Introduction

LING 570

Fei Xia

Week 1: 9/30/09

Outline

- Course overview
- Tokenization
- Homework #1
- Questionnaire

Course overview

General info

- Course url: **http://courses.washington.edu/ling570**
  - Syllabus: slides, hw, updated every week.
  - GoPost: for course-related discussion including pointers to the recording.
  - Collect it: for submitting your assignments and seeing TA's comments
  - GradeBook: for getting your grades

- Slides:
  - The slides will be online before class and may be revised afterwards.
  - "Additional slides" provide additional information. They are not required and not covered in class.

GoPost

- All course-related discussion should go to GoPost except for confidential issues (e.g., grades).

- Examples:
  - Can someone explain …?
  - My code is very slow: 3 hours/run. Do you run into the same problem?
  - How to solve the problem?
  - My results for Q1(a) are … Does that look right?

- GoPost is the main venue for collaboration outside classroom.
  - Post your questions
  - Reply to others' questions

Emails

- Mailing lists: (created automatically according to SDB, not EOS)
  - ling570a_au09@u.washington.edu, ling570b_au09@u.washington.edu, ling570c_au09@u.washington.edu
  - Do not send emails to these mailing lists unless for emergency.
  - I will use them only for emergency
  - Did you receive a message from me this morning at 1:15am?

- Emails:
  - Use it only for confidential issues
  - All non-confidential questions should be sent to GoPost

- Please check your emails and GoPost at least once per day.

- If we (Fei or David) emails you, please reply within 24 hours.

Homework submission

- Use “Collect it”: submit the tar file.
  - **tar –cvf hw1.tar hw1_dir**
- Due date: every Wed at 11:45pm.
- The submission area is closed **4** days after the due date.
- There is 1% penalty for every hour after the due date.
- Programming languages: C, C++, Java, Perl, or Python.
- Please follow the instructions in the assignments about the naming convention.

Homework Submission (cont)

- Each submission includes
  - a note file: hw1.(txt|doc|pdf) for hw1.
    - If your code does not work, explain in the note file what you have implemented so far.
  - a set of shell scripts: e.g., eng_token.sh
  - source code: e.g., eng_token.C
  - binary code (for C/C++/Java): eng_token.out
  - data files if any.

- Time spent on an assignment: 15-20 hours/week

➔ I would appreciate it if you could tell me the time you spent on the homework.

Grades

- Assignments: 80-90%
- Quizzes: 0-10%
- Class participation: 10%

- No mid-term or final exams

- Grades are on GradeBook
- TA’s feedback on assignments is on CollectIt

Course policies

- **No** “incomplete” or extension unless you can prove your case.
- Late assignments: 1% penalty for every hour after the deadline.
- No submission will be accepted 4 days after the due day.

Recording

- Distance learning
- You can listen to the recording afterwards

- Remind me to
  - record the class
  - check the chat box
  - repeat the questions (if needed)

- Try to stay close to the mic

- The links to the recordings will be on GoPost.

Fei's contact info

- Email: fxia@uw.edu
  - Subject line should include "**ling570**"
  - Use emails only for confidential matters (e.g., grades)
  - If I don't reply within 24-48 hours, you can send me a reminder.

- Office hour:
  - Location: Padelford A-210G
  - Time: 11am - noon

TA's contact info

- TA: David Goss-Grubbs
- Email: davidgg@uw.edu
- Office hour:
  - Location: ART 337
  - Possible time:
    (1) M: 12:30-1:30
    (2) M: 2:20-3:20
    (3) W: 2:20-3:20
    (4) F: 2:20-3:20

Course description

Prerequisites for ling570

- CS 326 (Data Structures) or equivalent:
  - Ex: hash table, array, tree, …
- Stat 391 (Prob. and Stats for CS) or equivalent: Basic concepts in probability and statistics
  - Ex: random variables, chain rule, Bayes' rule
- Regular grammars/languages, FSA/FST
- Programming in Perl, C, C++, Java, or Python
- Basic unix/linux commands (e.g., ls, cd, ln, sort, head): tutorials on unix

Prerequisites (cont)

- Placement tests: required for CLMA students
- LING 473 with 3.0+ grade: CLMA, NLT
- Single-course enrollment, state-funded programs

➔ A take-home questionnaire

Reading

- Textbook: Manning & Schutze
  - Get it from UW bookstore or amazon.com, etc.
- Reference book: Jurafsky and Martin (2008)

Topics covered in Ling570

- Unit #1: Formal language and FSA (2.5 weeks)
  - Formal language and formal grammar: 0.5
  - FSA: 0.5
  - FST: 0.5
  - Morphological analysis: 1

- Unit #2: ngram models and HMM (3 weeks)
  - ngram LM and smoothing: 1
  - Part-of-speech (POS) tagging and HMM: 1
  - ngram tagger: 1

Topics covered in ling570 (cont)

- Unit #3: Classification (2.5 weeks)
  - Introduction to classification: 1
  - POS tagging with classifiers: 0.5
  - Chunking: 0.5
  - Named-entity (NE) tagging: 0.5

- Other topics (2 weeks)
  - Introduction, tokenization, probability theory: 0.5
  - Clustering: 0.5 (??)
  - Information extraction (IE): 0.5 (??)
  - Summary: 0.5

CompLing courses offered in the ling dept

- Core courses on CompLing: Ling570-573
  - Ling570: Shallow processing
  - Ling571: Deep processing
  - Ling572: Statistical methods for NLP: focusing on supervised learning
  - Ling573: System/applications

- Other courses on CompLing:
  - Grammar engineering: ling566 and ling567
  - Seminar (ling575): information extraction, semantic search, etc.

