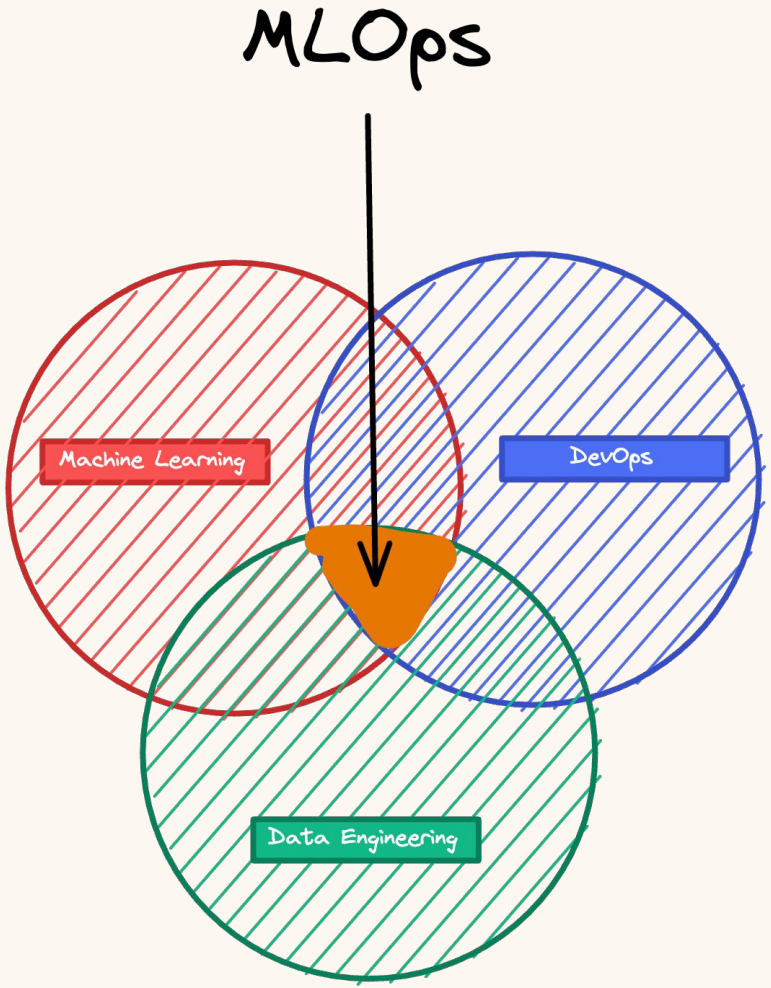
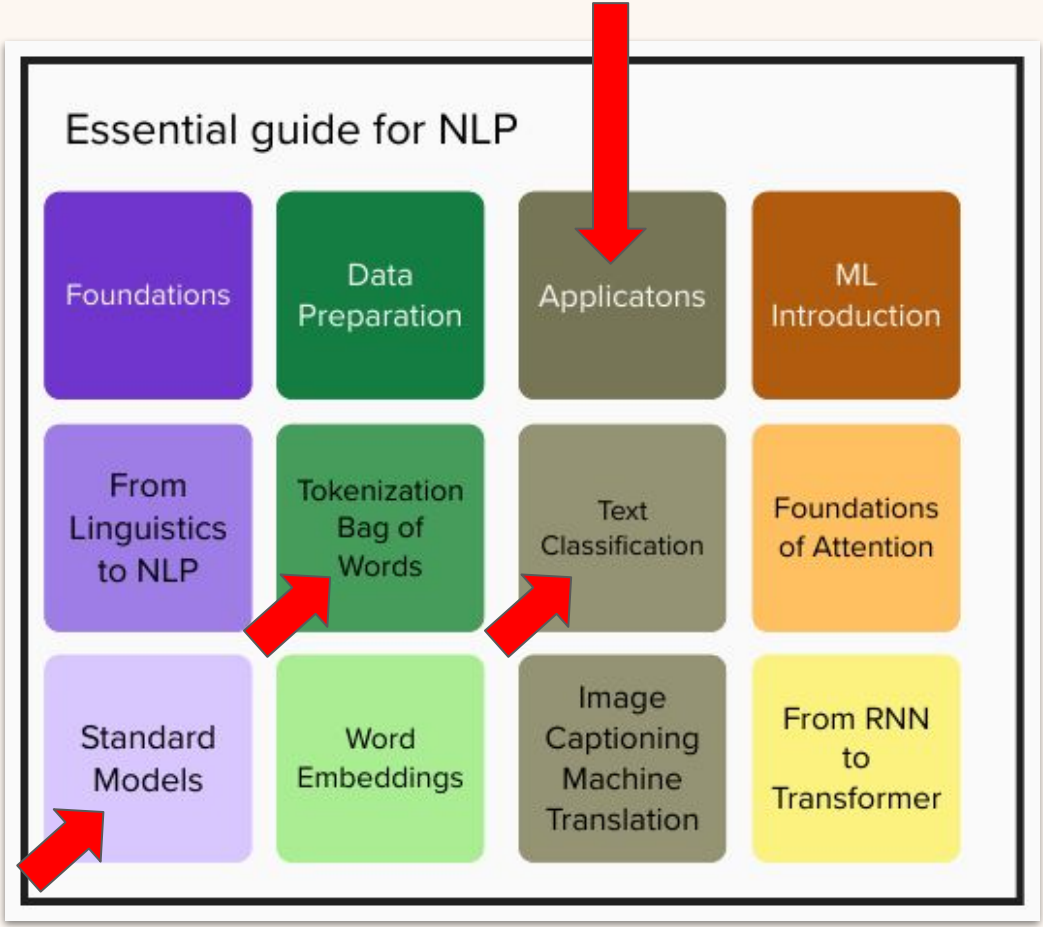
Essential Guide for NLP

