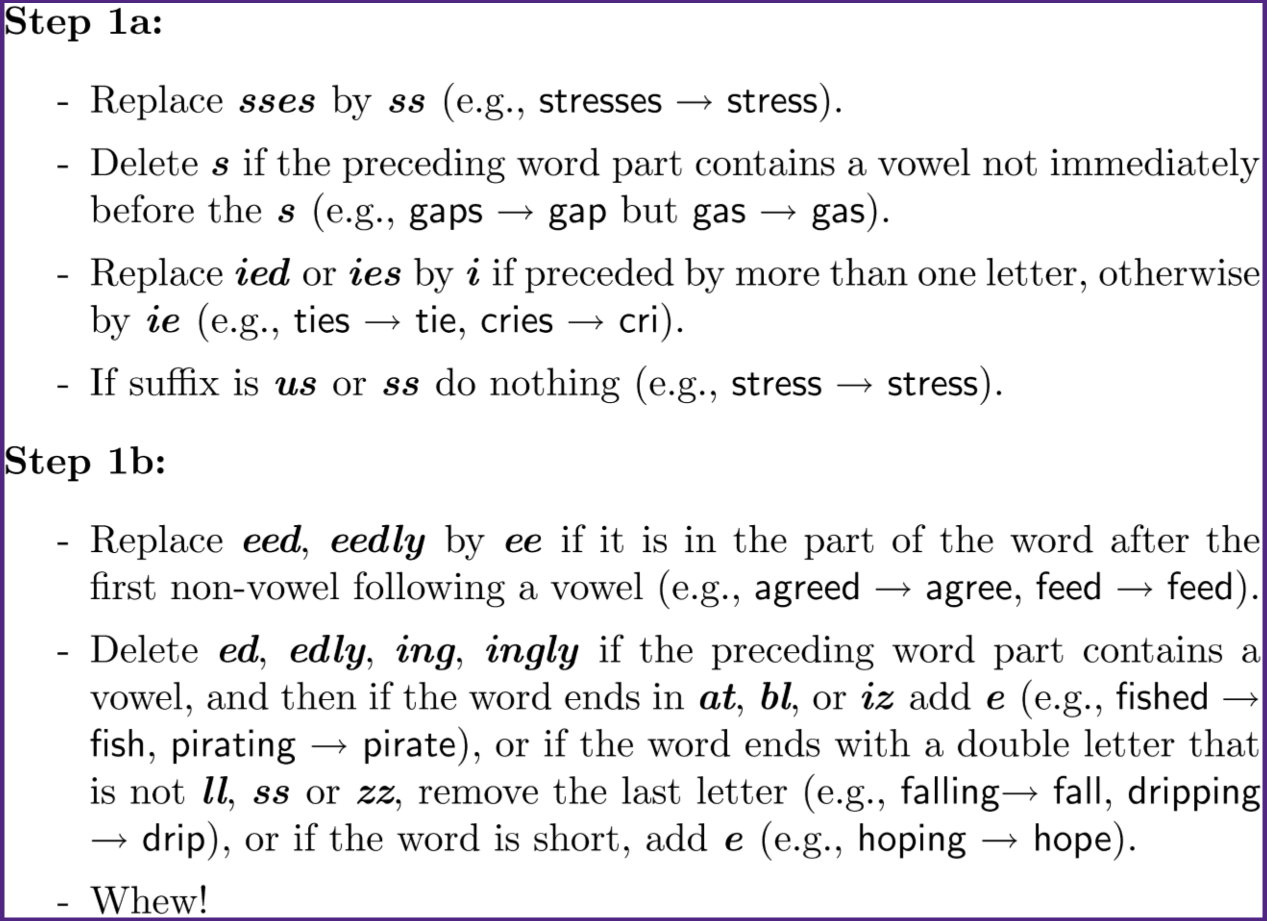
kitab	*a book*
kitabi	*my book*
alkitab	*the book*
kitabuki	*your book*
kitabuka	*your book*
kitabuhu	*his book*
kataba	*to write*
maktaba	*library*
maktab	*office*

