Text Pre-processing (I)

Text and Web Mining (H6751)

WKW School of Communication and Information

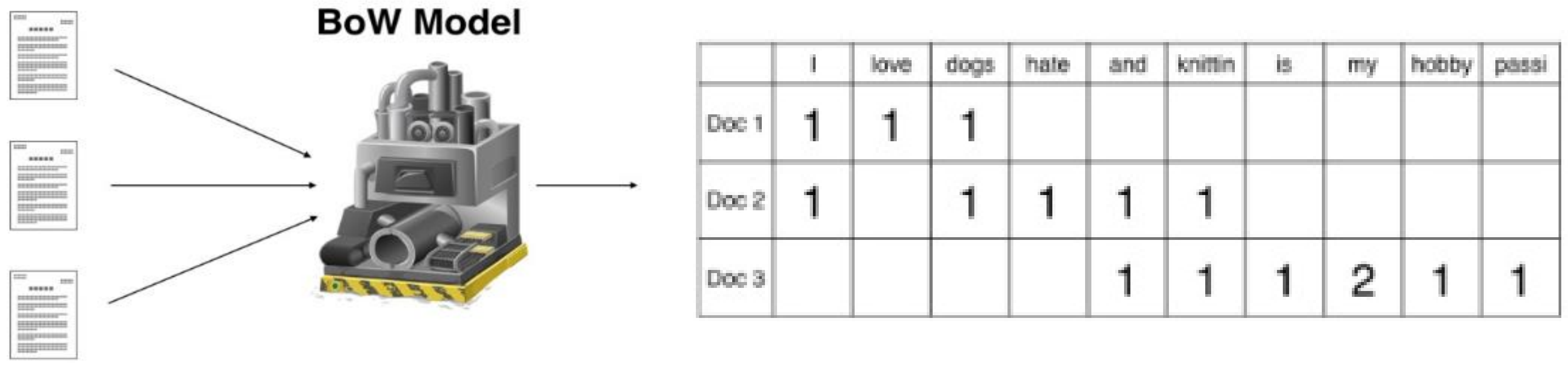
Why Pre-process Text?

- Tokenization illustration

1. I love dogs.

2. I hate dogs and knitting.

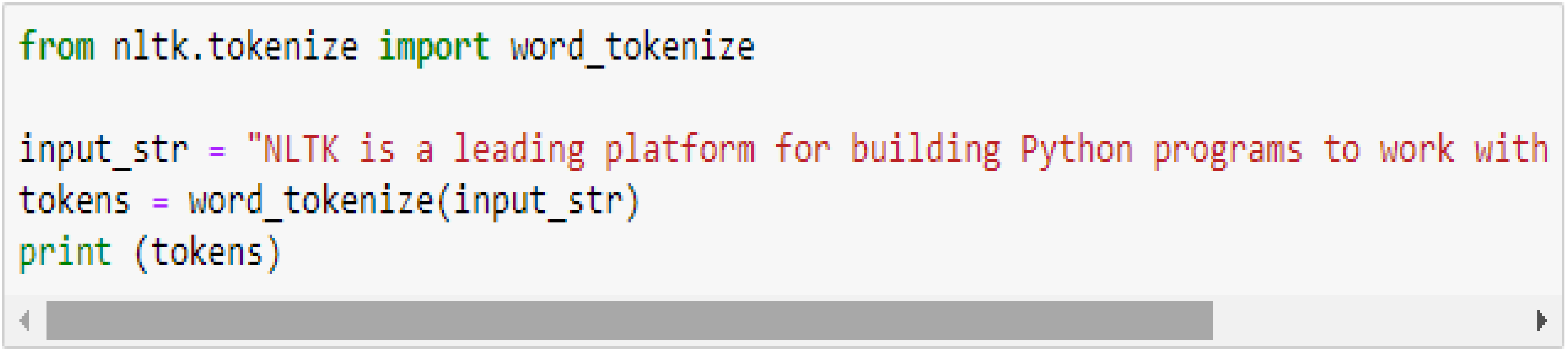
3. Knitting is my hobby and my passion.



	I	love	dogs	hate	and	knittin	is	my	hobby	passi
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

Why Pre-process Text?

Extract Significant Features / Remove stop words



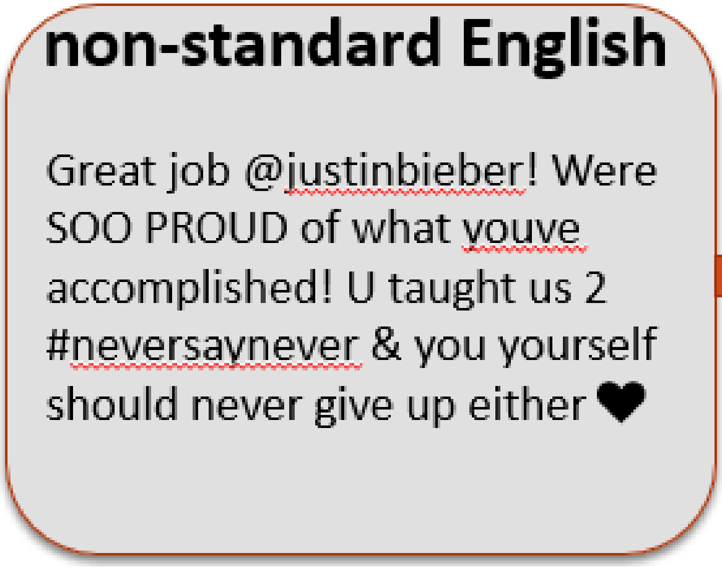
```
from nltk.tokenize import word_tokenize

input_str = "NLTK is a leading platform for building Python programs to work with
tokens = word_tokenize(input_str)
print (tokens)
```

```
['NLTK', 'is', 'a', 'leading', 'platform', 'for', 'building', 'Python', 'progr
ams', 'to', 'work', 'with', 'human', 'language', 'data', '.']
```

NLTK leading platform building Python programs work human language data

Why Pre-process Text?

- Normalization



non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either ♥

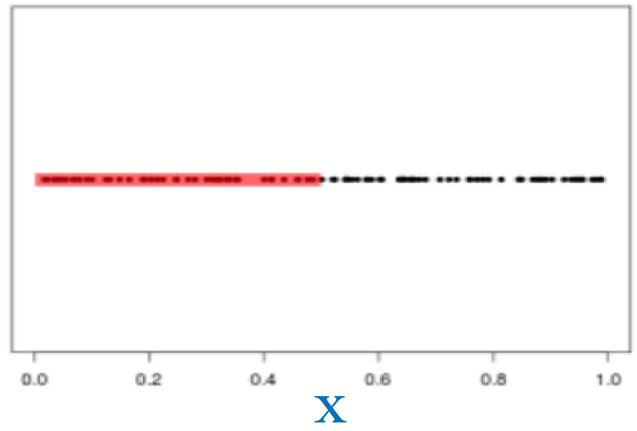
Standardization

great job justin bieber! we are so proud of what you have accomplished! you taught us to never say never and you yourself should never give up either love

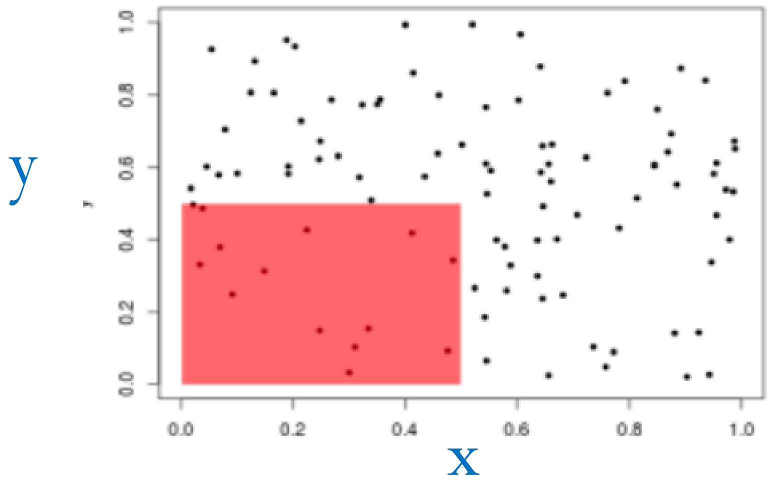
Curse of Dimensionality

- The number of data points that are captured by a **fixed length** rapidly diminishing as dimension increases.
- Performance drops when "capturing" required data points with increasing dimensions

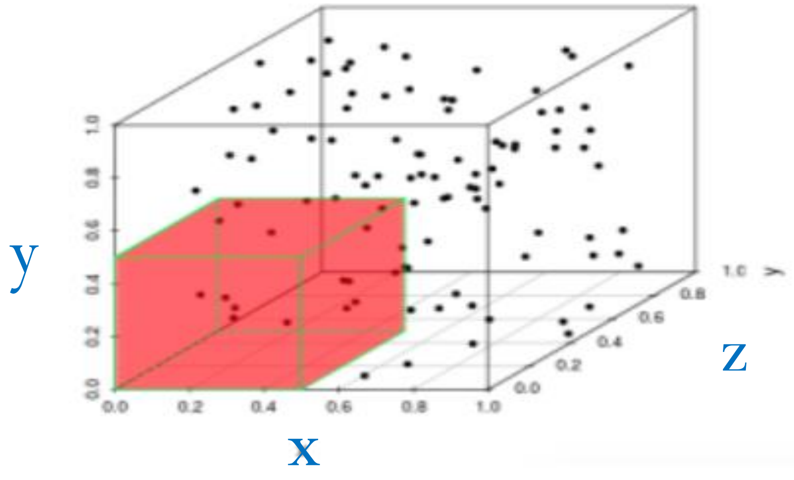
1-D **42%** of data captured



2-D **14%** of data captured



3-D **7%** of data captured

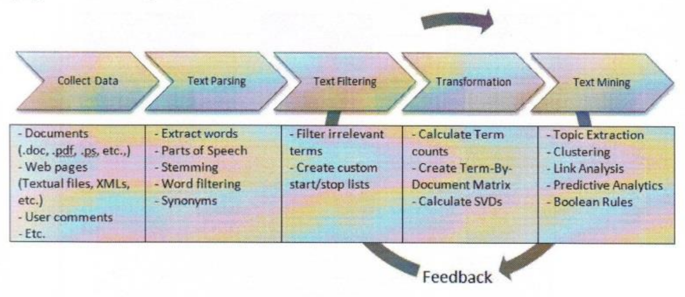


From Textual Information to Numerical Vectors

- In text mining - first need to process it into a form that data mining procedures can use.
- Determine the nature of the columns (i.e. **the features**) of the spreadsheet.
- Some useful features are easy to obtain (e.g., a **word** as it occurs in text) and some are much more difficult (e.g., the **grammatical function** of a word in a sentence such as subject, object, etc.)
  - E.g., ***I like the iphone -> I:subject, like:verb, the iphone:object***
- How to obtain the kinds of features commonly generated from text?

