Obedkova Maria, HW1 - Exploring Entropy and Language Modeling

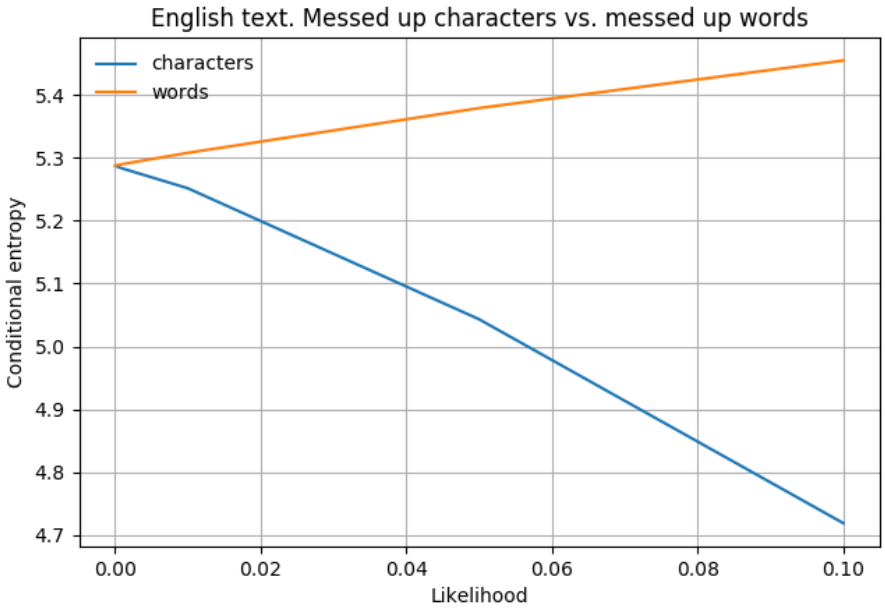
1. Entropy of a Text

	English text	Czech text
Conditional entropy	5.287	4.748
Perplexity	39.055	26.868

	English text	Czech text
Word count	221 099	222 413
Unique word count	3 812	26 316
Word form count	9 608	42 827
Bigram count	73 246	147 136
Unique bigram count	49 600	125 007

	English text			Czech text		
Likelihood	Min	Max	Average	Min	Max	Average
0.1	4.635	4.640	4.637	5.454	5.461	5.458
0.05	4.697	4.700	4.698	5.377	5.382	5.379
0.01	4.738	4.741	4.740	5.306	5.308	5.307
0.001	4.747	4.748	4.747	5.289	5.290	5.289
0.0001	4.747	4.748	4.748	5.287	5.288	5.288

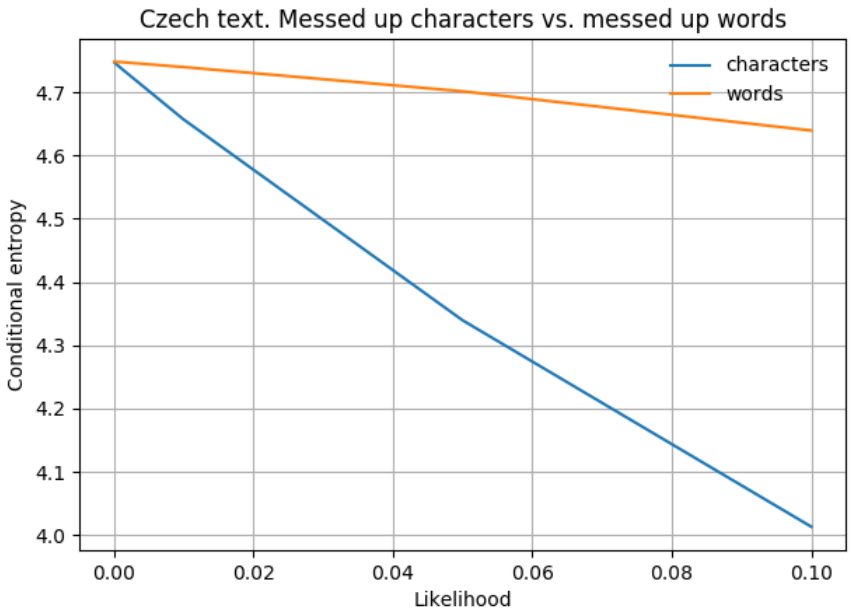
English text



We can see that the more we mess up **words**, the higher conditional entropy becomes. English text has little number of distinct word forms in comparison to overall number of words which can be explained by the presence of many function words in the language such as articles, auxiliary verbs, particles and so on. While messing up, words are replaced by words from a dictionary where every word from the text is present only once. That is why, more functional words are replaced which results in decreasing number of bigrams in overall. The less bigrams we have, the worse we can predict the next word in a text.

