# In-Class Assignment: NLP Pipeline

*DATA 5420/6420*

## Name: Dallin Moore

In this in-class assignment we're going to run through the entire NLP pipeline and apply some common cleaning and text normalizing steps. We'll start with a text that needs extensive processing to run through the battery of processing steps, then we'll do the same on a much more simple text that requires less effort.

What steps you ned to do will depend on the text and the task at hand!

## Basic Outline of Steps:

1. Import text
2. Remove HTML (if applicable)
3. Case conversion
4. Contractions
5. Stemming/Lemmatization
6. Removing Stopwords
7. Tokenize text
8. Text Output

It's important to note that this list is NOT exhaustive, does NOT need to be done in this order, and which steps you choose WILL depend on the task at hand. The point of this exercise is to show you one procedure for cleaning/processing a text and show two options of output. This will vary based on a given text and what you want to do with it after!

Here, we're going to be using lots of familiar libraries and packages, but we'll also introduce some new ones including the popular and useful `spacy` library! We'll also need `nltk`, `re`, `pprint`, `BeautifulSoup`, `contractions`, `pandas`, and `numpy`.

```
import nltk, re, pprint

from urllib import request
from bs4 import BeautifulSoup

!pip install contractions
import contractions
from string import punctuation

import spacy
!python -m spacy download en_core_web_sm                # OR in Jupyter download in terminal

from nltk.tokenize.toktok import ToktokTokenizer
from nltk.corpus import stopwords
tokenizer = ToktokTokenizer()
from nltk import word_tokenize

import pandas as pd
import numpy as np
```

```
Collecting contractions
  Downloading contractions-0.1.73-py2.py3-none-any.whl (8.7 kB)
Collecting textsearch>=0.0.21 (from contractions)
  Downloading textsearch-0.0.24-py2.py3-none-any.whl (7.6 kB)
Collecting anyascii (from textsearch>=0.0.21->contractions)
  Downloading anyascii-0.3.2-py3-none-any.whl (289 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 289.9/289.9 kB 2.9 MB/s eta 0:00:00
Collecting pyahocorasick (from textsearch>=0.0.21->contractions)
  Downloading pyahocorasick-2.0.0-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.mar
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 110.8/110.8 kB 5.6 MB/s eta 0:00:00
Installing collected packages: pyahocorasick, anyascii, textsearch, contractions
Successfully installed anyascii-0.3.2 contractions-0.1.73 pyahocorasick-2.0.0 textsearch
2024-01-26 05:59:25.125752: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:92
2024-01-26 05:59:25.125814: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:60
2024-01-26 05:59:25.127484: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1
2024-01-26 05:59:26.723431: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-T
Collecting en-core-web-sm==3.6.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 74.4 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.7.0,>=3.6.0 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/di
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist
Requirement already satisfied: typer<0.10.0,>=0.3.0 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: pathy>=0.10.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.10/dis
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.10/dist-packages
```

```
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/py
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from s
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.10/dist
Requirement already satisfied: pathlib-abc==0.1.1 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dis
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.10/dis
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-package
✓ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
```

## 1) Import Text - UTF-8 Encoded

For this example we'll run a `Helpful Hints for Halloween` text through the NLP pipeline. Why this text? Well it's pretty messy and provides a good opportunity to demonstrate different processing functions, plus I love Halloween.

```
url = "https://www.gutenberg.org/cache/epub/68984/pg68984-images.html"
response = request.urlopen(url)

raw = response.read().decode('utf-8-sig')
raw
```

```
'<!DOCTYPE html>\r\n<html lang="en">\r\n<head>\r\n<meta charset="utf-8"><style>\r\n#pg-
header div, #pg-footer div {\r\n    all: initial;\r\n    display: block;\r\n    margin-
top: 1em;\r\n    margin-bottom: 1em;\r\n    margin-left: 2em;\r\n}\r\n#pg-footer div.ag
ate {\r\n    font-size: 90%;\r\n    margin-top: 0;\r\n    margin-bottom: 0;\r\n    text
-align: center;\r\n}\r\n#pg-footer li {\r\n    all: initial;\r\n    display: block;\r\n
margin-top: 1em;\r\n    margin-bottom: 1em;\r\n    text-indent: -0.6em;\r\n}\r\n#pg-foo
ter div.secthead {\r\n    font-size: 110%;\r\n    font-weight: bold;\r\n}\r\n#pg-footer
#project-gutenberg-license {\r\n    font-size: 110%;\r\n    margin-top: 0;\r\n    margi
n-bottom: 0;\r\n    text-align: center;\r\n}\r\n#pg-header-heading {\r\n    all: inheri
t;\r\n    text-align: center;\r\n    font-size: 120%;\r\n    font-weight:bold;\r\n}\r\n
#pg-footer-heading {\r\n    all: inherit;\r\n    text-align: center;\r\n    font-size:
120%;\r\n    font-weight: normal;\r\n  …'
```

