Natural Language Processing
CSE 325/425

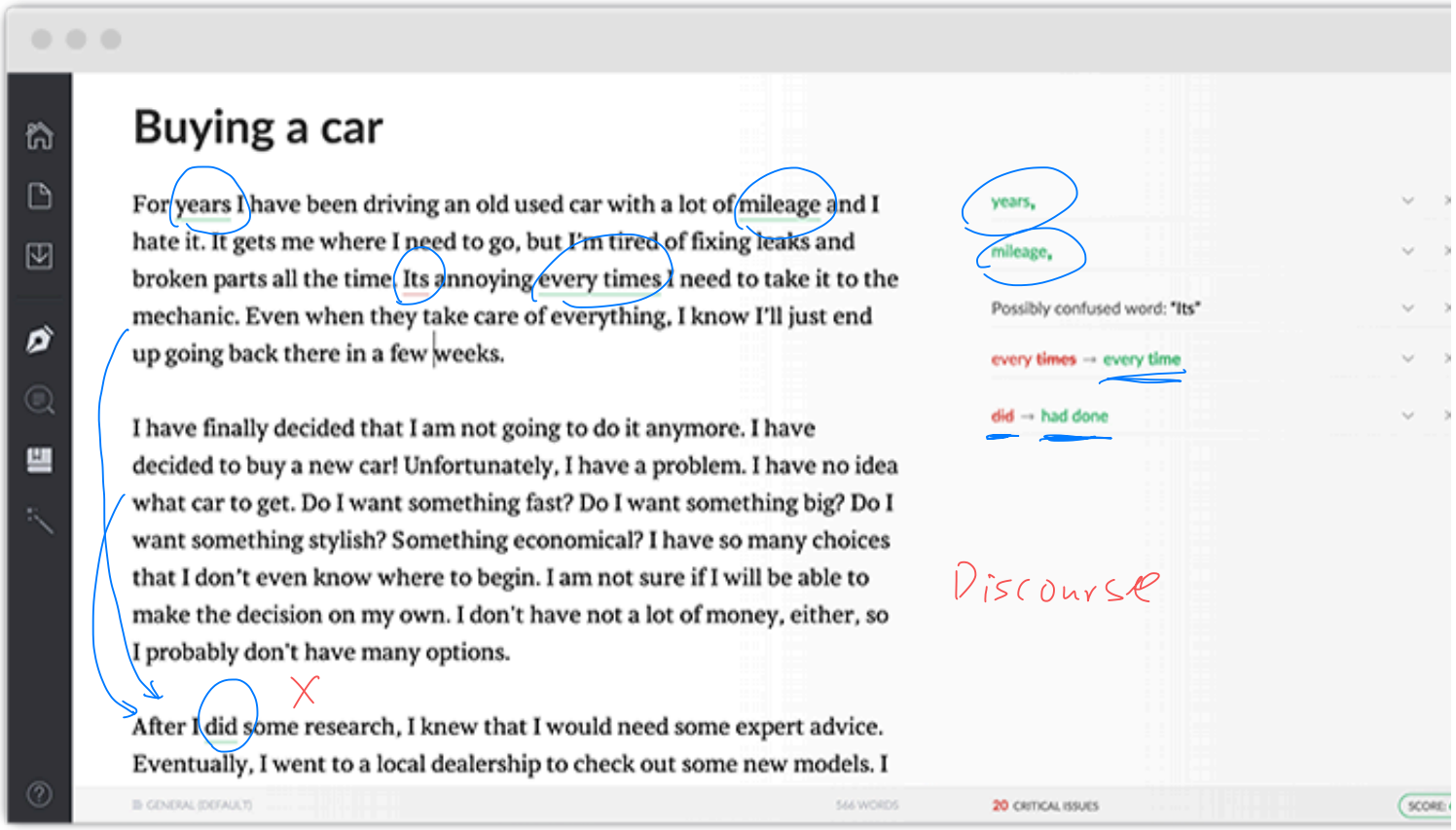
Sihong Xie

Lecture 1:

- Introduction to NLP
- Course outline
- Language models

NLP Applications

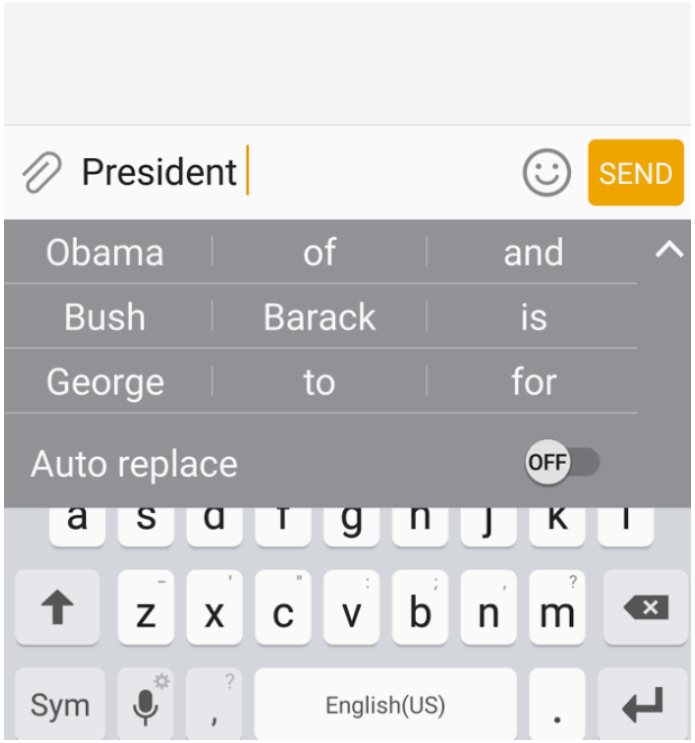
- Spell checking

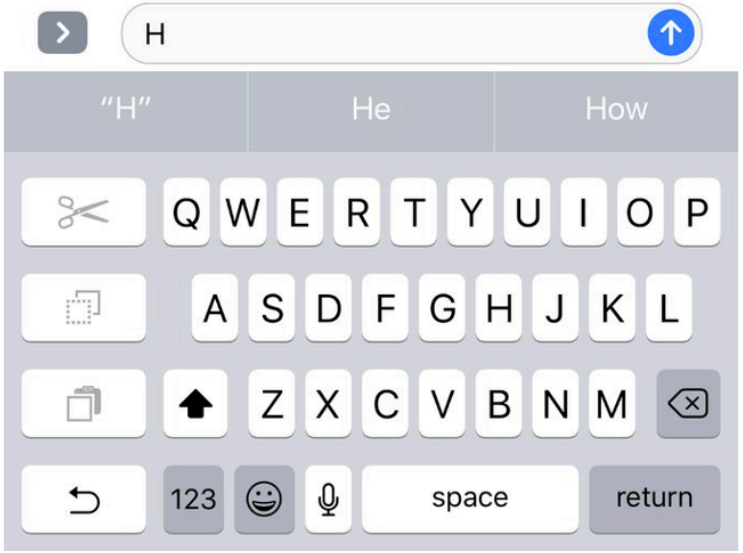


NLP Applications

- Auto completion.