It can be seen that the more we mess up **characters**, the lower probability becomes. This happens due to the fact that, while messing up characters, we create new words which are usually unique and, which is more important for counting conditional entropy, more unique bigrams. The more unique bigrams we have (bigrams with frequency 1), the better we can predict the following word in a text.

Czech text

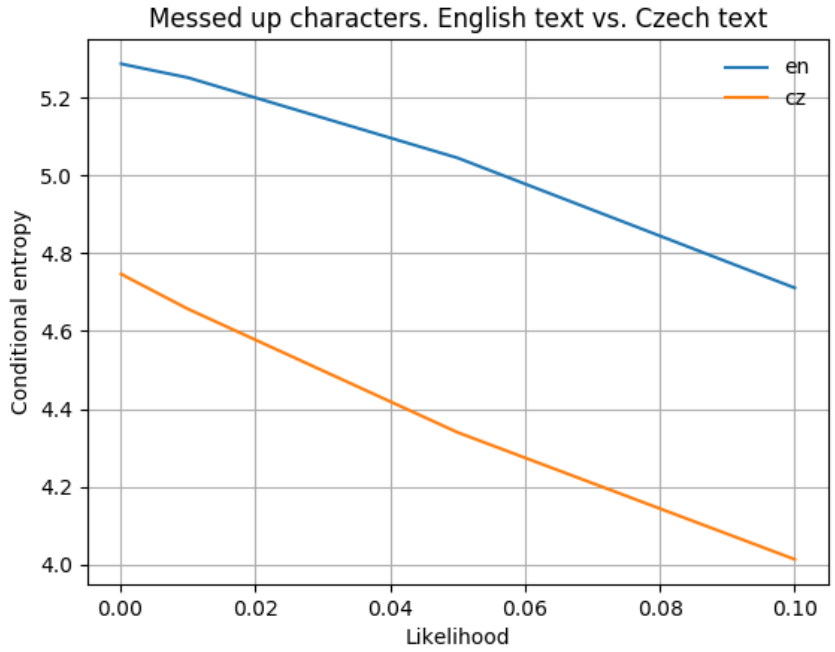


We can see that the more we mess up **words**, the lower conditional entropy becomes. Czech has rich morphology that guarantees to be many word forms in a text. When we mess up words might create more new bigrams and, which is more important, more unique bigrams. Thus, we have lower conditional entropy when we have more unique bigrams.

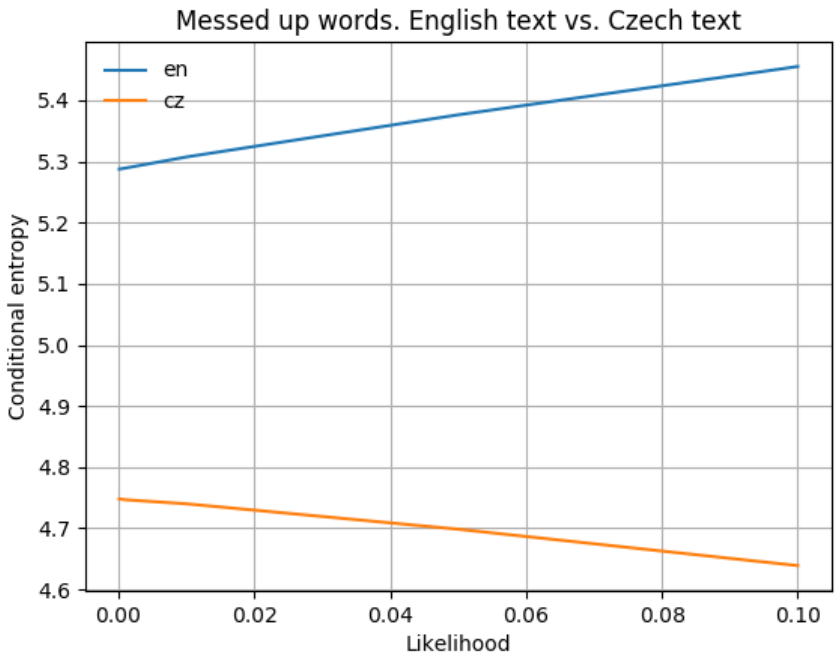
The reason for conditional entropy to decrease when we mess up **characters** is the same as for English language.