**It's clear that we want to remove the HTML tags, and we can use `html.parser` to do that. But that's not going to get rid of all unwanted characters. Let's remove the html and then figure out what else needs to be removed...**

## 2) Remove HTML Tags + Unwanted Characters & Trim Text

Let's start by defining a function to remove unwanted html tags, and then we'll build it out based on other characters we want to remove:

```
def text_cleaner(text):
    soup = BeautifulSoup(text, 'html.parser')
    [s.extract() for s in soup(['iframe', 'script'])]
    stripped_text = soup.get_text()
    stripped_text = re.sub('[\r|\n|\r\n]+', '\n', stripped_text)
    stripped_text = re.sub('’',"'",stripped_text)
    stripped_text = re.sub(r"[^'\w\s\.]+", '', stripped_text)
    stripped_text = re.sub(r'\d+\.|\d+', '', stripped_text)
    stripped_text = re.sub(r"HALLOWE'EN|[hH]allowe'en",'halloween', stripped_text)
    # iteratively add cleaning steps here
    return stripped_text

clean_text = text_cleaner(raw)
```

```
clean_text[0:5000]
```

```
'\n      The Project Gutenberg eBook of Helps and Hints for halloween by Laura Rountree
Smith.\n    \nThe Project Gutenberg eBook of Helps and hints for halloween\nThis ebook
is for the use of anyone anywhere in the United States and\nmost other parts of the wor
ld at no cost and with almost no restrictions\nwhatsoever. You may copy it give it away
or reuse it under the terms\nof the Project Gutenberg License included with this ebook
or online\nat www.gutenberg.org. If you are not located in the United States\nyou will
have to check the laws of the country where you are located\nbefore using this eBook.\n
Title Helps and hints for halloween\nAuthor Laura Rountree Smith\nRelease date Septembe
r   eBook \nLanguage English\nOriginal publication United States March Brothers \nCredi
ts Charlene Taylor and the Online Distributed Proofreading Team at httpswww.pgdp.net Th
is file was produced from images generously made available by The Internet ArchiveAmeri
can Libraries.\n START OF THE PROJECT GUTE…'
```

**Now let's find the beginning and end of the text and trim it:**

```
print("[", clean_text.find("START OF THE PROJECT GUTENBERG"), ":", clean_text.rfind("END OF
```

```
[ 958 : 67070 ]
```

```
clean_text = clean_text[958:67070] # trim the text
```

## 3) Lowercase

**Next in the pipeline is setting all characters to lowercase. Why do we care about doing this?**

To standardize the text and reduce the amount of tokens that we are working with.

```
def lowercase(text):
  sents_lower = text.lower() # fill in
  return sents_lower

lower_text = lowercase(clean_text) # apply to clean_text
lower_text
```

```
'start of the project gutenberg ebook helps and hints for halloween \n\nhelps and hints
\nfor\nhalloween\nby\nlaura rountree smith\nmarch brothers publishers\n   wright ave. l
ebanon ohio\n\ncopyright  by\nmarch brothers\n\ncontents\npage\nintroduction\n\nparty s
uggestions\nnutcrack night\n\nhalloween stunts\na shadow play\n\nthe black cat stunt\n
\na pumpkin climbing game\n\nexercises\nhalloween acrostic\n\ntake care tables are turn
ed\n\ndrills\nclown drill and song\n\nautumn leaf drill\n\ncattail drill\n\nmuff drill
\n\ndialogs and plays\nthe halloween ghosts\n\non halloween night\n\njack frost's surpr
ise\n\nan historical halloween\n\nthe witch's dream\n\na halloween carnival and waxwork
show\n\nthe play of pomona\n\nhalloween puppet play\n\n\nnote\nsend for our complete\nc
atalog in which will be\nfound all the accessories\nneeded in carrying out the\nideas g
iven in this book.\nmarch brothers publishers\n   wright ave. lebanon ohio\n\nintroduct
ion\nhist be still 'tis halloween\nwhen fairi…'
```