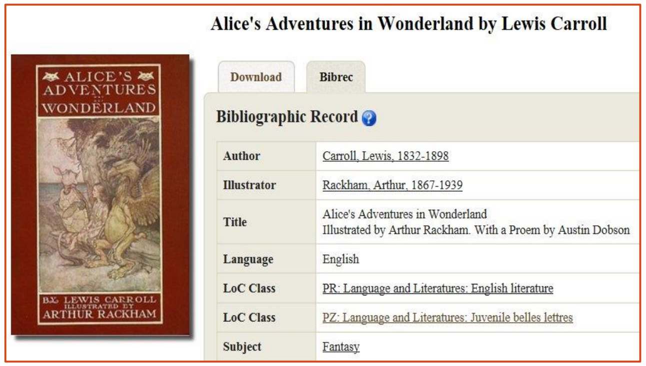
DocID	Apple	Bear	Durian	..	…	..	…	…	Zoo	Animal?
1	0	1	0	0	0	0	0	0	1	1
2	1	0	1	0	0	0	0	0	0	0
…										

Display 1.3: Text Mining Process Flow


Collect Data
Text Parsing
Text Filtering
Transformation
Text Mining
- Documents (.doc, .pdf, .ps, etc.,) - Web pages (Textual files, XMLs, etc.) - User comments - Etc.
- Extract words - Parts of Speech - Stemming - Word filtering - Synonyms
- Filter irrelevant terms - Create custom start/stop lists
- Calculate Term counts - Create Term-By-Document Matrix - Calculate SVDs
- Topic Extraction - Clustering - Link Analysis - Predictive Analytics - Boolean Rules
Feedback

Collecting Documents

- The first step in text mining is to collect the ***data***. (i.e. the relevant documents).



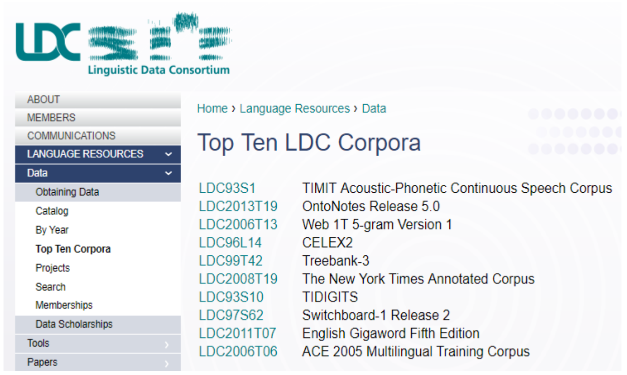
Alice's Adventures in Wonderland by Lewis Carroll

ALICE'S ADVENTURES IN WONDERLAND
BY LEWIS CARROLL ILLUSTRATED BY ARTHUR RACKHAM
Download
Bibrec

Bibliographic Record

Author	Carroll, Lewis, 1832-1898
Illustrator	Rackham, Arthur, 1867-1939
Title	Alice's Adventures in Wonderland Illustrated by Arthur Rackham. With a Proem by Austin Dobson
Language	English
LoC Class	PR: Language and Literatures: English literature
LoC Class	PZ: Language and Literatures: Juvenile belles lettres
Subject	Fantasy

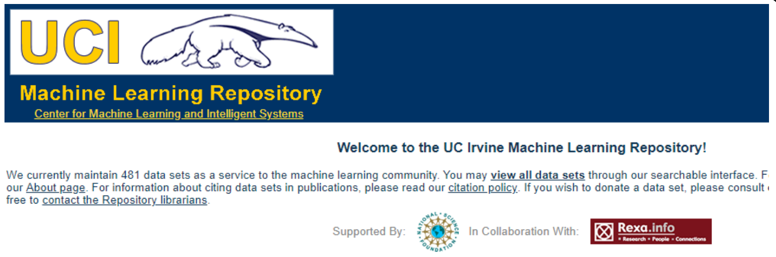
- In some applications, need to have a data collection process.
  - Web application - **a Web crawler** that collects the documents.
  - **E-mail audit application -** log all incoming and outgoing messages at a mail server for a period of time.
- For research and development of text-mining techniques, more generic data may be necessary, usually called a **corpus**.
  - E.g., **Gutenberg Project** (https://www.gutenberg.org), a very large collection of literary and other texts put into machine-readable form as the material comes out of **copyright**.
  - The **Linguistic Data Consortium (LDC)** (https://www.ldc.upenn.edu) provides various data for information extraction, parsing, etc.



Top Ten LDC Corpora

LDC93S1	TIMIT Acoustic-Phonetic Continuous Speech Corpus
LDC2013T19	OntoNotes Release 5.0
LDC2006T13	Web 1T 5-gram Version 1
LDC96L14	CELEX2
LDC99T42	Treebank-3
LDC2008T19	The New York Times Annotated Corpus
LDC93S10	TIDIGITS
LDC97S62	Switchboard-1 Release 2
LDC2011T07	English Gigaword Fifth Edition
LDC2006T06	ACE 2005 Multilingual Training Corpus

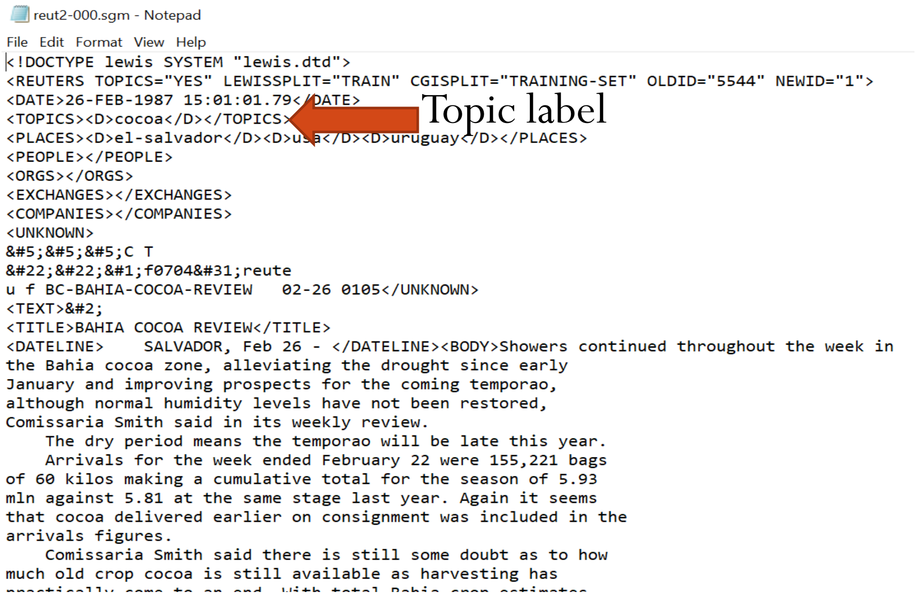
Collecting Documents


UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Welcome to the UC Irvine Machine Learning Repository!

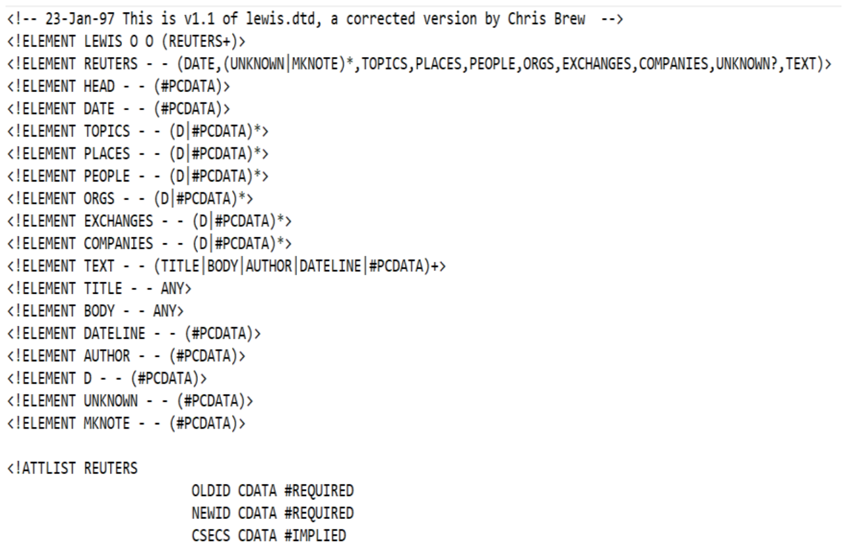
We currently maintain 481 data sets as a service to the machine learning community. You may **view all data sets** through our searchable interface. F
our About page. For information about citing data sets in publications, please read our citation policy. If you wish to donate a data set, please consult
free to contact the Repository librarians.

- Available **Corpus**. (Cont.)
  - **The UC Irvine Machine Learning Repository** currently maintain 481 data sets as a service to the machine learning community. http://archive.ics.uci.edu/ml/ - reuters news, amazon reviews, email spam, and sentiment-labelled sentences.
    - https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection
    - Different formats (e.g., standard generalized markups), understanding data dictionary/description



```
<!DOCTYPE lewis SYSTEM "lewis.dtd">
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="5544" NEWID="1">
<DATE>26-FEB-1987 15:01:01.79</DATE>
<TOPICS><D>cocoa</D></TOPICS>
<PLACES><D>el-salvador</D><D>usa</D><D>uruguay</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN>
&#5;&#5;&#5;C T
&#22;&#22;&#1;f0704&#31;reute
u f BC-BAHIA-COCOA-REVIEW   02-26 0105</UNKNOWN>
<TEXT>&#2;
<TITLE>BAHIA COCOA REVIEW</TITLE>
<DATELINE>    SALVADOR, Feb 26 - </DATELINE><BODY>Showers continued throughout the week in
the Bahia cocoa zone, alleviating the drought since early
January and improving prospects for the coming temporao,
although normal humidity levels have not been restored,
Comissaria Smith said in its weekly review.
    The dry period means the temporao will be late this year.
    Arrivals for the week ended February 22 were 155,221 bags
of 60 kilos making a cumulative total for the season of 5.93
mln against 5.81 at the same stage last year. Again it seems
that cocoa delivered earlier on consignment was included in the
arrivals figures.
    Comissaria Smith said there is still some doubt as to how
much old crop cocoa is still available as harvesting has
practically come to an end. With total Bahia crop estimates
```

