Natural Language Processing (CS5803)

Lecture 7
(Contextual Embedding)

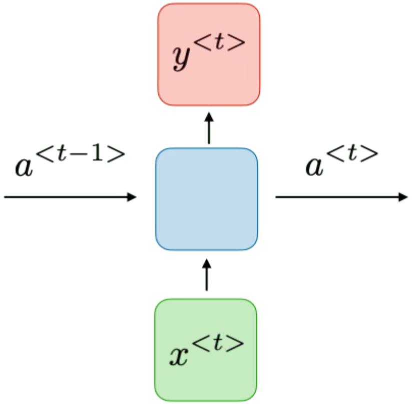
Context is key

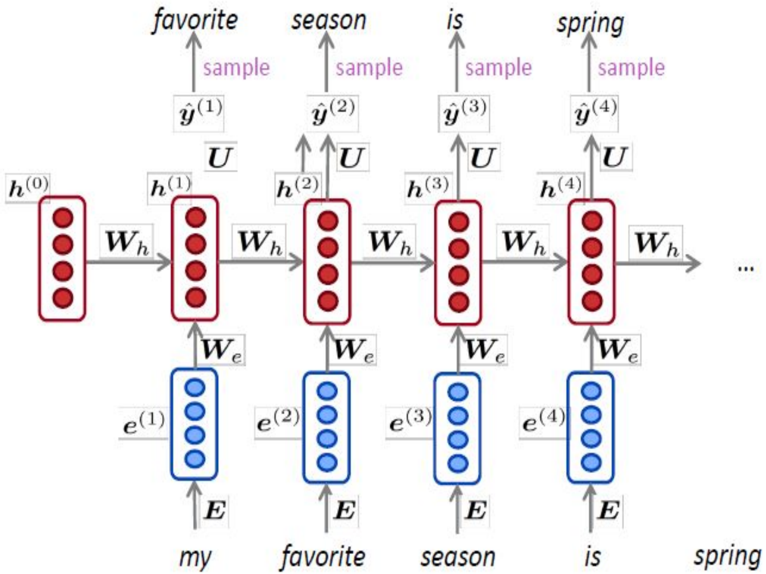
- Context is <u>key</u>
- I lost the <u>key</u> somewhere in the garden
- You need to generate an SSH <u>key</u> pair

- Kids went to <u>play</u> outside
- Croatia <u>play</u> Germany in the next match
- I never acted in any <u>play</u>

- Context is important
- **How?**
    - Contextual Word Embedding/ Contextual Representations

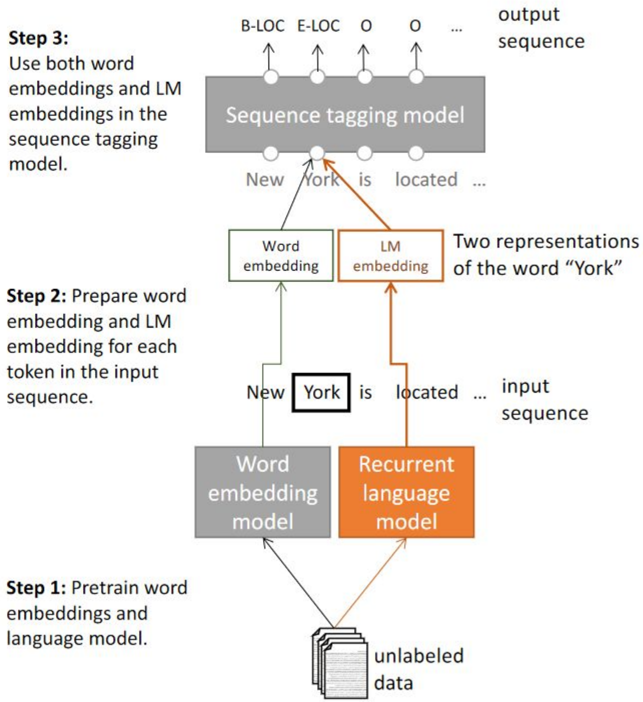
Different representations in different contexts?





- In LSTM, representation in one cell (token) is affected by the representation of the previous cell (token)
- Combine pre-trained representation and LM representation

Different representations in different contexts?



- Combine representations
  - Non-contextual
  - Contextual
- Pass it to next architecture block
- Was shown to work well for Sequence labeling task

Peters, M., Ammar, W., Bhagavatula, C. and Power, R., 2017, July. Semi-supervised sequence tagging with bidirectional language models. ACL 2017 (pp. 1756-1765).

