

---

## lazynlp

DOI [10.5281/zenodo.2582057](https://doi.org/10.5281/zenodo.2582057) license [MIT](#)

A straightforward library that allows you to crawl, clean up, and deduplicate webpages to create massive monolingual datasets. Using this library, you should be able to create datasets larger than the one used by OpenAI for GPT-2.

### Setup

This library uses Python 3.

1. Clone this library and cd into the lazynlp folder:

```
1 git clone https://github.com/chiphuyen/lazynlp.git
2 cd lazynlp
```

2. Install dependencies

```
pip3 install -r requirements.txt
```

3. Install the library `pip3 install .`

If you want to uninstall the library, use:

```
pip3 uninstall lazynlp
```

### How to create a massive dataset using lazynlp:

#### Step 1. Obtain URLs of the webpages you want to crawl

There are several major dumps of URLs available that you can use.

**Reddit URLs** This is the link to all submissions to Reddit by months. You can download the raw dump and process to get the links. Keep in mind that each of these dumps is huge (100MB - 1GB).

@jcpeterson is kind enough to provide a list of deduplicated links with at least 3 karma that you can download [here](#).

There are about 23M URLs from between 2015-06 to 2018-10, of which around 40 - 60 % are bad URLs (URLs no longer exist or aren't scraper-friendly). It means that after you've downloaded and cleaned all good URLs from this, you should have approx 10M webpages or 50GB of pure text.

---

**Gutenberg** You can download the list of all URLs to US Gutenberg books here. There are 50K books, which convert to about 14GB of pure text.

You can also run `lazynlp.get_us_gutenberg_links()` to get the same list. For example, if you want to get all the Gutenberg URLs and store it in the file `us_gutenberg.urls`, run the following command. This might take half a day.

```
lazynlp.get_us_gutenberg_links('us_gutenberg.urls')
```

You can download the list of all URLs to Australian Gutenberg books here. There are 4k books, which convert to about 1GB of pure text.

You can also run `lazynlp.get_aus_gutenberg_links()` to get the same list. For example, if you want to get all the Gutenberg URLs and store it in the file `aus_gutenberg.urls`:

```
lazynlp.get_aus_gutenberg_links('aus_gutenberg.urls')
```

**Wikipedia** You can download the Wikipedia dumps here.

## Step 2. Deduplicate URLs

You don't want to download the same URL multiple times. There are two functions that help you deduplicate all URLs:

```
lazynlp.dedup_lines(files, outfold)
```

This function takes in a list of files (in each file, each line is a URLs) and deduplicate each file against all previous files. Save all the deduplicated files in outfold.

```
lazynlp.dedup_lines_from_new_file(original_files, new_file, outfile)
```

This function allows you to deduplicate a new file against all previously deduplicated files (original\_files)

## Step 3. Download the URLs

If you want to download each webpage separately, call:

```
lazynlp.download_page(link, context=None, timeout=None)
```

If you want to download from a file that contains a list of URLs, call:

```
lazynlp.download_pages(link_file, folder, timeout=30, default_skip=True, extensions=[], domains=[])
```

---

```
1  """
2
3  link_file:
4
5      file contains links to webpages to crawl. Each line contains one
        URL.
6
7  folder:
8
9      folder that you want to contain your downloaded pages.
10
11  timeout:
12
13      seconds to wait for a page to respond before abandoning it.
14
15  default_skip:
16
17      set to True if you want to automatically skip all URLs that contain
        domains and extensions that are known to be scraper-unfriendly
        or NSFW.
18
19      You can see the list of excluded domains at lazynlp/exclude_domains
        .txt.
20
21      You can see the list of excluded extensions at lazynlp/
        exclude_extensions.txt
22
23  You can also add your own domains and extensions to skip with domains
        and extensions and arguments.
24
25  In the folder:
26
27      Each URL is downloaded into a file, indexed by the order in which
        it is downloaded. The first line of each file is the URL. The
        rest is the textual content of the page.
28
29      index.urls contains all the URLs that have been successfully
        downloaded.
30
31      bad.urls contains the URLs that are bad.
32
33      connection.urls contains the URLs that haven't been downloaded
        because of connection issues.
34
35      non_ascii.urls contains the URLs that haven't been downloaded
        because of bad encoding issues.
36
37      empty.urls contains the URLs that have empty textual content.
38
39  """
```

---

If you have a lot of URLs, you can divide the list into multiple files and call this function separately. I was able to run 40 scripts in parallel. I guess I could have parallelized the code. I just found this to be easier.

#### Step 4. Clean the webpages

You can get rid of all HTML tags, decode utf-8 into string, transliterate foreign characters, collapse white space, replace unprintable characters, unescape HTML, etc. using methods available in `lazynlp/cleaner.py`.

