Introduction to Computational Linguistics
APLN550

Adam Meyers
Montclair State University
9/8/2014 and 9/15/2014
Outline

• Grades, Exams, Policies, etc.
• Text Books and Suggested Reading
• A Survey of the Students
• Defining the Field
• CL Applications
• Types of Text Analysis used in CL
• A Practice Manual Annotation Task
• Summary and Syllabus
• Homework No. 1
Grades, Homework, Exams, Final Projects

• Grade Breakdown:
  – 1/3 Homework + 1/3 Midterm Exam + 1/3 Final Project

• Homework
  – ~10 Homeworks
  – Submit Homework electronically (details TBA)

• Final Project
  – Sample Topics Available by Oct 15
  – Final Project Proposal Due Nov 15 (counts as 1 homework)
  – Final Written Version Due Dec 8
    • Graded during Exam Period (Dec 12—18)
  – 15-20 minute presentations Dec 1 and 8
  – Special Rules for Multi-person projects

• Overall Structure/Curriculum
  – Subject to Change in the First 2-3 weeks of class
Succeeding in This Class

• Don't be Afraid
  – Ask questions
  – Work out examples from reading on paper
  – Try out and modify NLTK programs
    • break them, read the error messages, fix your bugs, repeat

• Homework: The point of the homework is to work out stuff. If you have trouble, try to state clearly what you do not understand, so I know to go over it in class.
  – You can get credit for getting right answers or clearly stating problems and identifying source of your confusion (you could uncover a valid criticism)

• Midterm: I will give you a practice test, analyze this and ask questions based on it.

• Final project: Do in stages
  – Proposal
  – Baseline System
  – Final System
Policies

• Late Homework
  – Natural Consequence: You fall behind and have trouble keeping up, leading to lower mark on midterm exam and final project
  – If HW is time sensitive, (e.g., I need to release answer key before next class), late HW gets no credit
  – Else if HW handed in before end of term
    • late HW is graded late (at my discretion, see natural consequences)
    • No official score penalty

• Missing Homework
  – I grade top 9 out of 10 homeworks, so 9 As and 1 F is still an A average

• Intellectual Integrity (context dependent):
  – http://www.cs.nyu.edu/webapps/content/academic/undergrad/academic_integrity
  – http://www.montclair.edu/dean-of-students/student-conduct/academic-integrity/
  – Usually, you may discuss HW with anyone, but your work should be your own.
    • If it is a problem, you should be prepared to solve it on your own after you submit your answer
    • If it is creative, 2 students should not have the same answers
    • Special Cases, e.g., experiments, where we test to see if people get same answer independently
  – Midterm – no help, other than explaining instructions or fixing my errors on questions
  – Final Project
    • research = your own (but you can get “normal” advice)
    • other people can help with experiments, e.g., annotation
    • multi-person projects are OK as long as each person's contribution is clear enough for grading purposes
Contact Info

• Adam Meyers, Research Scientist/Research Assist Prof at NYU
  – 719 Broadway, Room 702
  – meyers@cs.nyu.edu
  – 212 998 3482
  – http://nlp.cs.nyu.edu/people/meyers.html
  – Office Hours by appointment

• Adam Meyers at Montclair
  – Conrad J. Schmitt Hall, Room 240B
  – Mondays 3:30-4:30

• CLASS Room: Conrad J. Schmitt Hall, Room 105

• Class Website: http://nlp.cs.nyu.edu/meyers/montclair-class/

• Class Mailing List: TBA
Text Books

• SPEECH and LANGUAGE PROCESSING 2nd Edition
  – By Daniel Jurafsky and James H. Martin
  – http://www.cs.colorado.edu/~martin/slp.html
  – Great Overview of the Field, explanations of techniques, algorithms, etc.

• Natural Language Processing with Python
  – By Steven Bird, Ewan Klein, and Edward Loper
  – http://www.nltk.org/book (look at the rest of the website also)
  – Book is available on line (or you can purchase it)
  – Downloadable open source programs to try out various computational linguistics tools and inspect their code
  – Written in Python 2
More Stuff to Read/Download, etc.