Steps to Process Film Review Data for Sentiment Analysis

DCA0305
ivanovitch.silva@ufrn.br


MLOps
Machine Learning
DevOps
Data Engineering


Essential guide for NLP
Foundations
Data Preparation
Applicatons
ML Introduction
From Linguistics to NLP
Tokenization Bag of Words
Text Classification
Foundations of Attention
Standard Models
Word Embeddings
Image Captioning Machine Translation
From RNN to Transformer

Movie Review Dataset

Top 10 on IMDb this week

★ 8.1

1. The Fall of the House of Usher

+ Watchlist

▶ Trailer

★ 8.0

2. Gen V

Watch options

▶ Trailer

★ 8.2

3. Loki

+ Watchlist

▶ Trailer

★ 6.8

4. Reptile

+ Watchlist

▶ Trailer

★ 6.5

5. Fair Play

+ Watchlist

▶ Trailer

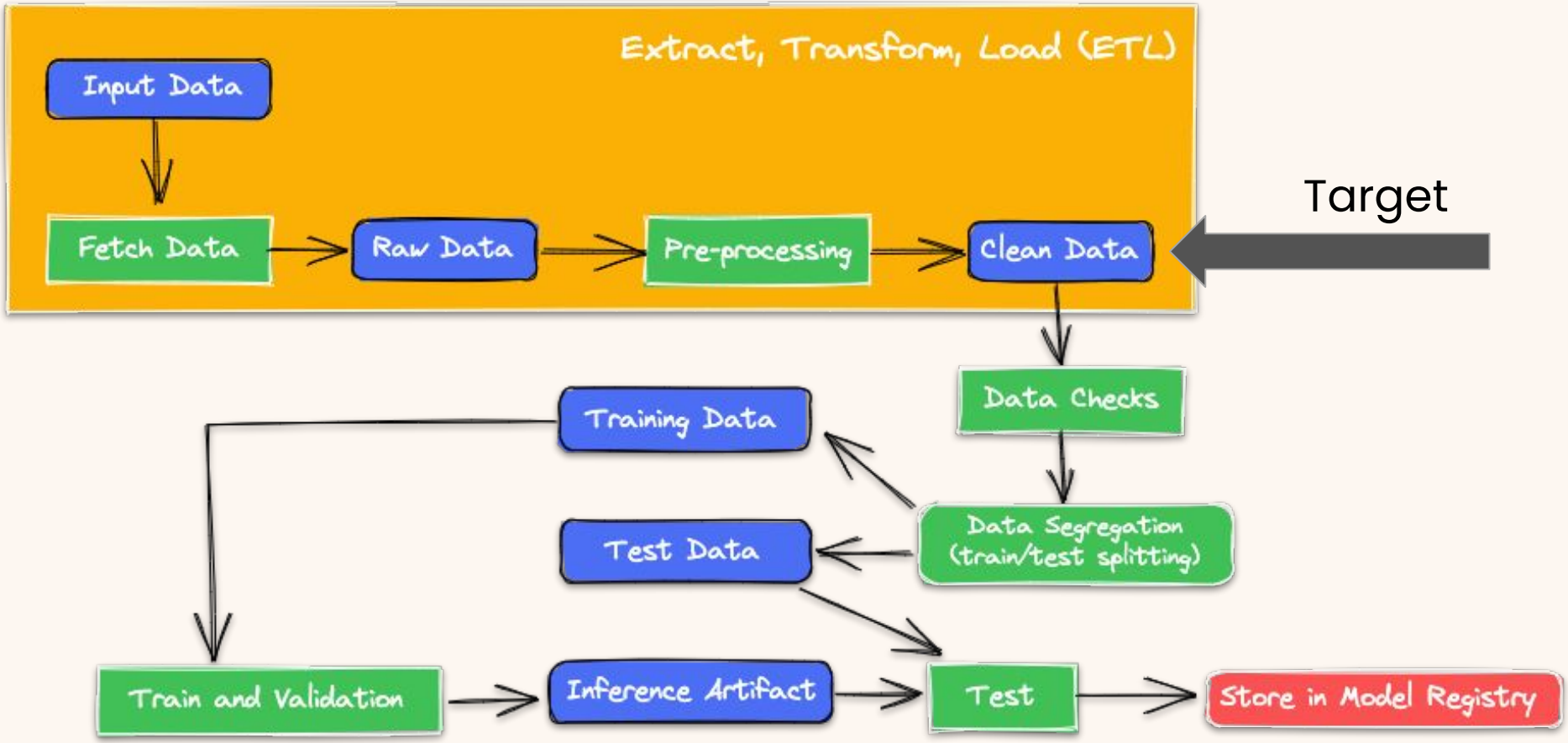
★ 6.6

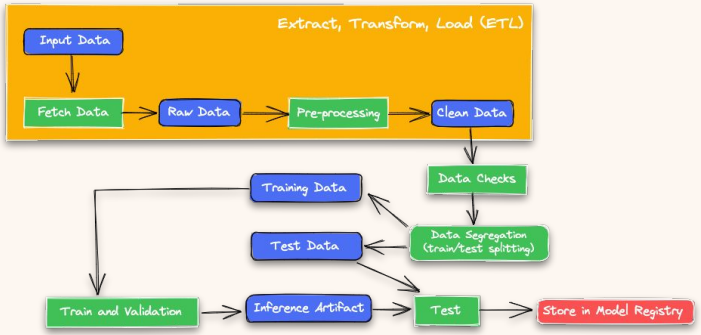
6. Totally Killer

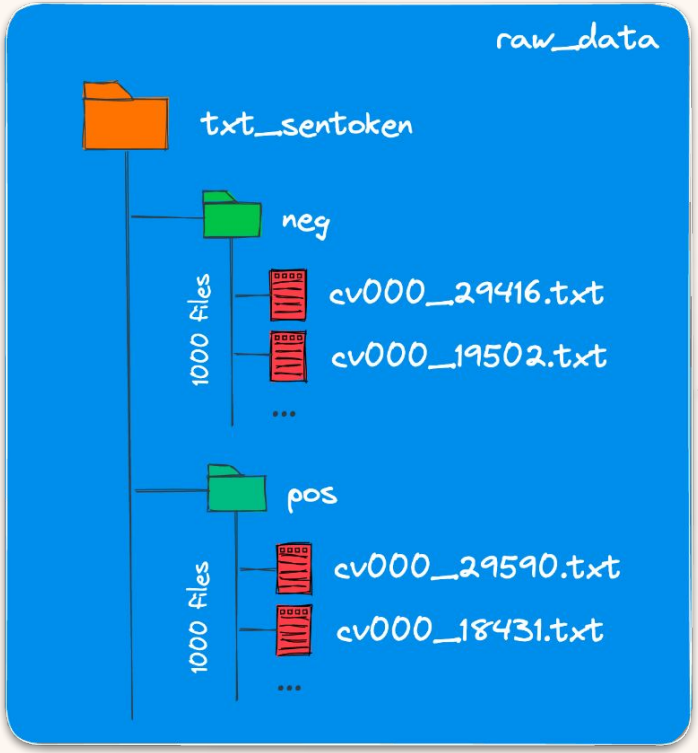
Watch options

▶ Trailer

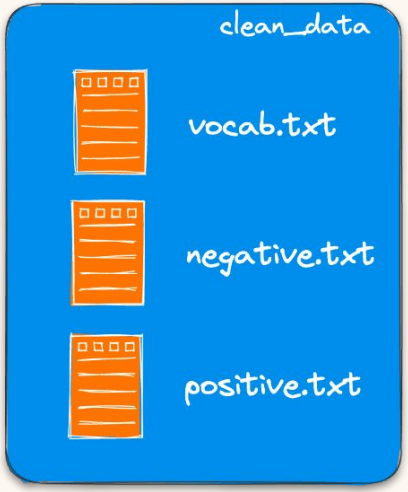
https://www.cs.cornell.edu/people/pabo/movie-review-data/


Extract, Transform, Load (ETL)
Input Data
Fetch Data
Raw Data
Pre-processing
Clean Data
Target
Data Checks
Training Data
Data Segregation
(train/test splitting)
Test Data
Train and Validation
Inference Artifact
Test
