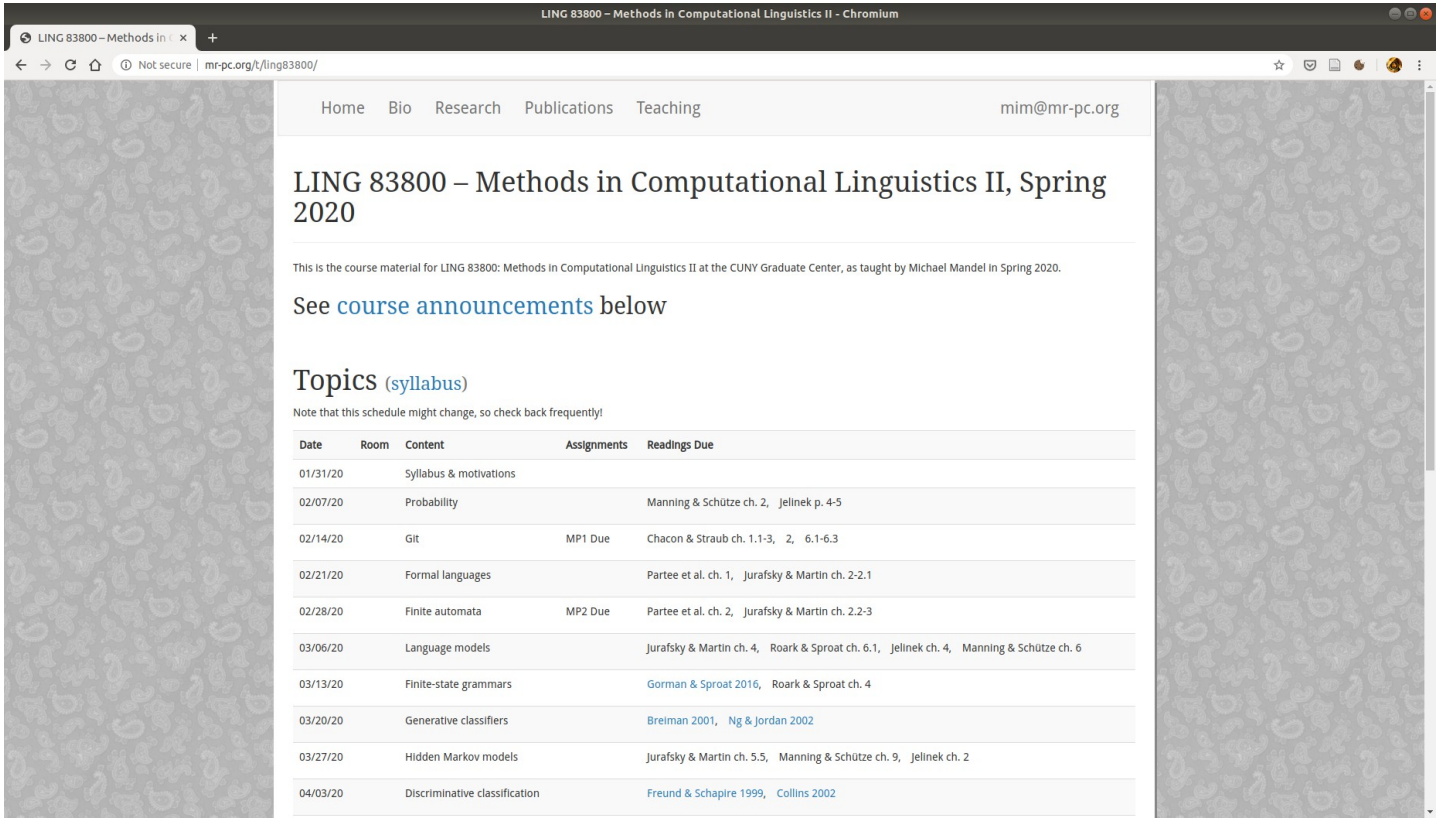
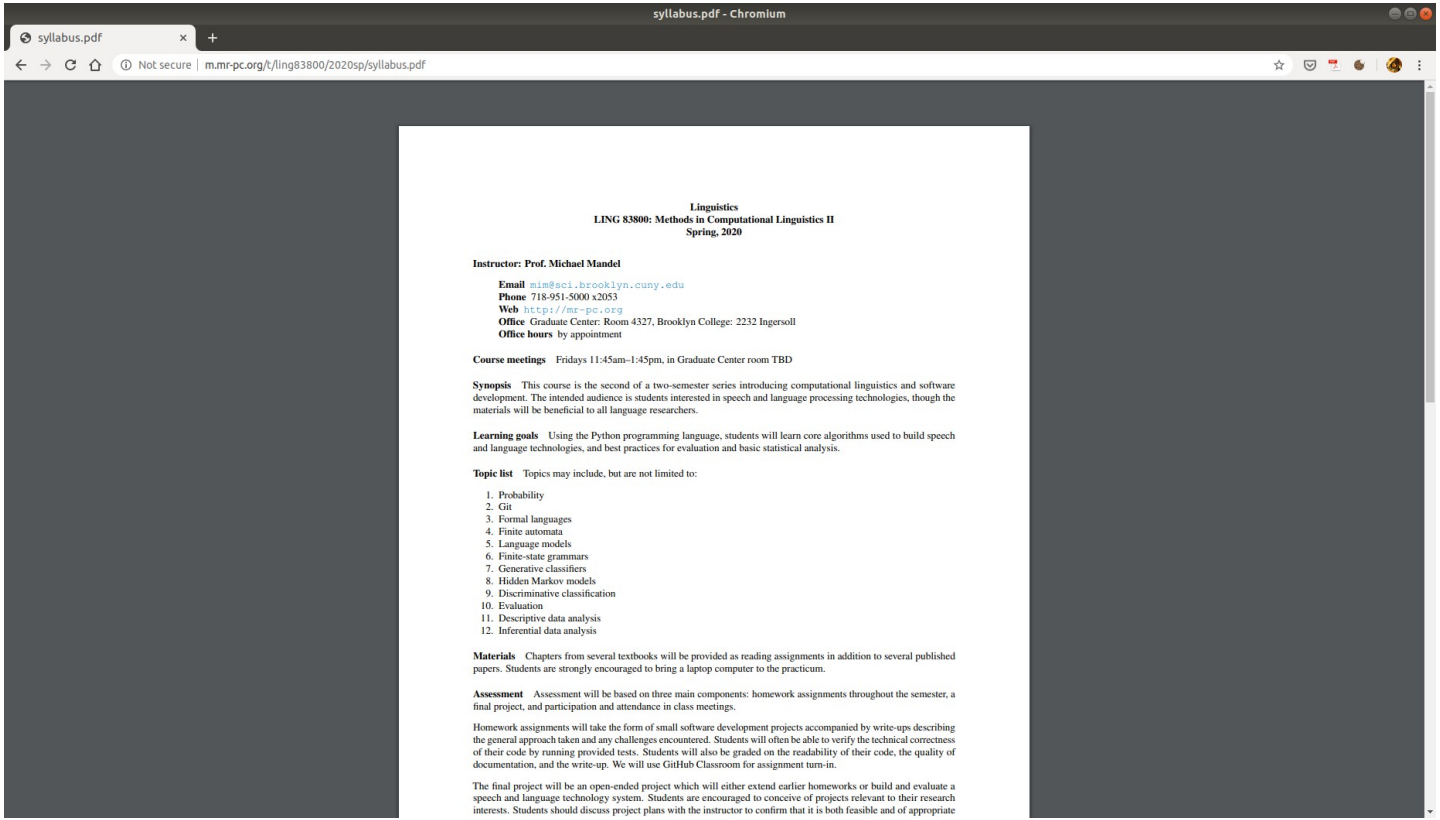
Intro: Methods in Computational Linguistics II