Stemming

- Two basic types
  - Dictionary-based: uses lists of related words (sometimes people consider these *lemmatizers* instead)
  - Algorithmic: uses program to determine related words

BCC Ch. 3.1.2

Porter Stemmer



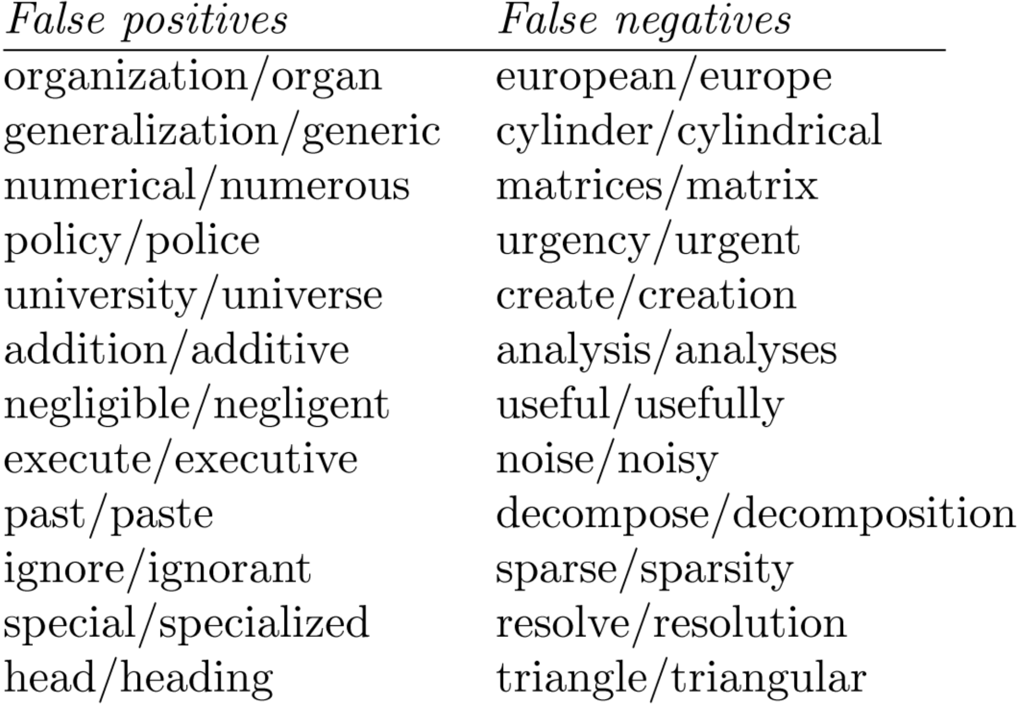
Step 1a:

- Replace ***sses*** by ***ss*** (e.g., stresses → stress).
- Delete ***s*** if the preceding word part contains a vowel not immediately before the ***s*** (e.g., gaps → gap but gas → gas).
- Replace ***ied*** or ***ies*** by ***i*** if preceded by more than one letter, otherwise by ***ie*** (e.g., ties → tie, cries → cri).
- If suffix is ***us*** or ***ss*** do nothing (e.g., stress → stress).

Step 1b:

- Replace ***eed***, ***eedly*** by ***ee*** if it is in the part of the word after the first non-vowel following a vowel (e.g., agreed → agree, feed → feed).
- Delete ***ed***, ***edly***, ***ing***, ***ingly*** if the preceding word part contains a vowel, and then if the word ends in ***at***, ***bl***, or ***iz*** add ***e*** (e.g., fished → fish, pirating → pirate), or if the word ends with a double letter that is not ***ll***, ***ss*** or ***zz***, remove the last letter (e.g., falling→ fall, dripping → drip), or if the word is short, add ***e*** (e.g., hoping → hope).
- Whew!