• Look at projects currently going on at NYU:
  – The Proteus website: http://nlp.cs.nyu.edu/
  – My website: http://nlp.cs.nyu.edu/people/meyers.html
    • GLARF: processing tool written in Common Lisp (for linux, but will soon be available for MAC)
    • NomBank: annotation project
    • COMLEX, NOMLEX: lexicon projects

• Other useful links (I will put more on a website):
  – http://cs.nyu.edu/courses/spring12/CSCI-GA.2590-001/
    • A similar class that I taught to CS grad students at NYU
  – http://aclweb.org/
  – http://www.cs.vassar.edu/sigann/
Technical Instructions for Montclair

• Connecting to the server
  – To be filled in
  – One advantage is that NLTK is installed
    • Alternative to installing it on your machine

• Other stuff:
  – ?????
Some Pointers for Installing NLTK on your own Machine

- **Linux**: NLTK is easy to install in *linux*
- **Apple**: it is possible, but to get all the bells and whistles, you may have to register as a developer and go through a little bit of pain
- **Windows**: There may be some limitations (last I checked), but most things relevant to this class will work.
  - I have not tested it, but Cygwin or AndLinux might be better for running NLTK
Purpose of Class Survey

• My current drafts of these lectures are based on a previous class taught in 2012
  – Students were all CS graduate students who were experienced programmers
  – Students had little or no linguistics background

• This class is different and I need to adjust my lectures and focus accordingly

• I believe that this class has more of a linguistics background and less of a CS background, but I would like to customize my lectures so they are a good fit.
Computer Background

• General Comfort Level
  – Installing software? Trying new applications?
  – Are you OK taking chances, letting things break and then reading the error messages and fixing things? Or will you only follow instructions?

• NonProgrammers
  – How many will have no programming experience other than what they learn in this class?
  – How many non-programmers are taking a programming class this semester?

• Programmers
  – How many people have written a computer program?
    • How many wrote a program of at least 100 lines of code?
  – How many have completed a programming class?

• UNIX experience?
  • Linux, Solaris, Using the Command line in Apple
  • Windows: Cygwin, Andlinux, ...
Linguistics Background

- Syntax:
  - Descriptive Linguistics, e.g., comprehensive grammar of English
  - Chomskyan Linguistics?
  - Non-Chomskyan Frameworks
    - LFG, HPSG, Categorial Grammar, Dependency Grammar, Systemic Grammar, Other
- Phonetics, Phonology
  - Acoustics, Articulatory, Phonetics, Phonology, Intonation
- Discourse, Pragmatics
- Psycho-Linguistics
- Lexicography
- Historical
- Any Other Area
Role of Linguistic Theory in Computational Linguistics

- Framework = Language for Expressing Theory
- Theory = Set of Statements in Framework
- Different Theories/Frameworks are typically designed with different interests/biases/etc.
  - Chomskian Linguistics: Meta Grammar for all languages, set of primitives,
- Computational Linguistics is Applied Field of Study
  - Theories/Frameworks are important to the extent to which they help make a successful application
  - Choice of theory/framework is secondary. Systems often designed to handle multiple theories/frameworks
    - It may simply depend on who writes the answer key
  - Descriptive Adequacy is more important than Explanatory Adequacy
- Frameworks that are popular in CL: Statistics-based Analysis (various), Dependency Grammar, Penn Treebank (based on 1980s Chomskian Linguistics), PropBank/Nombank (~ Relational Grammar), Frame Semantics (based on FrameNet), ...
- Only Broad Coverage Grammars are suitable, e.g., old theories with descriptive track records
- Proviso: there is a small niche within CL, in which researchers implement new theories
Defining Computational Linguistics

- AKA, Natural Language Processing (NLP), Language Engineering, ...