Store in Model Registry


Extract, Transform, Load (ETL)
Input Data
Fetch Data
Raw Data
Pre-processing
Clean Data
Data Checks
Training Data
Data Segregation (train/test splitting)
Test Data
Train and Validation
Inference Artifact
Test
Store in Model Registry

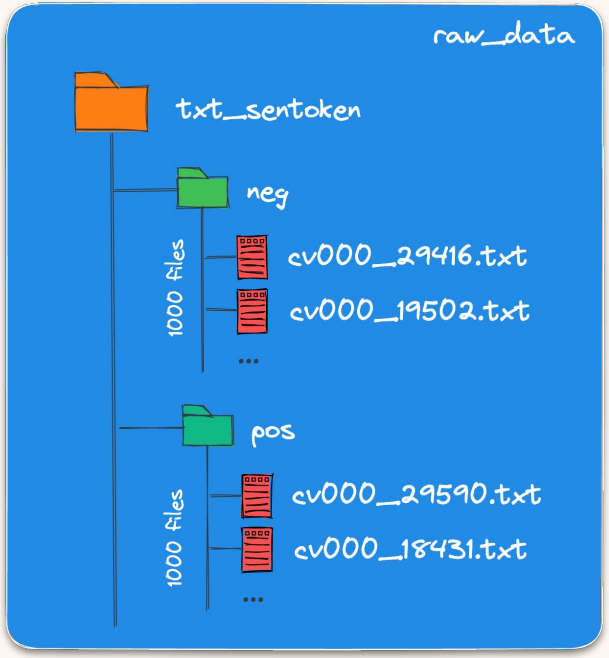

raw_data
txt_sentoken
neg
1000 files
cv000_29416.txt
cv000_19502.txt
...
pos
1000 files
cv000_29590.txt
cv000_18431.txt
...


Pre-Processing


clean_data
vocab.txt
negative.txt
positive.txt

vocab.txt

01 synopsis
02 meliassa
03 woman
04 likes
05 smoke
...
...
14801 sade
14802 mongkut
14802 rumpo