## 4) Contractions

Contractions are kind of an interesting thing to deal with; we often treat them as one entity but for NLP purposes we often want to separate them out into their two constituents. The `contractions` library contains a list of predefined contractions and their expansions. We will implement that here in the context of a `expand_contractions` function we will define.

```
contractions.contractions_dict # view dictionary of contractions
```

```
 we'd've :  we would have ,
'weren’t': 'were not',
'what’re': 'what are',
'what’d': 'what did',
'what’ve': 'what have',
'what’s': 'what is',
'what’ll': 'what will',
'what’ll’ve': 'what will have',
'when’ve': 'when have',
'when’s': 'when is',
'where’re': 'where are',
'where’d': 'where did',
'where’ve': 'where have',
'where’s': 'where is',
'which’s': 'which is',
'who’re': 'who are',
'who’ve': 'who have',
'who’s': 'who is',
'who’ll': 'who will',
'who’ll’ve': 'who will have',
'who’d': 'who would',
'who’d’ve': 'who would have',
'why’re': 'why are',
'why’d': 'why did',
'why’ve': 'why have',
'why’s': 'why is',
'will’ve': 'will have',
'won’t': 'will not',
'won’t’ve': 'will not have',
'would’ve': 'would have',
'wouldn’t': 'would not',
'wouldn’t’ve': 'would not have',
'y’all': 'you all',
'y’all’re': 'you all are',
'y’all’ve': 'you all have',
'y’all’d': 'you all would',
'y’all’d’ve': 'you all would have',
'you’re': 'you are',
'you’ve': 'you have',
'you’ll’ve': 'you shall have',
'you’ll': 'you will',
'you’d': 'you would',
'you’d’ve': 'you would have'}
```

```
text_1 = "I didn't even know it's a big deal."

# Add in comments
def expand_contractions(text):
    expanded_words = [] # create empty list
    for word in text.split(): # split text into individual words
        expanded_words.append(contractions.fix(word)) # identify contractions and replace wi
        expanded_text = ' '.join(expanded_words) # rejoin text
    return expanded_text

expand_contractions(text_1)
```

```
'I did not even know it is a big deal.'
```

```
expanded_text = expand_contractions(lower_text) # apply to lower_text
expanded_text
```

```
'start of the project gutenberg ebook helps and hints for halloween helps and hints for
halloween by laura rountree smith march brothers publishers wright ave. lebanon ohio co
pyright by march brothers contents page introduction party suggestions nutcrack night h
alloween stunts a shadow play the black cat stunt a pumpkin climbing game exercises hal
loween acrostic take care tables are turned drills clown drill and song autumn leaf dri
ll cattail drill muff drill dialogs and plays the halloween ghosts on halloween night j
ack frost's surprise an historical halloween the witch's dream a halloween carnival and
waxwork show the play of pomona halloween puppet play note send for our complete catalo
g in which will be found all the accessories needed in carrying out the ideas given in
this book. march brothers publishers wright ave. lebanon ohio introduction hist be stil
l it is halloween when fairies troop across the green on halloween when elves and witch
es are abroad we find it the custom over a…'
```

## 5) Removing Stopwords

Next, we'll define a function to filter out stop words based on a stopwords list from `nltk`. This process involves firs tokenizing the text, removing extra whitespace, removing tokens in the stopword list, and then finally rejoining all the remaining words back into a continuous string of text.

**Removal of stopwords isn't required, but it is common. Why do you think this is the case?**

They are not content words, it will only distract from the meaningful words in the text.

## Let's add some comments to see what we're doing here...

```
nltk.download('stopwords')
tokenizer = ToktokTokenizer()
stopword_list = set(stopwords.words('english'))

def remove_stopwords(text):
    tokens = [token.strip().lower() for token in tokenizer.tokenize(text)] # tokenize words,
    filtered_tokens = [token for token in tokens if token not in stopword_list] # fill in
    return ' '.join(filtered_tokens) # finish statement
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
stopword_list
```