Porter Stemmer

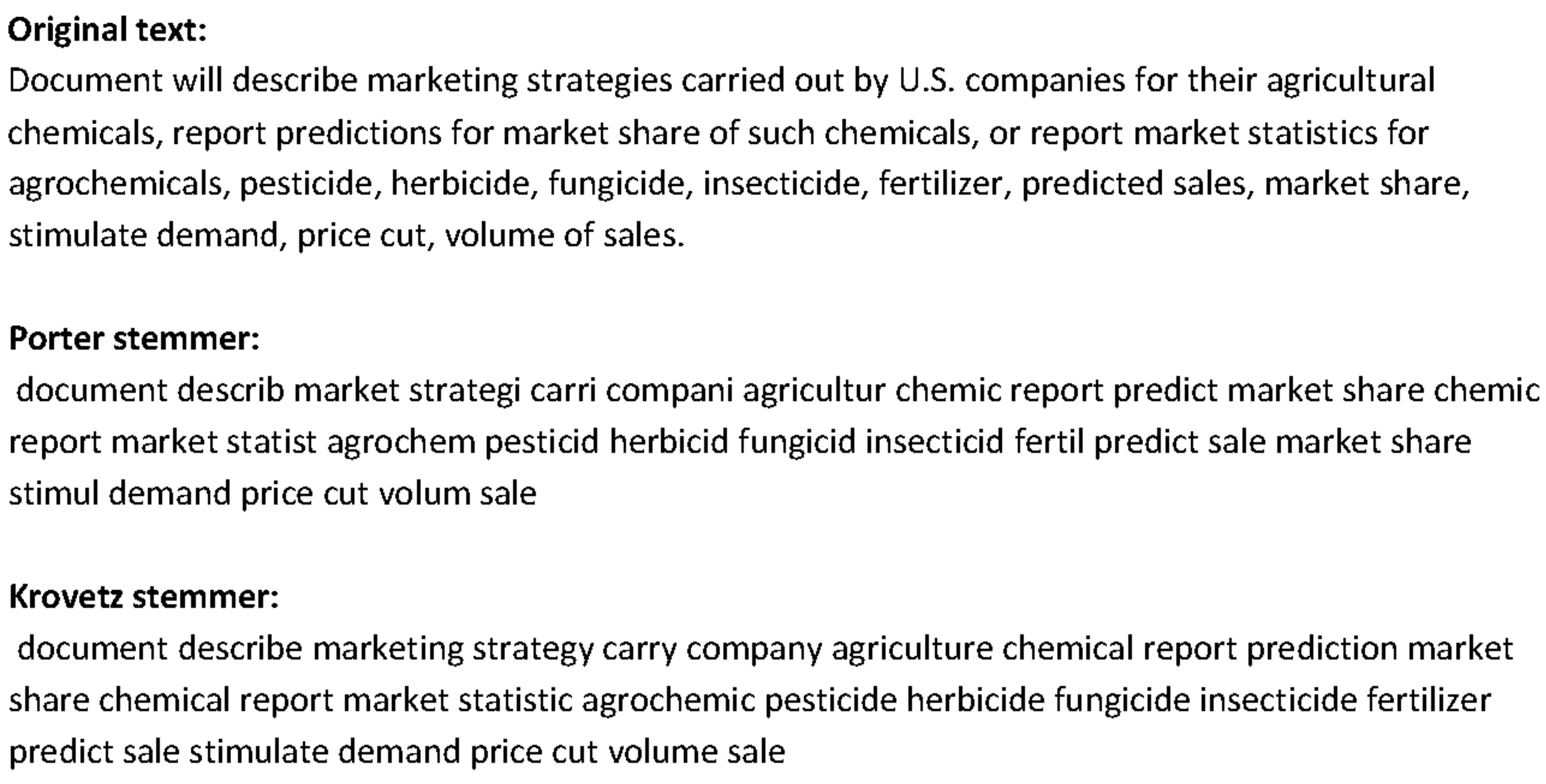


False positives	*False negatives*
organization/organ	european/europe
generalization/generic	cylinder/cylindrical
numerical/numerous	matrices/matrix
policy/police	urgency/urgent
university/universe	create/creation
addition/additive	analysis/analyses
negligible/negligent	useful/usefully
execute/executive	noise/noisy
past/paste	decompose/decomposition
ignore/ignorant	sparse/sparsity
special/specialized	resolve/resolution
head/heading	triangle/triangular

Krovetz Stemmer

- Hybrid algorithmic-dictionary
- Word checked in dictionary
    - If present, either left alone or replaced with “exception”
    - If not present, word is checked for suffixes that could be removed, after removal, dictionary is checked again
- Produces words not stems
- Comparable effectiveness
- Lower false positive rate, somewhat higher false negative

Stemmer Comparison



Original text:

Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales.

