Introduction and overview

LING 570
Fei Xia
Roadmap

• Course overview

• Tokenization

• Homework #1

• Questionnaire
Msg #1

• If you did not receive Msg #1 yesterday, please email me the following:
  – The section and the date that you registered
  – The UW email address
Course overview
Course Information

• Course web page:
  – [http://courses.washington.edu/ling570](http://courses.washington.edu/ling570)

  – Syllabus:
    • Schedule: links to slides and homework
    • Slides posted before class, but may be revised afterwards.

  – Catalyst tools:
    • GoPost: discussion board for class issues and links to class recordings.
    • CollectIt: places for homework submission and TA comments
    • Gradebook: for viewing all grades
GoPost Discussion Board

• Main venue for course-related questions, discussion
  – What not to post:
    • Personal, confidential questions; Homework solutions
  – What to post:
    • Everything else that is course-related
      – Can someone explain…?
      – Is this really supposed to take this long to run?
  – Key location for class participation:
    • Post questions or (partial) answers
    • Your discussion space: TA & instructors will not jump in unless needed
GoPost

• The 10-minute rule:
  – If you’ve been stuck on a problem for more than 10 minutes, post to the GoPost.

• Mechanics:
  – Please use your full name or UW NetID as your user id and upload a photo of yours.
  – Don’t wait until the last minute to ask questions
  – Decide how often you want to check GoPost and whether you will read all the posts.
GoPost (cont)
Email

• Should be used only for personal or confidential issues
  – Grading issues, extended absences, other problems
  – Please avoid long emails (e.g., one email per subject)

• General questions/comments go on GoPost

• Please check your UW emails at least once per day for any course-related announcements.

• Please send email from your **UW email address:**
  – Include Ling570 in the subject
  – If you don’t receive a reply in 24 hours, please follow-up.
Homework Submission

• All homework should be submitted through CollectIt
  – `tar -cvf hw1.tar hw1_dir`

• Homework due **11pm Thurs** (unless otherwise noted)

• Late homework receives 1% penalty for each hour after the deadline.

• No submission will be accepted after three days.

• Most major programming languages accepted
  – C/C++/C#, Java, Python, Perl, Ruby
  – If you want to use something else, please contact us first.

• Please follow the naming convention in each HW

• Expect to spend 10-25 hours/week, including HW, reading, and class time.
Grading

• Homework: 10 * 100 points
• Class participation: 100 points
  – GoPost participation counts as well.
• No midterm or final exams
• Lowest score is removed for the final grade

• Grades in Catalyst Gradebook
• TA feedback returned through CollectIt

• Incomplete: only if all work is completed up last two weeks
  – UW policy
Recording

• Everyone can listen to the recording afterwards

• Please remind me to
  – record the class
  – repeat the questions

• In-class students need to speak louder as the mic might not be very sensitive.

• The links to the recordings will be on GoPost on the same day by the TA.
Contact Info

• Instructor: Fei Xia  (fxia@uw.edu)
  ➢ Office hours:
    • Time: 2:15-3:15pm on Tues
    • Location: GUG 418A

• TA: David Inman (davinman@uw.edu)
  ➢ Office hour
    • Time: 11-noon on Thurs
    • Location: Treehouse at GUG

• Online students can attend office hour via skype (email us your skype id beforehand).

• Please do not ping us via skype outside the office hours except for true emergency.
Online Option

• Please check you are registered for correct section
  – NLT/SCE online or in-person: Section A
  – State-funded: Section B
  – CLMS online: Section C
  – CLMS in-person: Section D

• Online attendance for in-person students
  – Not more than 3 times per term (e.g., missed bus, snow)

• Please enter meeting room 5 mins before start of class
  – Try to stay online throughout class
Course description
Course Prerequisites

• Programming Languages:
  – Java/C++/Python/Perl/..

• Operating Systems: Basic Unix/linux

• CS 326 (Data structures) or equivalent
  – Lists, trees, queues, stacks, hash tables, …
  – Sorting, searching, dynamic programming,..

• Automata, regular expressions,…

• Stat 390 or 391 (Probability and statistics): random variables, conditional probability, Bayes’ rule, ….
Textbooks


- Used in both ling570 and ling571.

