EDAN20 - Assignment 2
Language models

Hugo Mattsson
hu5174ma-s

September 2024

1 Objectives and dataset

1.1 Objectives

The objectives of the assignment were to:

- Write a program to find n-gram statistics
- Compute the probability of a sentence
- Experiment with word completion and prediction, and sentence segmentation

1.2 Dataset

The dataset was a long concatenated text consisting of some of Selma Lagerlöf's works.

2 Method and program structure

The program was written inside a jupyter notebook structured into the following parts

2.1 Segmenting and tokenizing the corpus

To make the later processing stages easier, it begins with cleaning up the corpus by replacing any non-letter with a space using `r'[^\p{L}.;:?!]'`. The text is then segmented by marking the beginning and end of sentences with the tags <s> and </s>. The regex used for this was a substitution where `r'\p{P}\p{Z}+(\p{Lu})'` was replaced with r' </s>\n<s> \1'. Finally the resulting segmented string could be tokenized by considering spaces or linefeed characters as item separators.

2.2 Counting n-grams

The frequency of unigrams(words) and bigrams(word pairs) was computed from the corpus to later be used for probabilistic purposes.

2.3 Sentence probabilities

The probability of the sentence *Det var en gång en katt som hette Nils* was computed. First using a unigram model where the total probability was the product of the relative frequencies of the words, the using a bigram model where the probability took into account what the previous word was and how frequent this word pair was in the corpus. The results of these models applied to the test sentence can be seen in tables in the section *Results*.

2.4 Word prediction

Finally, using a bigram or trigram model, the program tries to predict what word is currently being typed or which would be most probable next word in the sentence.

3 Results

3.1 Unigram and bigram models

3.1.1 Det var en gång en katt som hette Nils

wi	**Ci**	**#words**	**P(wi)**
det	21108	1041559	0.02026577467046994
var	12090	1041559	0.01160759976151135
en	13514	1041559	0.012974781073371744
gång	1332	1041559	0.0012788521821615482
en	13514	1041559	0.012974781073371744
katt	16	1041559	1.536158777371229e-05
som	16288	1041559	0.01563809635363911
hette	97	1041559	9.312962587813076e-05
nils	87	1041559	8.352863351956059e-05
</s>	59047	1041559	0.05669097957964935
Prob. unigrams:			5.365167044398377e-27
Geo. mean prob:			0.0023602517310623073
Entropy rate:			8.726843547198847
Perplexity:			423.68362104745404

Table 1: Unigram Model Results

wi	wi+1	Ci,i+1	C(i)	P(wi+1—wi)
<s>	det	5672	59047	0.09605907158704083
det	var	3839	21108	0.1818741709304529
var	en	712	12090	0.058891645988420185
en	gång	706	13514	0.052242119283705785
gång	en	20	1332	0.015015015015015015
en	katt	6	13514	0.0004439840165754033
katt	som	2	16	0.125
som	hette	45	16288	0.002762770137524558
hette	nils	0	97	0.0 **backoff:* 8.352855332386037e-05
nils	</s>	2	87	0.022988505747126436
Prob. bigrams:				2.376169768780815e-19
Geo. mean prob:				0.013727382866049192
Entropy rate:				6.186799588766881
Perplexity:				72.84709764111099

Table 2: Bigram Model Results

3.1.2 För länge sedan bodde alla i grottor

wi	Ci	#words	P(wi)
för	9443	1041559	0.009066217084197822
länge	555	1041559	0.0005328550759006451
sedan	1092	1041559	0.001048428365555864
bodde	188	1041559	0.0001804986563411194
alla	2355	1041559	0.002261033700443278
i	16508	1041559	0.015849318185527657
grottor	4	1041559	3.840396943428073e-06
</s>	59047	1041559	0.05669097957964935
Prob. unigrams:			7.132727838017238e-24
Geo. mean prob:			0.0012783711370895558
Entropy rate:			9.611477543911716
Perplexity:			782.245445776163

