1 Collecting a corpus

Selma.txt was collected and "wc -w Selma.txt" unix command gave output: 965943. I.e. There are 965943 words in the corpus.

The concordance program gave example output below when run with command "python concord.py Selma.txt Nils 10".

> *—Lines containing Nils is listed below—*
>
> *Nils Holgersso*
> *os Holger Nilssons LV. A*
> *tt! Se på Nils gåsapåg!*
> *tt! Se på Nils Holgersso*
> *ll Holger Nilsson i Väst*
> *Jag heter Nils Holgersso*
> *det värt, Nils Holgersso*
> *att du är Nils gåsapåg,*
> *r han var Nils gåsapåg,*
> *r och hur Nils gåsapåg h*
> *...*

After tokenizer program was run it was sorted with unix command "sort selmaCorpus.txt — uniq". The list was now containing 38663 unique words.

2 Normalizing a corpus

Our program used the tokenizer program to create a list of all words. Then ¡s¿ was inserted on every first uppercase letter in sentence. ¡/s¿ was inserted where there where a . ! or ? in text. It was evaluated using regex101.com. This gave the last 5 sentences to be:

> *< s > hon hade fått större kärlek av sina föräldrar än någon annan han visste och sådan kärlek måste vändas i välsignelse < /s >*
> *< s > när prästen sa detta kom alla människor att se bort mot klara gulla och de förundrade sig över vad de såg< /s >*
> *< s > prästens ord tycktes redan ha gått i uppfyllelse< /s >*
> *< s > där stod klara fina gulleborg ifrån skrolycka hon som var uppkallad efter själva solen vid sina föräldrars grav och lyste som en förklarad< /s >*
> *< s > hon var likaså vacker som den söndagen då hon gick till kyrkan i den röda klänningen om inte vackrare< /s >*

3 Counting unigrams and bigrams

Output from count.py:

There are 39356 unigrams in training set.
Most common words:
och 37799
att 28914
han 22743
det 22087
i 17072
som 16790
hade 14955
på 14634
hon 14093

Output from count_bigrams.py:

There are 319877 bigrams in training set.
Some bigrams:
3 ('nils', 'holgerssons')
1 ('holgerssons', 'underbara')
1 ('underbara', 'resa')
4 ('resa', 'genom')
3 ('genom', 'sverige')
1 ('sverige', 'selma')
11 ('selma', 'lagerlöf')
2 ('lagerlöf', 'innehåll')

As there are 965 943 words in the corpus the maximum possible number of bigrams is 965 943 * 2 - 2 = 1 931 884. Their real value for selma corpus is, as printed above, 319 877. This gives a difference of (1 931 884 - 319 877 =) 1 612 007 "missing" bigrams. This is because of there is many bigrams that are occuring more than once. For example "selma lagerlöf" is occuring 11 times, as seen printed above. To cope with unseen bigrams in the corpus when calculating probability one can either use backoff where we take probability of the first word instead of the whole bigram or use a very small probability for the whole bigram.

4 Computing the likelihood of a sentence

Below are five made up sentences:

```
Unigram model
======================================================
wi C(wi) #words P(wi)
======================================================
['haddwad', 0, 1086836, 0]
['jaawd', 0, 1086836, 0]
['awddacc', 0, 1086836, 0]
['awger', 0, 1086836, 0]
['</s>', 63698, 1086836, 0.05860865852805759]
======================================================
Prob. unigrams:  0.0
Geometric mean prob.:  0.0
Entropy rate:  0
Perplexity:  1
======================================================
======================================================
Bigram model
======================================================
wi wi+1 Ci,i+1 C(i) P(wi+1|wi)
======================================================
['<s>', 'HaDDwad', 0, 63698, 1e-25]
['HaDDwad', 'jaawd', 0, 0, 1e-25]
['jaawd', 'awDDacc', 0, 0, 1e-25]
['awDDacc', 'awger', 0, 0, 1e-25]
['awger', '</s>', 0, 0, 0.05860865852805759]
======================================================
Prob. unigrams:  5.860865852805761e-102
Geometric mean prob.:  5.67012138399e-21
Entropy rate:  67.25711037242209
Perplexity:  1.763630674333816e+20
```

Figure 1: ”HaDDwad jaawd awDDacc awger”

```
Unigram model
======================================================
wi C(wi) #words P(wi)
======================================================
['hur', 2100, 1086836, 0.0019322142439153654]
['går', 655, 1086836, 0.0006026668236974116]
['bilen', 0, 1086836, 0]
['</s>', 63698, 1086836, 0.05860865852805759]
======================================================
Prob. unigrams:  0.0
Geometric mean prob.:  0.0
Entropy rate:  0
Perplexity:  1
======================================================
======================================================
Bigram model
======================================================
wi wi+1 Ci,i+1 C(i) P(wi+1|wi)
======================================================
['<s>', 'Hur', 0, 63698, 1e-25]
['Hur', 'går', 0, 0, 0.0006026668236974116]
['går', 'bilen', 0, 2969, 1e-25]
['bilen', '</s>', 0, 3, 0.05860865852805759]
======================================================
Prob. unigrams:  3.532149407627068e-55
Geometric mean prob.:  2.43786564049e-14
Entropy rate:  45.221374712366035
Perplexity:  41019487841774.53
```