Language model





NLP Applications

- Sentiment analysis

NLP Applications

- Machine translation

Input sentence:	*Translation (PBMT):*	*Translation (GNMT):*	*Translation (human):*
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

NLP Applications

- Chatbots

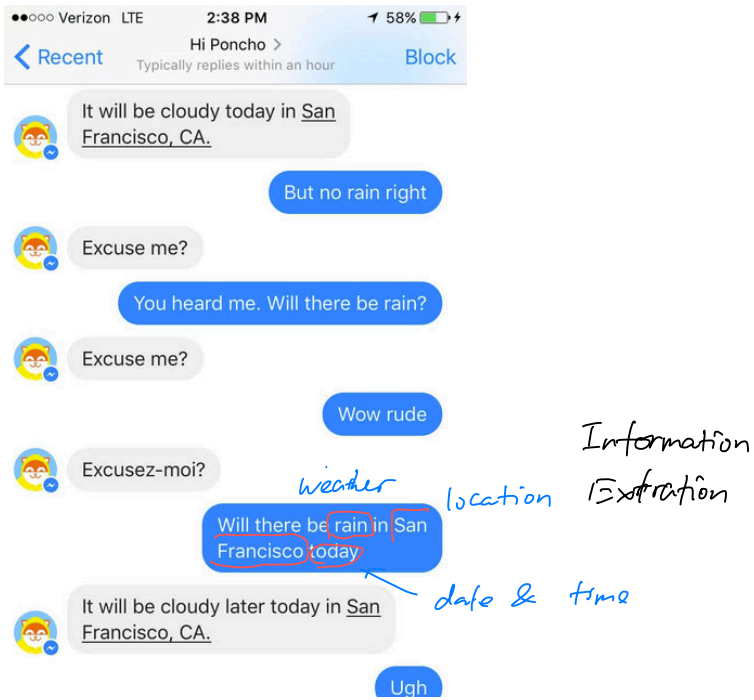
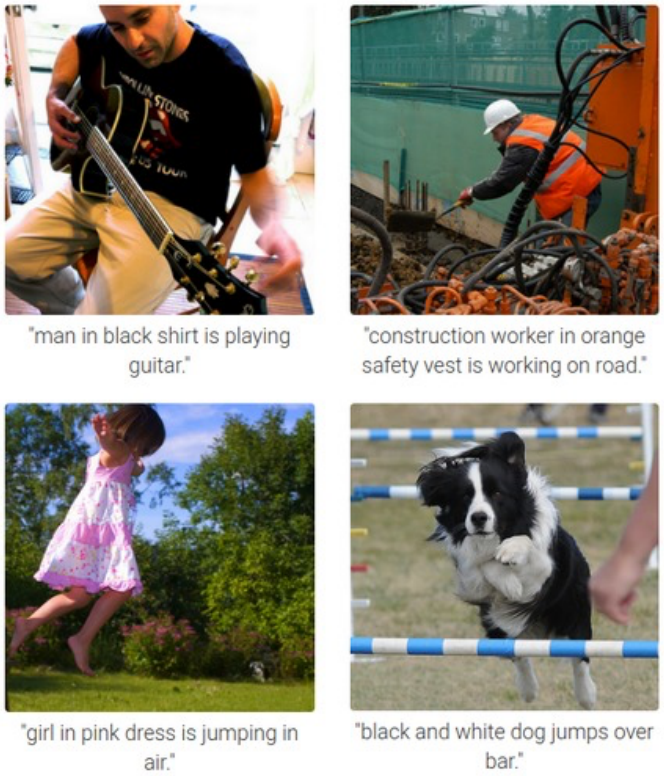
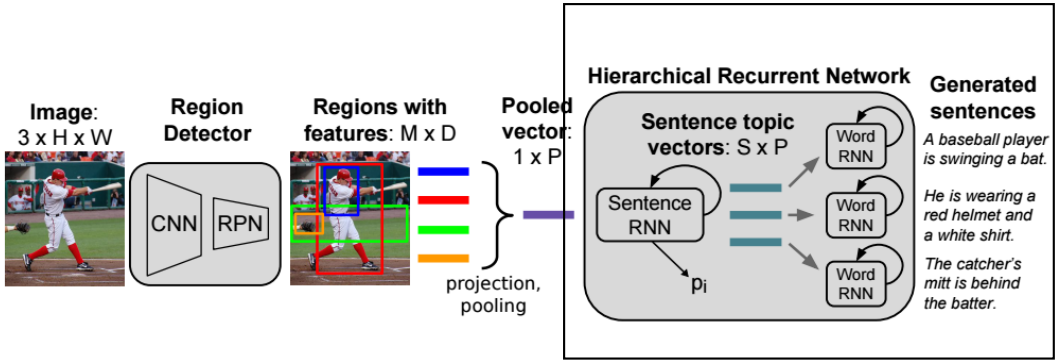