Michael Mandel

Course website



Syllabus



Linguistics
LING 83800: Methods in Computational Linguistics II
Spring, 2020

Instructor: Prof. Michael Mandel

Email mim@sci.brooklyn.cuny.edu
Phone 718-951-5000 x2053
Web http://mr-pc.org
Office Graduate Center: Room 4327, Brooklyn College: 2232 Ingersoll
Office hours by appointment

Course meetings Fridays 11:45am–1:45pm, in Graduate Center room TBD

Synopsis This course is the second of a two-semester series introducing computational linguistics and software development. The intended audience is students interested in speech and language processing technologies, though the materials will be beneficial to all language researchers.

Learning goals Using the Python programming language, students will learn core algorithms used to build speech and language technologies, and best practices for evaluation and basic statistical analysis.

Topic list Topics may include, but are not limited to:

1. Probability
2. Git
3. Formal languages
4. Finite automata
5. Language models
6. Finite-state grammars
7. Generative classifiers
8. Hidden Markov models
9. Discriminative classification
10. Evaluation
11. Descriptive data analysis
12. Inferential data analysis

Materials Chapters from several textbooks will be provided as reading assignments in addition to several published papers. Students are strongly encouraged to bring a laptop computer to the practicum.

Assessment Assessment will be based on three main components: homework assignments throughout the semester, a final project, and participation and attendance in class meetings.

Homework assignments will take the form of small software development projects accompanied by write-ups describing the general approach taken and any challenges encountered. Students will often be able to verify the technical correctness of their code by running provided tests. Students will also be graded on the readability of their code, the quality of documentation, and the write-up. We will use GitHub Classroom for assignment turn-in.

The final project will be an open-ended project which will either extend earlier homeworks or build and evaluate a speech and language technology system. Students are encouraged to conceive of projects relevant to their research interests. Students should discuss project plans with the instructor to confirm that it is both feasible and of appropriate

Practicum

- The course has a practicum (lab)
- It meets once a week for 2 hours
- Led by PhD student Arundhati Sengupta <asengupta2@gradcenter.cuny.edu>
- Let’s find a time that works for everyone

Introductions

- Name
- Hometown
- Background in linguistics and computer science
- Aspect of or problem in CL you are most interested in