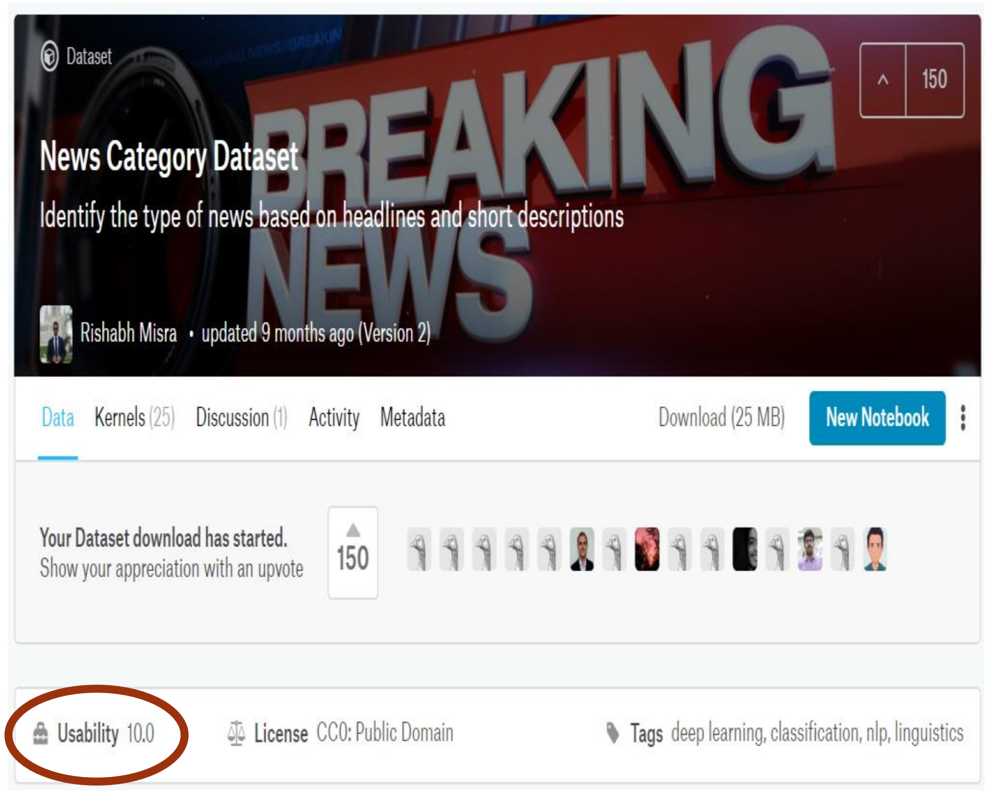
Document Type Declaration

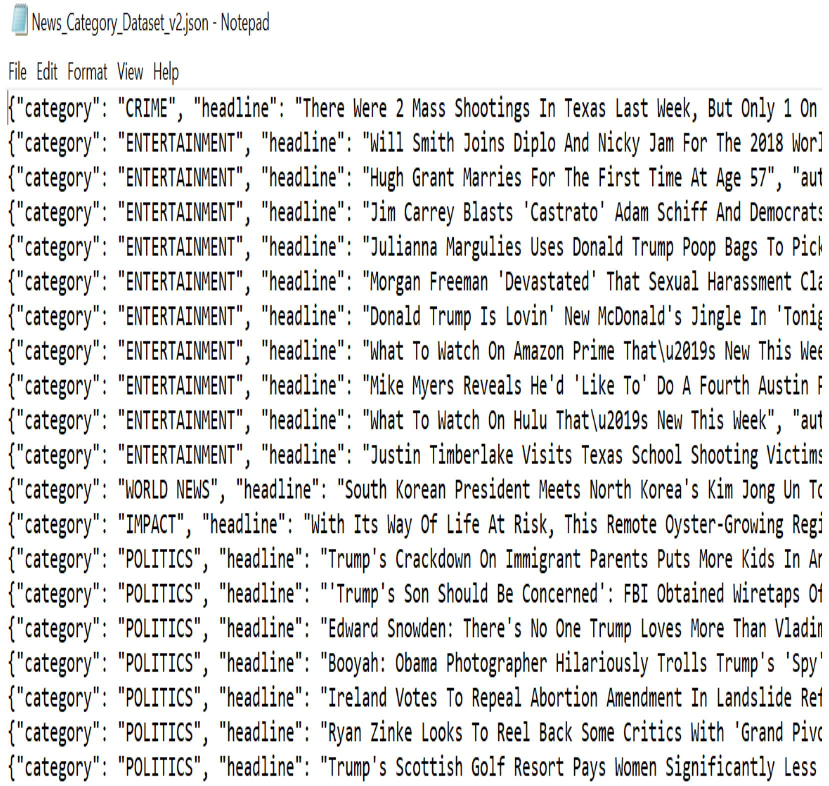


```
<!-- 23-Jan-97 This is v1.1 of lewis.dtd, a corrected version by Chris Brew  -->
<!ELEMENT LEWIS O O (REUTERS+)>
<!ELEMENT REUTERS - - (DATE,(UNKNOWN|MKNOTE)*,TOPICS,PLACES,PEOPLE,ORGS,EXCHANGES,COMPANIES,UNKNOWN?,TEXT)>
<!ELEMENT HEAD - - (#PCDATA)>
<!ELEMENT DATE - - (#PCDATA)>
<!ELEMENT TOPICS - - (D|#PCDATA)*>
<!ELEMENT PLACES - - (D|#PCDATA)*>
<!ELEMENT PEOPLE - - (D|#PCDATA)*>
<!ELEMENT ORGS - - (D|#PCDATA)*>
<!ELEMENT EXCHANGES - - (D|#PCDATA)*>
<!ELEMENT COMPANIES - - (D|#PCDATA)*>
<!ELEMENT TEXT - - (TITLE|BODY|AUTHOR|DATELINE|#PCDATA)+>
<!ELEMENT TITLE - - ANY>
<!ELEMENT BODY - - ANY>
<!ELEMENT DATELINE - - (#PCDATA)>
<!ELEMENT AUTHOR - - (#PCDATA)>
<!ELEMENT D - - (#PCDATA)>
<!ELEMENT UNKNOWN - - (#PCDATA)>
<!ELEMENT MKNOTE - - (#PCDATA)>

<!ATTLIST REUTERS
                        OLDID CDATA #REQUIRED
                        NEWID CDATA #REQUIRED
                        CSECS CDATA #IMPLIED
```

Collecting Documents

- **Kaggle** (data/text mining competition) also provides various data sets (https://www.kaggle.com/datasets?tags=11208-linguistics).


Dataset
BREAKING NEWS
150
News Category Dataset
Identify the type of news based on headlines and short descriptions
Rishabh Misra • updated 9 months ago (Version 2)
Data
Kernels (25)
Discussion (1)
Activity
Metadata
Download (25 MB)
New Notebook
Your Dataset download has started.
Show your appreciation with an upvote
150
Usability 10.0
License CC0: Public Domain
Tags deep learning, classification, nlp, linguistics


News_Category_Dataset_v2.json - Notepad
File Edit Format View Help

```
{"category": "CRIME", "headline": "There Were 2 Mass Shootings In Texas Last Week, But Only 1 On
{"category": "ENTERTAINMENT", "headline": "Will Smith Joins Diplo And Nicky Jam For The 2018 Worl
{"category": "ENTERTAINMENT", "headline": "Hugh Grant Marries For The First Time At Age 57", "aut
{"category": "ENTERTAINMENT", "headline": "Jim Carrey Blasts 'Castrato' Adam Schiff And Democrats
{"category": "ENTERTAINMENT", "headline": "Julianna Margulies Uses Donald Trump Poop Bags To Pick
{"category": "ENTERTAINMENT", "headline": "Morgan Freeman 'Devastated' That Sexual Harassment Cla
{"category": "ENTERTAINMENT", "headline": "Donald Trump Is Lovin' New McDonald's Jingle In 'Tonig
{"category": "ENTERTAINMENT", "headline": "What To Watch On Amazon Prime That\u2019s New This Wee
{"category": "ENTERTAINMENT", "headline": "Mike Myers Reveals He'd 'Like To' Do A Fourth Austin P
{"category": "ENTERTAINMENT", "headline": "What To Watch On Hulu That\u2019s New This Week", "aut
{"category": "ENTERTAINMENT", "headline": "Justin Timberlake Visits Texas School Shooting Victims
{"category": "WORLD NEWS", "headline": "South Korean President Meets North Korea's Kim Jong Un To
{"category": "IMPACT", "headline": "With Its Way Of Life At Risk, This Remote Oyster-Growing Regi
{"category": "POLITICS", "headline": "Trump's Crackdown On Immigrant Parents Puts More Kids In Ar
{"category": "POLITICS", "headline": "'Trump's Son Should Be Concerned': FBI Obtained Wiretaps Of
{"category": "POLITICS", "headline": "Edward Snowden: There's No One Trump Loves More Than Vladim
{"category": "POLITICS", "headline": "Booyah: Obama Photographer Hilariously Trolls Trump's 'Spy'
{"category": "POLITICS", "headline": "Ireland Votes To Repeal Abortion Amendment In Landslide Ref
{"category": "POLITICS", "headline": "Ryan Zinke Looks To Reel Back Some Critics With 'Grand Pivo
{"category": "POLITICS", "headline": "Trump's Scottish Golf Resort Pays Women Significantly Less
```

Text Normalization

- Every NLP task including text mining needs to do **text normalization**:
  - **Segmenting/tokenizing** words in running text
  - Standardize **word formats**
    - convert to **standard** or **common** forms.
  - Segmenting **sentences** in running text

Tokenization

- Breaks the stream of characters into **words** or **tokens**.
  - Trivial for a person familiar with the language structure.
- A computer program, though, being linguistically challenged, would find the task more complicated.
- The reason is that certain characters are sometimes **token delimiters** and sometimes not, depending on the application.
- The characters **space, tab,** and **newline** are always delimiters and are not counted as tokens, often collectively called **white space**.
- The characters **( )** < > **! ?** " are always delimiters and may also be tokens.

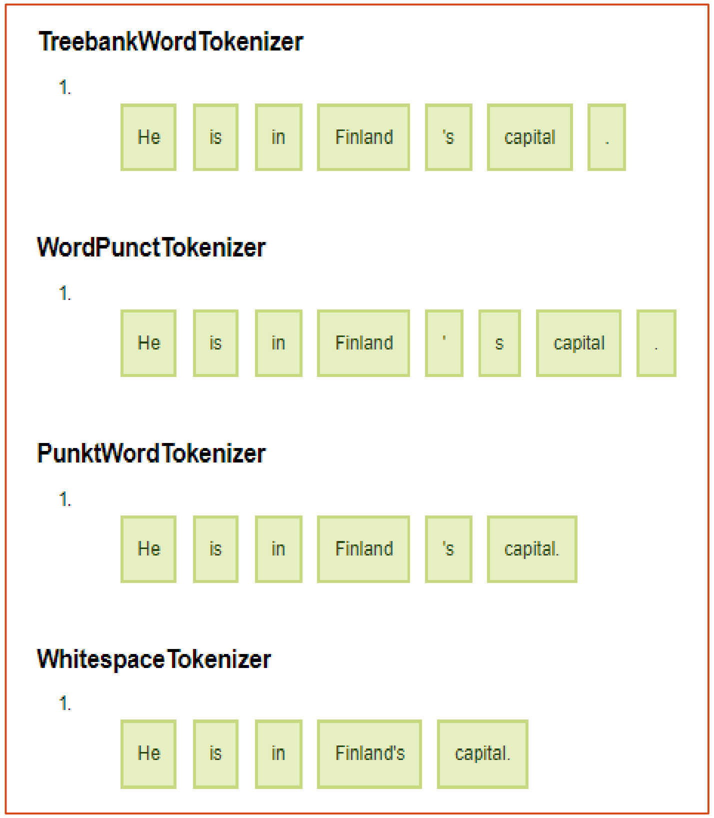
Tokenization

- The characters . , : - ' may or may not be delimiters, depending on their environment.
- Example cases
  - Numbers: 100,000 or 333-1221
  - Abbreviations: e.g. => for example
  - Part of the current token: isn't => [isn, 't] vs [is, n't]
  - Possessive: Tess'
- To get the best possible features, one may need to customize the tokenizer for the available text.
  - E.g., part:123-4567
- The tokenization process is language-dependent.

Example Issues in Tokenization

- `Finland's capital` → `Finland Finlands Finland's ?`
- `what're, I'm, isn't` → `What are, I am, is not ?`
- `Hewlett-Packard` → `Hewlett Packard ?`
- `state-of-the-art` → `state of the art ?`
- `San Francisco` → one token or two?