Multi-layer RNNs for Representations

BiLM Training Data Source: 1B English Word Benchmark from [1]

Language Support: English

Model Architecture: RNN

BiLM Objective Function:

A biLM combines both a forward and backward LM. Our formulation jointly maximizes the log likelihood of the forward and backward directions:

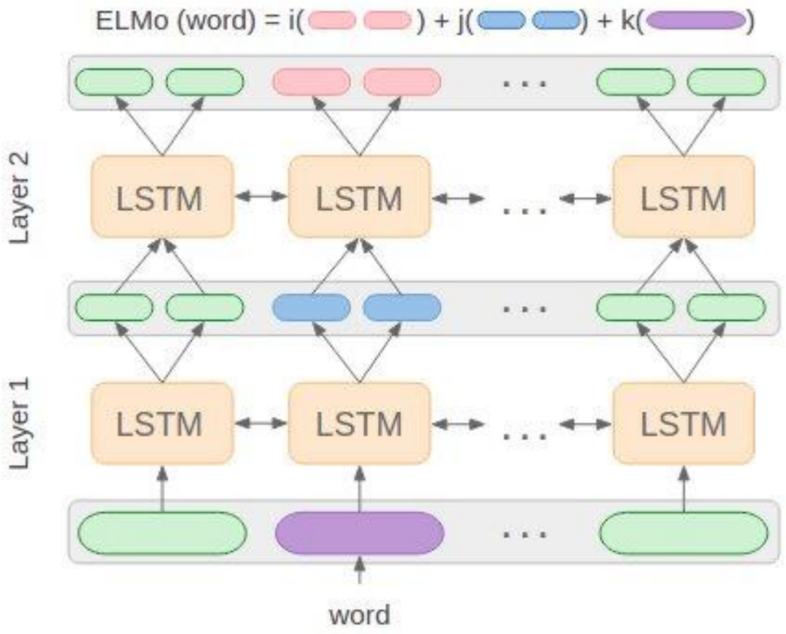
$$\sum_{k=1}^{N} \Big(\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \Big) .$$

We tie the parameters for both the token representation (Θ_x) and Softmax layer (Θ_s) in the forward and backward direction while maintaining separate parameters for the LSTMs in each direction.

Source: Peters, Matthew E., et al. "Deep contextualized word representations." NAACL 2018.

[1]: Ciprian Chelba, et al. 2014. One billion word benchmark for measuring progress in statistical language modeling. In INTERSPEECH.

Embeddings from Language Models (ELMO)

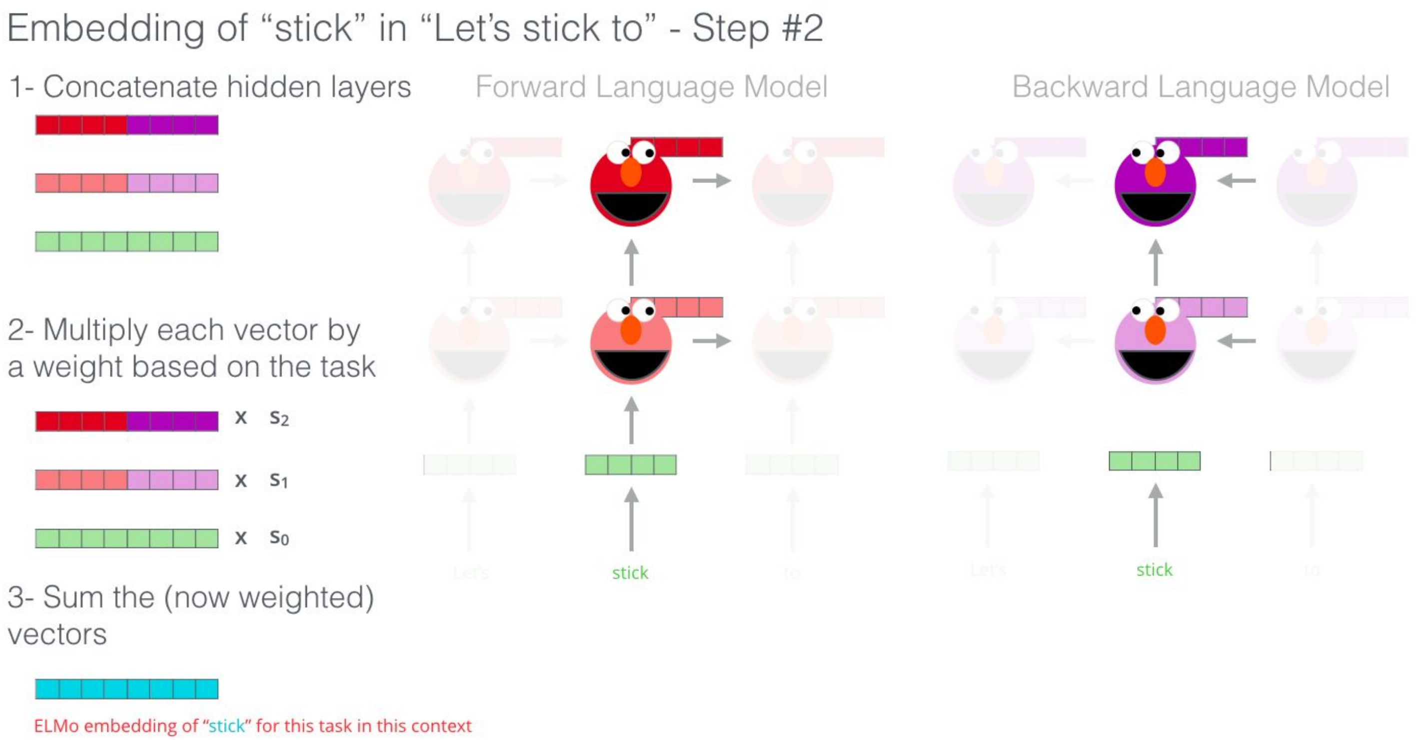


- Stacked LSTMs
- Each LSTM layer i gives a representation h_i of the token t_i
- Final representation h is a combination of the representations from different layers
  - $h=f(h_0,h_1,\ldots, h_L)$
- How to combine these representations?
- How to use in target tasks?

Ref: Peters, Matthew E., et al. "Deep contextualized word representations." NAACL 2018.

Figure from: Biesialska, K.et al. (2020). Sentiment analysis with contextual embeddings and self-attention. In *International Symposium on Methodologies for Intelligent Systems* (pp. 32-41).

Combining Representations from ELMO Layers



Picture courtesy: [The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)]() **by Jay Alammar**

