544 NLP Study Guide

Classic Models

Classification *L2*

Preprocessing

- Tokenization
- Other optional methods

Feature Extraction

- Vocabulary Creation
- Feature Representations :
    - Bag of Words (BoW): Limitations and Benefits
    - TF-IDF

TFidf captures the importance of an instance in the dataset: False

Classification Algorithms

- Generative vs, Discriminative (**Pro & Con**)
- Naive Bayes Classifier (**Assume Independence**)
- Linear Classifier

More on Generative Classification Models *L3*

- Perceptron
- SVM
- Logistic Regression

Evaluation Metrics

- Accuracy (**Limitation**)
- Precision
- Recall
- F1

Structured Prediction Tasks: Sequence Labeling *L4*

- Structured vs. Unstructured prediction
- Open vs. Closed classes

Subtasks:

- POS Tagging (**Challenges**)
- Named Entity Tagging (**Challenges**)

HMM

- Assumption: **The future state only depends on the current state**
- Initial state, Transition, and Emission prob
- The challenge of Unknown words
- Decoding *L5*
    - Greedy (Local)
    - Viterbi
- Limitations

MEMM

- Log-Linear Model
- Assumption: **the hidden states are generated from the observations**
- MEMM vs. HMM

CRF

- Same Markov Assumption
- More complex model
- Difference from MEMM

Rule-Based Tagging

Natural Language Representation *L6*

Word Similarity

- Use similar words to resolve sparsity

Latent Semantic Analysis

- Based on Co-occurrence, Document Level
- Features
- SVD byproduct of Matrix Factorization
- Limitations
    - *?Determining context is heuristic*

Word2Vec *L7*

- Based on neighboring words, Content Level
- The byproduct of context word prediction task using NN
- Models
    - Skip-Gram: Given center word, predict context words (Better with infrequent words)
    - Continuous BOW: Given a context word, predict the center word
- Optimization
    - simplify to a single sum and enable gradient descent
    - Negative Sampling

It contains contextual information from the sentence. False

Compare the above 2 approaches

- Depends on task
- Hyperparameter matters

Neural Network

- Capable of Non-linear tasks: XOR
- Universal Approximation Theorem

Computation

- Forward Pass: estimate prediction
- Backward Pass: Update weight based on loss between prediction and label

MLP for NLP

Problem: Input needs to be fixed length, but sentence lengths varies greatly
Solution:

- Pad and Truncate sentences to a fixed length
- Average the word embeddings to a sentence embedding

The emergence of Deep Learning

- Hardware Computational Power upgrade
- Dataset size
- ReLU function

RNN *L8*

- The current word depends on the previous words in the sequence
- Bp chain is long and dependent due to vanishing gradient
- Multiple Limitations on cost and optimization

- BiRNN: Solves the two-way dependency problem
- GRNN:
    - Solves long-range dependency
    - Reset Gate: Forget which element of previous state
    - Update Gate: Update which element to generate new state
    - New Gate: Uses the RG to control previous state, and add to new X
- LSTM *L9*
    - Solves long-range dependency due to vanishing gradient
    - Input Gate, Output Gate, Forget Gate
    - Long term memory cell
- Skip-thought Vectors
    - An implementation of RNN using an Encoder-Decoder Structure
    - For sentence-level classification tasks
    - Encoder: State-by-state accumulative update of sentence-level embedding
    - Decoder: Predict previous and next sentence

Pytorch

- Prepare you data
    - Write your own Dataset (inherit torch.nn.util.dataset)
- Create your model
    - A sequential module if your model is super easy and will not be reused.
    - A nn.Module module
- Write the loop (how many epoch/steps) to train your model:
    - Create a DataLoader that wraps the Dataset you provide
    - Set an optimizer
    - Set a loss for optimization
- For loop....
    - Forward pass
    - Zero-grad
    - Backward pass => Get gradient
    - Optimizer step to update your models' weight.

Probabilistic Language Models *L10*

- Comput the probability of a given sentence or sequence of words
- Given the previous words, predict the next word
- Local dependency assumption with Markov Simplification
    - Unigram
    - Bigram
    - N-grams

Evaluation Metric

- Extrinsic vs. Intrinsic
- Perplexity (Intrinsic): Inverse of the probability of a generated sequence

Limitations

- Generalization
- Long-distance dependency
- Unseen combinations
- Solutions:
    - Smoothing: Steal probabilities from known words
    - Laplace Smoothing: Add one to all counts

Applications of LMs *L11*

Spelling Correction

- Error types: Non-word Erros vs. Real-word Erros
- Detect error: Calculate the probability according to context
- Generate Candidates
    - Pronunciation
    - Spelling: Edit Distance
- Choose Best Candidate: Noisy Channel

Machine Translation

- Challenges
    - Lexical Ambiguity: One word with different meanings
    - Word Order
    - Syntactic Structure not preserved: One to many
- Rule Based
    - Dictionary word-translation
    - Use rules to rearrange
- Transfer-based
- Statistical (IBM Model): Requires parallel corpus
    - Distortion Parameter q (***(l+1)^m alignments***)
    - Translation Parameter t
    - EM algorithm to get those 2. *L12*

