Midterm Exam Study Guide | ENLP Fall 2016

This page provides a list of concepts you should be familiar with and questions you should be able to answer if you are thoroughly familiar with the material in the course. It is safe to assume that if you have a good grasp of everything listed here, you will do well on the exam. However, we cannot guarantee that only the topics mentioned here, and nothing else, will appear on the exam.

How to review

You should review the lecture slides, quizzes, and homework assignments. The readings should be helpful as well. If there are topics that you are not clear on from these resources, please ask on the discussion board, in office hours, or in the review session.

Exam procedures

The exam will be completed without use of a laptop, calculator, or textbook/reference materials.

Scope of the midterm

Everything in the course up to and including HMMs for part-of-speech tagging is fair game. We will not, however, test you on the operation of the Viterbi algorithm in the midterm.

Generative probabilistic models

We have discussed the following generative probabilistic models:

- Naïve Bayes classifier
- N-gram Language Model
- Hidden Markov Model

For each of these, you should be able to

- describe the imagined process by which data is generated, and say what independence assumptions are made (includes familiarity with terms like "Markov assumption" and “trigram model").
- write down the associated formula for the joint probability of hidden and observed variables (or just the observed variables if there are no hidden variables).
- compute the probability of (say) a tag-word sequence, document, or whatever the model describes (assuming you know the model parameters).
- for the naïve Bayes model, compute the most probable class for a particular input, hand-simulating any algorithms that might be needed (again assuming you know the model parameters).
- for the HMM, convert between a state transition diagram and transition table, and explain why a dynamic programming algorithm is needed for certain operations (decoding—Viterbi algorithm; marginalizing over tags—forward algorithm). You do not need to understand (for the midterm) how these algorithms work.
- explain how the model is trained.
- give examples of tasks the model could be applied to, and how it would be applied.
- say what the model can and cannot capture about natural language, ideally giving examples of its failure modes.

Discriminative classification models

We have covered:

- binary & multiclass Perceptron
- SVM
- multiclass Logistic regression/MaxEnt

For this model, you should be able to

- give examples of tasks the model could be applied to, and how it would apply (e.g., what features might be useful).
- explain at a high level what training the model aims to achieve, and how it differs from training a generative model.
- identify which models can be trained with early stopping, averaging, or regularization, and why these techniques are used.
- discuss the pros and cons of discriminative classifiers vs. Naïve Bayes.
- explain why optimization is required for learning, unlike with generative models.
- identify MaxEnt as probabilistic and the others as non-probabilistic. You should understand the formula for computing the conditional probability of the hidden class given the observations/features, and be able to apply that formula if you

are given an example problem with features and weights. You do not need to memorize the formula.
- explain how naïve Bayes can be expressed as a linear classifier.
- write and explain the decoding (classification) rule for any linear classifier given a vector of weights and a feature function.
- walk through the Perceptron learning algorithm for an example.
- explain how, for the Perceptron, decoding is embedded as a step within learning.

Other formulas

In addition to the equations for the generative and discriminative models listed above, you should know the formulas for the following concepts, what they may be used for, and be able to apply them appropriately. Where relevant you should be able to discuss strengths and weaknesses of the associated method, and alternatives.

- Bayes' Rule (also: definition of Condition Probability, law of Total Probability aka Sum Rule, Chain Rule, marginalization, and all other relevant formulas in the Basic Probability Theory reading)
- Noisy channel model
- Add-One / Add-Alpha Smoothing
- Interpolation (for language model smoothing)
- Dot product
- Precision, recall, and F1-score

Additional Mathematical and Computational Concepts

Overarching concepts:

- Zipf's Law and sparse data: What is Zipf's law and what are its implications? What does "sparse data" refer to? Be able to discuss these with respect to specific tasks.

- Probability estimation and smoothing: What are different methods for estimating probabilities from corpus data, and what are the pros and cons of each, and the characteristic errors? Under what circumstances might you find simpler methods acceptable, or unacceptable? You should be familiar at a high level at least with:

    - Maximum Likelihood Estimation
    - Add-One / Add-Alpha Smoothing
    - Lower-order smoothing

    - Interpolation
    - Backoff
    - Good-Turing Smoothing
    - Kneser-Ney Smoothing
    - Entropy and Cross-Entropy/Perplexity

  Except as noted under "Formulas" above, you do not need to memorize the formulas, but should understand the conceptual differences and motivation behind each method, and should be able to *use* the formulas if they are given to you.

