Ling/CSE 472:
Introduction to Computational Linguistics

3/31/20

Introduction, overview

Notes on privacy and Zoom

- This class is set to “auto-record” lectures.
    - That means every time someone visits the Zoom room, I get an email saying that was recorded. I will not ever post links to these spurious recordings.
    - I will post links to the recordings from the actual lectures; Olga will post links from section.
- Zoom saves the chat window including so-called “Private” chats and sends this to me as the instructor. If we make heavy use of the chat window for asking questions, I will plan to post these chat transcripts, without editing them.

Who’s here?

- A good class to work together --- everyone brings different skills
- I’m going to bring a lot to this class because...
- This is going to stretch me because...

Overview

- What is Computational Linguistics
- Who’s here
- Syllabus
- Showing the computer who’s boss

What is Computational Linguistics?

- Getting computers to deal with human languages
  - ... for practical applications (examples?)
  - ... for linguistic research (examples?)

Linguistic research

- Searching large corpora for patterns of use and linguistic examples
- Creating structured databases of information for typological research (Autotyp, ODIN)
- Creating ontologies for interoperable markup of linguistic resources (GOLD)
- Modeling human linguistic competence and performance (computational psycholinguistics, grammar engineering)
- Software to facilitate language documentation (Elan, FIELD, SIL FieldWorks, Grammar Matrix, AGGREGATION, EL-STEC)

Practical applications

- Speech recognition
- Speech synthesis
- Machine translation
- Information retrieval
- Natural language interfaces to computers
- Dialogue systems

Practical applications

- Computer-assisted language learning (CALL)
- Grammar checkers
- Spell checkers
- OCR (optical character recognition)
- Handwriting recognition
- Augmentative and assistive communication

Practical applications

- BioMedical NLP: Matching patients to clinical trials
- BioMedical NLP: Flagging electronic health records for urgent tests
- BioMedical NLP: Assistance in coding for insurance billing
- BioMedical NLP: Searching the biomedical literature for untested but promising things to study
- Legal domain: Electronic discovery

Practical applications

- B2B: Sentiment analysis for brand tracking
- Context-aware advertising
- Intelligence/national security: Monitoring social media, news, intercepted email/voice traffic
- ...

End-to-end applications are constructed from components that handle subtasks

- Each subtask has input and output
- Each subtask can be evaluated
  - often: precision, recall
  - intrinsic and extrinsic evaluation
- Output from one subtask is input to the next (in pipeline models)
- Many subtasks have “analysis” and “generation” variants
- Examples of subtasks?

Subtasks
(What's the input? What's the output?)

- Part of Speech tagging
- Named Entity Recognition
- Lemmatization
- Morphological analysis
- Parsing (constituent structure, dependency structure)
- Coreference resolution
- Word sense disambiguation
- Event detection
- Dialog act labeling
- Language modeling
- Alignment (of bitexts)
- ...

Statistical v. symbolic methods

- Statistical methods involve *training a stochastic model* on a body of data so it can predict the most probable label/structure/etc for new data
    - Knowledge comes from implicit patterns in naturally occurring language (unsupervised learning) or from hand-labeled data (supervised learning)
- Symbolic methods involve *knowledge engineering*, or hand-coding of linguistic knowledge which is then applied to tasks
- Statistical methods provide *robustness*, symbolic methods *precision*
- Statistical and symbolic methods can be combined

The World of CL: CL at UW

- Linguistics (CLMS)
- CSE
- ECE
- Biomedical informatics
- iSchool

The World of CL: CL in Seattle

- Microsoft (MSR)
- Amazon
- AI2
- Facebook
- Google
- ...

World of CL: ACL

- Association for Computational Linguistics; our chapter: NAACL
- Conferences: ACL, NAACL, EACL, IJCNLP, EMNLP, others
- Workshops
- Publications
  - *Computational Linguistics* and *TACL* journals
  - Conference and workshop proceedings: ACL Anthology https://www.aclweb.org/anthology/

World of CL: online communication

- @UW: cl-announce http://mailman.u.washington.edu/mailman/listinfo/cl-announce
- International: corpora http://mailman.uib.no/listinfo/corpora
- Twitter: #NLProc, conference hashtags

Learning Outcomes

- Be familiar with computational linguistic topics, tools, and resources, and how they are applied in research in both computational linguistics and other subfields

- Be able to conceptualize problems from the perspective of computational linguistics

- Be able to design and carry out a linguistically-informed error analysis of an NLP system

- Understand ways in which linguistic knowledge can be computationally encoded, to test linguistic hypotheses and strengthen NLP systems

- Be an informed conumser of NLP/speech technology and popular press reporting on NLP/speech technology

Why this class is weird

- Upper-division survey course
- Students with diverse backgrounds
- So why teach this as one cross-listed course?