- Online **Word Tokenization with Python NLTK**
  - http://text-processing.com/demo/tokenize/
  - E.g., “He is in Finland's capital.”
  - Spilt standard contractions – don’t => do n’t
- Replacing contractions with their expansions before Tokenization may help.
  - `isn't -> is not`
  - `Can't -> cannot`



TreebankWordTokenizer

1. He | is | in | Finland | 's | capital | .

WordPunctTokenizer

1. He | is | in | Finland | ' | s | capital | .

PunktWordTokenizer

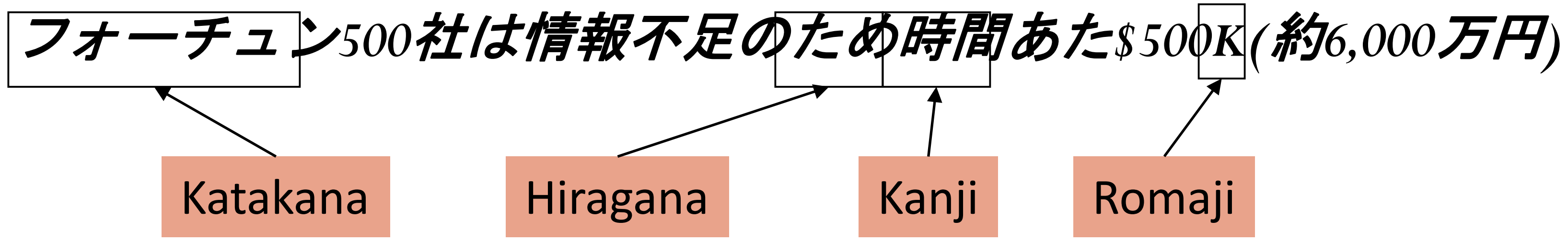
1. He | is | in | Finland | 's | capital.

WhitespaceTokenizer

1. He | is | in | Finland's | capital.

Tokenization: language issues

- 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。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
  - Sharapova now lives in US southeastern Florida
- Further complicated in Japanese, with multiple alphabets intermingled.



Word Tokenization in Chinese

- Also called **Word Segmentation**
- Chinese words are composed of characters.
  - Average word is 2.4 characters long.
- Standard baseline segmentation algorithm:
  - **Maximum Matching** (also called Greedy)

Maximum Matching Word Segmentation Algorithm

- Given a wordlist of Chinese (i.e. **dictionary**), and a string.

1) Start a pointer at the beginning of the string
2) Find the **longest word in dictionary** that matches the string starting at pointer
3) Move the pointer over the word in string
4) Go to 2

- **莎拉波娃**现在居住在美国东南部的佛罗里达。
- **莎拉波娃** 现在 **居住 在 美国** 东南部 **的 佛**罗里达
- Sharapova now lives in US southeastern Florida

Max-match segmentation illustration

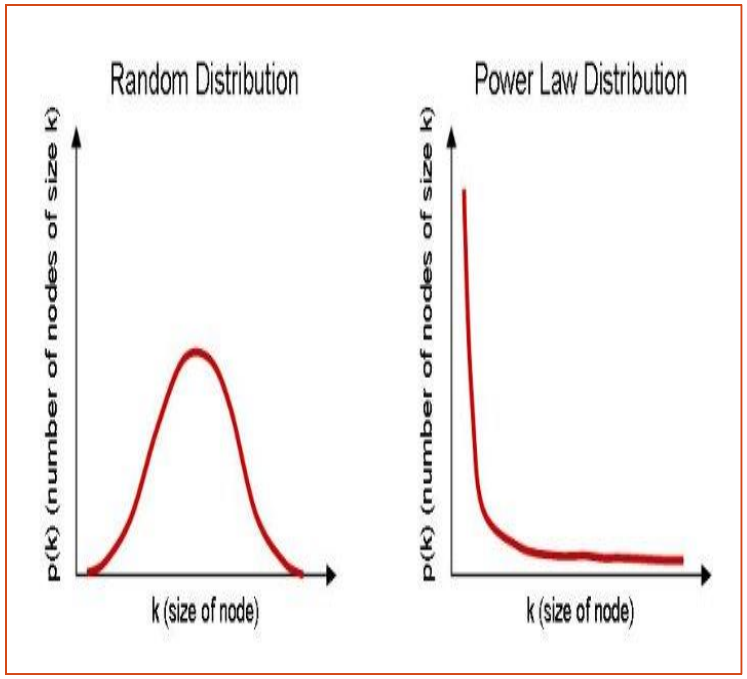
- Thecatinthehat — the cat in the hat
- Thetabledownthere — the table down there

  theta bled own there

- Doesn't generally work in English!
- But works well in Chinese
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
- Modern **probabilistic** segmentation algorithms even better.
  - E.g., "*the table*" has a higher chance than "*theta bled*".

Words Properties

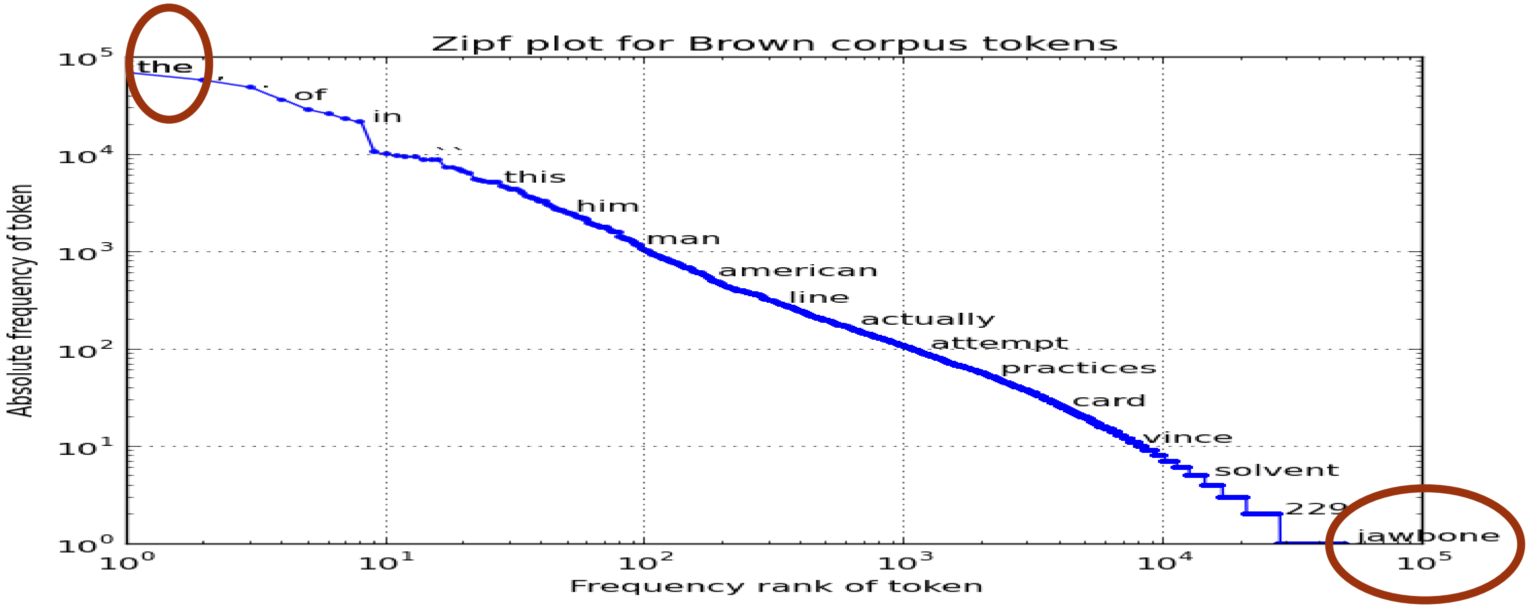
- Relations among word surface forms and their senses:
  - **Homonymy**: same form, but **different** meaning
    - E.g., bank: river bank and financial institution
  - **Polysemy**: same form, **related** meaning
    - E.g., man: the **human** species, **male** of the human species, and **adult males** of the human species
  - **Synonymy**: different form, same meaning
    - E.g., singer and vocalist
- Word frequencies in texts have **power law distribution**:
  - One quantity varies as a power of another
  - …small number of very important words
  - …big number of low importance words.
  - Also called Zipf's Law
  - Feature dimension becomes big, and have sparse matrix



Zipf's Law

- *Zipf's law* states that, if t_1 is the most common term in the collection, t_2 is the next most common, and so on, then the collection frequency cf_i (rank)of the *i*th most common term is proportional to $1/i$: $\mathrm{cf}_i \propto \frac{1}{i}$

=> given a large sample of words used, the **frequency of any word** is **inversely proportional** to its **rank** in the frequency table



Stop Words

- Stop-words are words that from non-linguistic view do not carry information.
  - They have mainly functional role.
  - Usually we remove them to help the methods to perform better.
    - Topic detection: may remove stop words

Why remove stop words?

- Natural language dependent – examples:
  - English: A, ABOUT, ABOVE, ACROSS, AFTER, AGAIN, AGAINST, ALL, ALMOST, ALONE, ALONG, ALREADY, ALSO, …
- Whether or not removing stop words is necessary is application-dependent.
  - Author detection; may keep stop words

Stop Words

- Example Stop words.
  - *Information Systems AsiaWeb - provides research, IS-related commercial materials, interaction, and even research sponsorship by interested corporations with a focus on Asia Pacific region.*
  - *Survey of Information Retrieval - guide to IR, with an emphasis on web-based projects. Includes a glossary, and pointers to interesting papers.*

Word Cloud

Google
social
media
project
today
opportunity
including
WordPress
platforms
Honrado
search
might
series
right
Dr
new
reason
listed
look
Doodle
surrounding
voice
forget
little
late
less
instead
reasons
roll
lost
take
themes
almost
facial
came
ton
provide
Bake-Off
sense
featured
home
based
along
included
entirely
group
earlier
good
Boston
activity
number
buttons
finding
Bake-Offs
surgeon
Los
page
now
purpose
running
conclusion
key
giving
determine
following
big
phase
attached
rolled
make
Angeles
Take
brand
scope
completed
eight
building
recently
also
Doodge
winning
fun
bit
sharing
breakdown
plastic
ongoing
members
hear
screengrab
addition
Group
Also
space
important
tiny
Theme
main
allowing
Carlo
time
viral
entire
really
favorite
course
theme
focused
spread
headline
summer
website
noticed
appears
just
color
Yelp
Places
major
feature
part
first
read
great
one

Normalization

- Need to **normalize** terms
  - Information Retrieval (IR): indexed text & query terms must have the same form.
    - We want to match ***U.S.A.*** and ***USA***
- We define equivalence classes of terms by making the two terms common (USA)
- Alternative: query expansion
  - Enter: ***window*** Search: ***window, windows***
  - Enter: ***windows*** Search: ***Windows, windows, window***
- Potentially more powerful, but less efficient

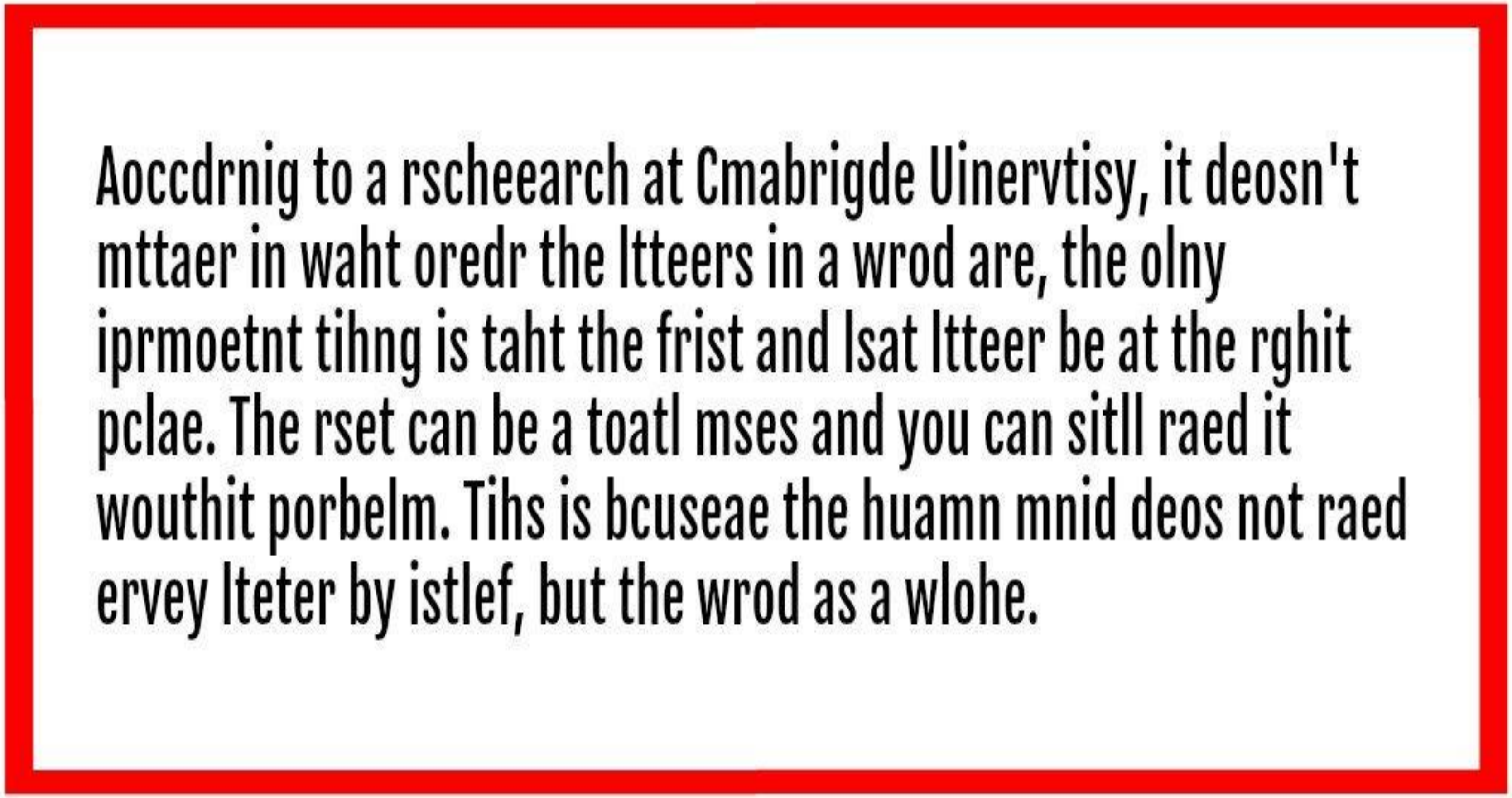
Normalization

- Converts each of the tokens to **a standard form**, a process usually referred to as ***stemming*** or ***lemmatization***.
- Whether or not this step is necessary is application-dependent.
  - Raw form may be useful => e.g., spam detection
- One effect of normalization is to **reduce the number of distinct types** (i.e. unique terms) in a text corpus and to **increase the frequency of occurrence of (significant) individual types**.
  - E.g., *types and typed* → *type*
- For classification algorithms that take frequency into account, this can sometimes make a difference.
  - May ease term scarcity issue (matrix is sparse, i.e. too many zero).
  - Reduce feature dimension (the number of features)

Normalization - Case folding

- Applications like IR: reduce all letters to lower case
  - Since users tend to use lower case
    - E.g., Car, CAR -> car
  - Possible exception: upper case in mid-sentence?
    - E.g.:
      - ***General Motors* vs. *general motors***
      - ***Fed*** vs. ***fed***
        - Fed : Federal Reserve
      - ***SAIL*** vs. ***sail***
        - SAIL: Stanford Artificial Intelligence Language, etc.
- For Sentiment Analysis and Information Extraction
  - Case is helpful (***US*** versus ***us*** is important).
  - E.g., "*US won a gold medal*"; "*They like US.*" *Vs.* "*They like us.*"

Can you read this?



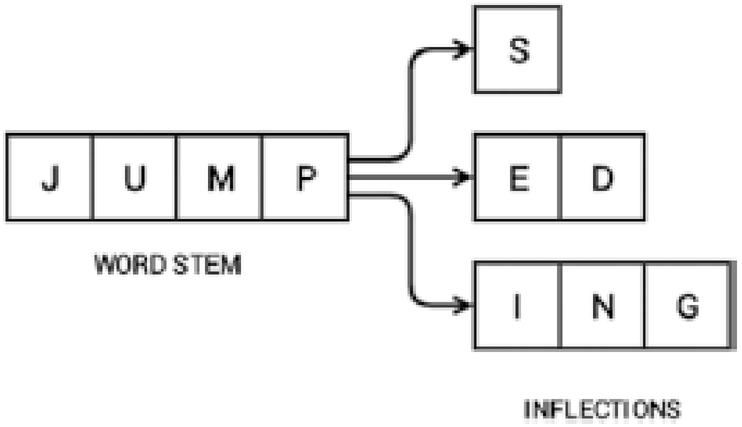
Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

Some Terms: Morphology

- **Morphemes**:
  - Morphemes consist of stems and affixes making up words with meaning.
  - **Stems**: The main part (base form) of a word that can create new words by attaching affixes – process called **inflection**.
    - E.g., **writ** is the stem of writes, writing, and written.
  - **Affixes**: Bits and pieces that adhere to stems (i.e. the prefix and suffix)
    - E.g., **un-**like**-ly**

Stemming

- Stemming – obtaining the base form of a word from its inflected form.



- *Stemming* is **crude chopping of affixes**
  - E.g., ***automate(s), automatic, automation*** 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

Porter's algorithm

- The best-known algorithm is the **Porter stemmer** (www.tartarus.org/~martin/PorterStemmer)
- (http://text-processing.com/demo/stem/)

Martin Porter

Step 1a

sses	→ ss	caresses	→	caress	
ies	→ i	ponies	→	poni	
ss	→ ss	caress	→	caress	
s	→ ø	cats	→	cat	

Step 1b

```
(*v*)ing → ø  walking   → walk
              loving    → lov
              sing      → sing
              king      → king
(*v*)ed  → ø  plastered → plaster
…
```

Note: v =[aeiou]

Step 2 (for long stems)

```
ational→ ate  relational→ relate
izer→ ize     digitizer → digitize
ator→ ate     operator  → operate
…
```

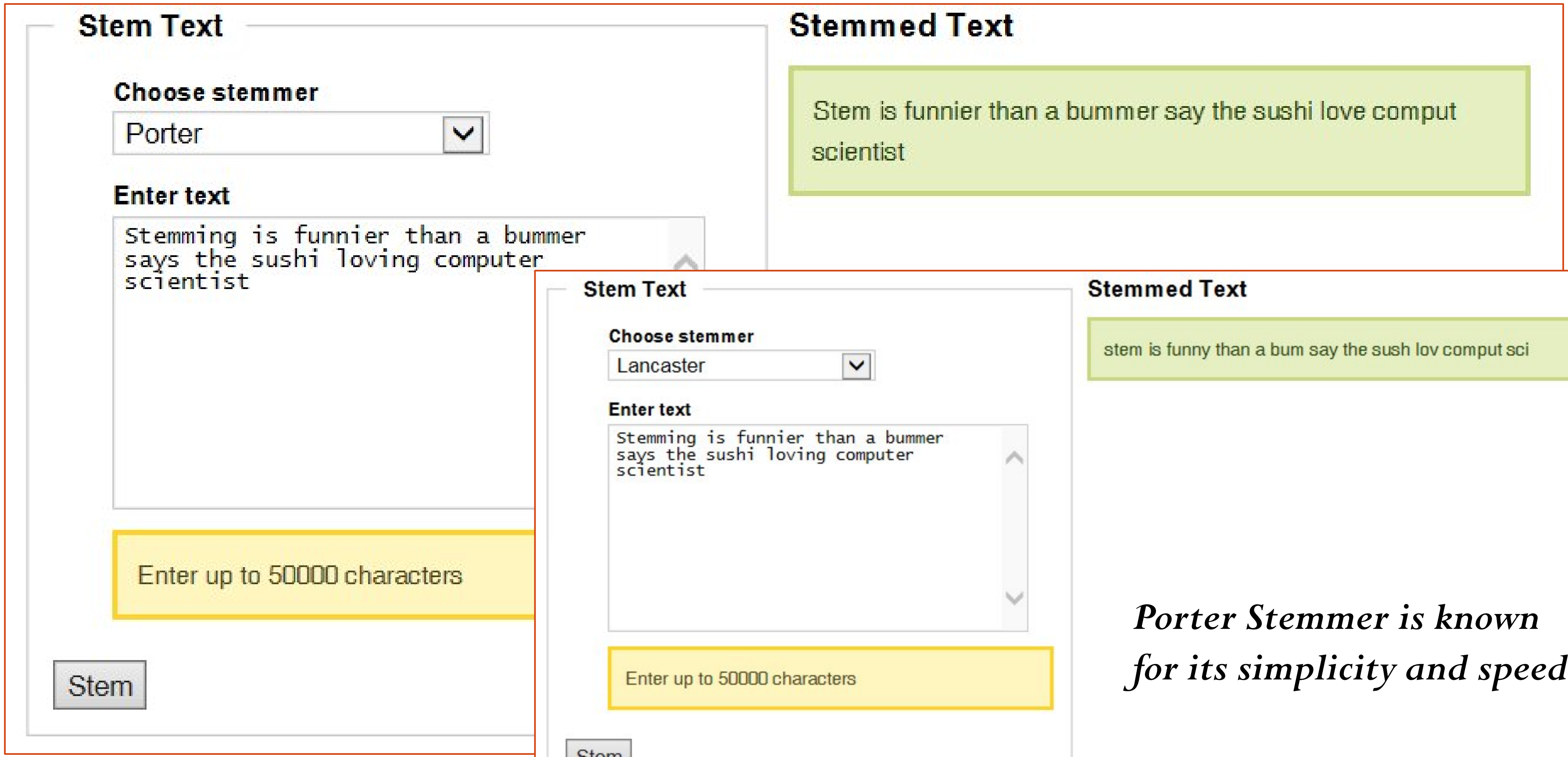
Step 3 (for longer stems)

```
al    → ø  revival    → reviv
able  → ø  adjustable → adjust
ate   → ø  activate   → activ
…
```

Snowball Stemmer – customizable stemmer

Porter's algorithm

- **Stemming with Python NLTK** (http://text-processing.com/demo/stem/).