- Training, development, and test sets: How are these used and for what reason? Be able to explain their application to particular problems.
- Cross-validation
- The distinction between parameters and hyperparameters
- The distinction between models and algorithms
- Objective function: Classification objective, Learning objective/Loss function

Linguistic and Representational Concepts

You should be able to explain each of these concepts, give one or two examples where appropriate, and be able to identify examples if given to you. You should be able to say what NLP tasks these are relevant to and why.

- Ambiguity (of many varieties, w.r.t. all tasks we've discussed)
- Part-of-Speech
    - especially, the terms: common noun, proper noun, pronoun, adjective, adverb, auxiliary, main verb, preposition, conjunction, punctuation
    - open-class vs. closed-class
- Word Senses and relations between them (synonym, antonym, hypernym, hyponym, similarity; homonymy vs. polysemy)
- Word order typology: SVO, VSO, etc.
- Dialect vs. language
- Phonetics, phonology, lexicon, morphology, syntax, semantics, pragmatics, orthography
- Synthetic vs. analytic language
- Inflectional vs. derivational morphology
- consonant, vowel, tone, syllable, prosody, stress
- morpheme, affix, prefix, suffix, compound
- International Phonetic Alphabet (what it is—not how to use it), Unicode

- Language families, e.g. Indo-European, Romance, Germanic, Slavic, Sino-Tibetan, Semitic
- Salient aspects of languages presented thus far (questions will be about groups of languages, so if you missed one or two presentations that shouldn't put you at a disadvantage)

Also, you should be able to give an analysis of a phrase or sentence using the following formalisms. Assume that either the example will be very simple and/or some set of labels is provided for you to use. (i.e. you should know some standard categories for English but you don't need to memorize details of specific tagsets etc.)

- label parts of speech
- label word senses given dictionary definitions
- label named entities given a list of classes
- label the sentiment of a document as well as some positive and negative cue words

Tasks

You should be able to explain each of these tasks, give one or two examples where appropriate, and discuss cases of ambiguity or what makes the task difficult. In most cases you should be able to say what algorithm(s) or general method(s) can be used to solve the task, and what evaluation method(s) are typically used.

- Tokenization
- Spelling correction
- Language modeling
- PoS-tagging
- Text categorization
- Word sense disambiguation
- Named entity recognition
- Sentiment analysis

Corpora, Resources, and Evaluation

You should be able to describe what linguistic information is captured in each of the following resources, and how it might be used in an NLP system.

- Penn Treebank
- WordNet

For each of the following evaluation measures, you should be able to explain what it measures, what tasks it would be appropriate for, and why.

- Cross-entropy/Perplexity
- Accuracy
- Precision, recall, and F1-score

In addition:

- Intrinsic vs. extrinsic evaluation: be able to explain the difference and give examples of each for particular tasks.
- Gold standard: what is it and what is it used for?
- Confusion matrix: what is it and what is it used for?

Text Processing

You should be able to write and interpret Python-style regular expressions with the following components:

- string delimiters: `^ $`
- optionality/repetition operators: `| ? * + {3,5}`
- the `.` operator (any character)
- character classes: e.g. `[xyz]`, `[^a-z0-9]` and the abbreviations `\w \W \s \S \d \D`
- groups: e.g. `([a-z][0-9])+`
- backslash-escaping for metacharacters

You should be familiar with the Unix text commands covered in class, including the concept of piping commands together and writing to stdout or redirecting output to files.

You should be familiar with basic Python functionality, esp. involving strings and data structures of the types: list, tuple, dict, Counter.

You will not be asked to write Python code or Unix commands from scratch, but you may be asked to choose which of several commands performs the desired function, for example.

You should be familiar with the file formats: TSV, JSON

You should be familiar with the concept of version control and its benefits. We will not test you on specific version control systems or commands.