1. <>ing -> <>
   <>er -> <>
   <>ee -> <>
   Other variations such as regular expressions in form of vowels and consonants have also been given full credit. Rules to overfitting just the given data set / having very low recall received partial credit.

2. Porter stemmer output: thinner thin thinker think think tinker water water emploi employe employ absent absente thee garden garden garden fee
   Porter stemmer has sophisticated rules to prevent issues occurring from naive stemming rules. It sometimes converts words to meaningless lexical forms to preserve semantic differences but the stemmed forms retain uniqueness to not clash with existing english words (employ -> emploi).

3. Yes. <>er -> <> :{Tinker, tink} (completely different meaning), <>ee -> <>, <>ee - > <>{employ, employer, employee} (moderately different)
   But, if the answer matched the rules you proposed, you received credit for them.

4. Yes. water -> wat, fee -> f, thee -> th
   Potential solution is to stem only when the resulting form is indexed pr ensure that the stemmed form is unique and doesn’t clash with existing English words. Limiting application of stemming by word length received partial credit but is not very generalizable.
   As in part 3, if the answer matched the rules you proposed, you received credit for them.
B.

1. There are 1 million documents and 10K distinct words, so the matrix would be of size 1 million * 10k, if using sparse representation. Using non-sparse representation, we only need to store the indices of nonzero entries of the matrix. Each word would have an index (row, col) indicating which row and which column it resides at. And in total we have 1 million * 1,000 1s in the matrix. So the space requirement is 1 million * 1K * (log(1 million) + log(1K)). [If you have 1 million * 1K, you are not considering the space needed to store each index]

2. According to Zipf’s law, a term’s rank and frequency are inversely proportional. Suppose a document has at most 1,000 unique terms and the vocabulary has 10K distinct terms total. Then, only 10% of the vocabulary is observed in that document’s term-incidence matrix using “1”s. The rest of the space, around 90% or more, depending on the number of unique terms in the document, is wasted with “0”s. By only storing the terms that actually occur in each document, space can be conserved more efficiently. This can be done by using an inverted list, which stores the documents that contain each word from the vocabulary. This can be implemented via a linked list with a look up table. Each word in the vocabulary can be linked to the documents that contain it. Let L denote the number of terms within one document and D denote the number of documents. The space complexity of the inverted list is O(D*L*log(D)) where. The number of bits needed to store the number of documents is log(D), and there are D*L entries in the list. If Zipf’s law holds true, space is conserved greatly and the new space requirement would be O(D*L*log(D)) = 1 million * 1K * log(1 million). (Credit to Anvitha Rayabhari)

[Note that you may also consider the size you use for 10K distinct words, then the result should be 1 million * 1K * log(1 million)+ log(10K)]
C.

1. “I”: 1:<1>; 2:<1>
   “student”: 1:<4>; 2:<4>

   “student I”: 1: < none >; 2: < (4,5) >;
   "I /2 student": 1: < (6,4) >; 2: < (5,4) >;
   "student /2 I": 1: < (4,6) >; 2: < (4,5) >;

3. No.
   [You can design any scheme as long as it is general enough for any query within a
   sentences. Below is one possible scheme]

   Change the scheme to include punctuations.
   To query “I /2 student” in one sentence:
     “/p I/2 student /p”
   where “/p” stands for any punctuations that ends a sentence. For example: “?”, "!", "."
D.

1. Precision = 3/10
   Recall = 3/20
2. Precision = 4 / 9
   Recall = 4 / 7
3. Recall is more important than precision:
   a. Before a company applies for a patent, they would like to know whether there are any relevant patents, which were already accepted.
   b. When a doctor is looking for side effects about a medicine
4. Precision is more important than recall:
   c. When a patient wants to find medicines to cure a disease.
   d. When someone’s tire is flat at night, they would want more refined results to know how to deal with the situation.

4.

<table>
<thead>
<tr>
<th></th>
<th>Relevant</th>
<th>Non-relevant</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved</td>
<td>A</td>
<td>B</td>
<td>A+B</td>
</tr>
<tr>
<td>Not retrieved</td>
<td>C</td>
<td>D</td>
<td>C+D</td>
</tr>
<tr>
<td>Total</td>
<td>A+C</td>
<td>B+D</td>
<td>A+B+C+D</td>
</tr>
</tbody>
</table>

Accuracy in the ad-hoc IR is the proportion of correctly classified items as relevant/irrelevant. Accuracy is not a good measure for IR, as it conflates performance on relevant items (A) with performance on irrelevant (uninteresting) items (D). For example, when a set of documents contain 99% false documents, a classifier can simply return false on every document to have a 99% accuracy.

