T-61.5020 Statistical Natural Language Processing
Answers 2 — Entropy and perplexity
Version 1.0

1. a) Let's use the definition of the entropy,

$$H(X) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)},$$

with given values:

$$\begin{aligned} H(X) &= \frac{3}{32}\log_2\frac{32}{3} + \frac{3}{16}\log_2\frac{16}{3} + \frac{7}{32}\log_2\frac{32}{7} \\ &\quad + \frac{1}{8}\log_2 8 + \frac{1}{8}\log_2 8 + \frac{1}{4}\log_2 4 \\ &= 2.50 \text{ bits} \end{aligned}$$

b) For the solution we need the probability $P(S = s)$ (a random substantive is s). It can be obtained from the right margin probability of the given table. In addition we need the probability

$$P(V = v|S = s) = \frac{P(S = s, V = v)}{P(S = s)}.$$

The entropy of the source, as we know that the previous symbol was a substantive, is

$$H(X_i|X_{i-1} \in S) = \sum_{S=\{\text{'kissa'},\text{'tuuli'},\text{'kiipeilijä'}\}} p(s = S)H(V|s = S).$$

For this we need to calculate the conditional entropy $H(V|s = S)$. For the word 'kissa',

$$\begin{aligned} H(V|s = \text{'kissa'}) &= \sum_{V=\text{'naukaisi'},\text{'tuivertaa'},\text{'katosi'}} p(v = V|s = \text{'kissa'}) \log_2(p(v = V|s = \text{'kissa'})^{-1}) \\ &= \sum_{V=\text{'naukaisi'},\text{'tuivertaa'},\text{'katosi'}} \frac{p(s = \text{'kissa'}, v = V)}{P(s = \text{'kissa'})} \log_2 \frac{P(s = \text{'kissa'})}{p(s = \text{'kissa'}, v = V)} \\ &= \frac{1}{8}\frac{16}{3}\log_2(8\frac{3}{16}) + \frac{1}{16}\frac{16}{3}\log_2(16\frac{3}{16}) \\ &= \frac{2}{3}\log_2\frac{3}{2} + \frac{1}{3}\log_2 3. \end{aligned}$$

As we place the probabilities for each word in S, we get

$$\begin{aligned} H(X_i|X_{i-1} \in S) &= \frac{3}{16}(\frac{2}{3}\log_2\frac{3}{2} + \frac{1}{3}\log_2 3) + \frac{3}{8}(\frac{1}{6}\log_2 6 + \frac{4}{6}\log_2\frac{6}{4} + \frac{1}{6}\log_2 6) \\ &\quad + \frac{7}{16}(\frac{1}{7}\log_2 7 + \frac{6}{7}\log_2\frac{7}{6}) \\ &= 0.90 \text{ bits.} \end{aligned}$$

What is the probability for a random word to be "kissa"? As both categories S and V are of equal probability, the result is

$$P(x = \text{'kissa'}) = P(x \in S)P(S = x) = 0.5 \cdot \frac{3}{16} = \frac{3}{32}$$

We note that the distribution in (a) is actually a marginal distribution for the joint distribution in (b).

To conclude, when we know the behavior of the source more accurately, the produced words are less suprising and we can code them with less bits (0.9 bit $<$ 2.5 bit).

c) In the sentences of the described language, the first word is always a noun and the second word is always a verb. The noun does not depend on the previous words, and the verb depends only on the previous noun.

Let's denote the probabilities of the language as $P(S, V)$, and the probabilities given by the model as $P_M(S, V)$. We want to calculate the expected coding length of a sentence when it is coded with the model:

$$E(-\log P_M(S, V)) = -\sum_{s\in S, v\in V} P(S = s, V = v) \log P_M(S = s, V = v).$$

This measure is called *cross-entropy*.

For the model, the noun and the verb of a sentence are independent, so $P_M(S = s, V = v) = P_M(S = s)P_M(V = v)$. By using that and writing the sum open first for the nouns and then for the verbs we get:

$$\begin{aligned}
& E(-\log P_M(S, V)) \\
= & -\sum_{s\in S, v\in V} P(S = s, V = v) \log P_M(S = s, V = v) \\
= & -\sum_{s\in S}\sum_{v\in V} P(S = s)P(V = v|S = s) \log(P_M(s)P_M(v)) \\
= & -P(S = \text{kissa}) \sum_{v\in V} P(V = v|S = \text{kissa}) \log(P_M(\text{kissa})P_M(v)) \\
& -P(S = \text{tuuli}) \sum_{v\in V} P(V = v|S = \text{tuuli}) \log(P_M(\text{tuuli})P_M(v)) \\
& -P(S = \text{kiipelijä}) \sum_{v\in V} P(V = v|S = \text{kiipelijä}) \log(P_M(\text{kiipelijä})P_M(v)) \\
= & -\frac{3}{16} \cdot [\frac{1}{8}\frac{16}{3} \log(\frac{3}{32}\frac{1}{8}) + \frac{1}{16}\frac{16}{3} \log(\frac{3}{32}\frac{1}{4})] \\
& -\frac{3}{8} \cdot [\frac{1}{16}\frac{8}{3} \log(\frac{3}{16}\frac{1}{8}) + \frac{1}{4}\frac{8}{3} \log(\frac{3}{16}\frac{1}{8}) + \frac{1}{16}\frac{8}{3} \log(\frac{3}{16}\frac{1}{4})] \\
& -\frac{7}{16} \cdot [\frac{1}{16}\frac{16}{7} \log(\frac{7}{32}\frac{1}{8}) + \frac{3}{8}\frac{16}{7} \log(\frac{7}{32}\frac{1}{4})] \\
= & \ 5.01
\end{aligned}$$

The average coding length (or cross-entropy) for a sentence is thus 5.01 bits.

Each sentence includes two words, so the average coding length for one word is 2.50 bits. The result equals to what was calculated in part (a). This is due to the fact that the distribution over which the expected value of the coding lengths is calculated is the same.

2. a) Each of the 30 elementary events has a probability of $\frac{1}{30}$. Just place these into the definition of entropy:

$$\begin{aligned} H(X) &= \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)} \\ &= \sum_{i=1}^{30} \frac{1}{30} \log_2(30) \\ &= \log_2(30) \approx 4.91 \text{ bits} \end{aligned}$$

b) To generate a one letter word, the given random language should generate two symbols, i.e. word boundary after something else. The probability for this is

$$P(s = t_1) = \frac{1}{30} \cdot \frac{1}{30}$$

and there are 29 words of this kind.
Respectively, the probability of a word of two letters is

$$P(s = t_1, t_1) = \frac{1}{30} \cdot \frac{1}{30} \cdot \frac{1}{30},$$

there are 29^2 words of this kind, and so on.
Let's calculate the entropy:

$$\begin{aligned} H(X) =& \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)} \\ =& \ 29 \times (\frac{1}{30})^2 \log_2(30^2) + 29^2 \times (\frac{1}{30})^3 \log_2(30^3) + 29^3 \times (\frac{1}{30})^4 \log_2(30^4) + \dots \\ =& \ \frac{1}{29} \left(\left(\frac{29}{30}\right)^2 \cdot 2 \cdot \log_2(30) + \left(\frac{29}{30}\right)^3 \cdot 3 \cdot \log_2(30) + \left(\frac{29}{30}\right)^4 \cdot 4 \cdot \log_2(30) + \dots \right) \\ =& \ \frac{\log_2(30)}{29} \left(-\frac{29}{30} + \sum_{i=0}^{\infty} i \cdot \left(\frac{29}{30}\right)^i \right) \end{aligned}$$

The sum term has a well-known solution. Let's quickly go through it:

$$\sum_{i=0}^{\infty} iq^i = q + 2q^2 + 3q^3 + 4q^4 + \dots \tag{1}$$

Multiply both sides by q.

$$q \sum_{i=0}^{\infty} i q^i = q^2 + 2q^3 + 3q^4 + 4q^5 + \ldots \tag{2}$$

Subtract equation 2 from equation 1.