Show hidden output

```
stopped_text = remove_stopwords(expanded_text) # apply to expanded_text
stopped_text
```

```
'start project gutenberg ebook helps hints halloween helps hints halloween laura rountr
ee smith march brothers publishers wright ave. lebanon ohio copyright march brothers co
ntents page introduction party suggestions nutcrack night halloween stunts shadow play
black cat stunt pumpkin climbing game exercises halloween acrostic take care tables tur
ned drills clown drill song autumn leaf drill cattail drill muff drill dialogs plays ha
lloween ghosts halloween night jack frost ' surprise historical halloween witch ' dream
halloween carnival waxwork show play pomona halloween puppet play note send complete ca
talog found accessories needed carrying ideas given book. march brothers publishers wri
ght ave. lebanon ohio introduction hist still halloween fairies troop across green hall
oween elves witches abroad find custom world build bonfires keep evil spirits night nig
hts entertain friends stunts similar performed two hundred years ago. night fortunes to
ld games played happens birthday falls nigh...'
```

## 6) Lemmatization

Lemmatization is another processing step that isn't required, but often implementd. Remember that lemmatization is different from stemming in that it attempts to reduce words to their roots (or lemmas), where as stemming simply cuts off suffixes and affixes.

Here we will implement a pretrained lemmatizer from `Spacy`.

**Why might we be interested in applying lemmatization?**

To reduce the amount of tokens and reduce the complexity of the text. This makes analyzing the text easier because there are less tokens.

```
nlp = spacy.load("en_core_web_sm")
lemmatizer = nlp.get_pipe("lemmatizer") # bring in spacy lemmatizer

def lemmatize_text(text):
  text = nlp(text)
  text = ' '.join([word.lemma_ if word.lemma_ != '-PRON-' else word.text for word in text])
  return text

lemmas = lemmatize_text(stopped_text) # apply to stopped_text
lemmas
```

```
'start project gutenberg ebook help hint halloween help hint halloween laura rountree s
mith march brothers publisher wright ave . lebanon ohio copyright march brother content
page introduction party suggestion nutcrack night halloween stunt shadow play black cat
stunt pumpkin climb game exercise halloween acrostic take care table turn drill clown d
rill song autumn leaf drill cattail drill muff drill dialog play halloween ghost hallow
een night jack frost ' surprise historical halloween witch ' dream halloween carnival w
axwork show play pomona halloween puppet play note send complete catalog find accessory
need carry idea give book . march brothers publisher wright ave . lebanon ohio introduc
tion hist still halloween fairy troop across green halloween elve witch abroad find cus
tom world build bonfire keep evil spirit night night entertain friend stunt similar per
form two hundred year ago . night fortunes tell game play happen birthday fall night ma
y even able hold converse fairiesso go an…'
```

## 7) Sentence Tokenize Text

Though we've applied word tokenization at other steps in the NLP pipeline and then rejoined our text, we are now ready to tokenize the text into sentences, so that we can put it into a structured format like a dataframe or list.

We will use the `PunktSentenceTokenizer` from `nltk` to perform this step:

```
punkt_st = nltk.tokenize.PunktSentenceTokenizer()

sents = punkt_st.tokenize(lemmas) # apply to lemmas
sents[3:15] # view some sentences
```

```
['lebanon ohio introduction hist still halloween fairy troop across green halloween
elve witch abroad find custom world build bonfire keep evil spirit night night
entertain friend stunt similar perform two hundred year ago .',
 'night fortunes tell game play happen birthday fall night may even able hold converse
fairiesso go ancient superstition careful halloween whenever come careful halloween
witch halloween origin old druid festival .',
 'druid keep fire burn year honor sungod .',
 'last night october meet altar fire burn put much pomp ceremony relighte they .',
 'take ember new fire return home kindle fire hearth .',
 'superstition home one fire burn constantly throughout year protect evil .',
 'later fire keep evil spirit away .',
 'country still witch fairy ghost agree night october st great time celebration .',
 'little book find useful school church home planning celebration halloween .',
```

```
 'air full magic let we write invitation hearty halloween night nutcrack party .',
 'party suggestion nutcrack night northern part england halloween still call nutcrack
night .',
 "nutcrack night party write invitation pumpkinshape booklet cut double face jacko '
lantern paint outside inside write nutcrack night meet fate please come eight late
mystery see know ' til halloween ."]
```

## 8) Deciding clean text output

Finally, we need to decide how to structure our cleaned text. This is going to depend on what we want to do with it next (which we'll cover in Topic 4). For now, let's store our sentence tokens in a dataframe, and then we'll store our vocab in a list.