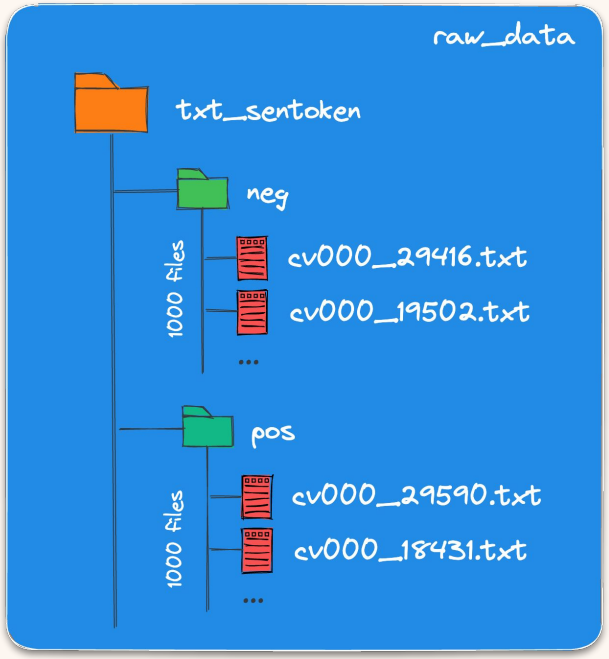


1. Opening and Reading Files
2. Cleaning the Content
3. Compiling a Preliminary Vocabulary
4. Processing Multiple Files
5. Refining the Vocabulary
6. Saving the Final Vocabulary

1. Opening and Reading Files
2. Cleaning the Content
3. Compiling a Preliminary Vocabulary
4. Processing Multiple Files
5. Refining the Vocabulary
6. Saving the Final Vocabulary

vocab.txt

01 synopsis
02 meliassa
03 woman
04 likes
05 smoke
...
...
14801 sade
14802 mongkut
14802 rumpo

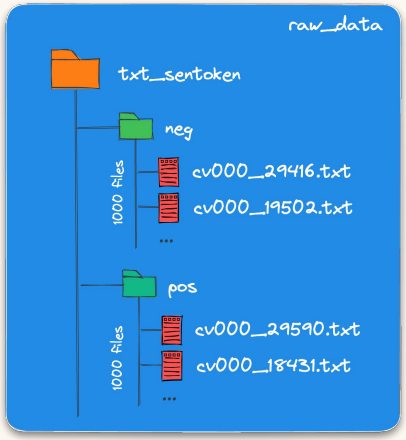



```python
# load doc into memory
def load_doc(filename):

    # open the file as read only
    file = open(filename, 'r')
    # read all text
    text = file.read()
    # close the file
    file.close()
    return text
```

vocab.txt

01 synopsis
02 meliassa
03 woman
04 likes
05 smoke
...
...
14801 sade
14802 mongkut
14802 rumpo

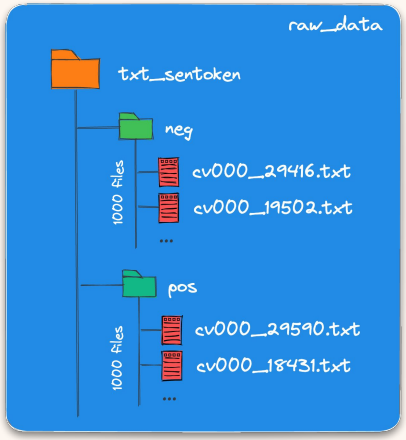


1. Opening and Reading Files
2. Cleaning the Content
3. Compiling a Preliminary Vocabulary
4. Processing Multiple Files
5. Refining the Vocabulary
6. Saving the Final Vocabulary

```
# turn a doc into clean tokens
def clean_doc(doc):
  # split into tokens by white space
  tokens = doc.split()
  # prepare regex for char filtering
  re_punc = re.compile('[%s]' % re.escape(string.punctuation))
  # remove punctuation from each word
  tokens = [re_punc.sub('', w) for w in tokens]
  # remove remaining tokens that are not alphabetic
  tokens = [word for word in tokens if word.isalpha()]
  # filter out stop words
  stop_words = set(stopwords.words('english'))
  tokens = [w for w in tokens if not w in stop_words]
  # filter out short tokens
  tokens = [word for word in tokens if len(word) > 1]
  return tokens
```

vocab.txt

01 synopsis
02 meliassa
03 woman
04 likes
05 smoke
...
...
14801 sade
14802 mongkut
14802 rumpo



1. Opening and Reading Files
2. Cleaning the Content
3. Compiling a Preliminary Vocabulary
4. Processing Multiple Files
5. Refining the Vocabulary
6. Saving the Final Vocabulary