Steeper decrease for characters can be explained by the fact that messing up characters more likely creates a new word -> a new bigram which cannot be said about messing up words.

English text vs. Czech text comparison



The mechanism for messing up **characters** for both languages are the same: we create more unique bigrams while messing up characters which makes conditional entropy to decrease. The difference is only in the conditional entropy for a language in general which depends entirely on the language type (that will be discussed further).



In comparison to English text, when we mess up **words** in Czech text, we get the decreasing conditional entropy with the growth of the likelihood. Czech language is a fusional language with many different inflection types while English has many identical words (especially it concerns function words). That explains why we get conditional entropy lower for Czech text than for English text: the more unique words -> unique bigrams we have, the easier we can predict the following word.

1.2. Paper-and-pencil exercise

The entropy of a text T = T_1 + T_2 depends entirely on the value of

$$P(T_1[-1], T_2[1]) * log_2(P(T_2[1] \mid T_1[-1]),$$

where $T_1[-1]$ – the last word of text T_1 and $T_2[1]$ – is the first word of text T_2. This is due to the fact that all conditional probabilities of two texts are the same in a new text and all joint probabilities are changed proportionally to the size of the new text which does not influence conditional entropy. The only question is what is happening on the junction of two texts.

If we do not consider this case, the entropy of the new text will be E.

$$P(T_1[-1], T_2[1]) = 1 / (T_1 \text{ size} + T_2 \text{ size})$$

and

$$P(T_2[1] \mid T_1[-1]) = 1 / T_1[-1] \text{ frequency}$$

So the equation above has the value which is really small to influence the value of conditional entropy. Thus, the conditional entropy will be somewhere near E.

2. Cross-Entropy and Language Modelling

	English text	Czech text
Cross-entropy	7.4667	10.2219
“Coverage” graph	0.9556	0.8646

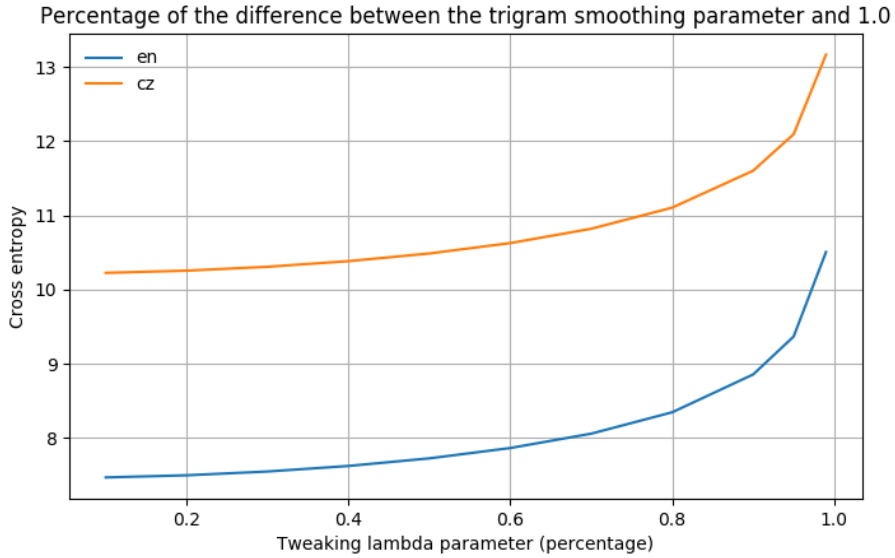
Cross-entropy for Czech text is much higher than cross-entropy for English text. That can be explained by the “coverage” graph which is smaller for Czech text than for English text. Every time when we do not have defined probability from the training data we replace it by uniform probability (quite small value) which makes cross-entropy higher. In overall, $p`_\lambda$ probability is smaller for Czech than for English due to many different word forms -> many different bigrams -> many different trigrams which makes cross-entropy higher.