Figure 2: ”Hur går bilen”

```
Unigram model
======================================================
wi C(wi) #words P(wi)
======================================================
['hon', 14093, 1086836, 0.012966999780713925]
['ryckte', 119, 1086836, 0.00010949214048853736]
['till', 9445, 1086836, 0.008690363587514583]
['</s>', 63698, 1086836, 0.05860865852805759]
======================================================
Prob. unigrams:  7.231395429779202e-10
Geometric mean prob.:  0.00518567777864
Entropy rate:  7.591251720121058
Perplexity:  192.83882313680908
======================================================
======================================================
Bigram model
======================================================
wi wi+1 Ci,i+1 C(i) P(wi+1|wi)
======================================================
['<s>', 'Hon', 0, 63698, 1e-25]
['Hon', 'ryckte', 0, 0, 0.00010949214048853736]
['ryckte', 'till', 36, 188, 0.19148936170212766]
['till', '</s>', 262, 12110, 0.021635012386457472]
======================================================
Prob. unigrams:  4.5361222002560064e-32
Geometric mean prob.:  1.45938939631e-08
Entropy rate:  26.030059882575564
Perplexity:  68521807.99213421
```

Figure 3: ”Hon ryckte till”

```
Unigram model
======================================================
wi C(wi) #words P(wi)
======================================================
['idag', 4, 1086836, 3.680408083648315e-06]
['är', 6378, 1086836, 0.005868410689377239]
['det', 22087, 1086836, 0.020322293335885082]
['måndag', 1, 1086836, 9.201020209120787e-07]
['</s>', 63698, 1086836, 0.05860865852805759]
======================================================
Prob. unigrams:  2.3669384074492462e-17
Geometric mean prob.:  0.000472974134731
Entropy rate:  11.0459510896766
Perplexity:  2114.2805210048605
======================================================
======================================================
Bigram model
======================================================
wi wi+1 Ci,i+1 C(i) P(wi+1|wi)
======================================================
['<s>', 'Idag', 0, 63698, 1e-25]
['Idag', 'är', 0, 0, 0.005868410689377239]
['är', 'det', 584, 29703, 0.019661313672019662]
['det', 'måndag', 0, 26764, 9.201020209120787e-07]
['måndag', '</s>', 0, 7, 0.05860865852805759]
======================================================
Prob. unigrams:  6.2220113221013024e-37
Geometric mean prob.:  5.73833844795e-08
Entropy rate:  24.054791698141408
Perplexity:  17426647.261584166
```

Figure 4: ”Idag, är: det$ måndag”

```
Unigram model
=======================================================
wi C(wi) #words P(wi)
=======================================================
['det', 22087, 1086836, 0.020322293335885082]
['var', 12852, 1086836, 0.011825151172762036]
['en', 13921, 1086836, 0.012808740233117047]
['gång', 1332, 1086836, 0.0012255758918548888]
['en', 13921, 1086836, 0.012808740233117047]
['katt', 15, 1086836, 1.3801530313681181e-05]
['som', 16790, 1086836, 0.015448512931113802]
['hette', 107, 1086836, 9.845091623759242e-05]
['nils', 84, 1086836, 7.728856975661462e-05]
['</s>', 63698, 1086836, 0.05860865852805759]
=======================================================
Prob. unigrams:  4.594552317604714e-27
Geometric mean prob.:  0.00232393683732
Entropy rate:  8.749213426641543
Perplexity:  430.3042939632865
=======================================================
=======================================================
Bigram model
=======================================================
wi wi+1 Ci,i+1 C(i) P(wi+1|wi)
=======================================================
['<s>', 'det', 5754, 63698, 0.09033250651511822]
['det', 'var', 4023, 26764, 0.15031385443132567]
['var', 'en', 753, 22780, 0.03305531167690957]
['en', 'gång', 695, 94886, 0.0073245789684463465]
['gång', 'en', 23, 2194, 0.010483135824977211]
['en', 'katt', 5, 94886, 5.269481272263558e-05]
['katt', 'som', 2, 488, 0.004098360655737705]
['som', 'hette', 50, 18172, 0.00275148580233326]
['hette', 'nils', 0, 109, 7.728856975661462e-05]
['nils', '</s>', 2, 121, 0.01652892561983471]
=======================================================
Prob. unigrams:  2.6161546826867327e-23
Geometric mean prob.:  0.00551780491694
Entropy rate:  7.5016898338305875
Perplexity:  181.23148879898764
```

Figure 5: ”det var en gång en katt som hette nils”

5 Reading

I wanted to test how it deals with old English that might not exist in the corpus and found one sentence in an old English poem from a unknown author. *”Hearken to me, gentlemen, Come and you shall heare; He tell you of two of the boldest brethren, That ever born y-were.”.*

First of only *heare* was not in corpus and *”hearken”*, *”boldest”* and *”brethren”* only occurred a couple of times. Norvig’s program replaced *”heare”* with here but the right replacement should probably be hear. Segmenting meaning turned out not to be possible with Norvig’s program and result was *”’H’, ’e’, ’a’, ’r’, ’k’, ’e’, ’n’, ’t’, ’o’, ’m’, ’e’, ’g’, ’e’, ’n’, ’t’, ’l’, ’e’, ’m’, ’e’, ’n’, ’C’, ’o’, ’m’, ’e’, ’a’, ’n’, ’d’, ’y’, ’o’, ’u’, ’s’, ’h’, ’a’, ’l’, ’l’, ’h’, ’e’, ’a’, ’r’, ’e’, ’H’, ’e’, ’t’, ’e’, ’l’, ’l’, ’y’, ’o’, ’u’, ’o’, ’f’, ’t’, ’w’, ’o’, ’o’, ’f’, ’t’, ’h’, ’e’, ’b’, ’o’, ’l’, ’d’, ’e’, ’s’, ’t’, ’b’, ’r’, ’e’, ’t’, ’h’, ’r’, ’e’, ’n’, ’T’, ’hat’, ’ever’, ’born’, ’y’, ’were’”*

The sentence probability for the original sentence was 1.26e-80. This could be explained with the words not being very common and non existing word was appearing.