This exam contains 8 pages (including this cover page) and 21 questions. There are a total of 100 points. Use the back page if you need more room for your answers. Describe any assumptions you make in solving a problem. We reserve the right to assign partial credit, and to deduct points for answers that are needlessly wordy.

Grade Table (for instructor use only)

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
1. (2 points) In this course, NLP stands for:
   A. No Likely Proof
   B. Non-Linear Proof
   C. Natural Language Processing
   D. National Lasagna Partners

2. (2 points) In this course, ACL stands for:
   A. Austin City Limits
   B. Association for Computational Linguistics
   C. Anterior Cruciate Ligament
   D. Association for Community Living

3. (4 points) List all the bigrams in the sentence below.
   Colorless green ideas sleep furiously
   Colorless green
   green ideas
   ideas sleep
   sleep furiously

4. (3 points) What does the probability $P(\text{nice}|\text{yesterday was})$ represent? (1-2 sentences)

   $P(\text{nice}|\text{yesterday was})$ represents the probability that "yesterday was nice" was found in the text. It is calculated by counting the number of times "yesterday was nice" is found in the corpora and dividing by the number of times "yesterday was" is found in the corpora.

5. (3 points) Why is it undesirable to have zero probabilities in a language model? Name one technique that is used to avoid this. (1-2 sentences)

   Because language models use the chain rule to estimate the probability of the whole sentence, a single zero probability in the sentence will cause the probability of the entire sentence to be zero. This is undesirable because language is very creative and novel sentences are always being produced, so we don't want to say these new sentences are impossible, which is what a probability of 0 would mean. A technique used to avoid this is Laplace smoothing.
6. (4 points) At an abstract level, what does the perplexity of a language model measure? (1-2 sentences)

   The perplexity of a language model measures how probable the text in a known corpus is, normalized by the number of words in the corpus. Because the corpus is language that we know exists, it should be fairly probably if our language model works correctly.

7. (6 points) How might language modeling help in predictive typing as seen on smart phones?

   Language modeling could help in predictive typing, by using the previous words and a list of some possible words to follow, and find their probabilities. For example, if the user typed "The team played so", we could chose between "well" and "bad" by seeing which one had a higher probability given the previous words.

8. (3 points) Besides spelling, what other application could the Noisy Chanel model be used for?

   A. Part of Speech Tagging
   B. Parsing
   C. Machine Translation
   D. Question Answering

9. (5 points) The noisy channel model can be written using the formula \( \arg \max P(O|w)P(w) \). What does the \( P(w) \) part of the formula represent? How can it be estimated? (1-2 sentences)

   \( P(w) \) represents the probability of the hypothesized correct word. It can be estimated by using a language model.

10. (5 points) Given the Finite State Transducer below, what is the output if the input is proper?

    ![Finite State Transducer Diagram]

    properly
11. (5 points) Imagine we use an FST to transform a word and a set of features into morphemes. For example, catch+PST outputs catch’d. Clearly this is not an English word. How might we go from catch’d to a valid English word?

To go from the morphemes of a word to the written representation of that word, we can use another FST.

12. (10 points) (a) (7 points) The recursive portion of the Viterbi algorithm is:

\[ v_t(q) = \max_{r=1}^{N} v_{t-1}(r)a_{rq}b_q(o_t) \]

Given the HMM representation and partially completed trellis below, what are the values for \( v_4(H) \) and \( v_4(L) \)? (Numerical expressions are fine, no need to do the calculation.)

\[
\begin{align*}
V4(H) &= \max \left[ V3(H) * P(H|H) * P(DOT|H), \right. \\
& \quad \left. V3(L) * P(H|L) * P(DOT|H) \right] \\
&= \max \left[ 0.04556 * 0.3 * 0.25, \\
& \quad 0.01001 * 0.6 * 0.25 \right] \\
&= 0.04556 * 0.3 * 0.25 \\
&= 0.003417 \\
V4(L) &= \max \left[ V3(H) * P(L|H) * P(DOT|L), \right. \\
& \quad \left. V3(L) * P(L|L) * P(DOT|L) \right] \\
&= \max \left[ 0.04556 * 0.5 * 0.9, \\
& \quad 0.01001 * 0.2 * 0.9 \right] \\
&= 0.04556 * 0.5 * 0.9
\]

(b) (3 points) What is the most probable tag sequence? (Hint: The transition to end state is equally likely)

HLLH
13. (5 points) Given the state diagram below, how might the transition probability between JJ and NNP be determined using a corpus (1-2 sentences)?

To find the transition probability between JJ and NNP, you need to find $P(\text{NNP} | \text{JJ})$. This can be calculated from a corpus using the counts of all "JJ NNP" pairs and the counts of all "JJ" tags in total.

14. (4 points) Given the two tag sequences below and the gold standard, which tagger appears to be more accurate, and why?

<table>
<thead>
<tr>
<th>TAGGER1</th>
<th>This/A is/B a/C test/D C C N N = 2/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAGGER2</td>
<td>This/L is/B a/D test/R N C C C = 3/4</td>
</tr>
<tr>
<td>Gold</td>
<td>This/A is/B a/D test/R</td>
</tr>
</tbody>
</table>

Tagger2 appears to be more accurate because it has 3 of the 4 tags correct, for an accuracy of 0.75.

15. (3 points) Is the grammar below in CNF Form?

A. True
B. False

A $\rightarrow$ C D
B $\rightarrow$ C F
C $\rightarrow$ the | a | an
D $\rightarrow$ book | movie | album
F $\rightarrow$ run | jump | dive | C D
16. (9 points) For each parse below, give the name of the type of parse it is.

A  **Constituent**
B  **Lexical**
C  **Dependency**

17. (10 points) Given the following grammar, fill in the shaded box in the CKY table, including any necessary backpointers. In addition, draw the backpointers in for the previously filled cells as necessary.

\[
\begin{align*}
S &\rightarrow NP \ VP \\
VP &\rightarrow V \ NP \\
VP &\rightarrow VP \ PP \\
VP &\rightarrow use \\
PP &\rightarrow P \ NP \\
NP &\rightarrow Det \ N \\
NP &\rightarrow we \\
NP &\rightarrow internet \\
NP &\rightarrow computer \\
V &\rightarrow use \\
P &\rightarrow on \\
P &\rightarrow in \\
N &\rightarrow we \\
N &\rightarrow internet \\
N &\rightarrow computer \\
Det &\rightarrow the \\
Det &\rightarrow a
\end{align*}
\]
18. (5 points) Draw all parse trees for the phrase below as indicated in the CKY table.

![Parse Tree Diagram]

19. (5 points) What is a benefit of probabilistic parsers that isn’t possible with a standard constituent parsers, such as a standard CKY parser? (1-2 sentences)

Compared to standard constituency parsers, which return all possible parses, probabilistic parsers only return one parse, which is the most likely parse. This is a benefit because it performs syntactic disambiguation for us.

20. (4 points) What are some reasons for the growing adoption of Dependency Parsers? What is an advantage they have? (1-2 sentences)

One reason dependency parsers are becoming more popular is they are very fast because they only have to go through the input one time. OR One reason dependency parsers are becoming more popular is that they are independent of word order, so they are useful on many different types of languages.
21. (3 points) Could the following dependency parse be generated by a vanilla shift-reduce based parser? Why or why not? ¹

No, this dependency could not be generated by a vanilla shift-reduce parser. This is because it contains overlapping arcs and is non-projective.

¹Sentence from 2010. Nivre, Joakim. Dependency Parsing. Language and Linguistics Compass