1 Information Theory

1.1 Entropy and Probability Distributions

a) The marginal distribution of X =

$$\{(\frac{1}{8}+\frac{1}{16}+\frac{1}{16}+\frac{1}{4}),(\frac{1}{16}+\frac{1}{8}+\frac{1}{16}+0),(\frac{1}{32}+\frac{1}{32}+\frac{1}{16}+0),(\frac{1}{32}+\frac{1}{32}+\frac{1}{16}+0)\}$$

$$=\{\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8}\}$$

The marginal distribution of Y =

$$\{(\frac{1}{8}+\frac{1}{16}+\frac{1}{32}+\frac{1}{32}),(\frac{1}{16}+\frac{1}{8}+\frac{1}{32}+\frac{1}{32}),(\frac{1}{16}+\frac{1}{16}+\frac{1}{16}+\frac{1}{16}),(\frac{1}{4}+0+0+0)\}$$

$$=\{\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\}$$

b) The Entropy of X, H(X) =

$$=-\frac{1}{2}log_2\frac{1}{2}-\frac{1}{4}log_2\frac{1}{4}-\frac{1}{8}log_2\frac{1}{8}-\frac{1}{8}log_2\frac{1}{8}$$

$$=.5+.5+.375+.375$$

$$=1.75\,bits$$

The Entropy of Y, H(Y) =

$$=4*(-\frac{1}{4}log_2\frac{1}{4})$$

$$=4*.5$$

$$=2\,bits$$

c)

$$H(X|Y)=\sum_{i=1}^{4}p(Y=i)H(X|Y=i)$$

$$=\frac{1}{4}H(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8})+\frac{1}{4}H(\frac{1}{4},\frac{1}{2},\frac{1}{8},\frac{1}{8})+\frac{1}{4}H(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4})+\frac{1}{4}H(1,0,0,0)$$

$$=\frac{1}{4}*\frac{7}{4}+\frac{1}{4}*\frac{7}{4}+\frac{1}{4}*2+\frac{1}{4}*0$$

$$=1.375\,bits$$

$$H(Y|X)=H(X|Y)-H(X)+H(Y)$$

$$=1.375-1.75+2$$

$$= 1.625\, bits$$

$$H(X,Y) = H(Y|X) + H(X)$$
$$= 1.625 + 1.75$$
$$= 3.375\, bits$$

d) The mutual information I(X;Y)

$$I(X;Y) = H(X) - H(X|Y) \tag{1}$$
$$= 1.75 - 1.375$$
$$= 0.375\, bits$$

If we calculate I(X;Y) as follows:

$$I(X;Y) = H(Y) - H(Y|X) \tag{2}$$
$$= 2 - 1.625$$
$$= 0.375\, bits$$

So, (1) and (2) are generating the same value which authenticates the validity of the symmetry property of the mutual information.

2 N -gram Language Models

2.1 Language Model Training

a) see ngram_LM.py

b) With the unigrams it is impossible to tell the genre sice we have no words which give us any context. In the bigrams we have 'the world', but with world as the only word which could give us a context it is still impossible to tell the genre. See Table 1 and 2 down below.

c) The Type Token ratio for both n is 7.65% since the two extra tokens do not change the ratio in a meaningful way.

d) see ngram_LM.py

e) Our implementation fullfills the test for both n.

f) See Table 3 down below.

unigram	count
the	734066
of	360504
to	330708
and	326187
in	250687
a	228170
that	154963
is	14967
s	121474
for	104848
it	85982
as	80903
be	74177
with	73328
on	69736

Table 1: Unigrams for part b)

bigram	count
of the	82750
in the	65399
to the	32188
and the	27717
on the	20190
it is	19828
to be	18774
that the	15456
for the	17383
the us	16979
the world	16123
with the	15745
by the	14295
of a	13571
at the	13429

Table 2: Bigrams for part b)

word	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
blue	collar	eyes	and	throat	$< /s >$	bond	sky	water	ribbon	or
natural	resources	gas	resource	disasters	selection	and	environment	capital	disaster	to
green	energy	economy	revolution	growth	light	jobs	and	technologies	investment	$< /s >$
artificial	intelligence	islands	and	life	$< /s >$	hyper	sweeteners	smile	photosynthesis	island
white	house	and	$< /s >$	men	collar	man	women	americans	paper	people
global	economy	financial	warming	economic	growth	trade	governance	gdp	climate	imbalance
black	sea	and	market	$< /s >$	hole	swan	hair	eyes	holes	carbo
domestic	demand	and	political	consumption	politics	investment	policy	economic	market	savings

Table 3: solution for part f)

2.2 Language Model Smoothing and Evaluation

a) The smoothing technique was applied to probability distribution in order to prevent zero-valued probabilities caused by unseen n-grams in the training corpus. The smoothing method was implemented in *estimate_smoothed_prob* function.

b) A *test_smoothed_prob* function was written with a view to check whether the probability mass of the distribution sums up to 1 when Lidstone smoothing is applied with $\alpha=0.5$. The test demonstrates that the probability mass of the smoothed distribution sums up to 1 with a precision 10^{-7} in both unigram and bigram distributions.

c) We applied bigram LM to assessment of translation hypotheses. Method *score_sentence* calculates average value of probability logarithms of sentence bigrams. The value can be interpreted as average word surprisal and estimates fluency of the sentence. Translation hypothesis with the lowest score is supposed to be the most appropriate.

According to the score values we found out that hypothesis #2 "Yesterday I was at home" should be preferred as the most fluent translation for German sentence "Gestern war ich zu Hause.".

Such kind of estimator assess how frequent are the bigrams that occur in the sentence. This approach gives understanding whether the sentence is similar to natural speech, so it can be useful in such tasks as:

- Evaluation of word order like in the example given.
- Spelling correction, as the model can suppose what word should stay in the beginning of the phrase or in collocations, for example.
- Word prediction.

However, there are restrictions of both the approach in general and using bigram LM in particular:

- Firstly, bigram model does not give the full view of the language. It considers closely neighboring words only, but words at a greater distance might be highly correlating. That relation of distant tokens is not assessed and it may cause inaccuracies.
- Considering the machine translation task, the approach does not allow evaluating word choice. In other words, hypotheses including different variants of translation of the same word cannot be estimated regarding the context. As a result, the translation may contain the variant which is more

common in the training corpus, but unsuitable for the topic. (However, we should note, that n-gram models could be very useful in case of choosing from synonymic variants.)

- N-gram language model may be trained on the corpus majoring in one topic (for example, medicine), so cannot be accurate on the texts with another topic (geology), because different fields have n-gram distributions that differ to some extent.