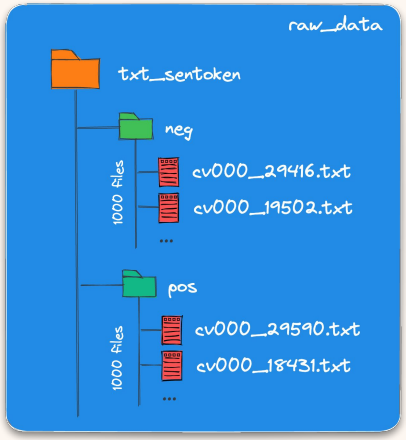
```
# load doc and add to vocab
def add_doc_to_vocab(filename, vocab):
  # load doc
  doc = load_doc(filename)
  # clean doc
  tokens = clean_doc(doc)
  # update counts
  vocab.update(tokens)
```

```
# load all docs in a directory
def process_docs(directory, vocab):
  # walk through all files in the folder
  for filename in listdir(directory):
    # skip files that do not have the right extension
    if not filename.endswith(".txt"):
      continue
    # create the full path of the file to open
    path = directory + '/' + filename
    # add doc to vocab
    add_doc_to_vocab(path, vocab)
```

```
# define vocab
vocab = Counter()
# add all docs to vocab
process_docs('txt_sentoken/neg', vocab)
process_docs('txt_sentoken/pos', vocab)
```

vocab.txt

01 synopsis
02 meliassa
03 woman
04 likes
05 smoke
...
...
14801 sade
14802 mongkut
14802 rumpo

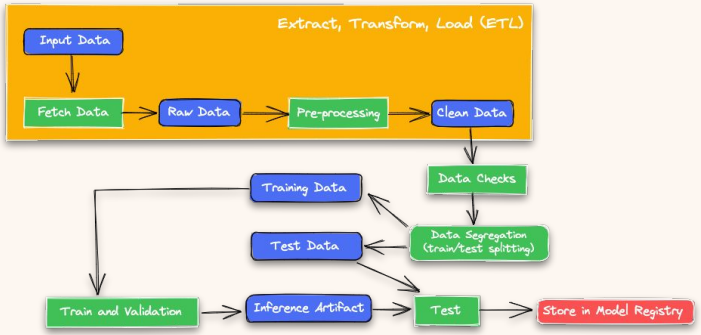


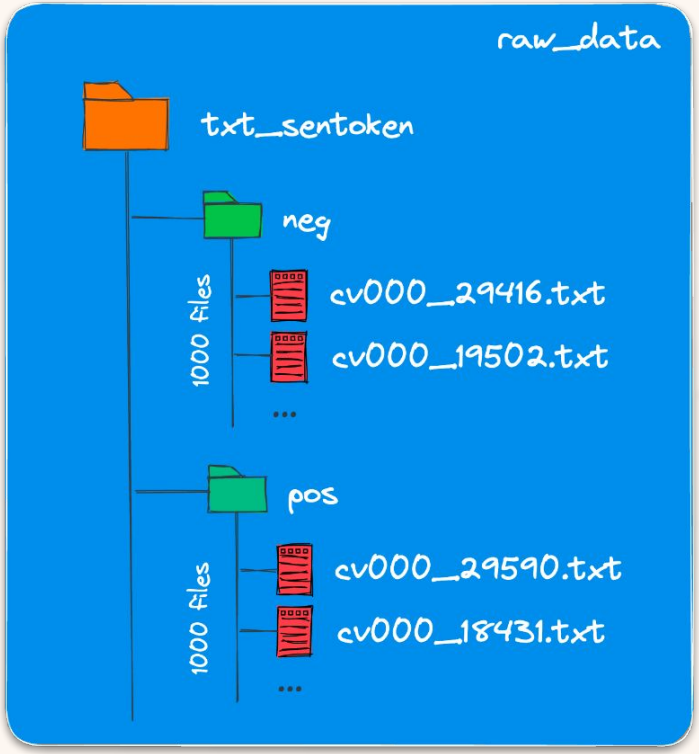
1. Opening and Reading Files
2. Cleaning the Content
3. Compiling a Preliminary Vocabulary
4. Processing Multiple Files
5. Refining the Vocabulary
6. Saving the Final Vocabulary

```
# save list to file
def save_list(lines, filename):
    data = '\n'.join(lines)
    file = open(filename, 'w')
    file.write(data)
    file.close()
```