Image captioning





Essentially a language model

predict the next word based on previous words

(Hard) NLP Tasks

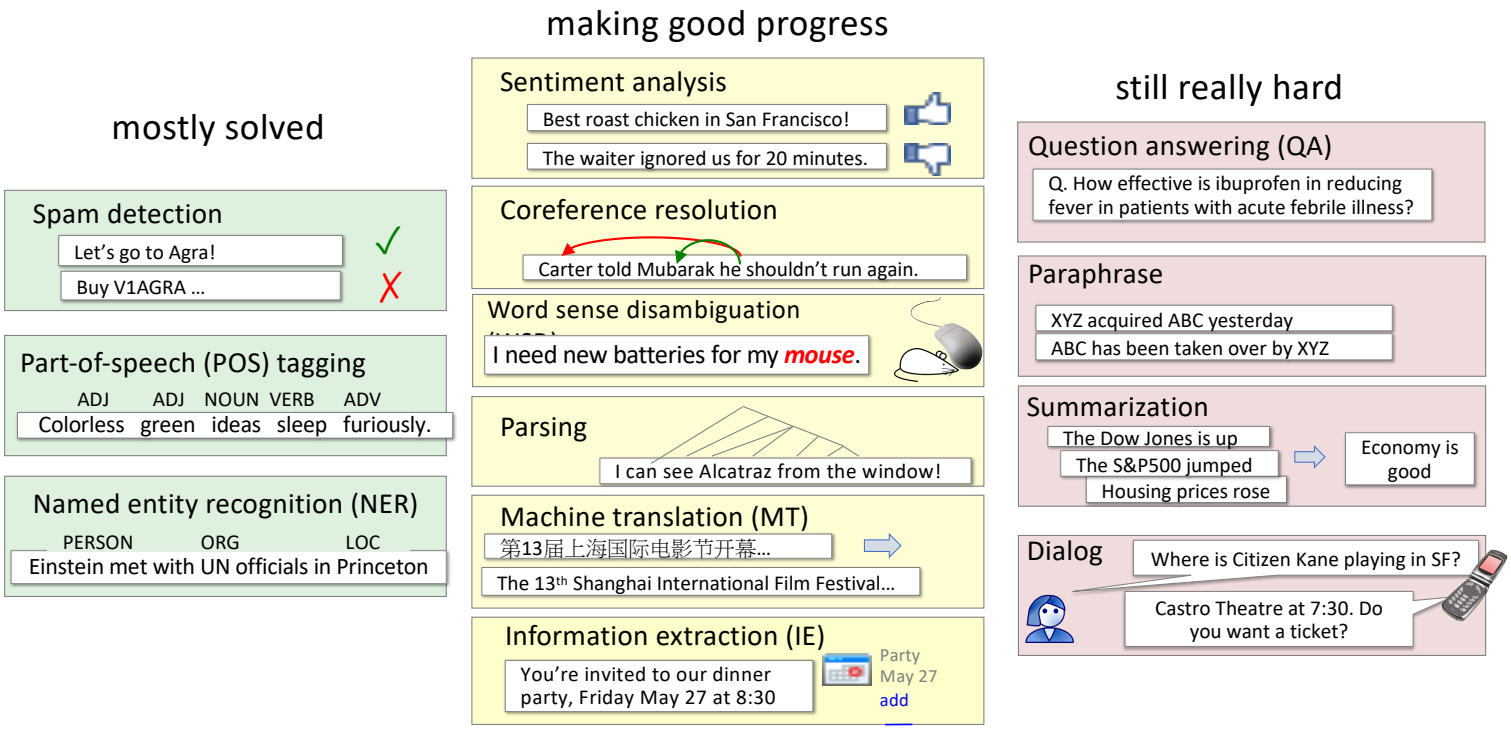
- Tasks on 4 Levels
  - Lexicon: segment "*New York-New Haven railroad*".
  - Syntax: structure of sentences (finding subject, object, and verb)
  - Semantics: I am a huge *fan*.
  - Discourse:
    - John went on a trip to NYC
    - He left on an early morning flight.
- Need knowledge of
  - the world,
  - the language,
  - how to combine them.

