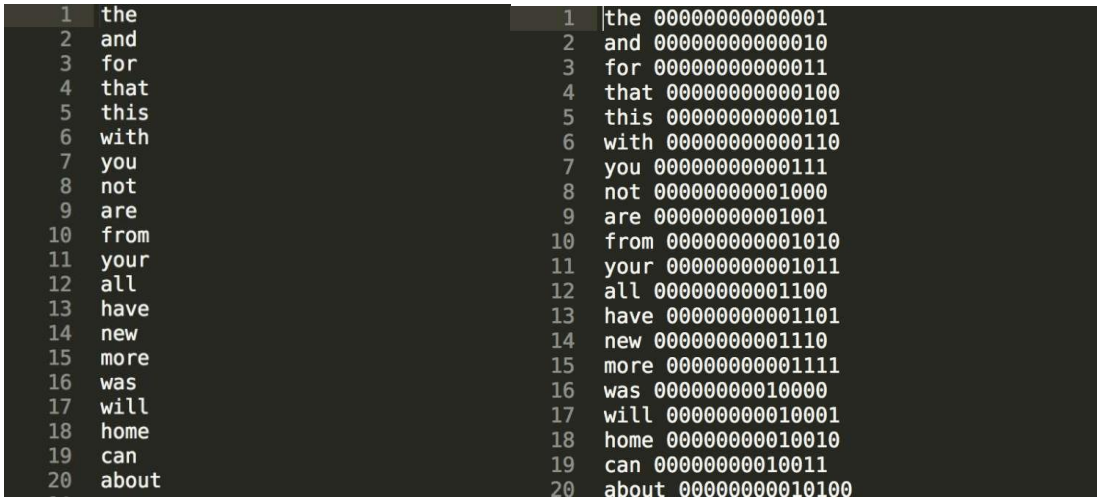
A New Algorithm for Compressing English Text

1 Introduction

Nowadays, people have a lot of demand for compression of data. Because it will enable data to be stored and transferred more easily. And when people need to see or use the data, they can decompress it back to the original format, without losing any information. A lot of data is written in English text, so we decided to implement a compression algorithm to efficiently compression English text.

2 Algorithm and Result

Our algorithm is based on a dictionary of most-used 10000 English words. We encode each word with 14 bits by setting up a bijection between those words and 0-10000's binary form.



```
1  the
2  and
3  for
4  that
5  this
6  with
7  you
8  not
9  are
10 from
11 your
12 all
13 have
14 new
15 more
16 was
17 will
18 home
19 can
20 about
```

```
1  the 00000000000001
2  and 00000000000010
3  for 00000000000011
4  that 00000000000100
5  this 00000000000101
6  with 00000000000110
7  you 00000000000111
8  not 00000000001000
9  are 00000000001001
10 from 00000000001010
11 your 00000000001011
12 all 00000000001100
13 have 00000000001101
14 new 00000000001110
15 more 00000000001111
16 was 00000000010000
17 will 00000000010001
18 home 00000000010010
19 can 00000000010011
20 about 00000000010100
```

Original dictionary and the dictionary after setting up bijection

After the dictionary is set up, for each word we just encode it by finding the corresponding 14-bits binary number.

In order to make the encoding faster, we also build a binary tree when setting up the dictionary, so when we look up the corresponding binary number, we don't have to do linear search in the dictionary, but do a binary search with the tree. This method improved the searching time from O(n) to O(logn).

