SNLP note

Introduction

Definition

- Building computer systems that **understand** and **generate** natural languages.
- Deep understanding of broad language.

or

- A collection of problems, techniques, ideas, frameworks, etc. that really are not tied together in any resonable way other than the fact that they have to do with NLP

Some Applications:

- Speech recognition
- Machine translation
- Personal assistants
- Information Extraction
- Summarization
- Generation
- Question Answering
- Sentiment analysis
- Machine Comprehension
- Cognitive Science and Psycholinguistics (认知科学与语言心理学)

Syllabus

- Structured prediction
- Preprocessing
- Generative learning
- Discriminative learning
- Weak supervision
- Representation and deep learning

NLP Tasks

- Tokenization, Segmentation
- Language modeling
- Machine translation
- Syntactic parsing (语法分析)
- Document classification
- information Extraction
- Textual entailment/Machine comprehension (文字蕴含，机器理解)

Structure Prediction

Problem Signature

- Given some input structure $x \in X$, such as a token, sentence, or documents....
- Predic an **output structure** $y \in Y$, such as a class label, a sentence or syntactic(句法) tree.

Recipe 1: Learn to Score

- Define a prametrized model $s_\theta(x, y)$ that measures the *match* of a given x and y using *representations* $f(x)$ and $g(y)$.
- Learn the paramenters θ from the trainning data D to minimise a loss.
- Given an input x find the highest-scoring output structure

$$y^\star = argmax_{y \in Y} s_\theta(x, y)$$

(a discrere optimization problem)

How to estimate θ

Let us define a **Loss Function**

$$l(\theta) = \sum_{(x,y) \in D} I(y \neq y^\star_\theta(x))$$

where

- $I(True) = 1$ and $I(False) = 0$
- $y^\star_\theta(x) \in Y$ is highest scoring translation of x

$$y^\star_\theta(x) = argmax_{y \in Y} s_\theta(x, y)$$

Learning is as simple as choosing the parameter with the lowest loss

$$\theta^\star = argmin_{\theta \in [0,2]} l(\theta)$$

Background Reading

- Noah Smith, Linguistic Structure Prediction
  - Free when logging in through UCL
  - Relevant:
    - Introduction
    - Dynamic Programming
    - Generative Models (and unsupervised generative models)
    - Globally Normalized Conditional Log-Linear Models