- Definition: Study of how to solve problems involving the interpretation and generation of human language text and speech

- Properties
  - As with applied science: the proof is in the pudding
  - Sometimes at odds with theoretical linguistics
    - Need not model human abilities and human methods
    - Need not correspond to published linguistic theories
    - But sometimes draws on one or both
  - Broad and changing domain influenced by available funding
CL Applications: Slide 1

• Machine translation
  – Methods are not at all based on how humans translate
  – Effective for gisting text, generating 1\textsuperscript{st} draft translations, but not for high-level translation
  – Works better for “controlled languages” – technical manuals (Microsoft, Catterpiller, etc.)

• Spoken Language
  – dictation (IBM ViaVoice, Dragon Naturally Speaking)
  – Telephone-based customer support (phone mazes)

• Information Retrieval (not like in the movie \textit{Brazil})
  – Web Searches (mostly statistics)
CL Applications Slide 2

• Information Extraction
  – Dealtime, Google Products, Monster.com (job search)
  – Some open source tools:
    https://opennlp.apache.org/
    http://alias-i.com/lingpipe/
  – NYU
    • Some tools on website
    • Example from disease domain
      http://nlp.cs.nyu.edu/info-extr/biomedical-snapshot.jpg

• Question Answering

• Summarization: http://newsblaster.cs.columbia.edu/

• Spelling/Grammar Checking, etc. https://languagetool.org/
Types of Analysis

• Phonetics/Phonology: speech recognition and speech synthesis (not in this class)
  – We will focus on text analysis
  – Text does not represent some phonological features
  – Text has punctuation

• Syntactic/Semantic: sentence splitting, tokenization, pos tagging, chunking, parsing, predicate/argument structure, sense disambiguation

• Discourse: anaphora, discourse argument structure, sentiment analysis

• Other: multi-lingual processing (including MT), summarization, IE, etc.
Lowest Level Syntactic Processing (text)

- Tokenization and Segmentation
  - Given a sentence, determine the words or word-like units that it consists of:
    - *They announced in unison, “We don't agree with each other.”*
    - Tokenization: *They | announced | in | unison | , | “| We | do | n't | agree | with | each | other | . |”*
      - Controversial parts: *n't, each other*
      - NLTK command: *nltk.word_tokenize(“this is a sentence”)*

- Part of Speech Tagging (modified PTB)
  - Apply a set of part of speech tags to a set of tokens
    - *They/PRP announced/VBD in/IN unison/NN ,/PU “/PU We/PRP do/VBP n't/RB agree/VB with/IN each/DT other/JJ ./PU ”/PU*
  - NLTK command: *nltk.pos_tag(tokens)*
Low Level Syntactic Processing

- Named Entity Tagging (with a little semantics)
  - Mark boundaries of names of type PERSON, ORGANIZATION, FACILITY, GPE, LOCATION, …
  - `<ENAMEX TYPE="PERSON"> Adam Meyers</ENAMEX> works for <ENAMEX TYPE="ORGANIZATION">New York University</ENAMEX>`
  - NLTK command: `nltk.chunk.ne_chunk(pos_tags)`

- Chunking -
  - mark verb groups and/or noun groups, convenient approximations of syntactic units (questionable theoretically).
  - `[NG The book] with [NG the blue cover] [VG was falling off] [NG the shelf].`
  - NLTK:
    - `sentence = 'The book with the blue cover was falling off the shelf.'`
    - `chunks = r"""
      NG: {(<DT|JJ|NN>)*(<NN|NNS>)}
      VG: {<MD|VB|VBD|VBN|VBZ|VBP|VBG>*<VB|VBD|VBN|VBZ|VBP|VBG><RP>?}
      """
    - `chucks_grammar = nltk.RegexpParser(chunks)`
    - `chucks_grammar.parse(nltk.pos_tag(nltk.word_tokenize(sentence)))`
Parsing

• (S (NP (DT the) (NN book))
  (VP (VBZ is)
   (PP (IN on)
    (NP (DT the) (NN shelf))))

Computational Linguistics Lecture 1
2014
Predicate/Argument Structure

- For thousands of years, linguists have employed systems to characterize predictable paraphrases, e.g., Pāṇini, a Sanskrit linguist from the 4th Century BC
- In 21st Century CL, semantic role labeling is popular
Sense Disambiguation

