CS5740: Natural Language Processing

Contextualized Word Representations

Instructor: Yoav Artzi

Slides adapted from Graham Neubig

Overview

- Contextualized word representations
- Models
  - context2vec
  - ELMo
  - BERT

Contextualized Word Representations

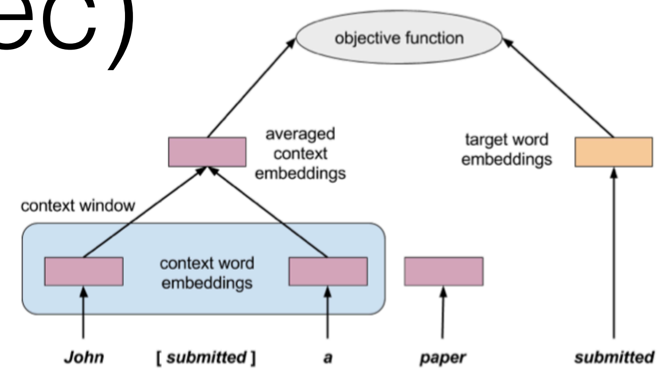
http://jalammar.github.io/illustrated-bert/

Contextualized Word Representations

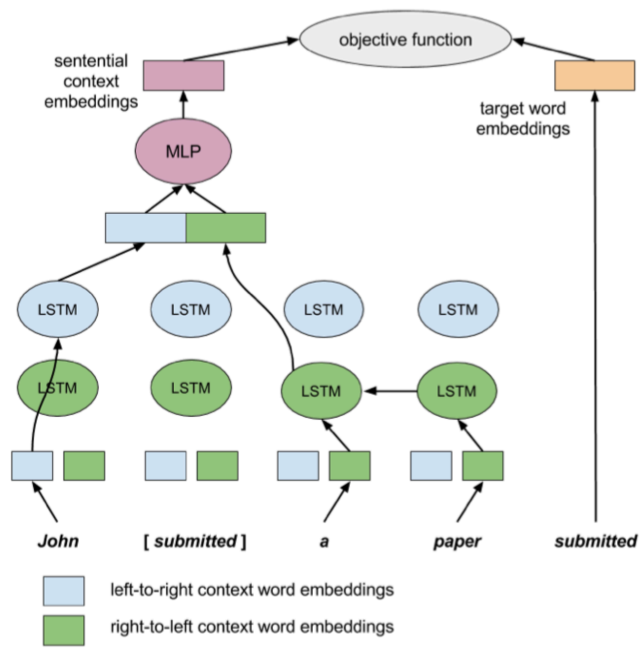
- word2vec, GloVe
  - Learn a vector for every word type
  - Always the same vector
- Instead: learn a different vector for word type in every usage
  - But: how do we define the space of uses? Isn't it too large?
  - Solution: use sentence encoders to create a custom vector for every instance of a word

Central Word Prediction Objective (context2vec)

- Model: bi-directional LSTM
- Objective: predict the word given context
- Data: 2B word ukWaC (English data) corpus
- Downstream: use vectors for sentence completion, word-sense disambiguation, etc.