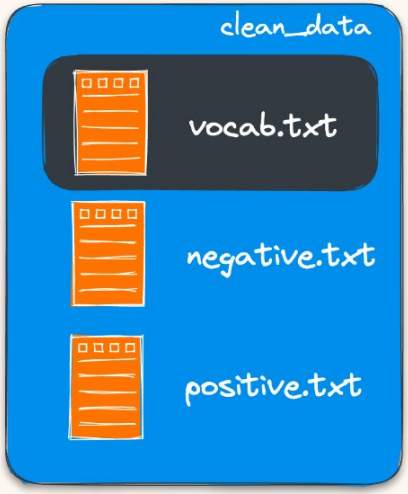
```
# keep tokens with > 5 occurrence
min_occurrence = 5
tokens = [k for k,c in vocab.items() if c >= min_occurrence]
print(len(tokens))

# save tokens to a vocabulary file
save_list(tokens, 'vocab.txt')
```


Extract, Transform, Load (ETL)
Input Data
Fetch Data
Raw Data
Pre-processing
Clean Data
Data Checks
Training Data
Test Data
Data Segregation (train/test splitting)
Train and Validation
Inference Artifact
Test
Store in Model Registry


raw_data
txt_sentoken
neg
1000 files
cv000_29416.txt
cv000_19502.txt
...
pos
1000 files
cv000_29590.txt
cv000_18431.txt
...


Pre-Processing


clean_data
vocab.txt
negative.txt
positive.txt

positive.txt

```
already feel hate letters pouring one folks loved wedding singer gets worse much mention titanic sentence he
psychic wounds family sitcom opening segment something film theres guys voice telling us tries imagine biolo
robert redford good playing characters incredible gifts able act like ordinary people natural played fallen
fully loaded entertainment review website coming july didnt really expect much rented stuart saves family mo
fugitive probably one greatest thrillers ever made takes realistic believable characters tells exciting stor
tired hot new releases gone time get video store com dedicated finding hidden gems lie shelves com reviews i
first thing notice movie cold placed minnesota north winter many scenes take place outside long scenes snowc
may years steven spielbergs success jaws years francis coppolas godfather risky ambitious young director nam
think first thing reviewer mention fan xfiles first let assure prior experience series required fully enjoy
present day three sisters reflect parents relationship trying define writerdirector tran scent green vertica
perhaps time say little reading habits really like read ive enjoyed many books lifetime problem im slow read
albert brooks saves day nick time poor summer movies brooks audiences looking cheer positive way may finest
clue unfairly ignored comedy similar murder death big screen version classic board game whats next motion pi
people enjoy science fiction often faced unpleasant surprises due novels stories comic books movies often sc
lisa high art intelligent quiet drama strongest quality aside topnotch central performances perceptive way f
```

```
fortunate enough attend advance screening upcoming thriller conspiracy theory course big deal reviewing movi
man presented us henry portrait serial killer comes wild tale set within elite white trash south coast plot
steven spielbergs amistad based true story group africans board slave ship captured taken america legal disp
capsule style heist film set present robert deniro stars wants retire form crime takes one last job request
zero effect gets title main character daryl zero bill pullman although dont understand truly means last line
asked see movie friend initial reaction hugh grant perhaps wrong harsh street hooker could picture romantic
damn trailers advertising film reveals far much contents would glued sand film retains value thanks excellen
youve got mail works alot better deserves order make film success cast two extremely popular attractive star
trekkies roger energetic hilarious documentary brings viewers world star trek conventions beauty film good o
set wild west carry around arrival rumpo kid sidney james cronies city dealings summary shootings judge burk
anxious see long time friend mine recommended crush neve campbell wanted prove shes hot thinks proved right
```

Main Execution

- The predefined vocabulary is loaded from the file *'vocab.txt'.*
- All negative reviews (from *'txt_sentoken/neg'* directory) are processed and saved to 'negative.txt'.
- Similarly, all positive reviews (from 'txt_sentoken/pos' directory) are processed and saved to 'positive.txt'.

```python
# load doc into memory
def load_doc(filename):
    # open the file as read only
    file = open(filename, 'r')
    # read all text
    text = file.read()
    # close the file
    file.close()
    return text
```

```python
# load vocabulary
vocab_filename = 'vocab.txt'
vocab = load_doc(vocab_filename)
vocab = vocab.split()
vocab = set(vocab)
# prepare negative reviews
negative_lines = process_docs('txt_sentoken/neg', vocab)
save_list(negative_lines, 'negative.txt')
# prepare positive reviews
positive_lines = process_docs('txt_sentoken/pos', vocab)
save_list(positive_lines, 'positive.txt')
```

