ECS763P
Natural Language Processing

Week 7
Review & Feedback

Matthew Purver
(with material from Jurafsky & Martin)

Coursework

- Most people finished
- Some people didn't do this bit:

When completed, you must submit two things:

- Your completed Python code;
- A PDF document describing what you did (instructions below).

- See QMPlus for example solutions with comments

Using QMPlus for Revision

- Each section has “learning outcomes”
  - Things you should be able to do in the exam
- If you can do everything on the list, you’re in good shape
- Otherwise, you know what to revise!

Week 1: Introduction

- After studying this section you should be able to:
  - Build a simple dictionary-based or classifier-based text classifier;
  - Explain the limitations of word- and ngram-based models;
  - Describe the co-occurrence-based approach to building vector models of word meaning.

Week 2: Text Classification

- After studying this section you should be able to:
  - Explain how to implement and apply a Naive Bayes classifier, and how to calculate a simple example;
  - Explain the principles and distinctive features of Naive Bayes and logistic regression classifiers, including the need for smoothing and regularisation;
  - Describe the main principles and features of advanced classification models such as support vector machines;
  - Explain the difference between generative and discriminative approaches;
  - Discuss issues in and basic approaches to word tokenisation, text normalisation, spelling correction and stemming;
  - Explain the use of these methods in text classification and sentiment analysis, and discuss their relative advantages and disadvantages;
  - Explain basic evaluation metrics (precision, recall and F-score) and discuss issues in evaluation and training for unbalanced datasets.

Week 3: Sequence Models

- After studying this section you should be able to:
  - Describe and explain the technique of n-gram language modelling;
  - Explain techniques for smoothing and interpolation of n-gram models, and discuss their advantages and disadvantages;
  - Explain and compare class-based language models and Hidden Markov Models;
  - Explain how Hidden Markov Models are used for likelihood estimation and sequence tagging, and how to calculate a simple example;
  - Describe the main principles and features of advanced sequence models such as conditional random fields and recurrent neural networks;
  - Explain the use of these methods in speech recognition, part-of-speech tagging, named entity recognition and dialogue act tagging, and discuss their relative advantages and disadvantages.

Week 4: Unsupervised Methods

- After studying this section you should be able to:
  - Explain the expectation-maximisation (EM) approach to unsupervised learning;
  - Explain a range of applications of EM, including k-means clustering, Brown clustering and the forward-backward algorithm;
  - Describe the main principles and features of latent variable models such as latent semantic analysis and latent Dirichlet allocation;
  - Discuss the use of these methods in topic modelling, HMM and grammar induction, and discuss their relative advantages and disadvantages.

Week 5: Formal Grammar

- In this lecture (Feb 7th), we will learn about:
  - some historical context for the concept of grammars and formal grammars of natural language
  - three main concepts of constituent, grammatical relations, and dependency relations, underlying the formalisation of grammar
  - the notion of constituency or phrase based grammars
  - the formalism of context free grammars (CFG's): its formal definition and examples thereof
    - direct derivations, derivations, and parse trees in a CFG
    - treebanks
    - a bracketed form for denoting parse trees in a compact form

Week 6: Syntactic Parsing

- In this lecture (Feb 13th), we will learn about:
  - two other widely used formalisms for grammar:
    - dependency grammars and categorial grammar
    - examples thereof, how they compare to CFG's, their advantages and disadvantages
  - applications of parsing
  - two algorithms for syntactic parsing by search:
    - the top-down algorithm
    - the bottom-up algorithm
  - the challenge of ambiguity for parsing, its various forms and examples thereof
  - ideas for solving the problem of ambiguity