Ling571 - 573

- Ling571: Parsing, semantics, discourse, dialogue, natural language generation, …
- Ling572: Decision tree, Naïve Bayes, Boosting, SVM, MaxEnt, …
- Ling573: Applications (e.g., Q&A, Dialogue systems, …)

Relation between ling570-573

- 570/571 are organized by tasks
- 572 is organized by learning methods and it focuses on statistical methods.
- 573 focuses on building an end-to-end system.

Questions?

Outline

- Course overview
- **Tokenization**
- Homework #1
- Questionnaire

Tokenization

The Task

- Given the input text, break it into tokens.
- A token is a word, a number, a punctuation mark, etc.
- An example:

  John said: “Please call me tomorrow.”

  ➔ John said : “ Please call me tomorrow . ”

A naïve algorithm

- Separating punctuation marks from word tokens.

- Problems:
  - Period: 1.23, Mr., Ph.D., U.S.A., etc.
  - Comma: 123,456
  - Single apostrophes: dog's, I'll
  - Parenthesis: (202)345-4568, (a), 1),
  - Hyphenation:
    - Line-breaking hyphens
    - e-mail, so-called, pro-life, text-based
    - data-base, database, data base
    - a take-it-or-leave-it offer

Whitespace is not always reliable

- Examples:
  - Names: Hong Kong
  - Numbers: 202 345 6787
  - Collocation: because of, pick up
  - Collocation mixed with hyphens:
    - The New York-based company
    - The New York-New Haven railroad

Word segmentation

- No whitespace between words: e.g., Chinese and Japanese

- Noun compounds in German:
  - Ex: Lebensversicherungsgesellschaftsangestellter

    “Life insurance company employee”

What counts as a word?

- In theory:
  - Phonological word, syntactic words, lexeme, etc.
- In practice:
  - $22
  - Hyphenated words
  - Named entity
  - …

➔ Design a set of annotation guidelines, and write a tokenizer based on that.

Homework 1

Patas

- All the files are under ~/dropbox/09-10/570/

- Hw1:
  - examples/: example files. They are NOT gold standard.
  - solution/:

Naming convention

- A tool called “eng_tokenizer.sh”
  - Source code: eng_tokenizer.(pl | py | C | java|cpp|h|…)
  - Shell script: eng_tokenizer.sh, which could look like

```
#!/bin/sh
eng_tokenizer.pl $@

Or
eng_tokenizer.pl abbr_file  $@
```

- More examples are under ~/dropbox/09-10/570/code-samples/
- Shell, perl, python, and binary code must be executable.
- Your code might be tested on new test data.

Q1-Q2: implementing a rule-based tokenizer for English

- Q1:
  - The shell command line should be

    cat input_file | ./eng_tokenizer.sh > output_file
  - Your code can use additional arguments.
  - The i^{th} input line should correspond to the i^{th} output line.
  - Don’t merge the tokens in the input.
  - Don’t spend too much time trying to make your tokenizer perfect.

- Q2: Explain how your tokenizer treats common cases (e.g., numbers, hyphenated words) and what are the remaining problems.

Q3: make_voc.sh

- Given the input text:
  The teacher bought the **bike.**
  The bike is expensive.

- The output should be

The	2
the	1
teacher	1
bought	1
bike.	1
bike	1
is	1
expensive.	1

- Q4: Run Q1 and Q3

- Q5: Probability
  - If you bet a 7 on a roulette wheel, there is a probability of 1/38 of winning.
  - What is the prob of winning 13 times out of 500 bets?
    - Binomial distribution
    - Poisson distribution

Coming up

- Questionnaire is due at 5pm tomorrow

- Next Monday:
  - Finish the quick review of probability theory.
  - Start on regular expression and FSA.

- Hw1 is due next Wed at 11:45 pm.

Your to-do list

- Finish the questionnaire by 5pm tomorrow
  - Submit via CollectIt: pdf, doc, txt, or image files, or
  - Drop a hardcopy in my mailbox at Padelford

- Finish the following by Friday noon
  - Make sure you have access to Syllabus (slides, hw), CollectIt, GoPost, and GradeBook.
  - Make sure you have access to Patas.

- Work on Hw1

Questionnaire

- Open book: You can check your notes and text books, but you should not google the Web or ask other people.

- Take no more than 3 hours

- Remember to write your name and UW email address on the first page

- Grades for Questionnaire are available by this Sat 5pm at GradeBook.

- Main parts:
  - Probability: 30
  - Formal grammars: 25
  - FSA and FST: 20
  - Unix: 10
  - Linguistics: 15
  - A list of questions