Next we generate a random text based on the dictionary. That being said all words can be found in the dictionary. We will use this random-generated file as the plaintext to encode.

```
1  proc dressing mailman. filed industry, specifications gray correction demand hitachi sensor bdsm catherine counseling
   breasts household constructed vote cas dover bestsellers bizarre humanity vegas telecharger. slow discrete reform
   grants authentic cotton insulation. rush botswana decimal cir. songs porsche families venue infant bennett relevant
   november, vip builder questionnaire surfaces shirt facing asbestos. basis limousines mayor hole briefly brush lookup
   kirk. prairie switzerland factors citations conducting necessity single, seats off, cottages union.
2  rogers dot reid responsibility neighborhood palestine depression nodes graduated. medicine.
3  ensuring purple issues, bowling therapeutic cruises treatment, tanks nor cast tapes phones.
4  asset cats winds tomorrow kinds counties task travelers hydraulic offshore tactics neighbor supplies, celebs nyc ted
   selective columns quoted disney postcards pumps cameron wallpaper monitor pdt surrounding awarded book, blade own, mac
   burke ratings.
5  query variety pointer fin egyptian native politicians ultimate spend tasks author, measure assign selecting ons
   illustration variable bargain danny come, saddam produced screenshots mississippi call, finances botswana clark conduct
   options, lately propecia till drawings factors novelty. trackbacks intel dialogue man, avi email, searching efficient
   asp existed. nerve caps cars, conservative balanced curtis vector relax steve tickets, yield rendered dat. assignment
   lit rob prominent cds now, team, cloth indoor notebook informative neil mobiles backup wellness concentrations entered
   administered officers directors reforms accepts plains norfolk monkey electronics, organizing districts kay beach,
   play, nurse summaries. associates community, wall divide skins harold edges occurring became wishing. epson inter paris
   icons uganda pirates horses smooth anytime speaking credits advertising, specifics mode.
6  allows feelings profile, battery bloomberg driver gui specially luxembourg aids anniversary eva adopted internship.
   cornwall lover lip welsh prepaid hurt return, architects suspended emails loved delhi widescreen lady cookbook
   experiments rest resorts stores, accountability retro wound minds comm housing printed str joins affiliated territory
   calling cathedral promoted. relations oak expect involved feed planned columnists nuke butler smallest. paint contrast
   hunter peoples est maintains crash haven posters alcohol screens binary indigenous homeless pack although, disclosure
   gratuit interpreted. weight, unavailable japanese valid contemporary oval decent editions advertiser wine root walter
   travel, matches earl linear prague shock abstracts hundreds des sparc comes.
7  electron addressing chicken lightweight iceland pda demanding. mug. multi component conflict minimize vista numerous
   plate scholars emotions retrieval huntington. segments saver quotations lips msie themes targeted indicate lounge
   entrepreneurs. dogs coordinated. enlargement dust respected. emphasis mhz join, pump june, johnson indians silver.
```

