Problem 4

- Named entity recognition: 30 points
  - Named entities: people names, organizations, locations, numerals, etc
  - Your objective is to build a machine learning named entity recognition system, which when given a new previously unseen text can identify and classify the named entities in the text. This means that your system should annotate each word in the text with one of the four possible classes.
  - You will be given labeled data sets to train and test your model.

Problem 5

- Text classification: 20 points
  - This data set contains 1000 text articles posted to each of 20 online newgroups, for a total of 20,000 articles. For documentation and download, see http://www-2.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html.
  - The "label" of each article is which of the 20 newsgroups it belongs to. The newsgroups (labels) are hierarchically organized (e.g., "sports", "hockey").
  - You should provide model evaluation results and discuss the reasons of the results.

Problem 6

- Web Content Identification: 20 points
  - This dataset contains webpages from 4 universities, labeled with whether they are professor, student, project, or other pages. For data and documents, see http://www-2.cs.cmu.edu/~webkb/
  - Project ideas:
    - Learning classifiers to predict the type of webpage from the text
- Additional points: you will get additional 20 points if you can improve accuracy by exploiting correlations between pages that point to each other

Problem 7

- Detecting sentiment polarity: 30 points
  - Given text about movie reviews
  - Can we detect sentiment, like whether a comment is
    - Positive?
    - Negative?
  - Can we tell to what extent is a comment positive of negative?
- Data:
  - 5331 positive snippets
  - 5331 negative snippets
- Other resources:
  - The Subjectivity Lexicon

Problem 8

- Word sense disambiguation: 30 points
  - Implement the simplified word sense disambiguation algorithm, and apply it to disambiguate a target ambiguous word in context.
  - For evaluation, use the dataset provided and the sense definitions provided by Wikipedia.
  - Note that you have to apply your own pre-processing to the content of the Wikipedia page (e.g., include the entire page or only certain sections; include the titles of the linked articles or not; etc.).
  - The quality of the pre-processing may affect the quality of your results. Report the accuracy of each word (i.e., number of instances correctly disambiguated).

Problem 9

- Parser: 40 points
  - In this assignment, you will build an English treebank parser. You will consider both the problem of learning a grammar from a treebank and the problem of parsing with that grammar.
  - The data is from the Penn Treebank, you can divide the data into the training data, the development data, and the blind test data as required.
  - You are recommended to build an array-based CKY parser, but you are also free to build an agenda-driven PCFG parser.

Problem 10

- Sentence matching: 30 points
  - In this assignment, you will build a model to compute the similarity of two sentences. You will consider both the problem of sentence meaning representation and the problem of similarity computing with that representation.
  - The data is from a clinical record. The model you build should make medical diagnosis through comparing the similarity of a new symptom with known symptoms. You can divide the data into the training data, the development data, and the blind test data as required.
  - You are recommended to build an array-based CKY parser, but you are also free to build an agenda-driven PCFG parser.

Problem 11

- Chatbot: 40 points
  - In this assignment, you will build a chatbot that can converse with you naturally with respect to some movies.
  - The chatbot is You can The data is collected from a clinical record. The model you build should make medical diagnosis through comparing the similarity of a new symptom with known symptoms. You can divide the data into the training data, the development data, and the blind test data as required.
  - You are recommended to build an array-based CKY parser, but you are also free to build an agenda-driven PCFG parser.