(a) word2vec *CBOW*

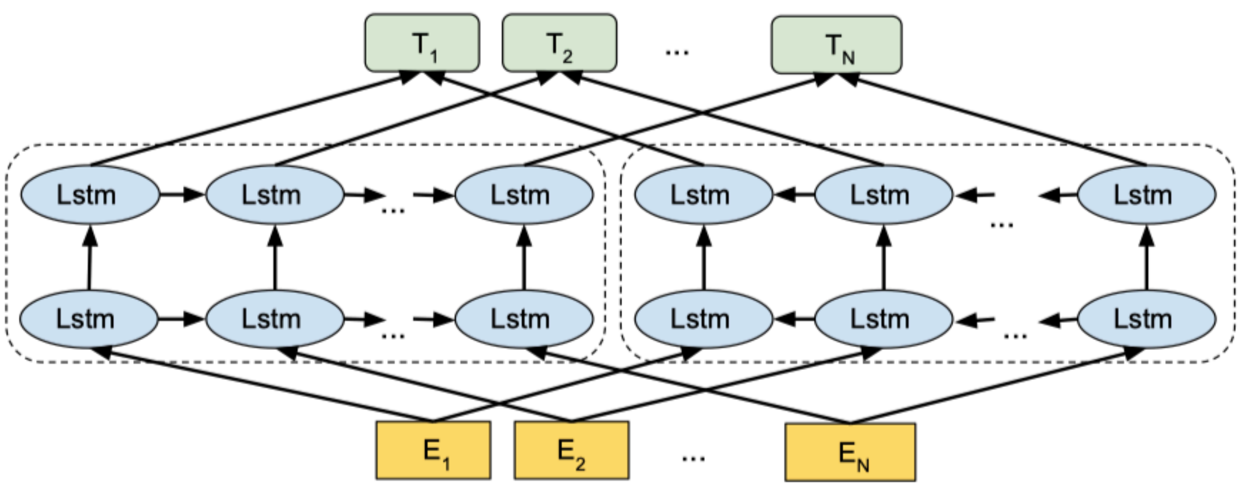


(b) *context2vec*

[Melamud et al. 2016]

Bi-directional Language Modeling Objective (ELMo)

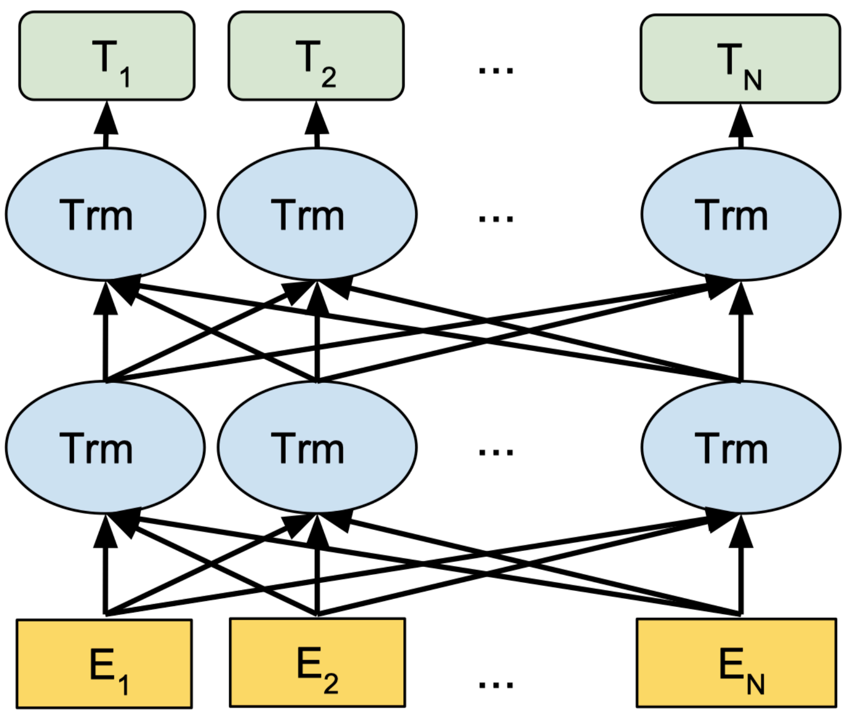
- Model: multi-layer bi-directional LSTM
- Objective: predict the next word left→right and next word right→left independently
- Data: 1B word benchmark LM dataset
- Downstream: fine-tune the weights of the linear combination of layers per task

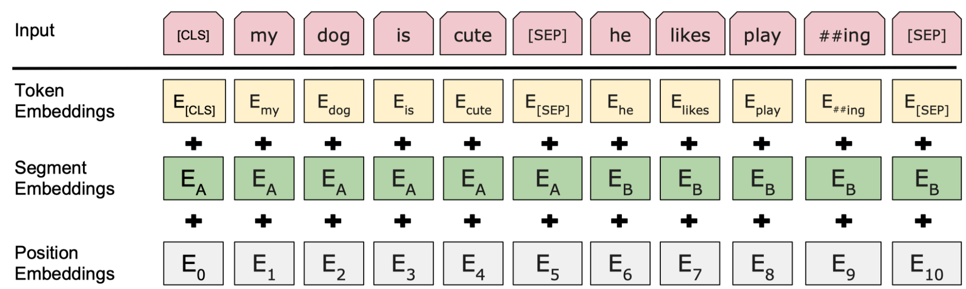


[Peters et al. 2018; figure from Devlin et al. 2018]

Masked Word Prediction (BERT)

- Model: multi-layer self-attention (Transformer), input sentence (or pair w/[CLS] token) and subword representation
- Objective: masked word prediction + next-sentence prediction
- Data: BookCorpus + English Wikipedia
- Downstream: fine-tune weights per task





[Devlin et al. 2018]

Masked Word Prediction

- Predict a masked word
  - 80%: substitute input word with “[MASK]”
  - 10%: substitute input word with random word
  - 10%: no change
- Like predicting the next word, but adapted for multi-layer self attention

[Devlin et al. 2018]

Consecutive Sentence Prediction

- Classify two sentences as consecutive or not
  - 50% of training data is consecutive

Input = `[CLS] the man went to [MASK] store [SEP]`

`he bought a gallon [MASK] milk [SEP]`

Label = `IsNext`

Input = `[CLS] the man [MASK] to the store [SEP]`

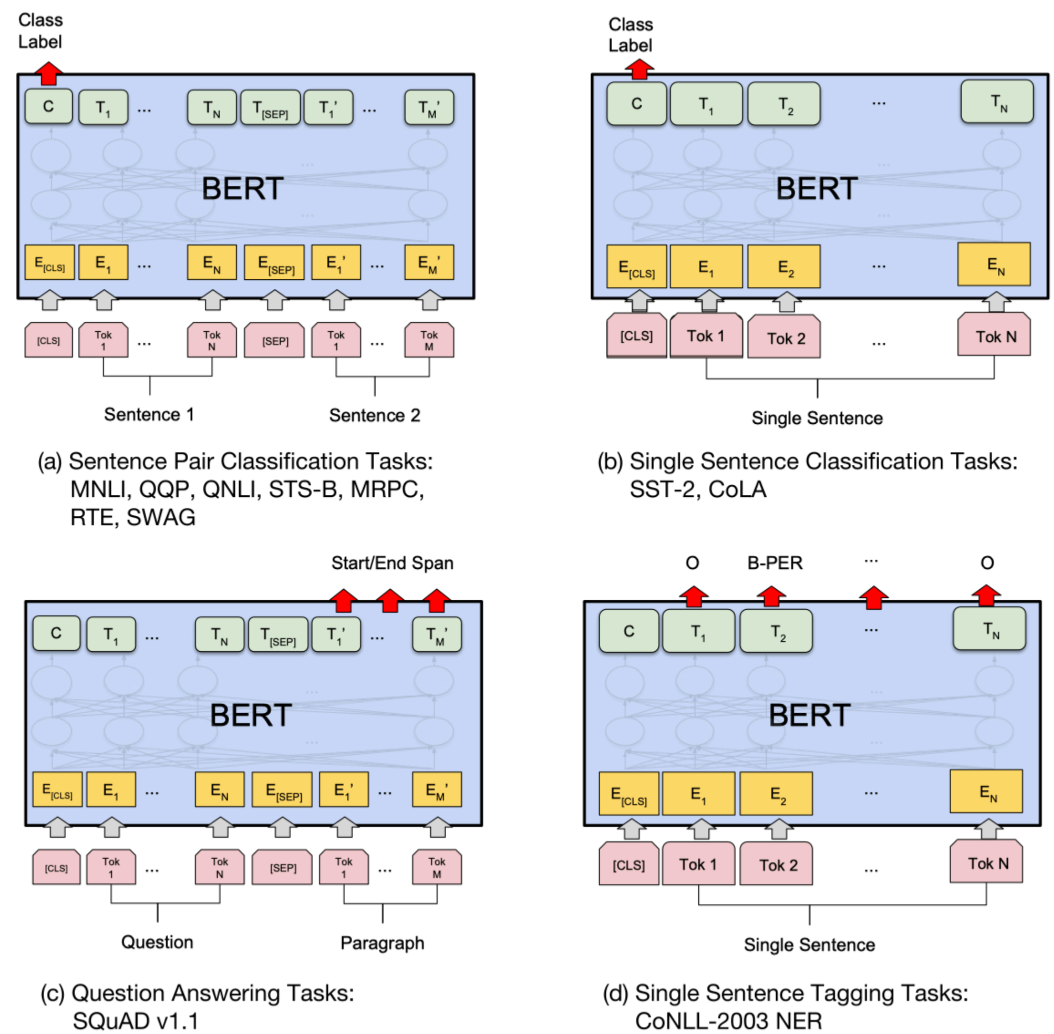
`penguin [MASK] are flight ##less birds [SEP]`

Label = `NotNext`

[Devlin et al. 2018]

Using BERT for Downstream Tasks

- Use the pre-trained model as the first “layer” of your final model
- Train with fine-tuning using your supervised data



