Building a Wikipedia Text Corpus for Natural Language Processing - KDnuggets

Wikipedia is a rich source of well-organized textual data, and a vast collection of knowledge. What we will do here is build a corpus from the set of English Wikipedia articles, which is freely and conveniently available online.

One of the first things required for natural language processing (NLP) tasks is a corpus. In linguistics and NLP, **corpus** (literally Latin for body) refers to a collection of texts. Such collections may be formed of a single language of texts, or can span multiple languages -- there are numerous reasons for which multilingual corpora (the plural of corpus) may be useful. Corpora may also consist of themed texts (historical, Biblical, etc.). Corpora are generally solely used for statistical linguistic analysis and hypothesis testing.

The good thing is that the internet is filled with text, and in many cases this text is collected and well oganized, even if it requires some finessing into a more usable, precisely-defined format. Wikipedia, in particular, is a rich source of well-organized textual data. It's also a vast collection of knowledge, and the unhampered mind can dream up all sorts of uses for just such a body of text.

What we will do here is build a corpus from the set of English Wikipedia articles, which is freely and conveniently available online.

Install gensim

In order to easily build a text corpus void of the Wikipedia article markup, we will use gensim, a topic modeling library for Python. Specifically, the `gensim.corpora.wikicorpus.WikiCorpus` class is made just for this task:

> Construct a corpus from a Wikipedia (or other MediaWiki-based) database dump.

In order to properly progress through the following steps, you will need to have gensim installed. It's a simple enough process; using pip:

```
$ pip install gensim
```

Moving on...

Download the Wikipedia Dump File

A Wikipedia dump file is also required for this procedure, quite obviously. The latest such files can be found here.

A warning: the latest such English Wikipedia database dump file is ~14 GB in size, so downloading, storing, and processing said file is not exactly trivial.

The file I aquired and used for this task was `enwiki-latest-pages-articles.xml.bz2`. Go ahead and download it or another similar file to use in the next steps.

Make the Corpus

I wrote a simple Python script (with inspiration from here) to build the corpus by stripping all Wikipedia markup from the articles, using gensim. You can read up on the WikiCorpus class (mentioned above) here.

The code is pretty straightforward: the Wikipedia dump file is opened and read article by article using the `get_texts()` method of the `WikiCorpus` class, all of which are ultimately written to a single text file. Both the Wikipedia dump file and the resulting corpus file must be specified on the command line.

```
"""

Creates a corpus from Wikipedia dump file.

Inspired by:

https://github.com/panyang/Wikipedia_Word2vec/blob/master/

"""


import sys

from gensim.corpora import WikiCorpus


def make_corpus(in_f, out_f):


"""Convert Wikipedia xml dump file to text corpus"""


output = open(out_f, 'w')

wiki = WikiCorpus(in_f)


i = 0

for text in wiki.get_texts():

output.write(bytes(' '.join(text), 'utf-8').decode('utf-8') + '\n')

i = i + 1

if (i % 10000 == 0):

print('Processed ' + str(i) + ' articles')

output.close()
```

```
    print('Processing complete!')


if __name__ == '__main__':

if len(sys.argv) != 3:

print('Usage: python make_wiki_corpus.py <wikipedia_dump_file
<processed_text_file>')

sys.exit(1)

in_f = sys.argv[1]

out_f = sys.argv[2]

make_corpus(in_f, out_f)
```

```
$ python make_wiki_corpus enwiki-latest-pages-
articles.xml.bz2 wiki_en.txt


Processed 10000 articles
Processed 20000 articles
Processed 30000 articles
Processed 40000 articles
Processed 50000 articles
Processed 60000 articles
Processed 70000 articles
Processed 80000 articles
Processed 90000 articles
Processed 100000 articles
...
```

After several hours, the above code leaves me with a corpus file named `wiki_en.txt`.

Check the Corpus

A second script then checks the corpus text file we just built.

Now, keep in mind that this large Wikipedia dump file then resulted in a very large corpus file. Given its enormous size, you may have dificulty reading the full file into memory at one time.

This script, then, starts by reading 50 lines -- which equates to 50 full articles -- from the text file and outputting them to the terminal, after which you can press a key to output another 50, or type 'STOP' to quit. If you *do* stop, the script then proceeds to load the entire file into memory. Which could be a problem for you. You can, however, verify the text by batches of lines, in order to satisfy your curiousity that something good happened as a result of running the first script.

If you are planning on working on such a large text file, you may need some workarounds for its large size in comparison to your machine's memory.

```
"""
Checks a corpus created from a Wikipedia dump file.
"""

import sys, time

def check_corpus(input_file):

"""Reads some lines of corpus from text file"""

while(1):

for lines in range(50):

print(input_file.readline())

user_input = input('>>> Type \'STOP\' to quit or hit Enter key for more <<< ')

if user_input == 'STOP':

break
```

```
def load_corpus(input_file):

"""Loads corpus from text file"""

print('Loading corpus...')

time1 = time.time()

corpus = input_file.read()

time2 = time.time()

total_time = time2-time1

print('It took %0.3f seconds to load corpus' %total_time)

return corpus


if __name__ == '__main__':

if len(sys.argv) != 2:

print('Usage: python check_wiki_corpus.py <corpus_file>')

sys.exit(1)


corpus_file = open(sys.argv[1],'r')

check_corpus(corpus_file)

corpus = load_corpus(corpus_file)
```

The corpus file must be specified at the command line to execute.

```
$ python check_wiki_corpus.py wiki_en.txt


...
best loved patriotic songs harperresource external
links mp and realaudio
recordings available at the united states library of
```

```
congress words sheet
music midi file at the cyber hymnal america the
beautiful park in colorado
springs named for katharine lee bates words archival
collection of america
the beautiful lantern slides from the another free
sheet music

>>> Type 'STOP' to quit or hit Enter key for more <<<
```

And that's it. Some simple code to accomplish what gensim makes a simple task. Now that you are armed with an ample corpus, the natural language processing world is your oyster. Time for something fun.

Related: