Part I: Multiple choice (circle the correct answer)

1) The IBM Model 3 ...
   a. Has a distortion model but no fertility model.
   b. Has a fertility model but no distortion model.
   c. Has both a distortion model and a fertility model. ← IBM Model 2 introduces the former, IBM Model 3 builds on that with the latter.
   d. Has neither a distortion model nor a fertility model.

2) The smoothing method that recursively approximates an \( N \)-gram probability with \((N-1)\)-gram probability if the \( N \)-gram does not appear in the training data is...
   a. Simple interpolation
   b. Maximum likelihood estimation
   c. Katz backoff ← it ‘backs off’ to a lower-order approximation
   d. Laplace (add-1) discounting

3) You have a dictionary where each word is associated with several possible part-of-speech tags. You also have a corpus without part-of-speech tags (just words). If you want to use a hidden Markov model (HMM) to perform part-of-speech tagging,
   a. You will find the task impossible.
   b. The states in your HMM will be part-of-speech tags. ← states are used to represent hidden data. Here, part-of-speech tags are not observed and are therefore hidden.
   c. The states in your HMM will be words.
   d. The states in your HMM will be bigrams.

4) Formants...
   a. Produce the spectrum from a continuous time series.
   b. Produce the spectrum from a discrete time series.
   c. Are concentrations of energy within a frequency band. ← by definition
   d. Are concentrations of frequencies within an energy band.

5) Given a French sentence \( F \), the noisy channel model of machine translation attempts to find the English translation \( E \) such that
   a. \( P(E|F) \) is maximized.
   b. \( P(F|E) \) is maximized.
   c. \( P(E|F)P(F) \) is maximized.
   d. \( P(F|E)P(E) \) is maximized. ← We need both \( P(F|E) \) for fidelity and \( P(E) \) for ‘naturalness’ in our translations.
Part II: Short answer (provide a brief written response)

1) What is mRMR?

A method for choosing an ‘optimal’ set of features for a classification task from a larger set. Bonus: it balances the desire for a minimally (internally) redundant set with the desire for a maximally relevant (to the task) set, often using mutual information.

2) Zipf’s law is a function of what two variables? Explain these variables with one sentence, each.

*Frequency* is the number of times a word appears in a corpus.
*Rank* is the position of the word among all words sorted from most frequent to least frequent.

3) Name 2 contentful parts-of-speech, 2 functional parts-of-speech, and give at least one example word of each.

There are many possible answers, e.g.,

**Content:** noun (“dog”), verb (“explode”), adjective (“wild”),
**Functional:** determiner (“the”), conjunction (“and”)...

4) The ID3 algorithm chooses a question at each iteration that maximizes what quantity?

*Information gain*

5) What is the relation between mutual information \(I(X; Y)\), entropy \(H(X)\), and conditional entropy \(H(X|Y)\)? Write the equation, or explain it, or draw the Venn diagram.

\[ I(X; Y) = H(X) - H(X|Y) \]
Part III: Worked-out problems

1) Imagine you have a language model with a vocabulary of $|\mathcal{V}| = 50,000$ words. In your training corpus you see the word *monkey* 10 times.
   - 6 times it was followed by the word *house*.
   - 4 times it was followed by other words: *wrench, farmer, around, throws*.

What is the Maximum Likelihood Estimate of $P(\text{house}|\text{monkey})$?
What is the add-2 estimate of $P(\text{house}|\text{monkey})$?
You can leave your answers as fractions, if necessary.

**Maximum Likelihood Estimation** is used when we have complete access to the data (i.e., no hidden variables) and we don’t do smoothing, therefore

\[
P(\text{house}|\text{monkey}) = \frac{\text{Count}(\text{monkey house})}{\text{Count}(\text{monkey})} = \frac{6}{10} = \frac{3}{5}
\]

The formula for the add-2 estimate is

\[
P^*(\text{house}|\text{monkey}) = \frac{\text{Count}(\text{monkey house})+\delta}{\text{Count}(\text{monkey})+\delta |\mathcal{V}|} = \frac{6+2}{10+2 \cdot 50000} = \frac{8}{100,010}
\]
2) Schrödinger’s cat can be in one of two states: alive and almost-alive. This cat will emit either of two different sounds depending on whether he’s alive or almost-alive. This cat is in a box, so you can’t see his state, but you know that somehow he can go back and forth between states. This entire system is described by the HMM parameters in the tables below.

Use the Viterbi algorithm to find the most likely sequence of states for this cat if you hear the sequence of sounds “meow, uuuuggghh” from the box. Show your work.

<table>
<thead>
<tr>
<th></th>
<th>alive</th>
<th>almost-alive</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi ) (alive)</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>( \pi ) (almost-alive)</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>end</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>alive</td>
<td>almost-alive</td>
</tr>
<tr>
<td>alive</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>almost-alive</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>state</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>word</td>
<td>alive</td>
<td>almost-alive</td>
</tr>
<tr>
<td>meow</td>
<td>0.8</td>
<td>0.0</td>
</tr>
<tr>
<td>uuuuggghh</td>
<td>0.2</td>
<td>1.0</td>
</tr>
</tbody>
</table>

This involves filling in the trellis for the Viterbi algorithm, which looks like this:

<table>
<thead>
<tr>
<th>State</th>
<th>alive</th>
<th>almost-alive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \pi(\text{alive})B_{\text{alive}}(\text{meow}) = 0.9 \cdot 0.8 = 0.72 )</td>
<td>( \Psi = \emptyset )</td>
</tr>
<tr>
<td></td>
<td>( \Psi = \emptyset )</td>
<td>( \Psi = \text{alive} )</td>
</tr>
<tr>
<td></td>
<td>( 0.72 \cdot A_{\text{alive},\text{alive}}B_{\text{alive}}(\text{uuuuggghh}) = 0.72 \cdot 0.4 \cdot 0.2 = 0.0576 )</td>
<td></td>
</tr>
<tr>
<td>almost-alive</td>
<td>( \pi(\text{almost-alive})B_{\text{almost}}(\text{meow}) = 0.1 \cdot 0 = 0 )</td>
<td>( \Psi = \emptyset )</td>
</tr>
<tr>
<td></td>
<td>( 0.72 \cdot A_{\text{alive},\text{almost}}B_{\text{almost}}(\text{uuuuggghh}) = 0.72 \cdot 0.6 \cdot 1.0 = 0.432 )</td>
<td></td>
</tr>
</tbody>
</table>

The most likely state sequence is highlighted in light grey.

Because there is 0 probability of starting in the almost-alive state, \([\delta_0(i)A_{ij}]\) will always be maximized for state \( i = \text{alive} \). Note that you did not need to compute the final \( \delta \) values in the last column – you can recognize that \( 0.72 \cdot 0.08 < 0.72 \cdot 0.6 \) and realize that the cat became almost-alive between time steps.