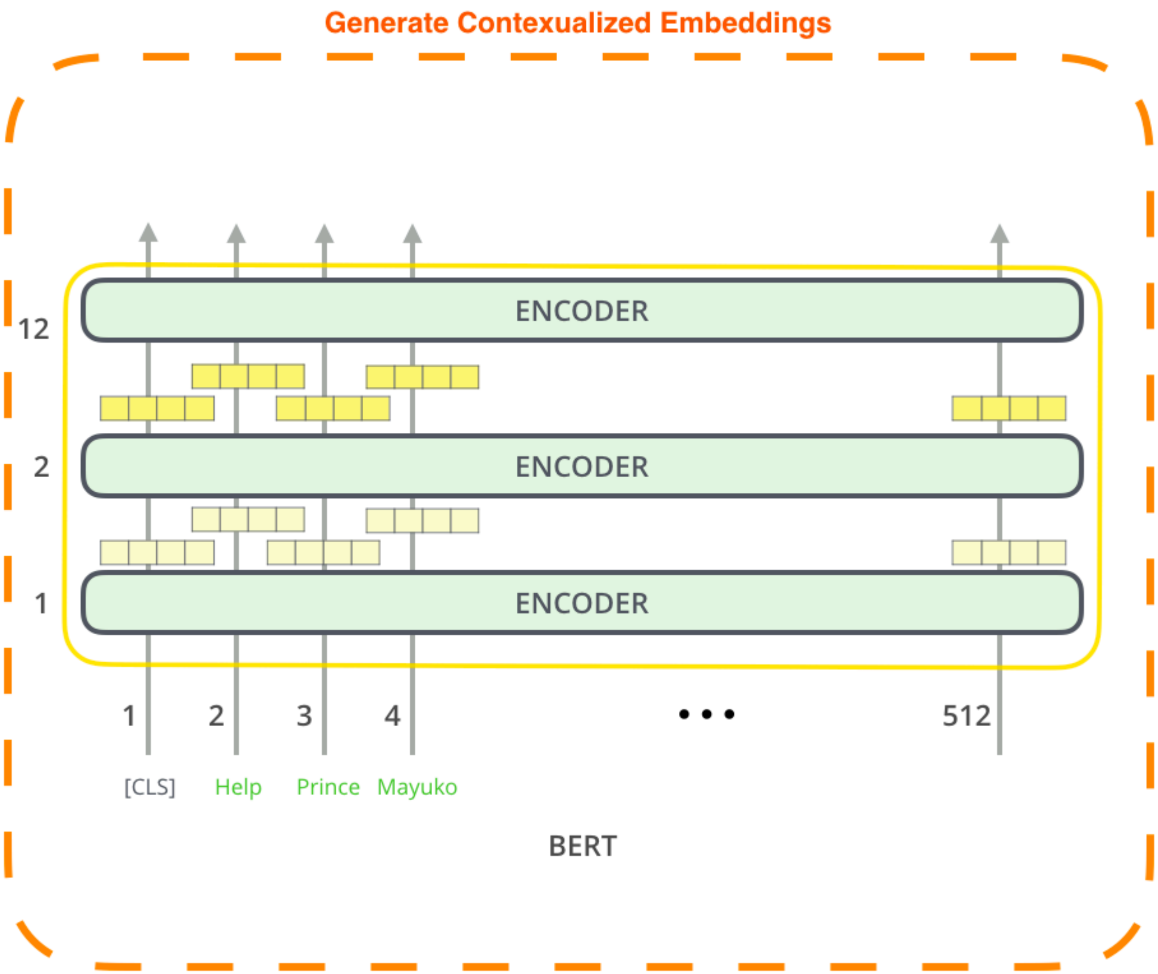
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

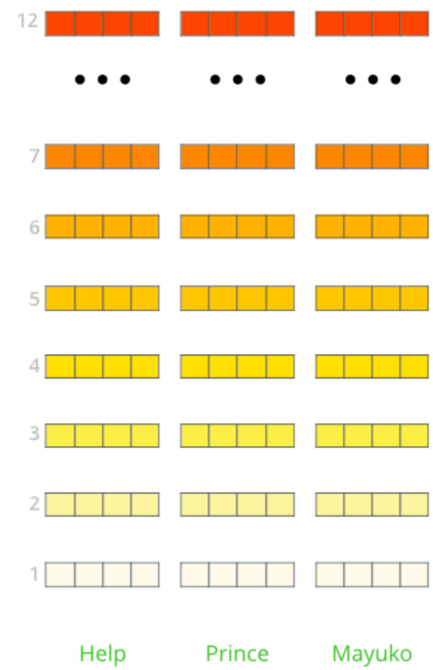
(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Using BERT for Feature Extraction



