

# Unit 1: Preprocessing

Florida State Summer Methods Workshop

Dr. Rochelle Terman

Department of Political Science  
University of Chicago

May 2019

**Goal:** Prepare texts into format used for computational text analysis

**Method:** Preprocessing recipe

**Decisions:** Feature selection, Non-english and multilingual issues.

## Key Terms:

- Corpus / document
- Encoding
- Preprocessing
- Tokens, grams
- Stemming / Lemmatize,
- Bag of Words
- Document-Term Matrix

## Key R Packages

- tm

# Preparing a Corpus

A **corpus** (pl: corpora) is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.

# Preparing a Corpus

A **corpus** (pl: corpora) is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.

Within each corpus we will have separate articles, stories, volumes, each treated as a separate entity or record. Each unit is called a **document**.

# Preparing a Corpus

A **corpus** (pl: corpora) is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.

Within each corpus we will have separate articles, stories, volumes, each treated as a separate entity or record. Each unit is called a **document**.

Documents come in a variety of formats, but **plain text** is best (e.g. .txt, .csv).

# Preparing a Corpus

A **corpus** (pl: corpora) is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.

Within each corpus we will have separate articles, stories, volumes, each treated as a separate entity or record. Each unit is called a **document**.

Documents come in a variety of formats, but **plain text** is best (e.g. .txt, .csv).

Plain text is **encoded** in different ways. UTF-8 is best.

# Preparing a Corpus

A **corpus** (pl: corpora) is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.

Within each corpus we will have separate articles, stories, volumes, each treated as a separate entity or record. Each unit is called a **document**.

Documents come in a variety of formats, but **plain text** is best (e.g. .txt, .csv).

Plain text is **encoded** in different ways. UTF-8 is best.

Corpora often come with **metadata** (e.g. author, date, label.)



# Preparing a Corpus

A **corpus** (pl: corpora) is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.

Within each corpus we will have separate articles, stories, volumes, each treated as a separate entity or record. Each unit is called a **document**.

Documents come in a variety of formats, but **plain text** is best (e.g. .txt, .csv).

Plain text is **encoded** in different ways. UTF-8 is best.

Corpora often come with **metadata** (e.g. author, date, label.)

**My preferred structure:** Each document a row, one column for text, and other columns for metadata.

# Preprocessing Texts

**One** (of many) recipe for preprocessing: retain **useful** information

# Preprocessing Texts

**One** (of many) recipe for preprocessing: retain **useful** information

- 1) Remove capitalization, punctuation

# Preprocessing Texts

**One** (of many) recipe for preprocessing: retain **useful** information

- 1) Remove capitalization, punctuation
- 2) Discard Word Order: (**Bag of Words** Assumption)

# Preprocessing Texts

**One** (of many) recipe for preprocessing: retain **useful** information

- 1) Remove capitalization, punctuation
- 2) Discard Word Order: (**Bag of Words** Assumption)
- 3) Discard stop words

# Preprocessing Texts

**One** (of many) recipe for preprocessing: retain **useful** information

- 1) Remove capitalization, punctuation
- 2) Discard Word Order: (**Bag of Words** Assumption)
- 3) Discard stop words
- 4) Combine similar terms: Stem, Lemmatize

# Preprocessing Texts

**One** (of many) recipe for preprocessing: retain **useful** information

- 1) Remove capitalization, punctuation
- 2) Discard Word Order: (**Bag of Words** Assumption)
- 3) Discard stop words
- 4) Combine similar terms: Stem, Lemmatize
- 5) **Output**: Document-Term Matrix, each element counts occurrence of a particular term in a particular document

# 1. Remove capitalization, punctuation, numbers

Assumption: capitalization, punctuation does not provide useful information.



# 1. Remove capitalization, punctuation, numbers

Assumption: capitalization, punctuation does not provide useful information.

Now we are engaged in a great civil war, testing whether that nation, or any nation

# 1. Remove capitalization, punctuation, numbers

Assumption: capitalization, punctuation does not provide useful information.

Now we are engaged in a great civil war, testing whether  
that nation, or any nation

now we are engaged in a great civil war testing whether  
that nation or any nation

# 1. Remove capitalization, punctuation, numbers

Assumption: capitalization, punctuation does not provide useful information.

Now we are engaged in a great civil war, testing whether that nation, or any nation

now we are engaged in a great civil war testing whether that nation or any nation

Caution

‘‘Turkey’’ = ‘‘turkey’’

## 2. Discard Word Order (Bag of Words) $\rightsquigarrow$ Tokenize

Assumption: Word Order Doesn't Matter.

