Lecture 1b: Text, terms, and bags of words

William Webber (william@williamwebber.com)

COMP90042, 2014, Semester 1, Lecture 1b

Corpus, document, term

- Body of text referred to as corpus
- Corpus regarded as a collection of discreet documents
- Document regarded as a collection of discreet terms
- Term not necessarily a “word” of a natural language

Complexities

- Corpus may be open-ended, poorly defined, growing
- Document boundaries may be unclear:
  - Non-trivial dependencies between documents (e.g. email in a thread of emails)
  - Internal subdivisions (e.g. chapter in a book; paragraphs in chapter)
- Terms may be “typed”, have internal structure
  - What are the terms in the email headers:

    ```
    From:  "William Webber"
    <william@williamwebber.com>
    Date:  Sun, 02 Mar 2014 23:26:33 +0000
    ```

Tokenizing

- Split document up into “tokens” (proto-terms)
- E.g. for alphabetical languages (like English):
  - Strip formatting (e.g. of HTML)
  - Strip punctuation
  - Break at whitespaces
- What about ideographic languages (e.g. CJK)?

Token processing

- Case folding: in bicameral (two-case) scripts (e.g. Latin, Greek), reduce all letters to one (e.g. lower) case

  United States → *united states*

- Stemming: in inflected languages, reduce words to base forms:

  stemming, stemmed, stems → *stem*

- Stopping: remove common words that do not carry meaning:

  ~~the~~, ~~a~~, ~~and~~, ~~of~~

- Other tokens may be suppressed (e.g. long numbers)

Notes

- All of the above are optional; pros and cons for each
- Different languages present different issues

Bag of words

- Represent document as "bag" of terms (known as "bag-of-words" or BOW representation)
- A "bag" is a multiset: a set in which each element can occur more than once (equivalently, each element occurs with a count)
- Bags do not retain sequential dependencies:
  - No phrase search!
  - No linguistic processing
  - No sense of proximity

Processing example

```
<h1>Welcome␣to␣COMP90042!</h1>

<p>Students␣taking␣this␣subject
<i>must</i>␣have␣taken␣the␣subject
COMP30018.␣␣All␣student␣work
--including␣projects---must
be␣programmed␣in␣Python.␣Email
<tt>william@williamwebber.com</tt>
with␣questions.</p>
```

Processing example

```
Welcome␣to␣COMP90042!

Students␣taking␣this␣subject
must␣have␣taken␣the␣subject
COMP30018.␣␣All␣student␣work
—–including␣projects——must
be␣programmed␣in␣Python.␣Email
william@williamwebber.com
with␣questions.
```

- Strip formatting

Processing example

```
Welcome␣to␣COMP90042

Students␣taking␣this␣subject
must␣have␣taken␣the␣subject
COMP30018␣All␣student␣work
including␣projects␣must
be␣programmed␣in␣Python␣Email
william␣williamwebber␣com
with␣questions
```

- Strip formatting
- Strip punctuation

Processing example

```
Welcome to COMP90042 Students
taking this subject must have
taken the subject COMP30018
All student work including
projects must be programmed in
Python Email william williamwebber
com with questions
```

- Strip formatting
- Strip punctuation
- Break at whitespace

Processing example

```
welcome to comp90042 students
taking this subject must have
taken the subject comp30018
all student work including
projects must be programmed in
python email william williamwebber
com with questions
```

- Strip formatting
- Strip punctuation
- Break at whitespace
- Case fold

Processing example

welcome to comp90042 student take this subject must have take the subject comp30018 all student work include project must be program in python email william williamwebber com with question

- Strip formatting
- Strip punctuation
- Break at whitespace
- Case fold
- Stem

Processing example

```
welcome comp90042 student
take subject must
take subject comp30018
student work include
project must program
python email william williamwebber
com question
```

- Strip formatting
- Strip punctuation
- Break at whitespace
- Case fold
- Stem
- Stop

Processing example

```
com:1 comp30018:1 comp90042:1
email:1 include:1 must:2 program:1
project:1 python:1 question:1
student:2 subject:2 take:2
welcome:1 william:1 williamwebber:1
```

- Strip formatting
- Strip punctuation
- Break at whitespace
- Case fold
- Stem
- Stop
- Bag of words

Term occurrences

Real-world vocabulary not closed, fixed (not just "dictionary words"), due to:

- Proper names
- Abbreviations, acronyms
- Serial numbers, URLs, other identifiers
- Terms from other languages
- Mispellings
- Slang
- . . .

One rule of thumb for processing web documents:

> *The size of your vocabulary equals the number of documents you have*

Distribution of term frequencies

- Distribution of term frequencies in collection (either total count, or document count) follows a power law (Zipfian, long-tailed) distribution
- Rank terms (or any other items) by decreasing frequency
- Zipf's law: The frequency f of the term at rank r is proportional to $r^{-}b$, with $b \approx 1$; i.e.:

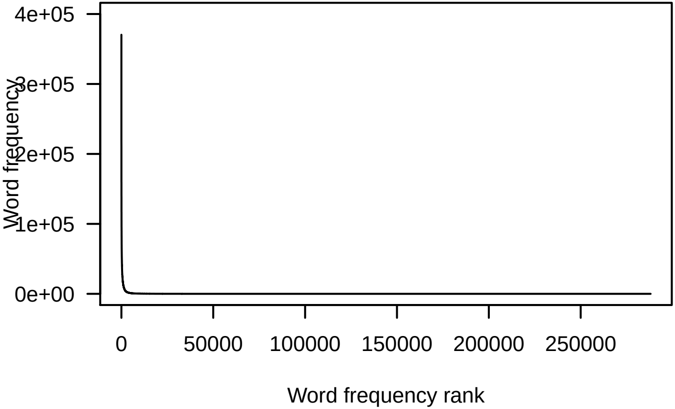
$$f \sim r^{-}b \tag{1}$$

- Zipfian distributions will look approximately straight in log–log graphs

Distribution of term frequencies (cont.)

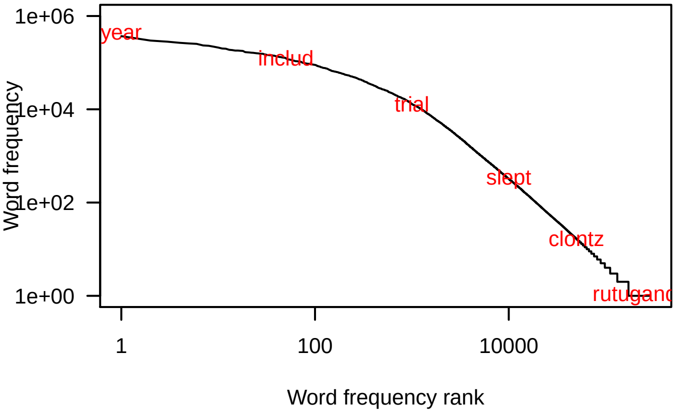
- Collection of 804,414 news articles (RCV1v2)
- Has (stemmed, stopped, case-folded) vocabulary of 288,062 terms. (The 20-volume OED lists 171,146 words in current use.)
- Count term frequency by number of documents a term appears in (see next slide)

Distribution of term frequencies (further cont.)



- Regular graph almost perfect L shape.

Distribution of term frequencies (further cont.)



- Regular graph almost perfect L shape.
- In log-log graph, a "bent" straight line (typical for such distributions)

Consequences of Zipf's law

Infrequent terms

- Most of the vocabulary will occur very infrequently
- For RCV1v2, 40% of vocabulary occurs only in single document
- Includes "terms" like `t04301`, `hrebenciuc`, `tluszczoweg`
- For **some** applications, such terms are uninformative

Frequent terms

- A few terms will appear in much of the collection (even if stopping is applied).
- For RCV1v2, `year` appears in 46% of documents, `percent` in 37%, and `million` in 35%
- For **some** applications, such terms are only weakly informative

Looking back and forward

In hortis Cardinalis de Medicis prope Villam Iulij III Pont. Max.

- The “Bag of Words” representation of a document is widely used, and will be used for most of this course
- Vocabularies are open-ended, and term frequencies long-tailed (Zipfian),
- Further modelling step required to make BOW useful
- We will look first at geometric models, then probabilistic

Further reading

- Chapter 2[1], Sections 1 ("Document delineation and character sequence decoding") and 2 ("Determining the vocabulary of terms"), of Manning, Raghavan, and Schutze, *Introduction to Information Retrieval* (on tokenization and term normalization)
- Introduction to Manning and Schutze, *Foundations of Statistical Natural Language Processing* (on term distributions and Zipf's law)
- Lada A. Adamic, "Zipf, Power-laws, and Pareto"[2] (on the relationship between these distributions)
- The Porter Stemmer[3], a standard English stemming algorithm
- A brief discussion[4] of the issues involved with CJK word segmentation

[1] http://nlp.stanford.edu/IR-book/pdf/02voc.pdf

[2] http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html

[3] http://tartarus.org/martin/PorterStemmer/

[4] http://www.hathitrust.org/blogs/large-scale-search/multilingual-issues-part-1-word-segmentation