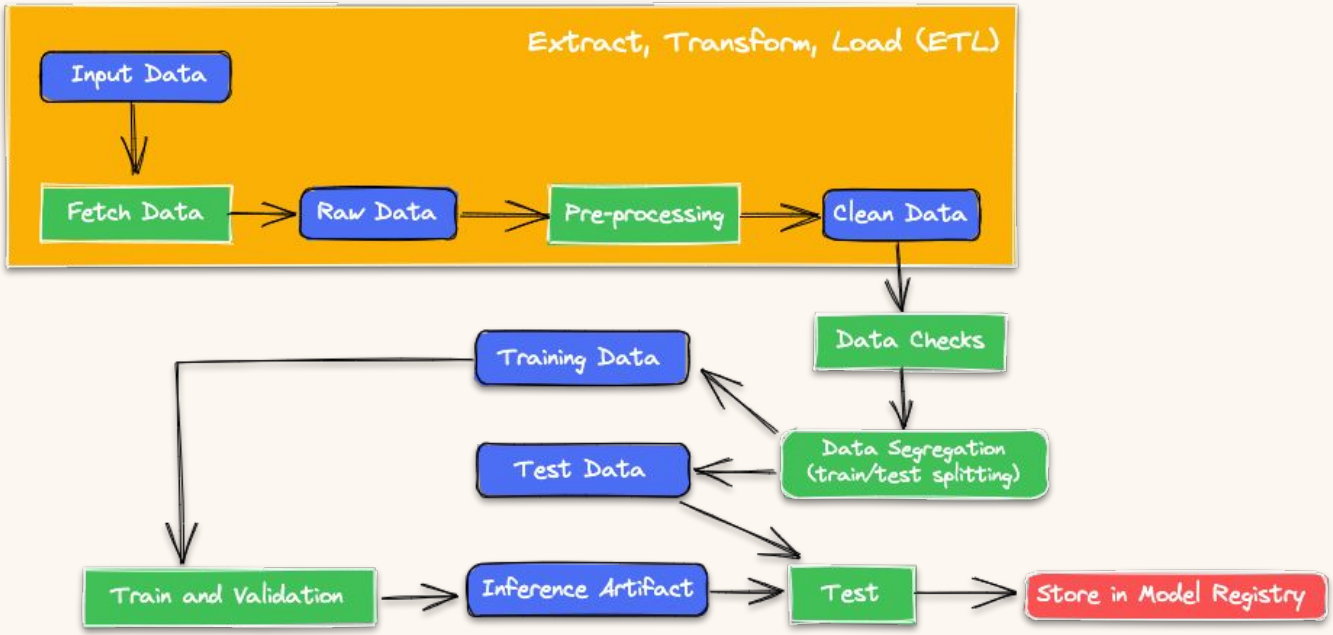
Processing All Documents in a Directory (*process_docs*)

- This function processes all ".txt" files in a given directory.
- It loads each file, cleans it, and keeps words from the predefined vocabulary.
- The cleaned content of each document is added to a list.

```python
# load all docs in a directory
def process_docs(directory, vocab):
    lines = list()
    # walk through all files in the folder
    for filename in listdir(directory):
    # skip files that do not have the right extension
    if not filename.endswith(".txt"):
        continue
    # create the full path of the file to open
    path = directory + '/' + filename
    # load and clean the doc
    line = doc_to_line(path, vocab)
    # add to list
    lines.append(line)
    return lines
```

```python
# load doc, clean and return line of tokens
def doc_to_line(filename, vocab):
    # load the doc
    doc = load_doc(filename)
    # clean doc
    tokens = clean_doc(doc)
    # filter by vocab
    tokens = [w for w in tokens if w in vocab]
    return ' '.join(tokens)
```

When we look at the pipeline, we realize that what was done to clean the dataset has a serious flaw, what would it be?



Cont.