### Output is a dataframe of sentences:

```
df = pd.DataFrame(sents, columns = ['Sentence'])
df
```

| | Sentence |
|---|---|
| **0** | start project gutenberg ebook help hint hallow... |
| **1** | lebanon ohio copyright march brother content p... |
| **2** | march brothers publisher wright ave . |
| **3** | lebanon ohio introduction hist still halloween... |
| **4** | night fortunes tell game play happen birthday ... |
| **...** | ... |
| **731** | punch judy punch judy merry time year often se... |
| **732** | call appear . |
| **733** | direction make puppet manipulation find puppet... |
| **734** | cent . |
| **735** | order publisher book . |

736 rows × 1 columns

## Output is a list of unique words:

```
words = nltk.wordpunct_tokenize(stopped_text)
text = nltk.Text(words)
```

```
vocab = sorted(set(text))
len(vocab)
```

```
1897
```

## Basic NLP Pipeline

We can also take a more basic approach and throw everything into one function, which can be helpful for less complicated texts.

```
url = "https://gutenberg.org/files/68667/68667-h/68667-h.htm"

html = request.urlopen(url).read()


raw = BeautifulSoup(html).get_text()
print(raw)
```

volunteer support.

Project Gutenberg™ eBooks are often created from several printed editions, all of which are confirmed as not protected by copyright in the U.S. unless a copyright notice is included. Thus, we do not necessarily keep eBooks in compliance with any particular paper edition.

Most people start at our website which has the main PG search facility: www.gutenberg.org.

This website includes information about Project Gutenberg™, including how to make donations to the Project Gutenberg Literary Archive Foundation, how to help produce our new eBooks, and how to subscribe to our email newsletter to hear about new eBooks.

```
print("[", raw.find("A LOVERS' PROLOGUE"), ":", raw.rfind("CHAPTER III"), "]")
```

```
[ 1435 : 363426 ]
```

```
raw = raw[1435:363426]
print(raw)
```

```
“Get up and pray now,” she said. “We have forgotten God in our deep
content—forgotten, in our bodies’ loves, the blows and anguish which
His flesh suffered to redeem them.”


He rose, unquestioning, and knelt by the bedside. He prayed that she
might not know, that his suspicions might be unfounded, that the
burden of that knowledge might never be hers—not that he might find
strength to ask her if it were. He prayed and prayed, until the
chillness of the night air seized his frail body with a very ague of
shivering. Then she, kneeling beside him, was smitten with remorse,
and blamed her thoughtlessness, and got him into bed again with all
speed, and watched beside him till he was once more warm and restful.
Then, his comfort was so great, her beauty so pitiful, he held out
rapturous arms to her, and wooed her to his heart. Shrinking,
reluctant, she surrendered passively. Had he not wounded his soul to
save hers? How could she deny him the fruits of that wild sacrifice.
She was a murderer’s wife.


There was even a thrill of ecstasy in the delirium of that thought—a
spark of new life struck out of a dead delusion. He could answer to a
provocation, after all—for her!


But later, when he had fallen into a deep sleep, she rose softly from
beside him, and crept to her oratory, and, kneeling on the icy stones
before the statue of the Holy Virgin, broke into prayer, and a passion
of tears,—


“O, Mother! show me how to love, and yet be clean!”
```

```
nltk.download('punkt')
def basic_text_cleaner(text):
    # Remove characters that are not letters, whitespaces, or periods
    text = re.sub(r'[^a-zA-Z\s\.]', '', text)
    # Tokenize and perform stopword removal, and casefolding
    tokens = word_tokenize(text)
    tokens = [token.lower() for token in tokens if token.lower() not in stopword_list]

    # Join tokens and trim extra whitespace
    cleaned_text = ' '.join(tokens).strip()

    return cleaned_text
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
cleaned_text = basic_text_cleaner(raw)
cleaned_text
```

```
'lovers prologue matter eternal dressing imagination world unconscious selfdelusion spi
rit . everything springs love love dreaming god . two figments endless sweet obsession
stood alonehigh slope alp time . born dream flesh thought owed flesha sacred debt . tru
th seemed plain pebbles brook lie round firm apparent shaking ripples . actual eyes whi
te mountains hoary glaciers pine woods foamy freshets eighteenth century le prieur . ac
tual ears whisper deathless confidence ever ever helps loves succession . loved therefo
re lived . man ten thousand ages pains prove love delusion still greets baby kitten nes
ting song birds hawthorn bush flower freshly latest expression newest product wisdom .
love delusion save shadows builds habitation . dust thou art said older god unto dust r
```