You can also just call the following function to do most of the processing.

```
lazynlp.clean_page(page)
```

**Note:** In this library, the function `lazynlp.download_pages()` does both the crawling and cleaning part, so the webpages you have are pure text, like this:

```
1 http://www.thecannabist.co/2017/03/02/jeff-sessions-russia-resign-
  democrats/74687/
2 Attorney general nominee Sen. Jeff Sessions, R-Ala., testifies on
  Capitol Hill in Washington on Jan. 10, 2017, in the first day of his
  confirmation hearing before the Senate Judiciary Committee. Top
  Democrats now say that because he misled the committee about his
  visits to Russia, he should resign. (Andrew Harnik, The Associated
  Press)
3
4 House Oversight and Government Reform Committee Chairman Jason Chaffetz
  , R-Utah, tweeted early Thursday that "AG Sessions should clarify
  his testimony and recuse himself."
5
6 Later, Sen. Rob Portman, R-Ohio, said in a statement, "Jeff Sessions is
  a former colleague and a friend, but I think it would be best for
  him and for the country to recuse himself from the DOJ Russia probe.
  "
7
8 House Majority Leader Kevin McCarthy, R-Calif., also initially said
  during an appearance on MSNBC's "Morning Joe" that Sessions should
  bow out.
9
10 Asked whether Sessions should recuse himself in this situation,
  McCarthy replied "I think the trust of the American people -- you
  recuse yourself in these situations, yes."
11
12 McCarthy was pressed a second time about whether he was calling for
  Sessions to recuse himself and he confirmed that he believed the
  situation required a recusal.
13
```

---

```
14 "I think it would be easier from that standpoint, yes," McCarthy said.
15
16 But McCarthy later said his comment had been misinterpreted, telling
    Fox News' "Fox and Friends," "I'm not calling on him to recuse
    himself. I was asked on 'Morning Joe,' if he needs to recuse himself
    as going forward. As you just heard, Attorney General Sessions said
    he would recuse himself going forward -- appropriate, and that's
    all my answer was."
17
18 The comments from prominent Republicans follow revelations that
    Sessions met with the Russian ambassador during election season.
    Under oath in front of the Senate Judiciary Committee for his
    confirmation hearing in January, Sessions had said that he had not
    met with any Russian officials.
19
20 Senate Minority Leader Charles Schumer, D-N.Y., joined growing
    Democratic calls for Sessions to either resign or at least recuse
    himself from any investigations into Russia's meddling in U.S.
    elections.
21
22 "Attorney General Sessions cannot possibly lead an investigation into
    Russian interference in our elections or come anywhere near it. With
    these revelations, he may indeed become the subject of it," Schumer
    told reporters. "Better for the country if he resigns, but let's
    get an investigation going."
23
24 Because the Department of Justice should be above reproach, for the
    good of the country, the Attorney General should resign.
```

## Step 5. Remove duplicated webpages

To avoid any piece of texts being over-represented, you want to only include pages that don't significantly overlap with other pages.

To estimate the amount of overlapping of target files with certain source files, use this function:

```
lazynlp.estimate_overlap(source_files, target_files, gran='word', n
=8, capacity=10000, error_rate=1e-5, header=0, interval=100000)
```

`gran` is the granularity of tokens: 'char' or 'word' level.

`n` is the n-gram.

`capacity` and `error_rate` are for the BloomFilter used.

`header`: number of lines of each file to skip. It's because in our format, the first line is the url

To estimate the amount of overlapping of a target file with an existing BloomFilter, use this function:

---

```
lazynlp.estimate_overlap_bf(bf, target_file, gran='word', n=8, header=0)
```

If given a list of files, e.g. cleaned webpages, to filter out all the files that contain more than `threshold` overlapping with other files, use this function:

```
lazynlp.filter_files(files, threshold=0.5, gran='word', n=8, capacity=100000000, error_rate=1e-7, header=0, interval=1000000)
```

Names of all the files that are deemed duplicated are stored in `dupped_files.list`

Names of all the files used for the dataset are stored in `clean_files.list`

### Some notes:

1. 1GB of text is about 1b characters. An English word has on average 4.5 characters, or 5.5 including whitespace. So 1GB of text is about 181M words.
2. When I ran 30 scripts in parallel, it took 3 hours to download and clean 1GB of pure text. So it'd take 5 days to get 50GB of pure text.
3. The OpenAI dataset has 40GB, which I estimate to contain about 7-8 billion words. If you download all the webpages from the good Reddit URLs and Gutenberg books, you should have a dataset bigger than OpenAI's WebText.
4. OpenAI, in their paper for GPT-2, didn't include Wikipedia articles for fear of overlapping. You can choose to include Wikipedia articles that have less than a certain amount of overlapping with the existing dataset using `lazynlp.estimate_overlap_bf(bf, target_file, gran='word', n=8)`.