## 2. Discard Word Order (Bag of Words) $\rightsquigarrow$ Tokenize

Assumption: Word Order Doesn't Matter.

now we are engaged in a great civil war testing whether  
that nation or any nation

## 2. Discard Word Order (Bag of Words) $\rightsquigarrow$ Tokenize

Assumption: Word Order Doesn't Matter.

now we are engaged in a great civil war testing whether  
that nation or any nation

[now, we, are, engaged, in, a, great, civil, war, testing,  
whether, that, nation, or, any, nation]

## 2. Discard Word Order (Bag of Words) $\rightsquigarrow$ Tokenize

Assumption: Word Order Doesn't Matter.

now we are engaged in a great civil war testing whether  
that nation or any nation

[now, we, are, engaged, in, a, great, civil, war, testing,  
whether, that, nation, or, any, nation]

[a, any, are, civil, engaged, great, in, nation, now, or,  
testing, that, war, we, whether]

## 2. Discard Word Order (Bag of Words) $\rightsquigarrow$ Tokenize

Assumption: Word Order Doesn't Matter.

now we are engaged in a great civil war testing whether  
that nation or any nation

[now, we, are, engaged, in, a, great, civil, war, testing,  
whether, that, nation, or, any, nation]

[a, any, are, civil, engaged, great, in, nation, now, or,  
testing, that, war, we, whether]

Tokenization



# Tokenization

**Unigrams** [now, we, are, engaged, in, a, great, civil, war, testing, whether, that, nation, or, any, nation]

# Tokenization

**Unigrams** [now, we, are, engaged, in, a, great, civil, war, testing, whether, that, nation, or, any, nation]

**Bigrams** [now we, we are, are engaged, engaged in, in a, a great, great civil, civil war, war testing, testing whether, whether that, that nation, nation or, or any, any nation]

# Tokenization

**Unigrams** [now, we, are, engaged, in, a, great, civil, war, testing, whether, that, nation, or, any, nation]

**Bigrams** [now we, we are, are engaged, engaged in, in a, a great, great civil, civil war, war testing, testing whether, whether that, that nation, nation or, or any, any nation]

**Trigrams** [now we are, we are engaged, are engaged in, engaged in a, in a great, a great civil, great civil war, civil war testing, war testing whether, testing whether that, whether that nation, that nation or, nation or any, or any nation]

# How Could This Possibly Work?

Speech is:

- Ironic

Thanks, Obama

- Subtle Negation (Source: Janyce Wiebe) :

They have not succeeded, and will never succeed, in  
breaking the will of this valiant people

- Order Dependent (Source: Arthur Spirling):

Peace, no more war

War, no more peace

# How Could This Possibly Work?

Three answers

- 1) **It might not**: Validation is critical (task specific)
- 2) **Central Tendency in Text**: Words often imply what a text is about  
war, civil, union or tone consecrate, dead, died, lives.  
Likely to be used repeatedly: create a theme for an article
- 3) **Proof in the pudding**: Bag-of-words assumption works for a number of applications.

### 3. Discard stop words

- **Stop Words:** English Language place holding words

### 3. Discard stop words

- **Stop Words:** English Language place holding words  
the, it, if, a, able, at, be, because...

### 3. Discard stop words

- **Stop Words:** English Language place holding words  
the, it, if, a, able, at, be, because...
- Add “noise” to documents (without conveying much information)



### 3. Discard stop words

- **Stop Words**: English Language place holding words  
the, it, if, a, able, at, be, because...
- Add “noise” to documents (without conveying much information)
- Discard stop words: focus on **substantive** words

### 3. Discard stop words

- **Stop Words**: English Language place holding words  
the, it, if, a, able, at, be, because...
- Add “noise” to documents (without conveying much information)
- Discard stop words: focus on **substantive** words

**Note of Caution**: Monroe, Colaresi, and Quinn (2008)

### 3. Discard stop words

- **Stop Words**: English Language place holding words  
the, it, if, a, able, at, be, because...
- Add “noise” to documents (without conveying much information)
- Discard stop words: focus on **substantive** words

**Note of Caution**: Monroe, Colaresi, and Quinn (2008)  
she, he, her, his

### 3. Discard stop words

- **Stop Words**: English Language place holding words  
the, it, if, a, able, at, be, because...
- Add “noise” to documents (without conveying much information)
- Discard stop words: focus on **substantive** words

**Note of Caution**: Monroe, Colaresi, and Quinn (2008)

she, he, her, his

**Many English language stop lists include gender pronouns**

### 3. Discard stop words

- **Stop Words**: English Language place holding words  
the, it, if, a, able, at, be, because...
- Add “noise” to documents (without conveying much information)
- Discard stop words: focus on **substantive** words

**Note of Caution**: Monroe, Colaresi, and Quinn (2008)

she, he, her, his

**Many English language stop lists include gender pronouns**

- Exercise caution when discarding stop words

### 3. Discard stop words

- **Stop Words**: English Language place holding words  
the, it, if, a, able, at, be, because...
- Add “noise” to documents (without conveying much information)
- Discard stop words: focus on **substantive** words

**Note of Caution**: Monroe, Colaresi, and Quinn (2008)

she, he, her, his

**Many English language stop lists include gender pronouns**

- Exercise caution when discarding stop words
- You may need to customize your stop word list↪ abbreviations, titles, etc.