Using the representations for target tasks

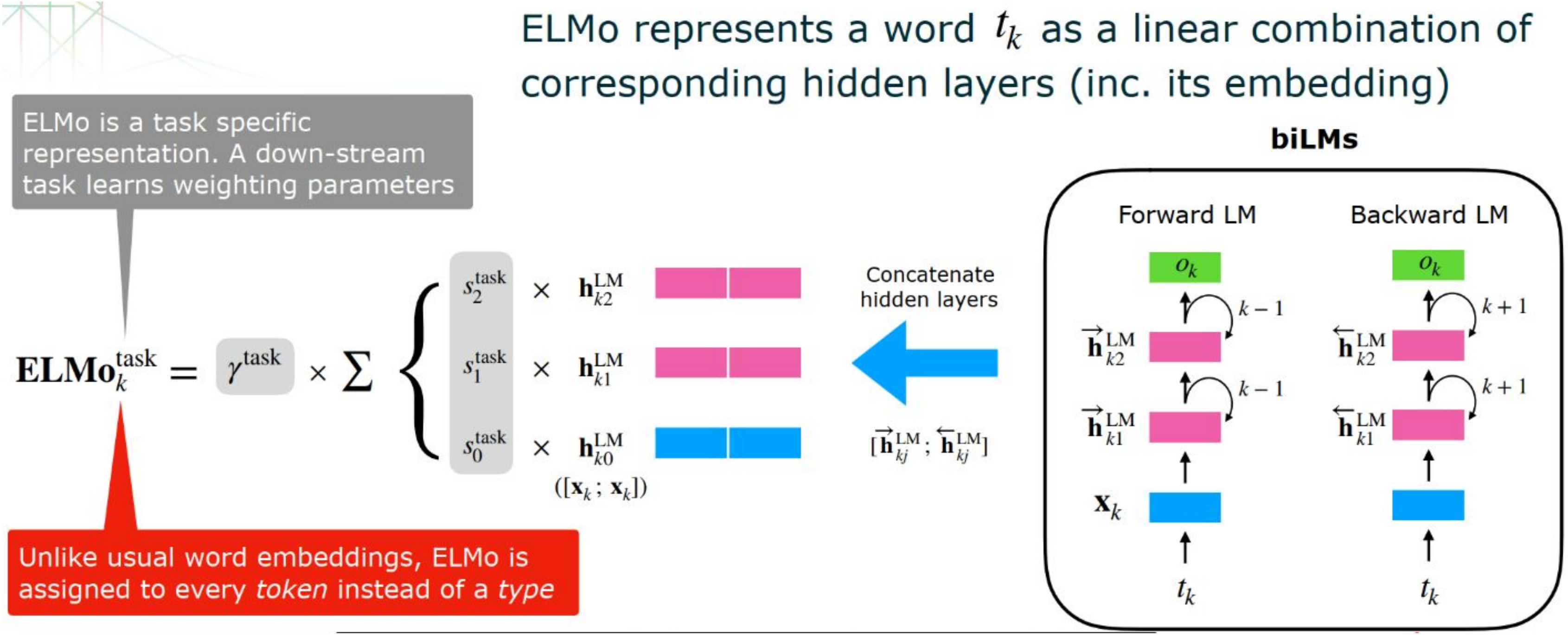
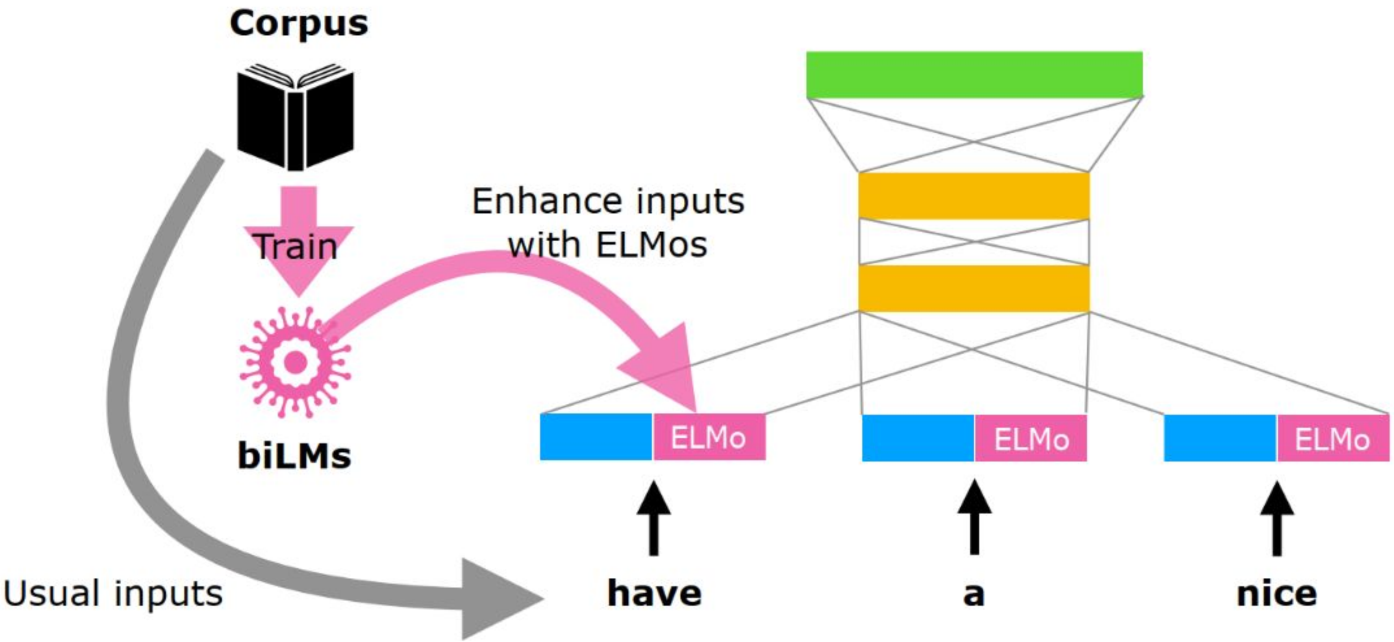


Image courtesy: From presentation slides by Hang Dong

Enhancing Representations with ELMO

ELMo can be integrated to almost all neural NLP tasks with simple concatenation to the embedding layer



Does it work?

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Ref: Peters, Matthew E., et al. "Deep contextualized word representations." NAACL 2018.