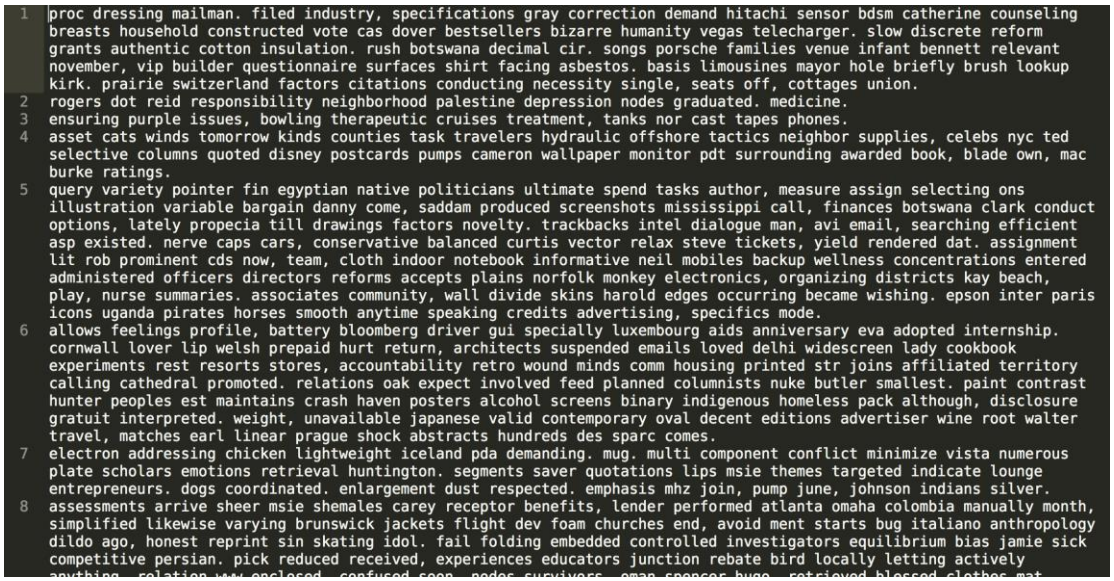
Randomly Generated File

Next we read through the file, file corresponding 14-bits binary number from the binary tree, and write to the encoded file directly as binary file, in order the minimize the encoded text size.

```
1  5dd6 1999 1725 6b26 381a a95b 0961 3445
2  2ea1 b1a0 2e48 cdb3 35ee d16d 3aa4 a935
3  2a05 2988 b22a 9500 db67 74d0 6c89 5065
4  6b2b 0e24 331c cbe8 5a7c c2f9 54e5 6b45
5  4585 9880 641e 95ac 62e7 bbc6 7247 092f
6  477a 06b9 0682 56c7 9b91 f97d d59d 81db
7  1288 926a 56b1 40d0 5543 ecb0 47aa d700
8  504a 4dd9 5ad9 5a32 a873 276c 5b73 8b48
9  2099 5b0f 7102 e256 c861 440d 95ac 0005
10 a70a e088 5057 a2f5 e09e 4269 51c8 d7e5
11 6b12 b656 b000 1666 4144 1ce9 5b17 0d64
12 40f2 80ce e56c 683c 78d2 3717 8411 8e56
13 b000 0ebd 41c5 3673 5188 a438 8878 969a
14 e215 5e36 06d8 9883 cb95 b206 24e6 d72e
15 8ac0 f1a5 cdca 927f cd80 36ca 0d9b 1bcc
16 c9f4 1950 a502 2e56 c52e 00f9 95b0 53b8
17 b3c4 b095 ac00 025b c620 531a 1c27 94c8
18 7b77 dc9d f2b5 4d97 05c2 56c2 4bdd f559
19 11b0 168a 8896 3715 c1f0 7065 6c6a c47a
20 85ca cb8f 051a 56c7 c618 5934 789c 807f
21 256c 8c86 2043 fa57 191c ca46 a95a ddf8
22 28a9 47f0 5aa5 6c6c 7405 595b 0bb3 3158
23 fa29 2a25 6b7d 6933 e0a4 656c 44ad 7b98
24 2a90 4c65 4c73 90bd 656c 4402 1699 2b65
25 6b45 29df 93df 9fdb 2000 0349 5b01 8c95
26 b176 d486 0c44 8039 4ce7 1c4b 9260 ad94
27 3300 5ce4 28b8 a5c8 01df 1971 515e a5c4
28 42bb 95b1 eda4 b51d 9f0a da56 c073 256c
29 3ea2 4169 5ac6 3303 4656 c144 1f2a 7399
30 f656 f060 6f20 0e55 095a cfef 4328 6384
31 935b 407e 0102 239d 1010 2fe4 91b0 ce25
32 6c75 2846 295a c000 16d1 5000 4fa5 6c17
33 ba01 0191 dd88 845d 7562 398e 607c 70c5
```

Encoded Binary File

To decode, very similar to encoding, we read the binary file, and divide it to sections, so that each section contains 14 bits. And we read each section, find the corresponding English word from the binary tree, and write it back to a decoded text file.



Decoded Text File

Note that the decoded text is exactly the same as the original plain text. So the whole encode and decode process will not lose any information.

```
[shun@ResNet-5-58:~/cs/Compression$ python compress.py
finished generating text!
finished modifying!
finished setting up dictionary!
Finished encoding!!!
average word length: 5.80
expected compression ratio: 0.257
finished compressing random_generated_file(797KB) to super_compressed_file(210K
B)!
compression ratio: 0.263
```

The Whole Program Output

3 Conclusion and Analysis

We have implemented an algorithm to encode and compress English text file. For the example we see that the program successfully compresses a 797KB text file to 210KB binary file, and decoded back without losing any data. The compression ratio is 0.263, which is about 1/4. Compared to Hoffman Encoding, which compress the data to about 2/3, our algorithm is much more efficient!

4 Further Improvement

We can potentially improve the program in these aspects:

1) we can use a more comprehensive dictionary, which will enable to program to deal with a larger range of English words.

2) we can add upper and lower case support.

3) we can make the program support more symbols, instead of only comma and period.