Overview

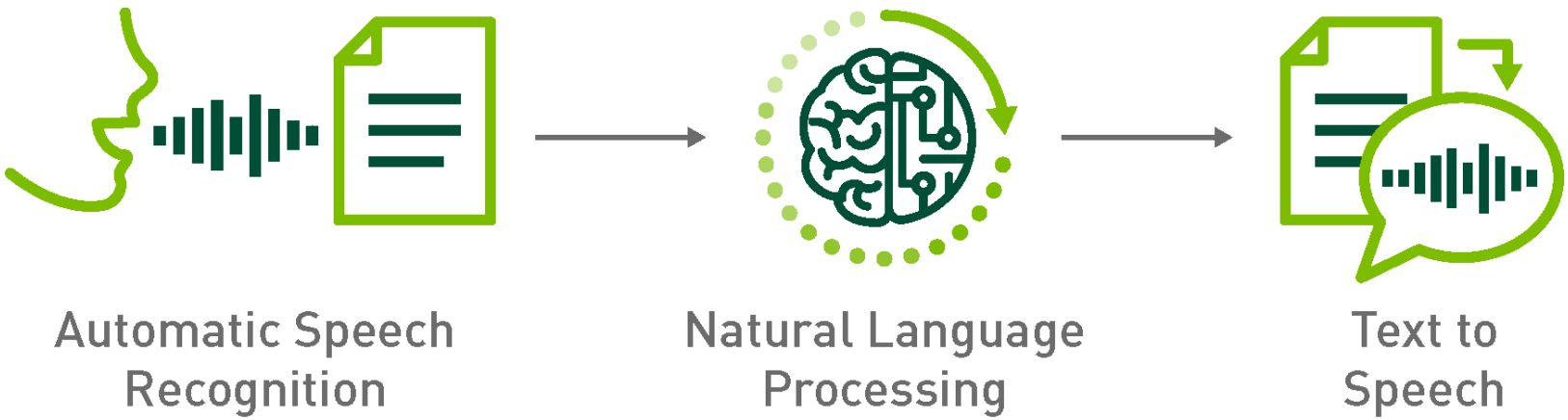
Computational Linguistics: The field

- Speech & language technology
  - *Speech technology*: technologies that work on audio
  - *Natural language processing* (NLP): technologies that work on text
- Formal models of human language
  - What type of "language" are human languages?
- Computational cognitive modeling

Speech & language technology

Speech technology

- *(Automatic) speech recognition* (ASR): audio to text
- *(Text to) speech synthesis* (TTS): text to audio



https://devblogs.nvidia.com/how-to-build-domain-specific-automatic-speech-recognition-models-on-gpus/

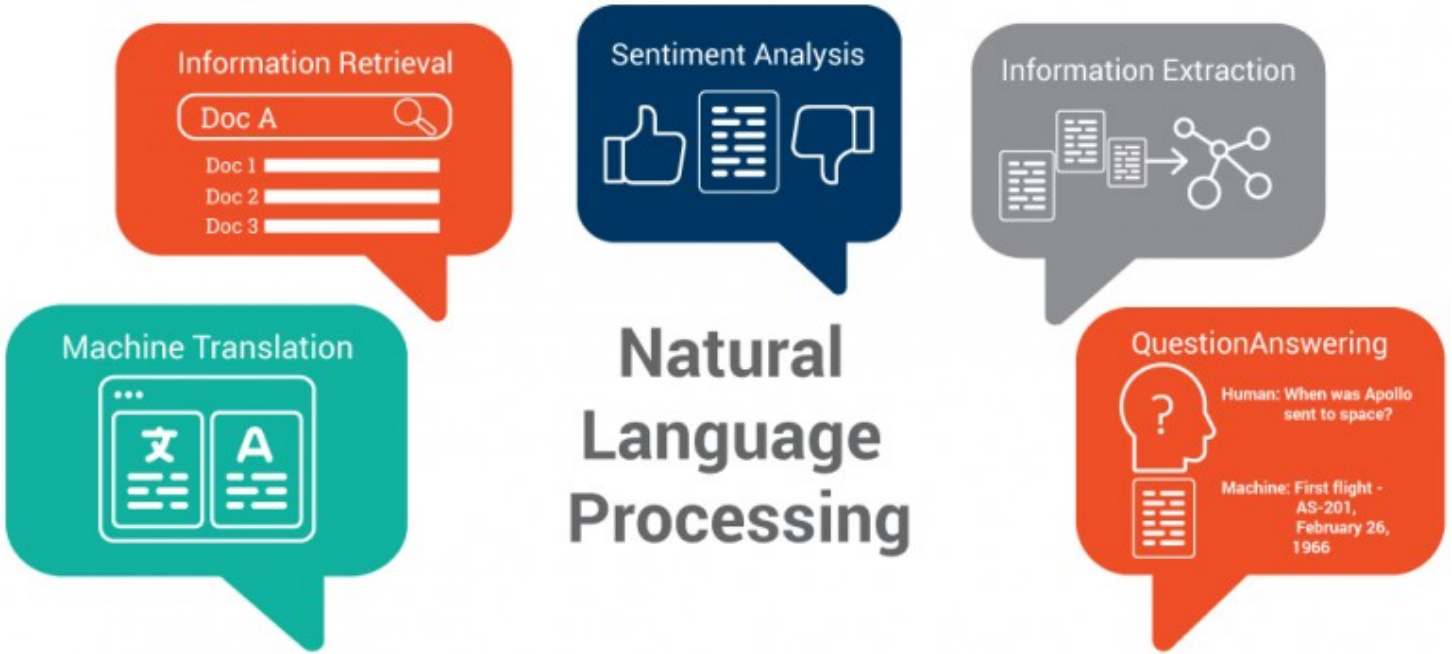
Natural language processing

Takes text as input, adds "annotations" or "labels" at various degrees of granularity.