"One morning I shot an elephant in my pajamas. How he got into my pajamas I'll never know."

~Groucho Marx
American comedian
1890-1977

CjB
www.cjbaxtergroup.com

Current state (-of-the-art, deep learning!)



Slides from Dan Jurafsky @ Stanford

Course outline (26 lectures)

1. Language models (4 lectures)
- N-gram and Laplacian smoothing.
- Word vectors.

2. Modeling sequences (10 lectures)
- HMM and CRF (5 lectures)
- RNN and LSTM (2 lecture)

3. Syntactic Parsing (10 lectures)
- Context-free grammar
- CYK algorithm
- PCFG and learning PCFG.
- Recurrent neural networks.

- 4. More recent deep learning
- Attention models (2 lecture)
- seq2seq models (2 lecture)

1.1 Machine learning basics (1 lecture)
- Data, loss, optimization (backprop and matrix calculus).

1.2 PyTorch basics (1 lecture)
- Computation graphs.
- Tensors, nn, functions.

5. Assignments

3 coding projects:
- Programming in Python
- All students are required to submit.

6 homework assignments:
- Written
- All students are required to submit.

Grading

- 6 homework (30%)
- No exam.
- 3 individual programming assignments (60%).
- Final online interview (10%).
- Active plagirism detection: severe penalty (from 0 credit for a detected submission to University level hearing committee).
  - see the Dean of Student office website for the details.

Programming environments

- Done with Python; no other languages will be accepted.
- Sketch of codes will be provided.
- If you need GPU for training neural network (first project):
  - Google Colab (https://colab.research.google.com/).
  - GPU-equipped, free of cost.
  - CUDA, Anaconda, and many machine learning packages pre-installed.

What this course is not about

- Data science (preliminaries).
- Data mining (hands-on).
- Machine learning (statistical and theoretical).
- Reinforcement learning.
- AI (traditional symbolic AI).
- Deep learning (we cover a rather good basic).

Sample space, events, random variable

- NLP is about using probability to model languages, and using optimization to find the right model.
- The universal set Ω contains all objects that can occur in a specific context.
  - Discrete: Ω = All words in a corpus (vocabulary)
  - Continuous: Ω =Sentimental score in [-1, 1]
- Event: subsets of Ω ($A \subseteq \Omega$)
  - A document: bag of words (subset of the vocabulary)
  - Positive sentimental score [0, 1]
- Random variable $X : \Omega \rightarrow \text{set of numbers}$
  - e.g., map a word to an integer index.
  - The sentimental score is a number already.
- Probability distribution of a random variable $P(X = 1) = P(\omega \in \Omega : X(\omega) = 1)$
- Probability of an event $A \subseteq \Omega$: $P(A) \rightarrow [0, 1]$

$\Omega = \{$ the, a, curry, boring $\}$

$X =$ 0, 1, 2, 3

$P(X=1) = P($ seeing 'a' in the corpus $)$

$= 5\%$

Joint Prob.

$P($ a boring curry $)$ Independence

$= P(a)\, P(\text{boring})\, P(\text{curry})$

decomposition

$P(a)\, P(\text{boring} \mid a)$ n-Gram

$P(\text{curry} \mid \text{boring}, a)$