Natural Language Processing

Summary

Pavel Rychlý

11 Dec 2023

Problems with NLP

- Problems with NLP
  - Zipf‘s law
  - Ambiguity
  - Variability
- Approaches
  - symbolic (rule-based)
    - no data available
  - statistical
  - neural (deep learning)
    - huge data available

Statistical NLP

- counts
- keywords
- collocations, multi-word units
- language modeling

Language Modeling

- probability of senteces, chain rule
- n-grams, Markov's assumption

  $p(W) = \prod_i p(w_i | w_{i-2}, w_{i-1})$
- maximum-likelihood estimation gives zero probabilities
- smoothing
- evaluation using cross entropy, perplexity

Text Classification

- applications
- Naive Bayes Classifier
- evaluation:
  - precision
  - recall
  - accuracy

Continuous Space Reprasentation

- words as vectors, word embeddings
- methods of learning vectors
- evaluation of words embeddings
- *optional homework: Stability of word embeddings*

HW: Stability of word embeddings

Choose one or more methods for creating word embeddings (word2vec, FastText, GloVe, …), run the traning on same data with different parameters (and/or epochs), evaluation stability.

Stability can be computed in several ways:

1. How many pair similarities are same. It can be computed on the whole vocabulary on a sample (for example: 10 words with frequences from [100, 400, 1600, 6400, 25600, …]).
2. Percentage of changes in analogy tasks. Same percentage in the taks doesn't mean the same succesful analogy items. Calculate how many items changed successful/unsuccessful estimation.
3. Percentage of changes in the Outlier Detection task

Neural Networks

- structure of NN
- matrix representation
- activation functions
- NN training
  - stochastic gradient descent
  - backpropagation
- sub-word tokenization
  - *opt. hw: subword coverage*

Recurrent NN

- language modeling using NN
- training RNN
- problems in training RNN
- LSTM
- Bidirectional, multi layer RNN

Simple NLP using NN

- Named Entity Recognition (NER)
- language modeling
- training
- evaluation
- *opt. hw: NN for adding accents*

Machine translation

- sequence to sequence RNN
- attention
- decoding, beam search
- MT evaluation: BLEU, ChrF++

Transformers

- encoder, decoder
- encoding positon
- attention structure

Pretrained models

- Encoder only
- Decoder only
- Encoder-decoder
- training objectives
- BERT, GPT, T5

Question Answering

- QA types
- usage
- reading comprehension
- applying NN for QA

Recipe for Training NN

- NN training fails silently

1. Become one with the data
2. Set up the end-to-end training/evaluation skeleton + get dumb baselines
3. Overfit
4. Regularize
5. Tune
6. Squeeze out the juice

Where to start

- Hugging Face
  - models
    - code
    - pre-trained, ready to use
  - datasets
    - sometimes with evaluaton
- transformers library
  - very complex
  - 3 implementations: Jax, PyTorch, TensorFlow

Pre-trained models

- OpenLLM
- llama.cpp
  - run the LLaMA model using 4-bit integer quantization on a MacBook
- optimizations
  - float16, bfloat16
  - quantization

Training from scratch

- nanoGPT
  - easy to read
  - minimal dependencies
- nanoT5
  - train T5 on 1xA100 GPU in less than 24 hours