Precision (A/(A+B)) is the proportion of relevant items among retrieved items, and recall (A/(A+C)) is the proportion of retrieved items among the relevant items. They show different goodness that are tied to the notion of relevance, and both should be considered when we evaluate the IR system.
E. (Answer from Michael Roth)

E. Evaluating Ranked Retrieval

Consider the following (ranked) retrieval order for a search result in a corpus with 20 documents (R is relevant, N is non-relevant):

Query 1: 8 relevant document of 20 total documents: R R N R N R R N N N N N N R N N N R
Query 2: 10 relevant documents of 20 total documents: R R N R N R R R N N N N N R N N N N

1. Create the confusion matrix (ground truth) for each query after 10 documents are retrieved

<table>
<thead>
<tr>
<th></th>
<th>Query 1</th>
<th>Query 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved</td>
<td>Not retrieved</td>
<td>Retrieved</td>
</tr>
<tr>
<td>Relevant</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Not relevant</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

2. Calculate precision@3, 5, 10 and recall@3, 5, 10 for each query

<table>
<thead>
<tr>
<th></th>
<th>Query 1</th>
<th></th>
<th>Query 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Position</td>
<td>Precision</td>
<td>Recall</td>
<td>Position</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>2/3</td>
<td>1/4</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3/5</td>
<td>3/8</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>3/5</td>
<td>3/4</td>
<td>10</td>
</tr>
</tbody>
</table>

3. Calculate the F-1 measure @ 10 for each query

\[
\text{Query 1 F1 measure @10} = 2 \cdot \frac{\text{precision \cdot recall}}{\text{precision} + \text{recall}} = 2 \cdot \frac{\frac{3}{5} \cdot \frac{4}{5}}{\frac{3}{5} + \frac{4}{5}} = \frac{2}{3}
\]

\[
\text{Query 2 F1 measure @10} = 2 \cdot \frac{\text{precision \cdot recall}}{\text{precision} + \text{recall}} = 2 \cdot \frac{\frac{4}{5} \cdot \frac{4}{5}}{\frac{4}{5} + \frac{4}{5}} = \frac{4}{5}
\]

4. Calculate the Mean Average Precision (MAP) of the system

\[
\text{Query 1 } = \frac{1}{8} \left( \frac{1}{1} + \frac{2}{2} + \frac{3}{4} + \frac{4}{6} + \frac{5}{7} + \frac{6}{8} + \frac{7}{16} + \frac{8}{20} \right) = \frac{9607}{13440} \approx 0.71
\]

\[
\text{Query 2 } = \frac{1}{10} \left( \frac{1}{1} + \frac{2}{2} + \frac{3}{3} + \frac{4}{6} + \frac{5}{7} + \frac{6}{8} + \frac{7}{9} + \frac{8}{10} + \frac{9}{15} + \frac{10}{16} \right) = \frac{19993}{25200} \approx 0.79
\]

Overall System MAP = \frac{1}{2} \left( \frac{9607}{13440} + \frac{19993}{25200} \right) = \frac{304049}{403200} \approx 0.75

5. Calculate the Mean Reciprocal Rank (MRR) of the system

Since the first document in the results for both queries is relevant, then the mean reciprocal rank is 1. The calculation is \(\frac{1}{2} \left( \frac{1}{1} + \frac{1}{1} \right) = 1\)
F. (Answer from Liang Zhang)

1. 

\[
\text{recall} = \frac{|\{\text{relevant documents} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}
\]

Since the numerator of query O is not less than (or equals to) the numerator of query A and the denominator of query O is equal to denominator of query A, then the recall of query A cannot be higher than the recall of query O. Thus, for the recall, query A cannot be higher than query O. So recall of A will always be smaller or equal to recall of O.

2. 

Counter example:
Suppose there are 4 documents.
Doc 1: Yesterday is either rainy or sunny.
Doc 2: Today is rainy
Doc 3: Tomorrow is sunny.
Doc 4: rainy,rainy,rainy,sunny,sunny,sunny.

Suppose Doc 1, 2, 3 are meaningful, and they are relevant documents while Doc 4 is not relevant document.
Precision = #relevant retrieved/ #retrieved
Query A: The precision rate of (rainy and sunny) = Doc 1 /Doc 1, Doc 4 = 1/2
Query O: The precision rate of (rainy or sunny)=(Doc 1, Doc 2, Doc 3) / (all four document) = 3/4
1/2 < 3/4, which implies query O is higher than query A.
So, this is a counter example.