$$\begin{aligned} (1-q) \sum_{i=0}^{\infty} i q^i &= q + q^2 + q^3 + q^4 + \ldots && (3) \\ \sum_{i=0}^{\infty} i q^i &= \frac{q + q^2 + q^3 + q^4 + \ldots}{1-q} && (4) \end{aligned}$$

Multiply equation 4 by q.

$$q \sum_{i=0}^{\infty} i q^i = \frac{q^2 + q^3 + q^4 + q^5 + \ldots}{1-q} \tag{5}$$

Subtract equation 4 from equation 5 to obtain the solution:

$$\begin{aligned} (1-q) \sum_{i=0}^{\infty} i q^i &= \frac{q}{1-q} && (6) \\ \sum_{i=0}^{\infty} i q^i &= \frac{q}{(1-q)^2} && (7) \end{aligned}$$

(In order to make the subtractions and the multiplications, $q \neq 0$ and $|q| < 1$.)

Applying this to the orginal problem, we obtain the result of

$$\begin{aligned} & \frac{\log_2(30)}{29}\left(-\frac{29}{30} + \frac{\frac{29}{30}}{(1-\frac{29}{30})^2}\right) \\ = \; & \log_2(30)(30 - \frac{1}{30}) \\ = \; & 147 \text{ bits} \end{aligned}$$

At first glance this may seem confusing: Shouldn't the result be same as in part (a)? A quick verification shows that it is indeed right: Expectation value for word length is 29, so entropy per symbol is approximately $147/(29+1) = 4.90$ bits.

There is also another reason for the results not to be exactly same: The first source may generate successively two word boundaries, the second source can not. In consequence, the second source has a bit lower entropy.

3. a) Let's mark the perplexities for the models as $Perp_1$, $Perp_2$ and $Perp_3$.

$$\begin{aligned}
&Perp_1(\text{'kissa'}, \text{'menee'}, \text{'puuhun'}) \\
&= P_1(\text{word}_1\text{='kissa'}, \text{word}_2\text{='menee'}, \text{word}_3\text{='puuhun'})^{-\frac{1}{3}} \\
&= (P_1(\text{word='kissa'})P_1(\text{word='menee'})P_1(\text{word='puuhun'}))^{-\frac{1}{3}} \\
&= (0.1 \cdot 0.1 \cdot 0.1)^{-\frac{1}{3}} = 10
\end{aligned}$$

The model always chooses one of ten different words with equal probabilities, so this is exactly what we should get.

$$\begin{aligned}
& Perp_2(\text{'kissa'}, \text{'menee'}, \text{'puuhun'}) \\
= \;& P_2(\text{word}_1\text{=subject}, \text{word}_2\text{=verb}, \text{word}_3\text{=object})^{-\frac{1}{3}} \\
= \;& (P_2(\text{word=subject})P_2(\text{word=verb})P_2(\text{word=object}))^{-\frac{1}{3}} \\
= \;& (0.33 \cdot 0.33 \cdot 0.33)^{-\frac{1}{3}} = 3
\end{aligned}$$

The model selects always one out of three options, so also this result seems reasonable.

$$\begin{aligned}
& Perp_3(\text{'kissa'}, \text{'menee'}, \text{'puuhun'}) \\
= \;& P_3(\text{word}_1\text{='kissa'}, \text{word}_2\text{='menee'}, \text{word}_3\text{='puuhun'})^{-\frac{1}{3}} \\
= \;& (P_3(\text{word='kissa'}|\text{word=first}) \\
& \cdot P_3(\text{word='menee'} \mid \text{previous_word} = \text{'kissa'}) \\
& \cdot P_3(\text{word='puuhun'} \mid \text{previous_word} = \text{'menee'}))^{-\frac{1}{3}} \\
= \;& (0.25 \cdot 0.33 \cdot 0.33)^{-\frac{1}{3}} = 3.32
\end{aligned}$$

The last model chooses from 3.32 words on average.