Syllabus

- Web page: http://courses.washington.edu/ling472
    - NB: Things are due already this week (RQ, Assignment 0)
- Slides will be posted (often before lecture)
- Using Canvas (http://uw.instructure.com) and Zoom (recordings will be included on Canvas page)
- Lab meetings (Fridays)

Course requirements

- Homework assignments (5 total, turned in via Canvas): 60%
  - Coding *and* writing: Writing will be 50% of the grade
- Final project: 30%
- Reading questions: 5%
- Blog assignment: 5%
- Up to 2% adjustment for:
  - Extra credit points for original clarification questions
  - In class participation
    - Other on-line participation
- Get set up: see course web page for server cluster accounts, lab access, reading assignments, etc.

Reading Questions: http://courses.washington.edu/ling472/rq.html

---

Background

The lectures for this class will assume that students have read the chapter before class. In order to reinforce that and in order to make the lectures more effective in deepening student understanding of the material, we have instituted a "reading questions" requirement.

Assignment

For each lecture with an assigned reading (see the schedule of topics and assignments), respond to the following prompt on Canvas:

> What in the reading was most confusing? If you can, articulate a question about it. If nothing was confusing, what further questions does this reading raise for you?

There are 16 lectures with associated reading assignments, so each student should submit 16 reading questions. These will collectively count for 5% of your course grade.

You are welcome to also answer questions from your peers, in addition to posting your own question.

Procedure

Submit RQ to the appropriate Canvas Discussion area by **11:59 pm the night before class**.

Note: We're asking you to post the reading questions because we think it will be of interest for you to see each others' questions. However, we do not expect you to go through the other reading questions before posting your own. Repeats/highly similar questions are expected and welcomed (but questions should arise from reading --- just paraphrasing something doesn't count ;-).

Blog Assignment: http://courses.washington.edu/ling472/blog.html

---

Background

Computational lingusitics is a rapidly developing field with a large (and ever-growing) literature. This is evidenced by the sheer size of the primary textbook we're using, which is way to large to cover in a one quarter class. In addition, the students in class bring a range of diverse expertise. The purpose of this assignment is to ask you to share that expertise with each other, so everyone can get a broader sense of the field than we would just reading on our own, and to practice writing accessible summarize of technical documents.

Assignment

This assignment has two parts:

1. Each student will pick ONE "blog options" reading for the whole quarter. By the date that reading is associated with, you will write a 300-500 word post on Canvas (in the "blog" discussion area for that day) summarizing the reading for your classmates. Ideally, these blog posts will convey the main point of the reading, plus a few choice details that you found particularly interesting. You can also add your own perspective, especially if you are skeptical of claims in the reading and/or excited about the findings/ideas therein.
2. Every student not blogging for a particular day is asked to respond to at least one blog post on that day, with a short but contentful comment. Our hope is that these blog responses will turn into discussions. Discussion comments are due by the following lecture.

Blog Assignment: http://courses.washington.edu/ling472/blog.html

---

Procedure

1. By Friday April 3, everyone should claim the reading they are going to be blogging about. Blog options are listed on the course web page, but you are also welcome to suggest your own reading (even something you have already read!) that is relevant to the topic of one of the lectures.
    - Claim your article by editing the page "Blog assignments" on the course Canvas. (You may change to a different article relevant to the same lecture up to the point that you write your post or switch places by mutual agreement with a classmate, but I want to have an initial schedule in place by April 3, if possible.)
2. By class on the day the reading is associated with, post your blog post to the Canvas discussion area for that date.
3. Every student not blogging for a particular day should post a reaction on the blogs for that lecture by following lecture, in the Cavnas discussion. These reactions can be comments or questions that engage with the content of the blog post or respond to previously posted comments or questions.

Blog posts and discussion comments will be graded on completion, but we are prepared to award extra credit for particularly outstanding blog posts.