• For interesting characterizations of word senses (and relation between senses), use WordNet (online or download it)
  – wordnet.princeton.edu/

• 2 obviously distinct senses of *bank*
  – *They took money out of the bank.*
  – *The water flooded over the bank of the river.*

• Difficult sense disambiguation
  – Senses 2, 3 and 5 on the next slide are arguably not distinct
  – Lexicographers are acutely aware of the merging vs. splitting problem of enumerating senses
  – CL systems usually collapse some WordNet distinctions
WordNet Noun entry for *table*

1. (52) *table*, *tabular array* -- (a set of data arranged in rows and columns; "see table 1")

2. (25) *table* -- (a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs; "it was a sturdy table")

3. (5) *table* -- (a piece of furniture with tableware for a meal laid out on it; "I reserved a table at my favorite restaurant")

4. *mesa*, *table* -- (flat tableland with steep edges; "the tribe was relatively safe on the mesa but they had to descend into the valley for water")

5. *table* -- (a company of people assembled at a table for a meal or game; "he entertained the whole table with his witty remarks")

6. *board*, *table* -- (food or meals in general; "she sets a fine table"; "room and board")
Anaphora

• Coreference
  – Though **Big Blue** won the contract, this official is suspicious of **IBM**.
  – **Mary** could not believe what **she** heard.

• Other Varieties
  – John ate a **sandwich** and Mary ate **one** also.
  – **The amusement park** is very dangerous. **The gate** has sharp edges. **The rides** have not been inspected for years.
  – **This book** is valuable, but **the other book** is not.
Discourse Argument Structure

- Adverbs, Subordinate/Coordinate Conjunctions, among other words link clauses
Role of Manual Annotation

• Used to create, test and fine-tune task definitions/guidelines.
  – For a task to be well-defined, several annotators must agree on classification most of the time.
  – If humans cannot agree, it is unlikely that a computer can do the task at all
  – Popular, but imperfect measurement of agreement:
    • \( \text{Kappa} = \frac{\text{Percent (Actual Agreement)} - \text{Prob (Chance Agreement)}}{1 - \text{Prob (Chance Agreement)}} \)

• Used to create answer keys to score system output
  – One set of measures are: recall, precision and f-score

  – \( \text{Recall} = \frac{|\text{Correct}|}{|\text{Answer Key}|} \)
  – \( \text{Precision} = \frac{|\text{Correct}|}{|\text{System Output}|} \)
  – \( \text{F-Score} = \frac{1}{\frac{1}{2} \left( \frac{1}{\text{Precision}} + \frac{1}{\text{Recall}} \right)} \)
Role of Manual Annotation in Supervised Statistical CL

- Divide the corpus into sub-corpora
  - A training corpus is used to acquire statistical patterns
  - A test corpus is used to measure system performance
  - A development corpus is similar to a test corpus
    - Systems are “tuned” to get better results on the dev corpus
    - Test corpora are used infrequently and system should not be tuned to get better results

- More annotated text often yield better results

- Different genres may have different properties
  - Systems can “train” separately on different genres
  - Systems can “train” on one diverse corpus
A Sample Annotation Task

• Hypothesis: changes in attributes may provide a useful way to summarize biological/medical documents

• Annotation (All items must be in the same sentence):
  – 1 signal indicating increase/decrease/change
  – 1 Noun plus adjective/noun left modifiers indicating the attribute that changed
    • Why no determiner?
    • Why no right modifiers?
  – Optional Arguments (Always mark if found)
    • Degree (how much it changed)
    • Cause
    • THEME (possessor of the attribute)
Sample Annotation

• The 23 patients who completed 4 or more weeks of medication showed significant improvement on all depression scales and in quality of life.