Stem Text

Choose stemmer: Porter

Enter text

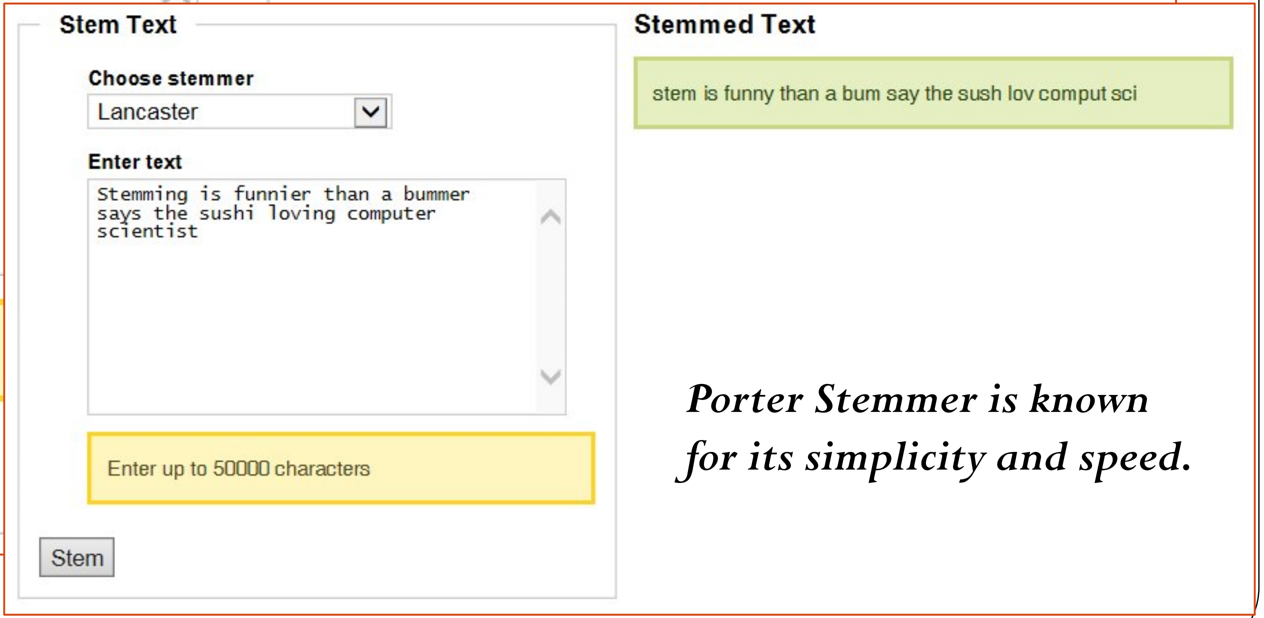
```
Stemming is funnier than a bummer
says the sushi loving computer
scientist
```

Enter up to 50000 characters

Stem

Stemmed Text

Stem is funnier than a bummer say the sushi love comput scientist



Stem Text

Choose stemmer: Lancaster

Enter text

```
Stemming is funnier than a bummer
says the sushi loving computer
scientist
```

Enter up to 50000 characters

Stem

Stemmed Text

stem is funny than a bum say the sush lov comput sci

Porter Stemmer is known for its simplicity and speed.

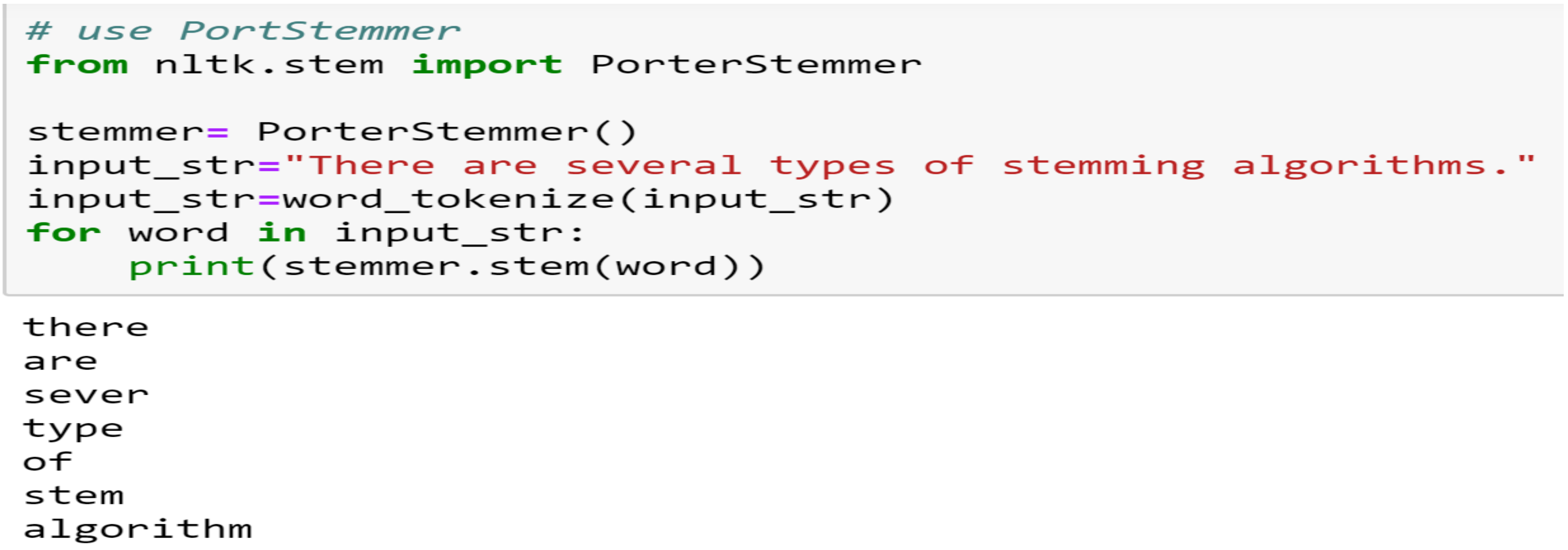
Lancaster stemming

- Lancaster stemming - an **iterative** algorithm containing about 120 rules indexed by the last letter of a suffix. On each iteration, it tries to find an applicable rule by the **last character of the word**. Each rule specifies either a deletion or replacement of an ending.
- Rules iterates until => (1) no rules matched, or (2) if a word starts with a **vowel** with **two letters left**, (3) or if a word starts with a **consonant** with **three characters left**.

Aggressive stemming reduces the number of types in a text collection very drastically, thereby making distributional statistics more reliable.

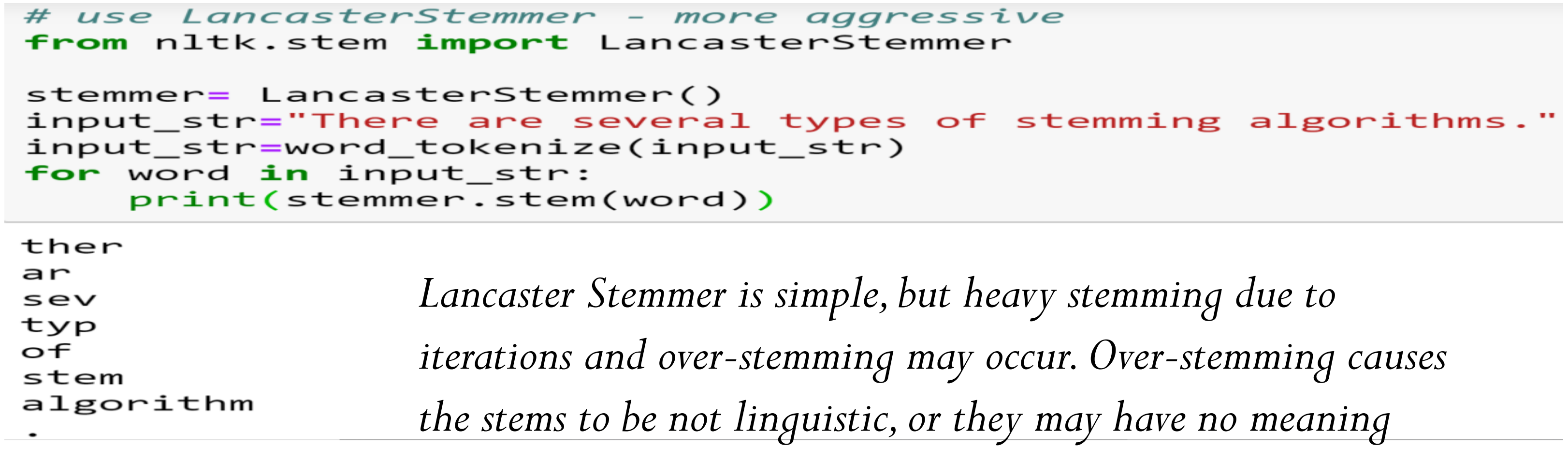
Word	Porter Stemmer	Lancaster Stemmer
friend	friend	friend
friendship	friendship	friend
friends	friend	friend
friendships	friendship	friend
destabilize	destabil	dest

Stemming - Python



```
# use PortStemmer
from nltk.stem import PorterStemmer

stemmer= PorterStemmer()
input_str="There are several types of stemming algorithms."
input_str=word_tokenize(input_str)
for word in input_str:
    print(stemmer.stem(word))
```

```
there
are
sever
type
of
stem
algorithm
```



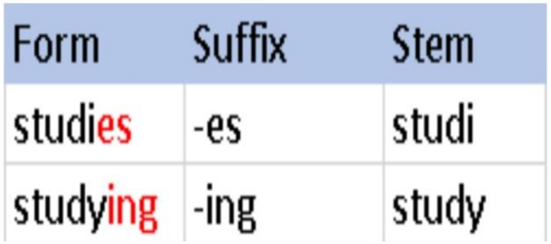
```
# use LancasterStemmer - more aggressive
from nltk.stem import LancasterStemmer

stemmer= LancasterStemmer()
input_str="There are several types of stemming algorithms."
input_str=word_tokenize(input_str)
for word in input_str:
    print(stemmer.stem(word))
```

```
ther
ar
sev
typ
of
stem
algorithm
.
```

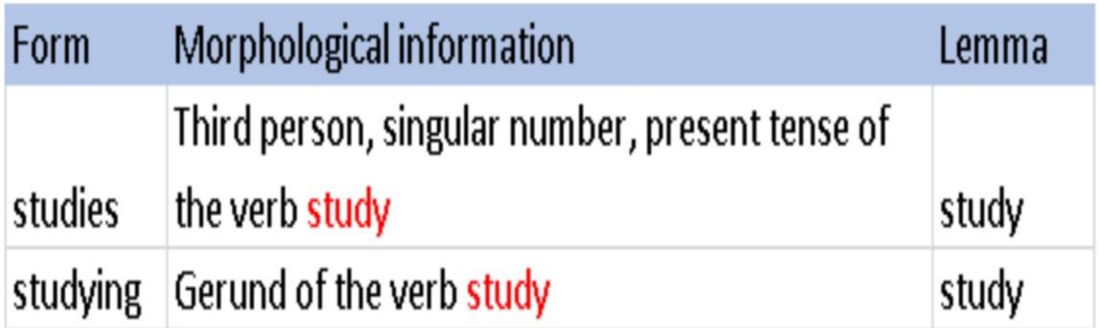
Lancaster Stemmer is simple, but heavy stemming due to iterations and over-stemming may occur. Over-stemming causes the stems to be not linguistic, or they may have no meaning

Lemmatization

- Lemmatization is very similar to stemming - remove word affixes to get to a base form of the word.
- However, in lemmatization the base form is the ***root* word** and not the **root stem**.
- The root stem may not always be a lexicographically correct word, i.e., it may not be present in the dictionary but the **root word**, also known as the lemma, will **always be present in the dictionary**.