Porter stemmer:

document describ market strategi carri compani agricultur chemic report predict market share chemic report market statist agrochem pesticid herbicid fungicid insecticid fertil predict sale market share stimul demand price cut volum sale

Krovetz stemmer:

document describe marketing strategy carry company agriculture chemical report prediction market share chemical report market statistic agrochemic pesticide herbicide fungicide insecticide fertilizer predict sale stimulate demand price cut volume sale

Lemmatization

- Map tokens to *lexemes*

- In English, a *lexeme* is a "word" in the sense of a dictionary entry.
  - bridge, n. (1) ... structure ... over a river
  - bridge, v. (1) ... to form a way by means of a bridge ...

- Stemming sometimes reduces different forms to same lexeme, sometimes not

- Often not possible to lemmatize tokens independently of one another

Lemmatization vs. Stemming

- The word "better" has "good" as its lemma
  - Stemming does not produce the lemma
- The word "walking" has "walk" as its lemma
  - Stemming produces the lemma
- "Meeting" can be either the base form of a noun, or a form of a verb ("to meet") depending on context
  - "in our last meeting"
  - "we are meeting again tomorrow"

Thinking Question

- Can you think of a task that would be made *worse* by removing stop words, or applying a stemmer or lemmatiser?

- Why do you think this would happen?

Tools for Text Cleaning

- iconv
- tr
- sed
- awk
- Simple Shell Scripts
- Python
- OpenRefine

iconv

- Converts text from one character encoding to another.

- Options for input from files or from standard input

- iconv [options] –f from-encoding –t to-encoding

- Example for cleaning social media data:
- iconv -f utf-8 -t ascii//TRANSLIT -c file.txt > ascii_only.txt

- ascii//TRANSLIT -> € becomes EUR
- ascii//IGNORE -> € is deleted
- ascii -> € raises error
- TRANSLIT IS IMPLEMENTATION DEPENDENT

tr

- translate characters
- makes user-defined character-by-character transformations
- echo “Hello World” | tr abcde 12345
  - H5llo World
- echo “Hello World” | tr [A-Z] [a-z]
  - hello world
- Other useful simplifying options; check the man page

sed

- A stream editor
- Sed has support for regular expressions, runs line by line, making additions, substitutions, deletions
- Can also count lines and insert and delete specific ones
- Similar to grep, but more flexible and can modify data
- https://www.linode.com/docs/guides/differences-between-grep-sed-awk/

sed

- Replacing text:

```
sed –e 's/2022/2023/g' index.html > modified.html
```

- Very powerful

- CAREFUL:
  - MacOS uses BSD sed, other systems use GNU sed syntax is a little different

https://www.linode.com/docs/guides/manipulate-text-from-the-command-line-with-sed/#finding-and-replacing-strings-within-files-using-sed

AWK

- A programming language for processing text, can do manipulation and arithmetic

- Processes one line at a time
- Automatically breaks line up using whitespace, assigns to $1, $2, …

- AWK program structure:
- CONDITION { actions }

- awk '($3 == "Toyota") {print}' names.txt
  - Print every line where the third token is Toyota

- https://www.linode.com/docs/guides/differences-between-grep-sed-awk/

Simple Shell Scripts

- Collect a selection (regex compatible) of files into a file:

```
ls expression > tmp
ls myfile-[0-9][0-9][0-9].txt > tmp
```

- To run a command on every file in a selection, we can use a shell while loop:

```
while read p;
do
        echo “Hello World”
        python3 myfile.py $p >> collectedoutput.txt
done < tmp
```

Python

- Text cleaning can be done by regex – usually the ‘re’ module/library, but there are others which expand support.