- Available at the Treehouse lab, UW Bookstore, Amazon, libraries, etc.
LING 473 (summer 2015)

- Probability (6): random variables, independence, probability distributions
- Algorithm (3): RegEx, complexity, edit distance, etc.
- Intro to Ling570 topics (4): FSM, grammar, LM, POS tagger, classifier, evaluation
- Misc (2)

⇒ see http://courses.washington.edu/ling473/
Topics covered in Ling570

- Unit #1: Formal language and FSA (2 weeks)
  - Formal language and formal grammar:
  - FSA
  - FST
  - Morphological analysis

- Unit #2: ngram models and HMM (2-3 weeks)
  - ngram LM and smoothing:
  - Part-of-speech (POS) tagging and HMM:
  - ngram tagger:
Topics covered in ling570 (cont)

• Unit #3: Classification (2-3 weeks)
  – Introduction to classification
  – POS tagging with classifiers
  – Chunking
  – Named-entity (NE) tagging

• Other topics (3 weeks)
  – Introduction, tokenization, probability theory
  – Clustering
  – Information extraction (IE)
  – Word embedding ??
  – Summary
NLP applications

• Applications of Speech and Language Processing
  – Call routing
  – Information retrieval
  – Question-answering
  – Machine translation
  – Dialog systems
  – Spam tagging
  – Spell- and Grammar-checking
  – Sentiment Analysis
  – Information extraction
  – ...

Shallow vs. Deep Processing

- Shallow processing (Ling 570)
  - Usually relies on surface forms (e.g., words)
    - Less elaborate linguistic representations
  - E.g. Part-of-speech tagging; Morphology; Chunking

- Deep processing (Ling 571)
  - Relies on more elaborate linguistic representations
    - Deep syntactic analysis (Parsing)
    - Rich spoken language understanding (NLU)
Language & Intelligence

• Turing Test: (1949) – Operationalize intelligence
  – Two contestants: human, computer
  – Judge: human
  – Test: Interact via text questions
  – Question: Can you tell which contestant is human?

• Crucially requires language use and understanding
Knowledge of Language

• What does HAL (of 2001, A Space Odyssey) need to know to converse?

  • *Dave: Open the pod bay doors, HAL.*
  • *HAL: I'm sorry, Dave. I'm afraid I can't do that.*

• Phonetics/Phonology (LING550) and speech recognition:
  – Sounds of a language, acoustics
  – Legal sound sequences in words
Knowledge of Language

• What does HAL (of 2001, A Space Odyssey) need to know to converse?

  • *Dave: Open the pod bay doors,* HAL.
  • *HAL: I'm sorry, Dave. I'm afraid I can't do that.*

• Tokenization and Morphology (Ling 570)
  – Recognize, produce variation in word forms
  – Contraction: I‘m → I am, can’t → can not
  – Singular vs. plural: doors → /door/ + PL V
  – Verb inflection: am → /be/ + 1st person, sg, present
Knowledge of Language

• What does HAL (of 2001, A Space Odyssey) need to know to converse?

• *Dave: Open the pod bay doors, HAL.*
• *HAL: I'm sorry, Dave. I'm afraid I can't do that.*

• Part-of-speech tagging (Ling 570)
  – Identify word use in sentence
  – “can” is a verb, not a noun.
Knowledge of Language

• What does HAL (of 2001, A Space Odyssey) need to know to converse?

  • *Dave*: Open the pod bay doors, HAL.
  • *HAL*: I'm sorry, Dave. I'm afraid I can't do that.

• Syntax
  – Syntactic structure: subject, verb, object, etc.
  – Ling 566: analysis
  – Ling 570: chunking
  – Ling 571: parsing
Knowledge of Language

• What does HAL (of 2001, A Space Odyssey) need to know to converse?

  • *Dave*: Open the pod bay doors, HAL.
  • *HAL*: I'm sorry, Dave. I'm afraid I can't do that.

• Semantics (Ling 571)
  – The meaning of sentences:
    • individual (lexical), combined (compositional)
    • ‘Open’: AGENT *cause* THEME to become *open*;
Knowledge of Language

- What does HAL (of 2001, A Space Odyssey) need to know to converse?
  - *Dave: Open the pod bay doors, HAL.* (request)
  - *HAL: I'm sorry, Dave. I'm afraid I can't do that.* (statement)

- Pragmatics/Discourse/Dialogue (Ling 571):
  - Interpret utterances in context
  - Speech act (request, statement)
  - Reference resolution: I = HAL; that = ‘open doors’
  - Politeness: I’m sorry, I’m afraid I can’t
Cross-cutting Themes

• Ambiguity
  – How can we select among alternative analyses?