https://www.kdnuggets.com/2018/10/main-approaches-natural-language-processing-tasks.html

Natural language processing

Takes text as input, adds "annotations" or "labels" at various degrees of granularity.



https://www.kdnuggets.com/2018/10/main-approaches-natural-language-processing-tasks.html

Machine learning

Machine learning (ML) is a theory of learning *decision functions* which *classify* (or assign labels or actions to) incoming data.

For this to allow machines to learn without explicit instruction, these must *generalize* appropriately to unseen data.

Machine learning in NLP

NLP without machine learning has been tried, and has been found wanting.

Human language is incredibly (temporarily and globally) ambiguous, and the machines locate structural ambiguities that humans do not normally notice.

Linguistic representations and resources act as a hypothesis space for ML.

Structural ambiguity: prepositional phrase attachment

Pope Francis on Saturday appointed a victim of sexual abuse and a senior cardinal known for his zero-tolerance approach to a new group charged with advising the Catholic Church on how to respond to the problem of sexual abuse of children. (Wall St. Journal, 2014-03-22)

- The prepositional phrase *on Saturday* is construed as a modifier of *Pope Francis* rather than of *appointed*.
- The phrase *to a new group charged with advising the Catholic Church on how to respond to the problem of sexual abuse of children* is construed as a modifier of *zero tolerance approach* rather than of *appointed*.

http://languagelog.ldc.upenn.edu/nll/?p=11286

Simple classification

- Tokenization:
  - *Sentence boundary detection* (or *sentence tokenization*)
  - *Word tokenization*
- Text classification:
  - *Document classification*: is it news or sports?
  - *Sentiment analysis*: is this a positive or negative movie review?
- Word classification:
  - *Word sense disambiguation*: does this instance of *bank* refer to a financial institution or the edge of a body of water?
  - *Homograph disambiguation*: should this instance of *bass* be pronounced as [beɪs] or [bæs]?

Structured prediction

In *simple classification* we have an observation *x*, a decision function *D* and we predict a label $\hat{y}$ so that:

$$\hat{y} = D(x)$$

However for NLP tasks, our predictions are not strictly independent of other nearby predictions. For instance, in a language like English, where determiners precede nominal complements, if the previous word is a determiner it makes the next word more likely to be a noun.

This setting is known as *structured prediction*, and it requires us to marry machine learning and custom *search algorithms*.

Sequence labeling tasks

- *Part-of-speech tagging*:

  PC/NNP Music/NNP is/VBZ a/DT record/NN label/NN and/CC art/NN collective/NN based/VBN in/IN London/NNP and/CC run/VBN by/IN producer/NN A./NNP G./NNP Cook/NNP ./

- *Named entity recognition*:

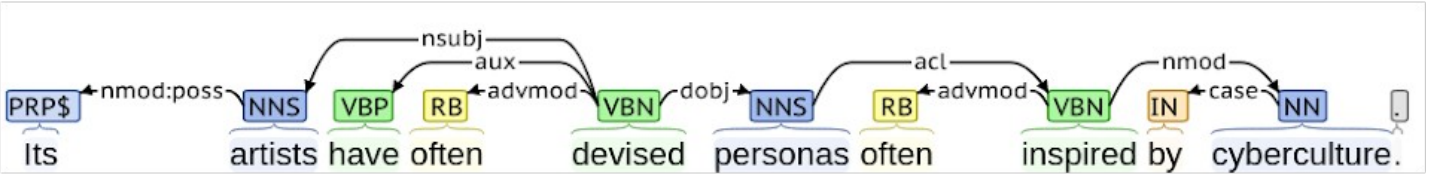
  Artists on its roster include [Hannah Diamond]$_{per}$, [GFOTY]$_{per}$, [Life Sim]$_{org}$, and [Danny L Harle]$_{per}$.