- For a given string object, there are built in .encode() and .decode() functions which are useful for conversion: https://docs.python.org/3/library/stdtypes.html?highlight=encode#str.encode

OpenRefine

- https://openrefine.org

- Open source software that assists cleaning

- Key concepts:
  - Faceting
  - Clustering

Faceting

- Faceting is basically aggregation by column

cityLabel	population	countryLabel
Shanghai	23390000	People's Republic of China
Beijing	21710000	People's Republic of China
Lagos	21324000	Nigeria
Dhaka	16800000	Bangladesh
Mumbai	15414288	India
Istanbul	14657434	Turkey
Tokyo	13942856	Japan
Tianjin	13245000	People's Republic of China
Guangzhou	13080500	People's Republic of China
São Paulo	12106920	Brazil

Faceting

- Facet using **countryLabel** gives row counts by **countryLabel**
- Also shows all unique countryLabels

Facet	**Count**
People's Republic of China	4
Bangladesh	1
Brazil	1
India	1
Japan	1
Nigeria	1
Turkey	1

Faceting

- Faceting can help catch typos and inconsistencies

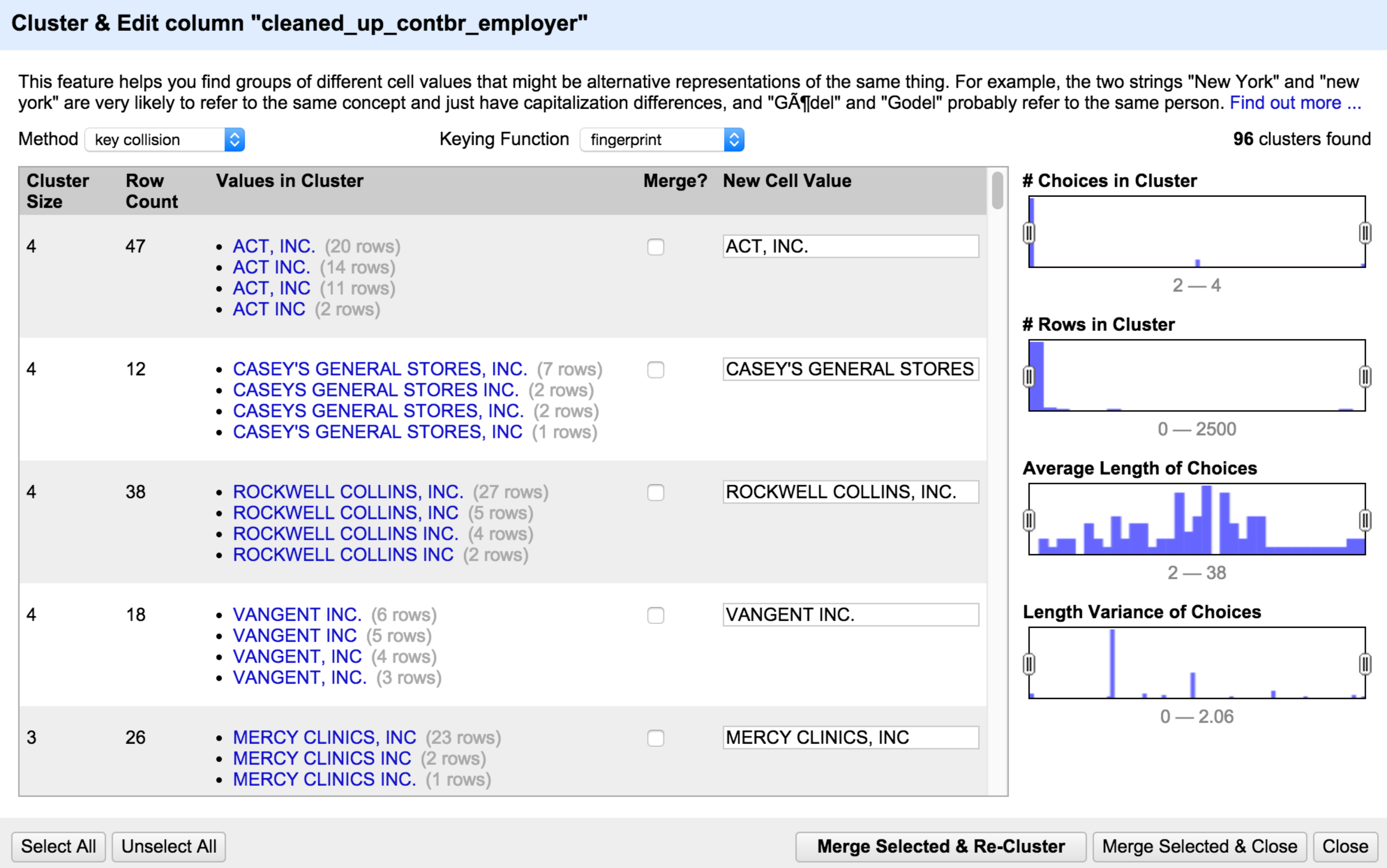
cityLabel	population	countryLabel
Shanghai	23390000	People’s Republic of China
Beijing	21710000	People's Republic of China
Lagos	21324000	Nigeria
Dhaka	16800000	Bangladesh
Mumbai	15414288	India
Istanbul	14657434	Turkey
Tokyo	13942856	Japan
Tianjin	13245000	Peoples Republic of China
Guangzhou	13080500	People's Republic of China
São Paulo	12106920	Brazil