Form	Suffix	Stem
studies	-es	studi
studying	-ing	study



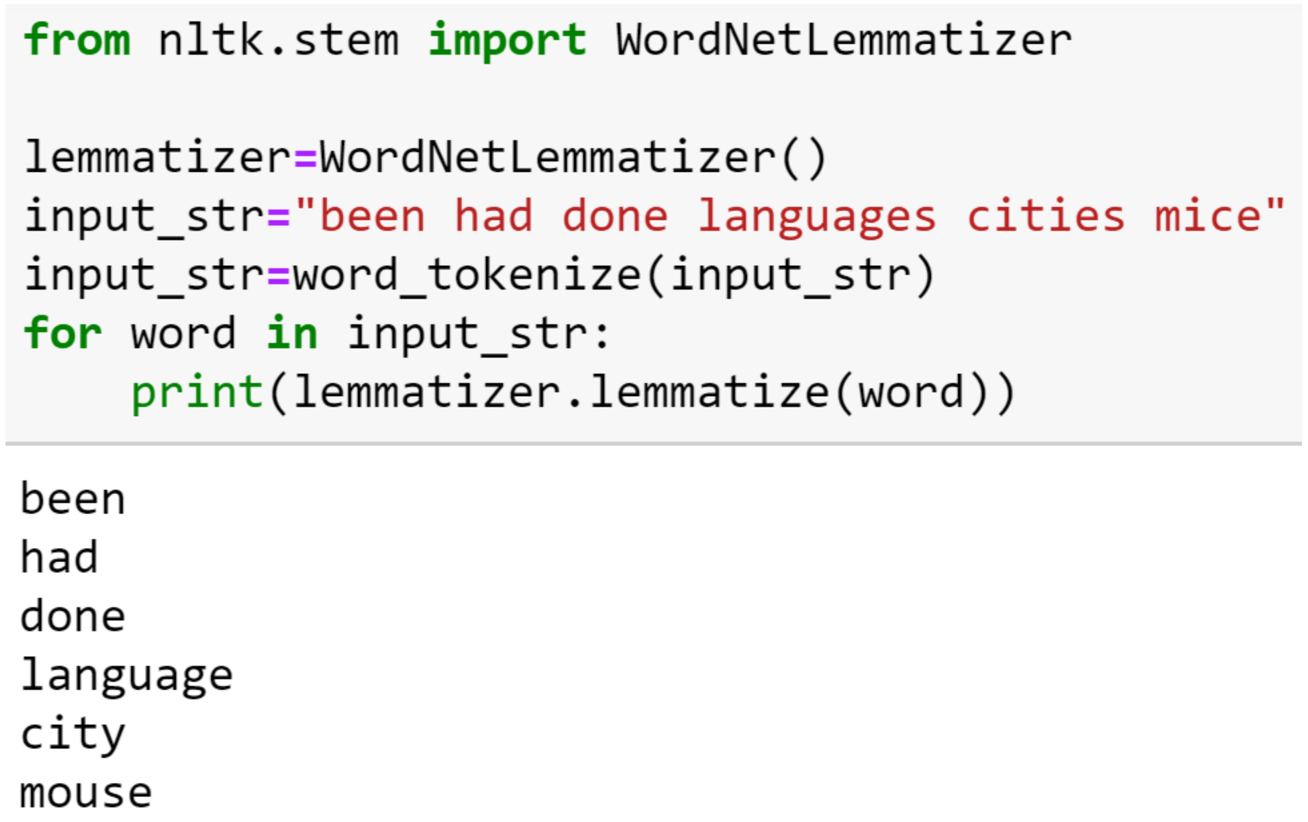
Form	Morphological information	Lemma
studies	Third person, singular number, present tense of the verb study	study
studying	Gerund of the verb study	study

- The lemmatization process is **slower** than stemming - additional step involved where the root form or lemma is formed by removing the affix from the word <u>if and only if the lemma is present in the dictionary</u>.

Lemmatization

- Converts to a **root form** with no inflectional or derivational prefixes and suffixes.
  - **Inflectional suffixes** are endings such as "-ed", "-ing", "s", etc.
    - Create **different grammatical forms** of the same word for example, changing singular to plural (dog ⟶ dogs), or changing present tense to past tense (walk ⟶ walked)
  - **Derivational suffixes** are endings such as "-ism", "-ful", "-fy", etc.
    - **Change the meaning** of the word
  - E.g., "denormalization" is reduced to the stem "norm".
  - E.g., "reapplied", "applications" -> "apply"

Lemmatization – Python



```python
from nltk.stem import WordNetLemmatizer

lemmatizer=WordNetLemmatizer()
input_str="been had done languages cities mice"
input_str=word_tokenize(input_str)
for word in input_str:
    print(lemmatizer.lemmatize(word))
```

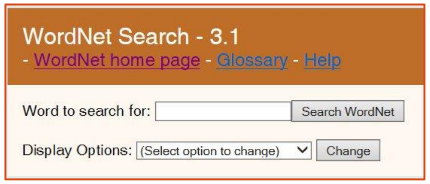
```
been
had
done
language
city
mouse
```

Phrases in the form of frequent N-Grams

- Simple way for generating phrases are frequent **n-grams**:
  - **N-Gram** is a sequence of n consecutive words (e.g., "machine learning" is 2-gram)
  - **"Frequent n-grams"** are the ones which appear in all observed documents Minimum Frequent or more times.
- The simple and efficient algorithm:
- Given:
  - Set of documents (each document is a sequence of words),
  - MinFreq (minimal n-gram frequency),
  - MaxNGramSize (maximal n-gram length)
- for Len = 1 to MaxNGramSize do
  - Generate candidate n-grams as sequences of words of size Len using frequent n-grams of length Len-1
  - Delete candidate n-grams with the frequency less then MinFreq

WordNet –a database of lexical relations

- WordNet is the most well developed and widely used lexical database for English
  - It consists from 4 databases (nouns, verbs, adjectives, and adverbs)
  - On-line version: http://wordnetweb.princeton.edu/perl/webwn

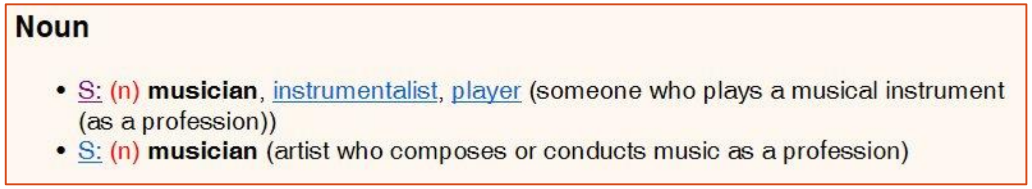


WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: [ ] Search WordNet

Display Options: (Select option to change) Change

- Each database consists of sense entries consisting from a set of synonyms (synsets), e.g.,:
  - musician, instrumentalist, player
  - person, individual, someone
  - life form, organism, being



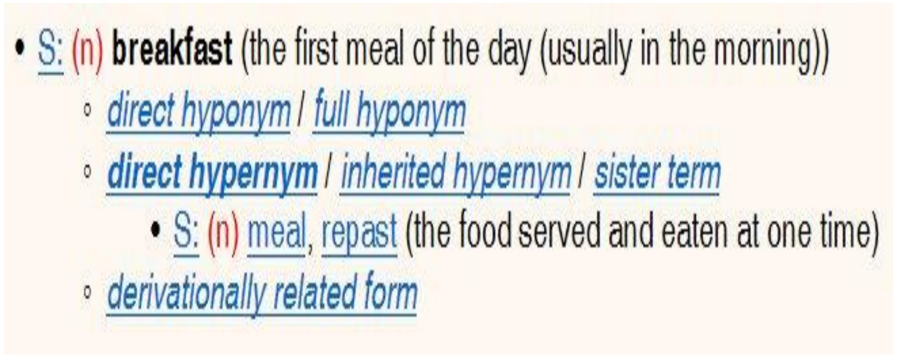
Noun

- S: (n) **musician**, instrumentalist, player (someone who plays a musical instrument (as a profession))
- S: (n) **musician** (artist who composes or conducts music as a profession)

Category	Unique Forms	Number of Senses
Noun	94474	116317
Verb	10319	22066
Adjective	20170	29881
Adverb	4546	5677

WordNet relations

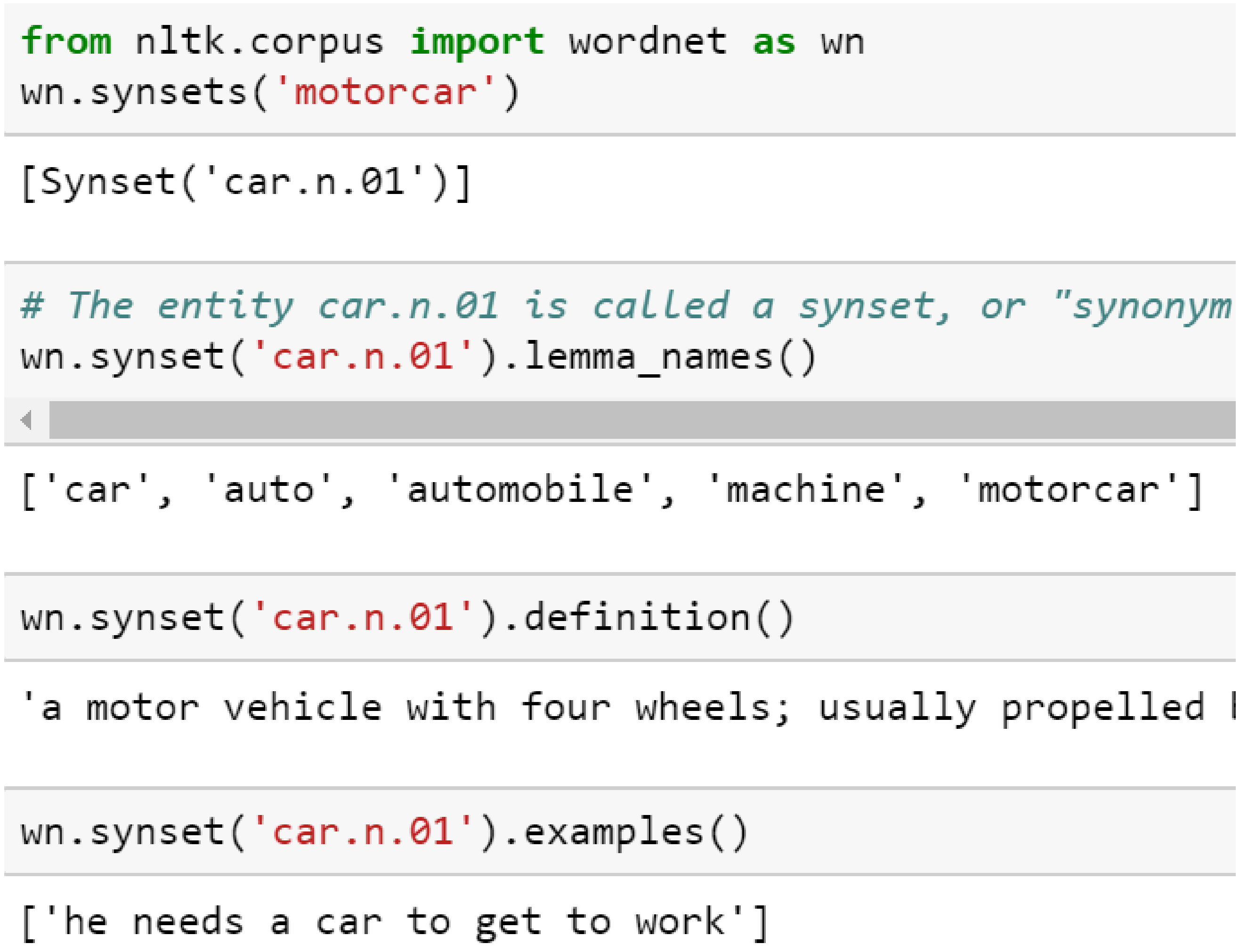
- Each WordNet entry is connected with other entries in a graph through relations.
- Relations in the database of **nouns**