<table>
<thead>
<tr>
<th>Change</th>
<th>Degree</th>
<th>Attribute</th>
<th>Theme</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>improvement</td>
<td>significant</td>
<td>depression scales</td>
<td>23 patients</td>
<td>medication</td>
</tr>
<tr>
<td>improvement</td>
<td>significant</td>
<td>quality</td>
<td>23 patients</td>
<td>medication</td>
</tr>
</tbody>
</table>
Observation about specification and example

• “no right modifiers” for Attribute
  – “23 patients who completed 4 or more weeks of medication” vs “23 patients”
  – “quality of life” vs “quality”

• Why might we want to modify the specifications to include right modifiers?
• Why might we not want to do so?
• What are some alternatives?
Annotation Experiment

• I am handing out 2 short sample biology and/or medical texts
• Please take a few minutes to annotate all instances of changing attributes that you can find.
• I will compare the results and discuss some annotation specification issues
Summary

• Computational Linguistics is an applied discipline with an increasingly large inventory of applications.

• A wide variety of levels of analysis are used to implement these applications.
  – Many, but not all of these levels are derived from or inspired by theoretical linguistics

• One popular paradigm for producing an analysis automatically involves manually annotating text
Syllabus of 2012 NYU Class

• Introduction (today)
• Formal Languages and Transducers
• Natural Language Syntax and Parsing
• POS Tagging and Hidden Markov Models
• Named Entities and Machine Learning
• Lexical Semantics and Semantic Role Labeling
• Information Extraction: Entities, Relations, Events, Time
• Anaphora: Coreference and Similar Phenomena
• Feature Structures and Representing Multiple Phenomena
• Machine Translation (Not Sure)
Why Modify Syllabus?

• Montclair Classes are Longer
  – At NYU, I had less than 1 hour 50 minutes/class
  – 2.5 Hours is enough time to have
    • 1 lecture
    • 1 co-operative project (program, annotation, etc.)

• Linguistics Students
  – The Descriptive Aspects of Comp Ling May be of more interest to Linguistics Students
  – Less CS background can be assumed, so more explanation may be necessary

• CS students
  – The formal and statistical aspects of Comp Ling may have been of more interest to CS students
  – Less Linguistics background was assumed, so more explanation may have been necessary
How will I Modify the Syllabus

• More Focus on Language Resource Creation
  – Corpus Collection, Corpus Annotation
  – Creation of Lexicons & Lexical resources like WordNet

• More Focus on Linguistics in NLP
  – Feature Structures, Dependencies, Syntactic/Semantic Annotation Frameworks, etc.

• Less of a Focus on Statistical Techniques
  – Possibly drop Statistical Models for Machine Translation

• More Discussion of Implementation Issues
  – Writing programs, etc.

• Questions:
  – Can I rely on students teaching themselves to program?
  – Could programming literacy be an obstacle?
Homework and Readings

• Jurafsky and Martin, Chapter 1
• NLTK Book – Install NLTK, read Chapter 1 and follow along with their examples.
• Do the following Annotation Task
  – Download about 200 words of text from Wikipedia or News (save as .txt file)
  – In a text editor, add “/JJ” after each adjective
    • Text editors = emacs, vi, ex, notepad, WordPad, ...
      – Windows Users: NotePad is bad for reading non-Windows txt files
• Initially use these Specifications for identifying adjectives
  – An adjective must be able to fill in the blanks
    • The ___ noun ... --- occur between “the” and a noun
    • The noun is ____ . --- a single word following “the” + noun and
      preceding the end of sentence marker (period)
  – An adjective CANNOT (comfortably) fill the following blank
    • The ___ is --- be the subject of “is”
    • ___s --- become a plural
• Modify the specifications to produce a better result. Better = (a) corresponds to some external definition of adjective (e.g., a list of adjectives); and (b) can be applied consistently to the text. Explain your modifications and produce new annotation based on your specifications.
Optional Independent Work

• Get ahead in NLTK book
  – Goal1: install/run all NLTK programs
  – Goal2: look at source code, copy files and edit copies: add slight modifications and run

• Teach Yourself Python?
  – Programs and Manuals: http://python.org/
  – Free Online Text books
    • http://openbookproject.net/thinkcs/python/english2e/index.html
    • http://gnosis.cx/TPiP/
  – Other Textbooks
    • Starting Out with Python by Tony Gaddis (NYU uses)