Term Project: http://courses.washington.edu/ling472/final_project.html

---

General remarks about the project

- Project components:
  1. find an NLP package described in a peer-reviewed paper which reports results using a quantitative evaluation metric, such as precision/recall;
  2. run it on the dataset that comes with it (or, in some exceptional cases, on a different dataset),
  3. and perform careful error analysis over the results (for ~100 errors + some number of correct outputs for comparison)

- Additional specifications:
  1. All papers (packages) must include quantitative evaluation (e.g. in terms of precision and recall).
  2. Find a package that you can get running easily and quickly.
  3. Perform an analysis of the specific test items that the system does not get right by categorizing the errors, counting how many fall into which category, and finally providing a meaningful discussion about them, including either hypothetical or directly observed reasons for why the system is making these errors, as well as what implications this behavior might have.
  4. Your error analysis: its method, categories, and results, will be described in detail in the **term paper**.
  5. The term paper quality must be representative of the work that you did in this upper division course.

- Further comments: The error analysis is the main point of the project, not reproducing the experiment, so beware of choosing a package which you will then waste a lot of time trying to run. Be especially careful about choosing a package which does not come with a dataset. Chances are you will not even get to error analysis this way, spending all your time making the tool run on new data. Also, we will not approve such a package!

Term Project: Milestones

- 4/15: Form project groups (2-3 people, with complementary expertise)
- 4/24: Milestone 1 - proposals of three possible packages w/datasets
- 5/15: Milestone 2 - complete project plan, 1st draft
- 5/29: Milestone 3 - complete project plan, revised
- 6/2: Milestone 4 - in-class presentation of completed error analysis
- 6/11: Milestone 5 - term paper (project write up)

Course requirements

- Homework assignments (5 total, turned in via Canvas): 60%
  - Coding *and* writing: Writing will be 50% of the grade
- Final project: 30%
- Reading questions: 5%
- Blog assignment: 5%
- Up to 2% adjustment for:
  - Extra credit points for original clarification questions
  - In class participation
    - Other on-line participation
- Get set up: see course web page for server cluster accounts, lab access, reading assignments, etc.

Policies

Students are expected to complete the assigned readings before each lecture. Lecture and Lab/Section will connect with the readings, but not everything in the readings will be covered in lecture. Homework assignments and exams may nonetheless cover material in the readings not gone over in class.

All homework assignments and the final project will include a significant writing component, weight at or near 1/2 of the assignment grade. Be sure to save time to do a careful job on your write up.

We expect all write ups to be turned in as pdf files, even if they started as plain text files that we gave you.

Policies

Collaboration policy: Students are encouraged to work with each other on the homework, both in small groups and by posting & answering questions on Canvas. However, each student must turn in their own answers (both code and write up). No copying or sharing code or prose is allowed. Also, students who have collaborated must acknowledge the collaboration in their write ups (e.g. "I discussed this problem with Kim Smith/with classmates on Canvas as we were working on it.").

Plagiarism policy: Plagiarism is strictly forbidden. The offender will get 0 points for the plagiarized assignment and will be reported to the University. NB: It is very easy to detect not only plagiarized text but also a (piece of a) program, or even a mathematical solution that was adapted from something posted on the internet. Just don't. Submit your own solution, and rest assured, it will be unique!

Late homework policy: Unless prior arrangements are made, homework turned in late but within 24 hours of the deadline will be graded at 80% credit, homework turned between 24 and 48 hours will be graded at 70% credit, and homework turned in later than that will not be graded. No late final projects or reading questions will be accepted.

Letting the computer know who's boss

- Computer 'literacy' is really a combination of experience and attitude
- Experience gives you the answers to many questions and a sense of what the possible space of answers is
- The important attitude boils down to confidence in one's ability to find the answer to a new question
- There are always new questions because:
  - The technology is always developing
  - There is too much for any one person to know it all

Letting the computer know who's boss

- Keep in mind:
  - It's always obvious once you know the answer
  - All pieces of software were designed by some person or people with some functionality in mind
- Places to look for answers:
  - On-line documentation (man, info, help)
  - Product websites (esp. discussion forums)
  - Google: websites, and especially newsgroups
  - Off-line documentation (i.e., books!)
  - Work together!
    - ... and post to the discussion boards in Canvas
  - 10 minute rule
    - It's ~~okay~~ critically important to ask questions!

Questions?

Overview

- What is Computational Linguistics
- Syllabus
- Who’s here
- Showing the computer who’s boss