- S: (n) **breakfast** (the first meal of the day (usually in the morning))
    - direct hyponym / full hyponym
    - **direct hypernym** / inherited hypernym / sister term
        - S: (n) meal, repast (the food served and eaten at one time)
    - derivationally related form

Relation	Definition	Example
Hypernym	From concepts to superordinate	breakfast -> meal
Hyponym	From concepts to subtypes	meal -> lunch
Has-Member (member meronym)	From groups to their members	faculty -> professor
Member-Of (member holonym)	From members to their groups	co-pilot -> crew
Has-Part (part meronym)	From wholes to parts	table -> leg
Part-Of (part holonym)	From parts to wholes	course -> meal
Antonym	Opposites	leader -> follower

Wordnet – Python



```
from nltk.corpus import wordnet as wn
wn.synsets('motorcar')
```

```
[Synset('car.n.01')]
```

```
# The entity car.n.01 is called a synset, or "synonym
wn.synset('car.n.01').lemma_names()
```

```
['car', 'auto', 'automobile', 'machine', 'motorcar']
```

```
wn.synset('car.n.01').definition()
```

```
'a motor vehicle with four wheels; usually propelled
```

```
wn.synset('car.n.01').examples()
```

```
['he needs a car to get to work']
```

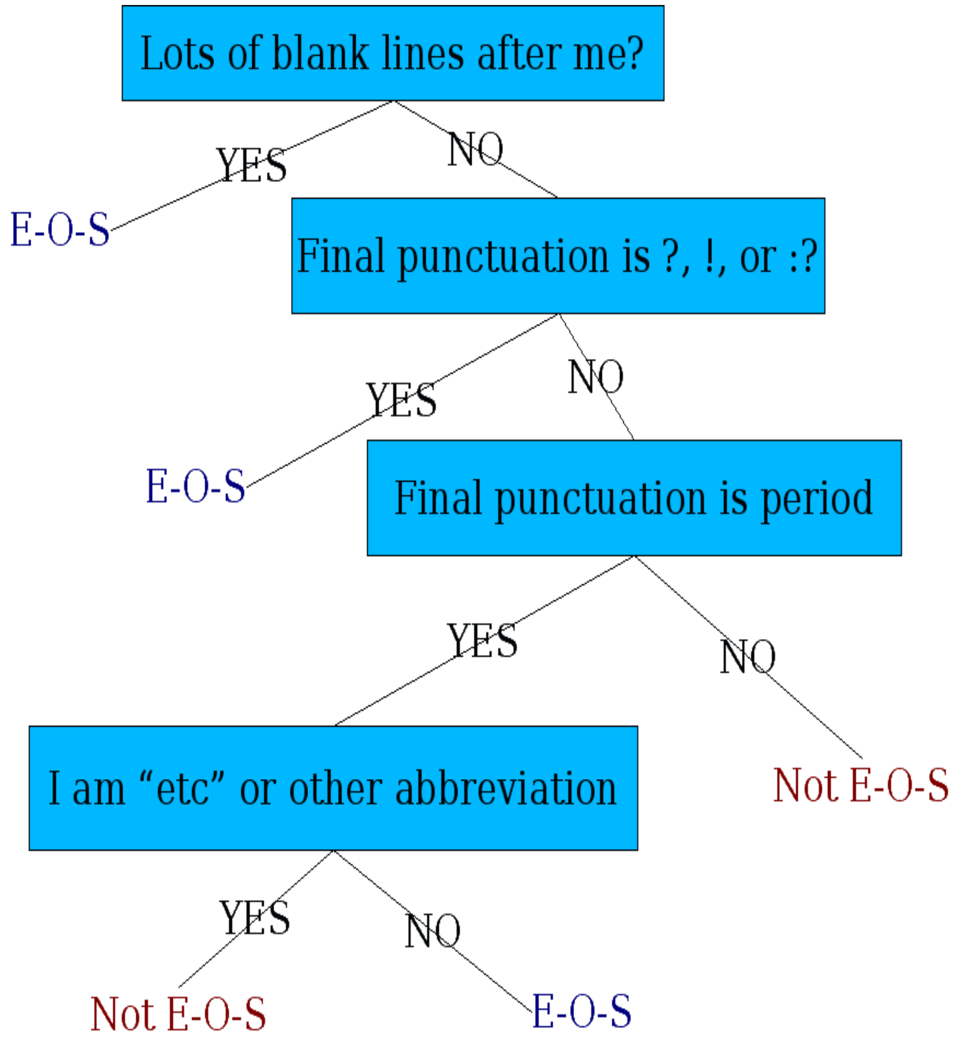
Sentence Boundary Determination

- For more sophisticated linguistic parsing, the algorithms often require **a complete sentence as input**.
  - E.g., Sentence-level sentiment analysis
- We shall also see other information extraction algorithms that operate on a sentence at a time.
- Tokenization function also uses a complete sentence as input.
- Sentence boundary determination is essentially the problem of deciding ***which instances of a period (.) followed by whitespace are sentence delimiters and which are not*** since we assume that the characters ? and ! are unambiguous sentence boundaries.

Sentence Segmentation

- !, ? are relatively unambiguous.
- Period "." is quite ambiguous
  - Sentence boundary
  - Abbreviations like Inc. or Mr.
  - Numbers like .02% or 4.3
- Build a binary classifier
  - Looks at a "."
  - Decides EndOfSentence/NotEndOfSentence
  - Classifiers: hand-written rules, regular expressions, or machine-learning

Determining if a word is end-of-sentence: a Decision Tree

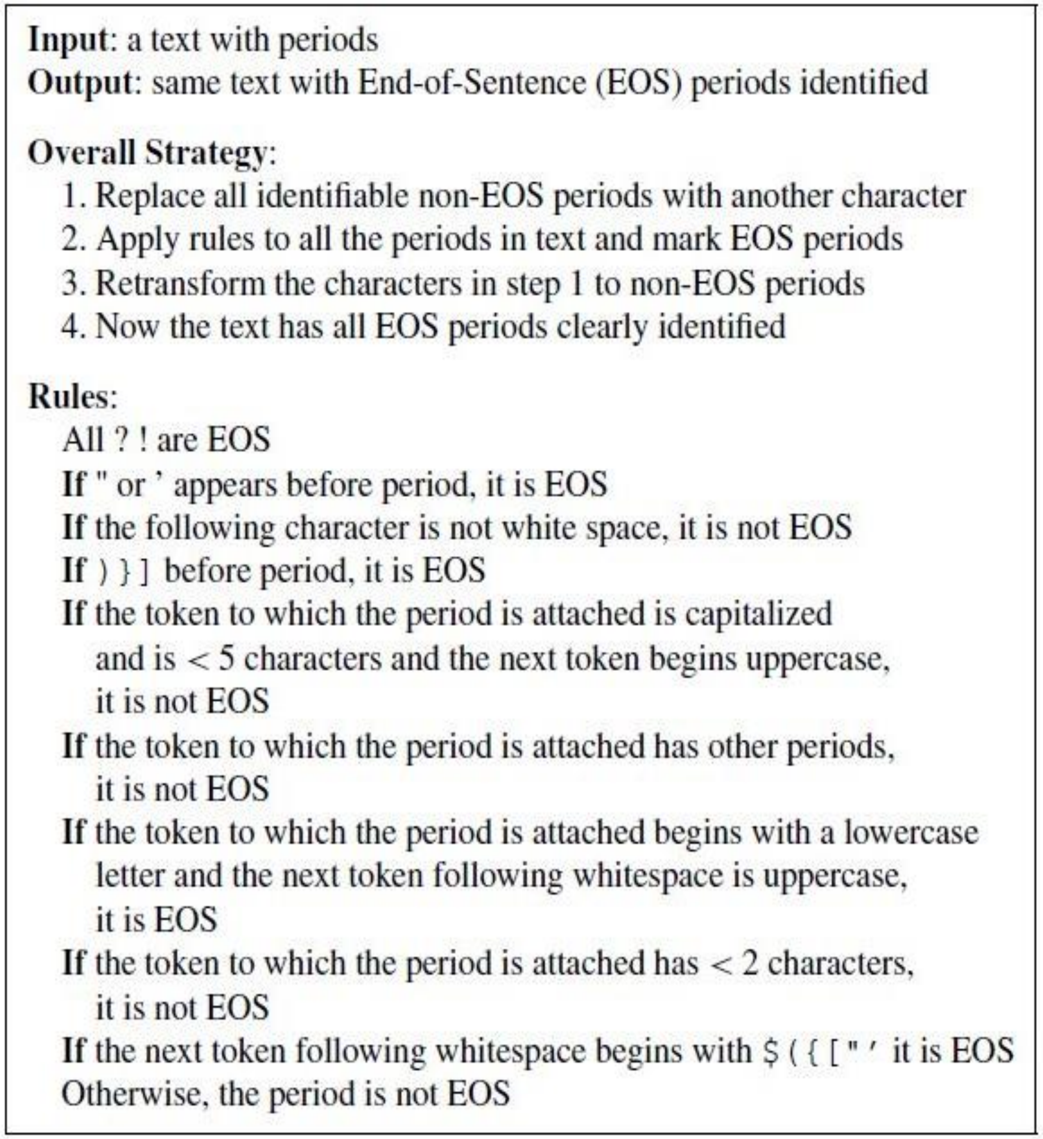

Lots of blank lines after me?
YES
NO
E-O-S
Final punctuation is ?, !, or :?
YES
NO
E-O-S
Final punctuation is period
YES
NO
Not E-O-S
I am "etc" or other abbreviation
YES
NO
Not E-O-S
E-O-S

Implementing Decision Trees

- A decision tree is just an if-then-else statement.
    - We can think of the questions in a decision tree.
- The interesting research is choosing the features.
- Setting up the structure is often too hard to do by hand.
    - Hand-building only possible for very simple features, domains
        - For numeric features, it's too hard to pick each threshold.
    - Instead, structure usually learned by machine learning from a training corpus.
- The features could be exploited by any kind of classifier.
    - SVM, Neural Nets, Logistic regression, etc.

Sentence Boundary Determination

- Sentence Detection Algorithm.
  - **Hand-written rules**
  - Examples of EOS
    - …”.　　…’.
    - …).　　…}.
    - …].　　…x.Y
    - . $　　. (
    - . {　　. [
    - . “　　. ‘
  - Examples of not EOS
    - Ph.D.
    - www.google.com
    - i.e.



Input: a text with periods
Output: same text with End-of-Sentence (EOS) periods identified

Overall Strategy:
1. Replace all identifiable non-EOS periods with another character
2. Apply rules to all the periods in text and mark EOS periods
3. Retransform the characters in step 1 to non-EOS periods
4. Now the text has all EOS periods clearly identified

Rules:
All ? ! are EOS
If " or ' appears before period, it is EOS
If the following character is not white space, it is not EOS
If) }] before period, it is EOS
If the token to which the period is attached is capitalized and is < 5 characters and the next token begins uppercase, it is not EOS
If the token to which the period is attached has other periods, it is not EOS
If the token to which the period is attached begins with a lowercase letter and the next token following whitespace is uppercase, it is EOS
If the token to which the period is attached has < 2 characters, it is not EOS
If the next token following whitespace begins with $ ({ [" ' it is EOS
Otherwise, the period is not EOS

Referenced Materials

- Fundamentals of Predictive Text Mining, Sholom M. Weiss, Nitin Indurkhya, and Tong Zhang, Springer.
  - Chapter 2
- Natural Language Processing, Dan Jurafsky and Christopher Manning, http://www.stanford.edu/~jurafsky/NLPCourseraSlides.html