Expectation Maximization algorithm can be used to compute the global maximum likelihood given incomplete data. False

    - Limitations
- Phrase-Based Translation Models *L13*
    - Solves Many-to-many and local context problems
    - Distortion Parameter: Distance relative to the previous phrase

    - Stages:
        i. Start from IBM
        ii. Extract phrase pairs
        iii. Compute params
- Decoding
    - Challenges
    - Translation is bad vs. Search is bad
    - Risk Free Recombination
    - Ricky Pruning

Neural Models

- In classic models, performance is dependent on feature complexity

Models Introduction *L14*

- FNN
- Non-linearity
- CNN: FNN dealing with various lengths using a sliding window
- RNN
- Training a Deep Learning Model

Neural Networks for Structured Prediction *L15*

RNN

- Limitations
- Variants

Sequence Labeling

- Getting rid of structured models on dependencies
- BiLSTM+CNN

Dependency Parsing

- Constituency (Context-Free) vs. Dependency
- BiAffine: not guaranteed to be a tree
- NeuroMST
- Main Task and Probing Task

Syntax structures cannot resolve semantic ambiguities

Seq2seq *L16*

- Difference from Sequence Labeling: Length of Y

Neural Machine Translation

- Encoder-Decoder Architecture
    - Encoder: Convert input seq to a seq of vectors
    - Decoder: Convert encoded vector into a seq
- Classifier Decoding
    - Brute: complexity V^T
    - Greedy
    - Beam Search: Only K best remain for each step
- Limitations

Attention

- Solves Vanishing Gradient and Bottleneck
- Attention Score
- Softmax

Advanced NMT with Transformers *L18*

- Semi-Supervised: Given limited amount of parallel corpus, use monolingual corpus to help

Target Side (Back-Translation)

- Synthetic sentences with noise improves encoder
- More monolingual data improves decoder

Source Side

Multi-Lingual NMT

- Benefit small languages
- Many-to-one Interlingual
    - Shared vocab challenges
    - Solution: Byte-pair encoding
    - Pros and cons
- One-to-many with prefix
- Many-to-many

Non-Autoregressive MT

- Iterative Decoding: Deleting words unsure and re-predict
- Latent Variable Models

Evaluation Metrics

- Criterions
- Automatic Evaluation
- Drawbacks

Transformer *L17*

Key Components

- (Masked) Multi-head Self Attention
    - Self-Attention: Solves long-distance dependency, enables parallel computation
    - Multi-head Attention: Solves low complexity
    - Masked Attention: For autoregressive decoder
    - Positional encoding
- Layer Normalization
- Position-wise FFN: For Non-linearity

Architecture

Pros and Cons

Efficient Attention Mechanisms *L19*

- Solves time and memory consumption
- Longformer: sliding window Attention
- Decoupling Attention Matrix: Matrix computation simplified
- Low-Rank Approximation: Decrease context matrix size
    - Luna
- Complexity
- Performance
- Limitations

Inductive Bias

- CNN: Local Dependencies and time-invariant kernel
- RNN: Sequential dependencies and time-invariant recurrence
- Mega Attention
    - EMA: local dependencies that decay exponentially
    - Single-head Gated Attention
    - Mega-chunk: Reducing quadratic to linear complexity

Pretraining *L20*

Objectives for Pretraining tasks

- Easy to collect larget amount of data requiring no label
- General and Semantic

Pretrained Encoder

- Training Task: Masked Language Modeling
- Probing Task: Classification and Labeling
- Models: BERT
- How to use: Fine-Tuning

Pretrained Decoder

- Training Task:
- Probing Task: Autoregressive Generation
- Models: GPT
- How to use: Promting

Pretrained Ecoder-Decoder

- Training Task: Denoising Seq2seq Pre-training
- Probing Task: Seq2Seq Generation
- Models: BART

Prompting

Procedure

- Prompt addition
- Answer Prediction
- Answer-label mapping

Types

- Cloze (encoder)
- Prefix (Decoder)

Parameter-Efficient Fine-Tuning *L21*

Existing Methods

- Adapter

- Prefix Tuning
- Lora

Reinforcement Learning with Human Feedbacks

- Imroves prompting on direct communication
- Reinforcement Learning with Human Feedbacks

Deep Generative models

- Requiring no labeled data while learning well
- Distribution-based vs. Non-distribution based

Distribution based

- Pros and Cons of Closed-form Analytic Solution
- Autoregressive Models and Limitations
- Exact Density Estimation
    - Genrative Flows: by inverting to and from a gaussian distribution
    - Limitations
- Approx Density Estimation: solves wasting model space
    - VAE: Approximate using variational inference
        - ELBO
    - Diffusion: Multi-step VAE that solves weak correlation between x and z