The models 1 and 3 are comparable, as they operate with the same set of symbols. Of these two, model 3 seems to be much better.

Model 2 is not comparable to others, as it operates on a smaller set of symbols and gets a low entropy because of that. An extreme example would be a model for which every word goes to the same category. This kind of model would never be "surprised", and the perplexity would be one.

b) Let's examine the next test sentence. For the first model,

$$\begin{aligned}
& Perp_1(\text{'valas'}, \text{'on'}, \text{'kala'}, \text{'paitsi'}, \text{'ettei'}) \\
= \;& (P_1(\text{word='valas'})P_1(\text{word='on'})P_1(\text{word='kala'}) \\
& \cdot P_1(\text{word='paitsi'})P_1(\text{word='ettei'}))^{-\frac{1}{5}} \\
= \;& (0.1 \cdot 0.1 \cdot 0.1 \cdot 0 \cdot 0)^{-\frac{1}{5}} \\
= \;& \frac{1}{0^{\frac{1}{5}}} = \infty.
\end{aligned}$$

We note that perplexity cannot be calculated, if the model gives a probability of zero for any word in the test set. Often those words are excluded. This way the result is

$$\begin{aligned}
& Perp_1(\text{'valas', 'on', 'kala'}) \\
= \quad & (P_1(\text{word='valas'})P_1(\text{word='on'})P_1(\text{word='kala'}))^{-\frac{1}{3}} = 10.
\end{aligned}$$

To report a meaningful result, the perplexity is not enough, but we should also count the words that were not recognized by the model. In this case, $\frac{2}{5} \cdot 100\% = 40\%$ words were out of model's vocabulary. For the next model,

$$\begin{aligned}
& Perp_2(\text{'valas', 'on'}) \\
= \quad & (P_2(\text{word=subject})P_2(\text{word=verb}))^{-\frac{1}{3}} \\
= \quad & (0.33 \cdot 0.33)^{-\frac{1}{2}} = 3
\end{aligned}$$

This model misses 60% of the words.

Also model 3 recognizes only the two first words.

$$\begin{aligned}
& Perp_3(\text{'valas', 'on'}) \\
= \quad & (P_3(\text{word='valas'}|\text{word=first}) \\
& \cdot P_3(\text{word='on'} \mid \text{previous_word} = \text{'valas'})))^{-\frac{1}{3}} \\
= \quad & (0.25 \cdot 0.33)^{-\frac{1}{2}} = 3.5
\end{aligned}$$

The out-of-vocabulary (OOV) rate is 60%.

As before, model 2 is not comparable with the rest. Models 1 and 3 can be compared, as long as we take into account the out-of-vocabulary rates. Model 1 covers more vocabulary, but model 3 gives better perplexity. Creating a language model is often balancing between these properties.

To conclude, perplexity can be used to compare two language models, if the results are calculated similarly and the OOV rates are announced. When comparing results from several sources, both issues must be carefully observed to prevent wrong conclusions.

As a final conclusion of these exercises, let's list the different entropy measures:

- **Entropy**

  $H(X) = E(-\log P(X)) = \sum_x P(x) \log \frac{1}{P(x)}$

  Interpretation: Self-information of the source, or the average coding length that is needed to send a message with the optimal coding

- **Cross-entropy**

$H_M(X) = E(-\log P_M(X)) = \sum_x P(x) \log \frac{1}{P_M(x)}$

Interpretation: Average coding length needed to send the message using the model M for the coding

- **Relative entropy** or **Kullback-Leibler divergence**

  $D(P(X)||P_M(X)) = E(-\log \frac{P(X)}{P_M(X)}) = \sum_x P(x) \log \frac{P(x)}{P_M(x)} = H_M(X) - H(X)$

  Interpretation: How many bits on average are lost if the message is coded with the model M

- **Perplexity**

  $Perp_M(X) = 2^{H_M(X)} = \prod_x (\frac{1}{P_M(x)})^{P(x)}$

  Interpretation: Average branching factor of the model M for the data given by the source