Faceting

- In the faceted view, you can directly edit the names of the facets to correct them and correct the original data.

Facet	Count
People's Republic of China	3
Peoples Republic of China	1
Bangladesh	1
Brazil	1
India	1
Japan	1
Nigeria	1
Turkey	1

Clustering



Cluster & Edit column "cleaned_up_contbr_employer"

This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, the two strings "New York" and "new york" are very likely to refer to the same concept and just have capitalization differences, and "GÃ¶del" and "Godel" probably refer to the same person. Find out more ...

Method: key collision — Keying Function: fingerprint — **96** clusters found

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
4	47	• ACT, INC. (20 rows) • ACT INC. (14 rows) • ACT, INC (11 rows) • ACT INC (2 rows)	□	ACT, INC.
4	12	• CASEY'S GENERAL STORES, INC. (7 rows) • CASEYS GENERAL STORES INC. (2 rows) • CASEYS GENERAL STORES, INC. (2 rows) • CASEY'S GENERAL STORES, INC (1 rows)	□	CASEY'S GENERAL STORES
4	38	• ROCKWELL COLLINS, INC. (27 rows) • ROCKWELL COLLINS, INC (5 rows) • ROCKWELL COLLINS INC. (4 rows) • ROCKWELL COLLINS INC (2 rows)	□	ROCKWELL COLLINS, INC.
4	18	• VANGENT INC. (6 rows) • VANGENT INC (5 rows) • VANGENT, INC (4 rows) • VANGENT, INC. (3 rows)	□	VANGENT INC.
3	26	• MERCY CLINICS, INC (23 rows) • MERCY CLINICS INC (2 rows) • MERCY CLINICS INC. (1 rows)	□	MERCY CLINICS, INC

# Choices in Cluster 2 — 4

# Rows in Cluster 0 — 2500

Average Length of Choices 2 — 38

Length Variance of Choices 0 — 2.06

Pre-processing and Provenance

- "Provenance" – how did we end up with the final version of the data we are using?
- Key component of *reproducibility* in research
- OpenRefine keeps track for you; in your own work, scripts and detailed notes are very important.
    - What you did
    - Why you did it

Summary

- Text (pre-)Processing
  - Stop words
  - Stemming
  - Lemmatization
- Tools for text cleaning
  - iconv
  - tr
  - sed
  - awk
  - shell scripts
  - python
  - OpenRefine
- Provenance