• Evaluation
  – How well does this approach perform:
    • On a standard data set?
    • When incorporated into a full system (e.g., E2E)?

• Multi-linguality
  – Can we apply this approach to other languages?
  – How much do we have to modify it to do so?
Outline

• Course Overview
• Tokenization
• Homework #1
• Questionnaire
Tokenization

• Given input text, split into words or sentences

• Tokens: words, numbers, punctuation

• Example:
  – Input: Sherwood said reaction has been “very positive.”
  – Output: Sherwood said reaction has been “very positive.”

• Why tokenize?
  – Identify basic units for downstream processing
Tokenization

- Proposal 1: Split on whitespace

- Good enough? No

- Why not?
  - Multi-linguality:
    - Languages without white space delimiters: Chinese, Japanese
    - Agglutinative languages (Hungarian, Korean)
      - meggazdagíthatnok
    - Compounding nouns (German):
      - Lebensversicherungsgesellschaftsangestellter
        “Life insurance company employee”
  
  - Even with English, the approach does not handle punctuation properly.
Tokenization - again

- Proposal 2: Split on white space and punctuation
  - For English

- Good enough? No

- Problems: Non-splitting punctuation
  - 1.23 → 1 . 23 1,234,456 → 1 , 234 , 456
  - don’t → don ‘ t E-mail → E - mail

- Problems: no-splitting whitespace
  - Names: New York; Collocations: pick up

- What’s a word?
Tokenization in Chinese, Japanese, Thai, etc.

- No whitespace in written languages

- Baseline: maximum match
  - Use a dictionary
  - Cannot handle unknown words or ambiguity

- Current approach: treat it as a sequence labeling task
  - Input: c1c2c3c4c5c6
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.
Sentence Segmentation

• Proposal: Split on period, !, ?

• Problems?
  – Non-boundary periods:
    • 1.23
    • Mr. Sherword
    • U.S.A.
    • Ph.D.
    • Etc.

• Solutions?
  – Rules, dictionaries esp. abbreviations
  – Marked up text + machine learning

• What if the text is ASR output?
Homework #1
Patas

- All the files are under
  ~/dropbox/15-16/570/ on patas.

- For each hw subdir (e.g., hw1/):
  - examples/: example files. They are NOT gold standard.
  - solution/: will be provided afterwards
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/15-16/570/code-samples/

• Shell, perl, python, and binary code must be executable.

• Your code may be tested on new test data.
Tokenizer with arguments

Write “eng_tokenizer.pl file1 file2”

Option 1:
•  
  
  eng_tokenizer.sh
  #!/bin/sh
  Eng_tokenizer.pl $@
•  To run it: cat input | eng_tokenizer.sh file1 file2 > output

Option 2:
•  
  
  eng_tokenizer.sh
  #!/bin/sh
  eng_tokenizer.pl file1 file2
•  To run it: cat input | eng_tokenizer.sh > output
Q1-Q2: implementing a rule-based tokenizer for English

• Q1: write a tokenizer
  – 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 probability of winning 13 times out of 500 bets?
    • Binomial distribution
    • Poisson distribution
Questionnaire
• Part 1: prerequisites
  – Probability:
  – Formal grammars
  – FSA and FST
  – Unix
  – Linguistics

• Part 2: questions about some topics in ling570
  – Language model
  – HMM
  – POS tagging
  – ...
Part 3: 1st question

- Three rooms: one has gold; the other two do not.
- You choose one door, let’s call it door A.
- The host is required to open door B or C.
- He opens door B and there is no gold in it.
- Question: what is the probability of Room A having gold before and after the host opens door B?
• Semi-open book: You can check your notes or text books, but you should not google the Web or ask other people.

• Spend no more than 3 hours on Part I.

• Part III is optional.

• Submit your questions to CollectIt.

• Deadlines:
  – Submission deadline: 11pm Sat (10/3)
  – Feedback are available by 11pm Tues at CollectIt.
Coming up

• Go to course website to check whether you have access to CollectIt, GoPost, syllabus, etc.
  – If not, let me know by noon tomorrow.

• Your questionnaire answers are due at 11pm Sat.

• Next Tues:
  – Finish the quick review of probability theory.
  – Start on regular expression and FSA.

• Hw1 is due next Thurs at 11pm.