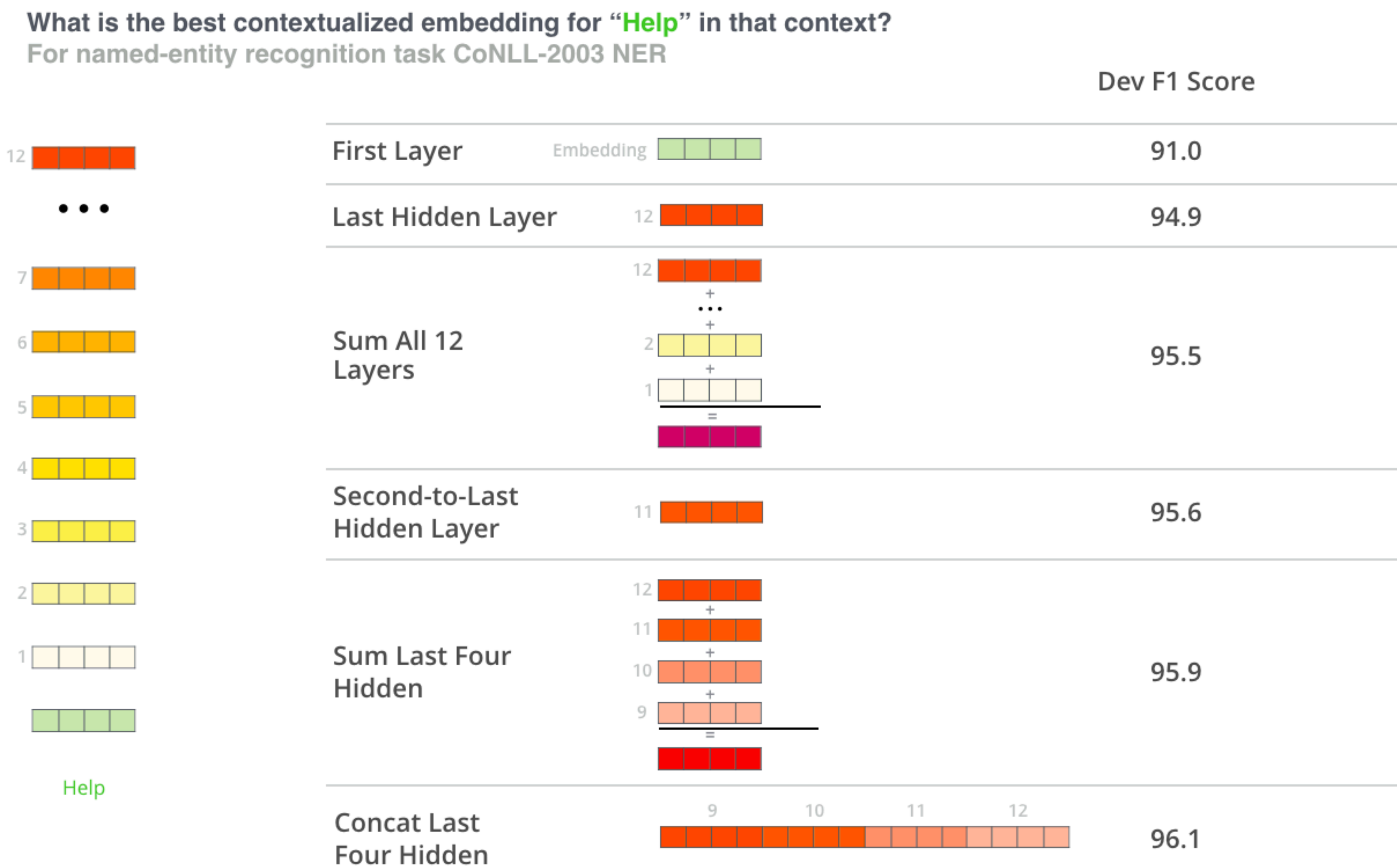
The output of each encoder layer along each token's path can be used as a feature representing that token.



But which one should we use?

http://jalammar.github.io/illustrated-bert/

Using BERT for Feature Extraction



http://jalammar.github.io/illustrated-bert/