GTU Department of Computer Engineering
CSE 484 / 654 - Fall 2022
Homework #2 Report

Hasan Mutlu
1801042673

1.) Seperating Words into Syllables

In order to seperate the words into syllables, this Python package is used. Also, every space and dots are tokenized by the names “spc” and “nkt” as well. After that, the file with seperated syllables is divided into two files containing %95 and %5 of all the data.

2.) N-Gram Tables

In the N-Gram tables, the number of occurances are stored. N-Gram probabilities aren’t stored in the tables because of the risk of floating point underflow.

Unigram table is constructed using two arrays, one storing the syllable and the other storing the counts of the syllable at the same row. Here are some of the mostly encountered tokens:

- ka 1392354
- gi 1435298
- ki 1438432
- ti 1450907
- ni 1604898
- ra 1618126
- ta 1621717
- ma 1696196
- ve 1805055
- 1877272
- di 1913364
- i 1973229
- ya 1986513
- a 2155891
- li 2296008
- de 2453887
- da 2574334
- si 2784701
- le 2802227
- ri 2813571
- la 3092544
- nkt 3622494
- spc 48866659

Also, the syllable array of the unigram table is used as a reference to all unique syllables in the document. 2218 unique syllables are found in total of 180,439,320 syllables in the %95 part of the file.

In order to store bigram and trigram tables more efficiently, Python dictionary is used and tuples are used as keys. The tuples were the indices of syllables. For example,

bigramArr.readValue((indexDict[“la”], indexDict[“ri”])

is used to find the bigram value of “la-ri”.

The syllable strings aren’t directly used as keys at it extended the hashing process. Only the table entries with value greater than zero are stored in this way. When the desired syllable couple tuple’s key isn’t found in the dictionary, it means that its value is 0.

Here are how bigram and trigram tables are kept:

```
(1996, 1618): 1792
(1618, 2217): 4724
(2217, 1934): 21032
(1934, 2217): 26658
(2217, 1600): 9818
(1600, 2217): 4287
(2217, 361): 493
```

```
(1996, 1618, 2217): 1621
(1618, 2217, 1934): 713
(2217, 1934, 2217): 6536
(1934, 2217, 1600): 1
(2217, 1600, 2217): 1506
(1600, 2217, 361): 14
(2217, 361, 2217): 356
```

3.) Smoothing

GT Smoothing is used in unigram and bigram tables. The number of new syllables/couples are counted and saved for using in perplexity measurements.

It took too long to calculate the new syllable triplets in the trigram table so Laplace Smoothing have to be used for trigram tables.

4.) Perplexity

Perplexities of all the sentences in the remaining %5 of the file are calculated. The output file is big in size so not all of them are included in the submission. Here are some examples of perplexity calculations:

- Sentence: letonya sovyet sosyalist cumhuriyeti bayragi letonya tarafindan ocak tarihinde kullanilmaya baslanmistir.
- Perplexity Unigram: 119.69174299467919
- Perplexity Bigram: 17.815296564776578
- Perplexity Trigram: 5.065506683932393

- Sentence: kabushiki gaisha japonya merkezli bir video oyunu ve muzik sirketidir.
- Perplexity Unigram: 107.58449032629392
- Perplexity Bigram: 29.38180722459873
- Perplexity Trigram: 9.365064279496034

- Sentence: gose bira gose kokeni almanyanin goslar kentine dayanan minimum oraninda day malti iceren eksimsi ve tuzlu bir bira turudur.
- Perplexity Unigram: 89.52808017588973
- Perplexity Bigram: 42.69362804549076
- Perplexity Trigram: 20.24032279391091

- Sentence: ada sili tarafindan yilinda ele gecirilmistir.
- Perplexity Unigram: 61.479724848866475
- Perplexity Bigram: 10.174449288946775
- Perplexity Trigram: 6.227219613853876

- Sentence: bu genellikle yeni gelenlerin kendilerini ispatlamasina tahsis edilirdi.
- Perplexity Unigram: 97.99128289319388
- Perplexity Bigram: 17.311155128110812
- Perplexity Trigram: 7.3880705294239135

5.) Random Sentences

Unigram Sentences: In unigram sentences, the sentence starts with one of the most encountered 5 syllables. After that, next most encountered syllable is added to queue. Space is also used as a probability in every measurement. After having at least 4 spaces in the sentence, the sentence ending dot is included as a probability for next token selection and when dot is encountered, the sentence is ended.

As expected, the words in the generated sentences didn't make any sense. Here are some examples:

- sileli rideya aovemaladaniitiditakisabirneraegibu kanintedir naci.
- risidele dayalidiveoma rani atiki ta.
- ladale ria yalidiomairadevetigi takanisibir kitenedir sabuninreena yese.
- sidaleadeilaove yadilinimatagika kiti rabir terinenin buesame.
- ledadelalisia o iridi ranitima yakasanebir kie tevedirna tareginin me se yeubucemisgelanrin micibiluyi.

Bigram Sentences: In bigram sentences, the first syllable is chosen randomly out of all the syllables. After that, 5 random syllables are added to the candidate queue. Then, all the table entries starting from the last syllable in the sentence are traversed and the ones with the highest values are added to the candidate queue. Finally, one of them is chosen randomly to be added to the sentence.

Space is always used as a probable next token candidate because otherwise, the words were getting too long and losing catched manings. Similarly to unigram sentences, although dot always being a probability from bigram table, the terminating dot is included in probabilities and the sentence is ended when that dot is encountered.

Here are some examples:

- width ozel i o.
- rizmdir.rum ileri olan veren o.
- ve olandakiyenilan arafinlan i i olanma.
- pnin arasin a oyun i a.
- sehza i veyayinla veren.
- tepsinedenge ikisina veri verilmesi a ise veyapi.
- pullamahal alamalama a iki ola ikiyeniversininde veren.
- nancasine veren verenme veren ileme o.
- lirli arada ve ve a ala.
- id adigiliz i ozelliklemeri arafindan ozellinindeki.

Trigram Sentences: In trigram sentences, the first syllable is chosen randomly. The second syllable is chosen from the bigram table as the most probable next syllable. Then, next syllables are chosen with the same procedure as bigram sentence generation.

Here are some examples:

- nevresimleriy ice adinin yasa daha icin olus.
- yis da ameli bir maci.
- boylelikle bir kara km. km.
- jay yikildi icindirmekle gele iseanin.
- va veyadelerine baglanmisti.platannpopnab a.
- queenstelaslanmaya sahiptirkoydensedecek o kmye baslar.
- kerry king veyahutmehrkirchra yikildikca i icinde. ayrilmaya kazan.
- sonunaturabildik ame ve sanatcisinin ya.
- bap veyada bulunur ice olasilinindekileri.
- bbc te verenginebis yilinacakti ta.

Output Files

In order to keep this report clean, 5-10 examples are given from outputs. All outputs are included in the output_files directory.

There are perplexity calculation results of first 1000 sentences in the remaining %5 of file in perplexity.txt and 100 examples in each N-Gram sentence generations.