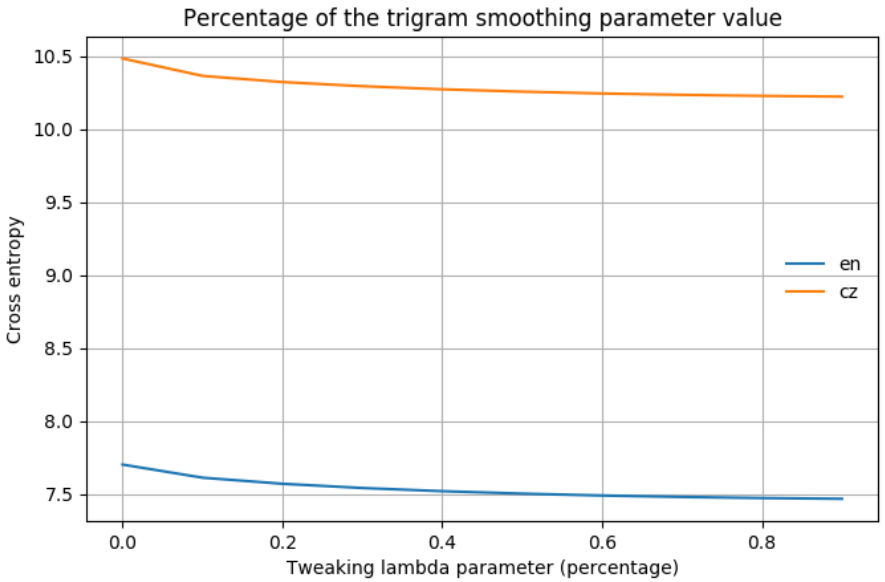
		λ_0	λ_1	λ_2	λ_3
English text	Heldout data	0.0727	0.2518	0.4821	0.1934
	Training data	0.0002	0.0026	0.0381	0.9590
Czech text	Heldout data	0.1495	0.4202	0.2464	0.1838
	Training data	0.0000	0.0021	0.0741	0.9238

As we can see, if we use the same data for extracting probabilities and computing lambdas for any of given languages, we get lambdas almost equivalent to the vector (0, 0, 0, 1). That can be explained that trigram distribution can the best describe the data because it was computed exactly on the same data whereas other distribution can describe data correctly but lack information that has trigram distribution. That is why, the weight of these trigram probabilities is prevailing in general proportion.

For English text, the biggest lambda is λ_2 and, for Czech text, it is λ_1. It seems that for Czech language bigram and trigram distributions are not quite reliable to predict the next word. Perhaps, it is due to many unique bigrams and trigrams in Czech which is, in turn, due to many word forms in the language. For English text, bigram distribution contributes the most. The most probable reason is that there are much less unique bigrams than trigrams for English text.

	English text		Czech text	
	Tweaking lambda parameter	Cross entropy	Tweaking lambda parameter	Cross entropy
Increasing λ_3	0.1	7.47119570318	0.1	10.2264558164
	0.2	7.4999495309	0.2	10.2560477356
	0.3	7.55101164972	0.3	10.308262679
	0.4	7.62565604462	0.4	10.3841258067
	0.5	7.72819949162	0.5	10.4877236019
	0.6	7.86742741399	0.6	10.627513063
	0.7	8.06092374726	0.7	10.820443834
	0.8	8.34883929637	0.8	11.105070693
	0.9	8.85806412646	0.9	11.6022934904
	0.95	9.36914723791	0.95	12.0942519668
	0.99	10.5071300723	0.99	13.1656556859
Decreasing λ_3	0.9	7.46984307388	0.9	10.2249776242
	0.8	7.47497883347	0.8	10.2299418079
	0.7	7.48233309649	0.7	10.2370046851
	0.6	7.49225205071	0.6	10.2465098879
	0.5	7.50522862563	0.5	10.258952701
	0.4	7.52200324025	0.4	10.2750909076
	0.3	7.54378166787	0.3	10.2961922434
	0.2	7.57280502905	0.2	10.3247090346
	0.1	7.61445043947	0.1	10.3669258748
	0	7.70429253092	0	10.4860946639





For both languages, when we change the value of λ_3 cross-entropy increases which is predictable because computation of lambdas gives us the most optimal value of λ_3 minimizing cross-entropy.

If we increase λ_3, cross-entropy increases much more than when we decrease λ_3. The probable explanation is that trigram distribution actually does not bear a lot of additional information for us and we can safely mostly rely on bigram and unigram probabilities.