Table 3: Unigram Model Results

wi	wi+1	Ci,i+1	C(i)	P(wi+1—wi)
<s>	för	293	59047	0.004962148796721256
för	länge	27	9443	0.002859260828126654 7
länge	sedan	21	555	0.03783783783783784
sedan	bodde	3	1092	0.0027472527472527475
bodde	alla	0	188	0.0 **backoff:* 0.002261031529628634
alla	i	20	2355	0.008492569002123142
i	grottor	1	16508	6.05766900896535e-05
grottor	</s>	1	4	0.25
Prob. bigrams:				4.288838933228556e-19
Geo. mean prob:				0.005058741260182629
Entropy rate:				7.627005833261002
Perplexity:				197.67763334151985

Table 4: Bigram Model Results

3.1.3 När man går på fest bör man klä sig fint

wi	Ci	#words	P(wi)
när	2772	1041559	0.0026613950817956544
man	2322	1041559	0.002229350425659996
går	633	1041559	0.0006077428162974925
på	14250	1041559	0.01368141411096251
fest	18	1041559	1.728178624542633e-05
bör	38	1041559	3.648377096256669e-05
man	2322	1041559	0.002229350425659996
klä	6	1041559	5.760595415142109e-06
sig	9250	1041559	0.008880917931677418
fint	54	1041559	5.184535873627898e-05
</s>	59047	1041559	0.05669097957964935
Prob. unigrams:			1.0426877649571743e-35
Geo. mean prob:			0.00066043823340696
Entropy rate:			10.564288737871903
Perplexity:			1514.1461372418814

Table 5: Unigram Model Results

wi	wi+1	Ci,i+1	C(i)	P(wi+1—wi)
`<s>`	när	867	59047	0.014683218453096687
när	man	62	2772	0.022366522366522368
man	går	8	2322	0.0034453057708871662
går	på	9	633	0.014218009478672985
på	fest	0	14250	0.0 **backoff:* 1.728176965321249e-05
fest	bör	0	18	0.0 **backoff:* 3.64837359345597e-05
bör	man	1	38	0.02631578947368421
man	klä	0	2322	0.0 **backoff:* 5.760589884404163e-06
klä	sig	1	6	0.16666666666666666
sig	fint	1	9250	0.00010810810810810811
fint	`</s>`	7	54	0.12962962962962962
Prob. bigrams:				3.59143878571538e-30
Geo. mean prob:				0.002104777654600298
Entropy rate:				8.892116447355255
Perplexity:				475.1095669484872

Table 6: Bigram Model Results

3.2 Next word prediction

Input	Prediction
De	det, de, den, detta, denna
det var en	stor, liten, gammal, god, sådan
det var en g	gammal, god, gång, ganska, glädje

Table 7: Next word prediction results

4 Conclusion

This was a good assignment to learn more about n-grams and their usages.

5 Answer to possible questions

5.1 N-grams, seen and unseen

In the used corpus, there existed, with mixed case, 44 256 unique words which means that the amount of possible bigrams that could exist are $44\,256^2 = 1\,958\,593\,536$(almost two billion). However this corpus only contains 320 122 bigrams, which is only 0.016% of what's possible. The simple reason for this is that many words aren't used in conjunction and might not even make sense when doing so. One example is the expressions *strong tea* or *powerful computer*, which is something you would say. But *powerful tea* or *strong computer* is at best very rare to say. Already in this simple example a few bigrams could be identified that most likely would be unseen in normal text.

The amount of possible 4-grams is ridiculously large: roughly $44\,256^4 \approx 3.836 \cdot 10^{18}$.

So how does a model deal with bigrams that it has never seen before? Here's three examples of techniques that could be used: Backoff, Laplace and Good-Turing, each with their own pros and cons. The one used in this assignment is backoff, were the probability for an unseen bigram gets replaced with the relative frequency of the last word.

5.2 Norvig

The sentence i chose to test Norvig's segmenting functions was:

> Sometimes to understand the meaning of a word you need more than a definition; you need to see the word used in a sentence.

Which became this

> sometimes to understand the meaning ofawordyouneed more than a definition you need to seethewordusedina sentence

using the unigram model, and this

> sometimes to understand the meaning of a word you need more than a definition you need to see the word used in a sentence

when using the bigram segmenter.

As can be seen the bigram model used clearly had more relevant information available which meant that it could correctly segment the test sentence, while the unigram model didn't do quite as well