## 4. Combine similar terms

Reduce dimensionality further

## 4. Combine similar terms

Reduce dimensionality further  $\rightsquigarrow$  combine similar terms (tense and number).



## 4. Combine similar terms

Reduce dimensionality further  $\rightsquigarrow$  combine similar terms (tense and number).

- Words used to refer to same basic concept

## 4. Combine similar terms

Reduce dimensionality further  $\rightsquigarrow$  combine similar terms (tense and number).

- Words used to refer to same basic concept  
family, families, familial  $\rightarrow$  famili

## 4. Combine similar terms

Reduce dimensionality further  $\rightsquigarrow$  combine similar terms (tense and number).

- Words used to refer to same basic concept  
family, families, familial  $\rightarrow$  famili
- Stemming/Lemmatizing algorithms: Many-to-one mapping from words to stem/lemma

# Comparing Stemming and Lemmatizing

Stemming algorithm:

# Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms

# Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word

# Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word
- Porter stemmer, Lancaster stemmer, Snowball stemmer

# Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word
- **Porter** stemmer, **Lancaster** stemmer, **Snowball** stemmer

Lemmatizing algorithm:



# Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word
- Porter stemmer, Lancaster stemmer, Snowball stemmer

Lemmatizing algorithm:

- Condition on part of speech (noun, verb, etc)

# Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word
- Porter stemmer, Lancaster stemmer, Snowball stemmer

Lemmatizing algorithm:

- Condition on part of speech (noun, verb, etc)
- Verify result is a word

# Other common steps

- Remove sparse terms (rare words)

## Other common steps

- Remove sparse terms (rare words)
- Remove other terms (e.g. proper nouns).

# Other common steps

- Remove sparse terms (rare words)
- Remove other terms (e.g. proper nouns).
- Weight some terms more than others (tf-idf)

# All together now...

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

# All together now...

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

Step 1: Remove capitalization and punctuation:

# All together now...

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

Step 1: Remove capitalization and punctuation:

four score and seven years ago our fathers brought forth on this continent a new nation conceived in liberty and dedicated to the proposition that all men are created equal



# All together now...

## Step 1: Remove capitalization and punctuation:

four score and seven years ago our fathers brought forth on  
this continent a new nation conceived in liberty and  
dedicated to the proposition that all men are created equal

## Step 2: Tokenize:

# All together now...

## Step 1: Remove capitalization and punctuation:

four score and seven years ago our fathers brought forth on  
this continent a new nation conceived in liberty and  
dedicated to the proposition that all men are created equal

## Step 2: Tokenize:

four, score, and, seven, years, ago, our, fathers, brought,  
forth, on, this, continent, a, new, nation, conceived, in,  
liberty, and, dedicated, to, the, proposition, that, all,  
men, are, created, equal

# All together now...

Step 1: Remove capitalization and punctuation:

Step 2: Tokenize:

four, score, and, seven, years, ago, our, fathers, brought,  
forth, on, this, continent, a, new, nation, conceived, in,  
liberty, and, dedicated, to, the, proposition, that, all,  
men, are, created, equal

Step 3: Remove stop words:

# All together now...

Step 1: Remove capitalization and punctuation:

Step 2: Tokenize:

four, score, and, seven, years, ago, our, fathers, brought,  
forth, on, this, continent, a, new, nation, conceived, in,  
liberty, and, dedicated, to, the, proposition, that, all,  
men, are, created, equal

Step 3: Remove stop words:

four, score, seven, years, ago, fathers, brought, forth,  
continent, new, nation, conceived, liberty, dedicated,  
proposition, men, created, equal

# All together now...

Step 1: Remove capitalization and punctuation:

Step 2: Tokenize:

Step 3: Remove stop words:

four, score, seven, years, ago, fathers, brought, forth,  
continent, new, nation, conceived, liberty, dedicated,  
proposition, men, created, equal

Step 4: Applying Stemming Algorithm

# All together now...

Step 1: Remove capitalization and punctuation:

Step 2: Tokenize:

Step 3: Remove stop words:

four, score, seven, years, ago, fathers, brought, forth,  
continent, new, nation, conceived, liberty, dedicated,  
proposition, men, created, equal

Step 4: Applying Stemming Algorithm

four, score, seven, year, ago, father, brought, forth,  
contin, new, nation, conceiv, liberti, dedic, proposit,  
men, creat, equal

# All together now...

Step 1: Remove capitalization and punctuation:

Step 2: Tokenize:

Step 3: Remove stop words:

Step 4: Applying Stemming Algorithm

four, score, seven, year, ago, father, brought, forth,  
contin, new, nation, conceiv, liberti, dedic, proposit,  
men, creat, equal

Step 5: Create Count Vector

Stem	Count
ago	1
brought	1
seven	1
creat	1
conceiv	1
men	1
father	1
⋮	⋮

# All together now...

Step 1: Remove capitalization and punctuation:

Step 2: Tokenize:

Step 3: Remove stop words:

Step 4: Applying Stemming Algorithm

Step 5: Create Count Vector

Stem	Count
ago	1
brought	1
seven	1
creat	1
conceiv	1
men	1
father	1
⋮	⋮



# Document-Term Matrices

		Word1	Word2	Word3	...	WordP
$\mathbf{x} =$	Doc1	1	0	0	...	3
	Doc2	0	2	1	...	0
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
	DocN	0	0	0	...	5

# Document-Term Matrices

		Word1	Word2	Word3	...	WordP
$\mathbf{X} =$	Doc1	1	0	0	...	3
	Doc2	0	2	1	...	0
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
	DocN	0	0	0	...	5

$\mathbf{X} = N \times P$  matrix

- $N$  = Number of documents

# Document-Term Matrices

		Word1	Word2	Word3	...	WordP
$\mathbf{X} =$	Doc1	1	0	0	...	3
	Doc2	0	2	1	...	0
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
	DocN	0	0	0	...	5

$\mathbf{X} = N \times P$  matrix

- $N$  = Number of documents
- $P$  = Number of features

# Document-Term Matrices

		Word1	Word2	Word3	...	WordP
$\mathbf{X} =$	Doc1	1	0	0	...	3
	Doc2	0	2	1	...	0
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
	DocN	0	0	0	...	5

$\mathbf{X} = N \times P$  matrix

- $N$  = Number of documents
- $P$  = Number of features
- $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iP})$

# Document-Term Matrices

		Word1	Word2	Word3	...	WordP
$\mathbf{X} =$	Doc1	1	0	0	...	3
	Doc2	0	2	1	...	0
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
	DocN	0	0	0	...	5

$\mathbf{X} = N \times P$  matrix

- $N$  = Number of documents
- $P$  = Number of features
- $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iP})$

$\mathbf{X}$  = main input for many computational text analysis applications.

# Multi-language Issues

## Non-English languages pose specific challenges:

- Tokenization: Some languages, like Chinese, Japanese, and Lao, do not have spaces between words and cannot be parsed into individual units.

# Multi-language Issues

## Non-English languages pose specific challenges:

- Tokenization: Some languages, like Chinese, Japanese, and Lao, do not have spaces between words and cannot be parsed into individual units.
- Stop words: Each language has its own list of stop words.

# Multi-language Issues

## Non-English languages pose specific challenges:

- Tokenization: Some languages, like Chinese, Japanese, and Lao, do not have spaces between words and cannot be parsed into individual units.
- Stop words: Each language has its own list of stop words.
- Stemming/Lemmatization: Not all languages require stemming (Chinese), and others require more complex lemmatization (Hungarian)



# Multi-language Issues

## Non-English languages pose specific challenges:

- Tokenization: Some languages, like Chinese, Japanese, and Lao, do not have spaces between words and cannot be parsed into individual units.
- Stop words: Each language has its own list of stop words.
- Stemming/Lemmatization: Not all languages require stemming (Chinese), and others require more complex lemmatization (Hungarian)

## Solutions

- 1 Language-specific processing and software (e.g. `tm`, `txtorg`).

# Multi-language Issues

## Non-English languages pose specific challenges:

- Tokenization: Some languages, like Chinese, Japanese, and Lao, do not have spaces between words and cannot be parsed into individual units.
- Stop words: Each language has its own list of stop words.
- Stemming/Lemmatization: Not all languages require stemming (Chinese), and others require more complex lemmatization (Hungarian)

## Solutions

- 1 Language-specific processing and software (e.g. `tm`, `txtorg`).
- 2 Translate everything into English or other common language (e.g., Google Translate), especially if doing cross-language work

To the R code!