Parsing tasks

- *Coreference resolution*:

  [The label]$_i$ is known for its$_i$ surreal or exaggerated take on pop music, often featuring pitch-shifted, feminine vocals and bright, synthetic textures

- *Dependency parsing*:



http://nlp.stanford.edu:8080/corenlp/

Evaluation

How do we compare (machine learned) speech & NLP models on real data?

Computational formal linguistics

Logical Structure of Linguistic Theory (I)

> *Customarily, the linguist carrying out grammatical analysis disregards all questions of frequency and simply notes the occurrence or nonoccurrence of each element in each context of his observed materials. A consequence of this approach is that the resulting grammar sets up a sharp division between a class* G *of grammatical sentences and a class* G' *of ungrammatical sequences.* [...] *The grammatical approach thus contrasts with a statistical approach that leads to an ordering of sequences from more to less probable, rather than a sharp division into two classes within which no such gradations are marked.* [...] *If we somehow rank sequences of English words in terms of their probability, we will find grammatical sequences scattered freely throughout the list.* (Chomsky 1956 [1975]:145)

Logical Structure of Linguistic Theory (II)

> *We might thus be tempted to identify grammaticalness in English with high order of approximation to English, and nongrammaticalness with low order of approximation. But if we do, though we will be characterizing something, it will not be grammaticalness, in the presystematic sense of the term. Perfectly grammatical English sentences can have a reasonable probability of occurrence only in zero-order approximations to English, and as we move to higher orders of approximation, we simply exclude more and more grammatical utterances.* (Chomsky 1956 [1975]:102)

The famous example

1) Colorless green ideas sleep furiously.
2) Furiously sleep ideas green colorless.

Syntactic Structures

> *Despite the undeniable interest and importance of semantic and statistical studies of language, they appear to have no direct relevance to the problem of determining or characterizing the set of grammatical utterances*. I think *we are forced to conclude* that grammar is autonomous and independent of meaning, and *that probabilistic models give no particular insight into some of the basic problems of syntactic structure*. (Chomsky 1957:17)

Modeling colorless green sentences

As we progress through various NLP models I will at times call your attention to whether these models may be brought to bear on the "*colorless green problem*".

Language models

Language models generate sequences of words as a function of immediately preceding token (e.g., how often is *colorless* followed by *green*?).

P(colorless green ideas sleep furiously) = P(colorless | <s>) ×
P(green | colorless) ×
P(ideas | green) ×
P(sleep | ideas) ×
P(furiously | sleep) ×
P(</s> | furiously)

Hidden Markov models

Hidden Markov models generate sequences of words in a two-step procedure:

- POS tags are generated as conditioned on the preceding tags

$$P(\text{JJ JJ NNS VBP RB}) = P(\text{JJ} \mid \langle s \rangle) \times P(\text{JJ} \mid \text{JJ}) \times P(\text{NNS} \mid \text{JJ}) \times P(\text{VBP} \mid \text{NNS}) \times P(\text{RB} \mid \text{VNP}) \times P(\langle /s \rangle \mid \text{RB})$$

- Then, each POS tag then generates a word:

$$P(\text{colorless green ideas sleep furiously}) = P(\text{JJ JJ NNS VBP RB}) \times P(\text{colorless} \mid \text{JJ}) \times P(\text{green} \mid \text{JJ}) \times P(\text{ideas} \mid \text{NNS}) \times \ldots$$