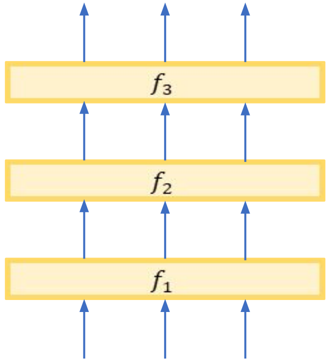
How to Measure Contextuality?

- Words in different contexts should not have similar representations
- Word w is present in sentences: $\{s_1, s_2, s_3, \cdots, s_n\}$
- Position of the word in these sentences: $\{i_1, i_2, i_3, \cdots, i_n\}$

- f_l (s,i) is a function that maps s[i] to its representation in layer l of model f
- Measures: IntraSim and SelfSim

$$IntraSim_\ell(s) = \frac{1}{n}\sum_i \cos(\vec{s}_\ell, f_\ell(s,i))$$

$$SelfSim_\ell(w) = \frac{1}{n^2 - n}\sum_j \sum_{k \neq j} \cos(f_\ell(s_j, i_j), f_\ell(s_k, i_k))$$

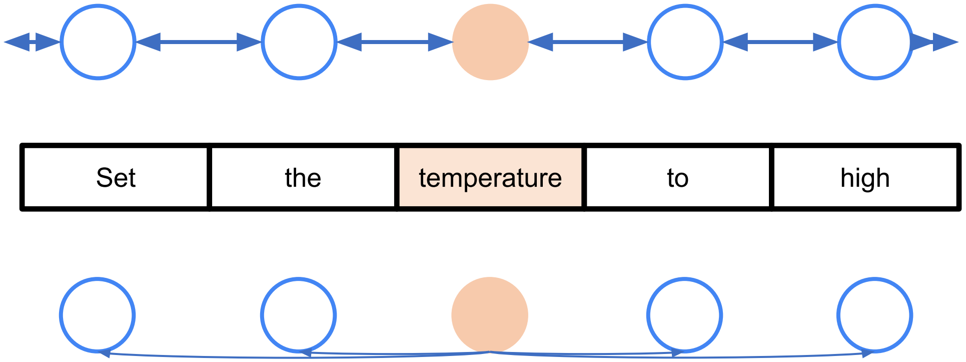


s_1= Context is key
s_2= I lost the key
s_3= Generate an SSH key pair

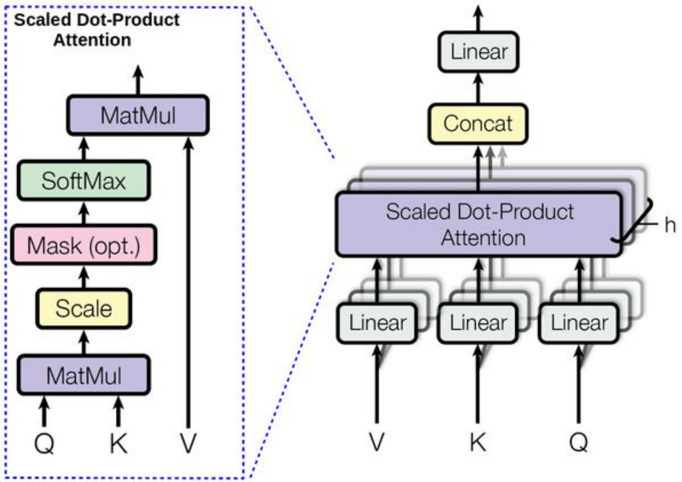
Ethayarajh, Kawin. "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings." *EMNLP-IJCNLP 2019*.

Moving Forward: Transformer

- Sequential connections make the training slow.
- Can we
  - Remove the sequential connections
  - Capture dependence through attention
- Transformer model
  - Has encoder and decoder
- Trained for the MT task
- Transformer blocks are used as part of other architectures too



Attention in Transformer



- Each token has the following representations
  - Query (Q) [To match others]
  - Key (K) [To be matched]
  - Value (V) [Information to be used]
- Idea: For a given word, its representation will be governed by Value vectors of similar vectors
- Take a word, get its query (Q) vector
- Get "similar" keys (K)
- Take weighted combinations of corresponding V's

$$A(q,K,V) = \sum_i \left(\frac{\exp(q,k_i)}{\sum_j \exp(q,k_j)} v_i \right)$$

- Combined equation after scaling

$$A(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Transformer: Encoder and Decoder

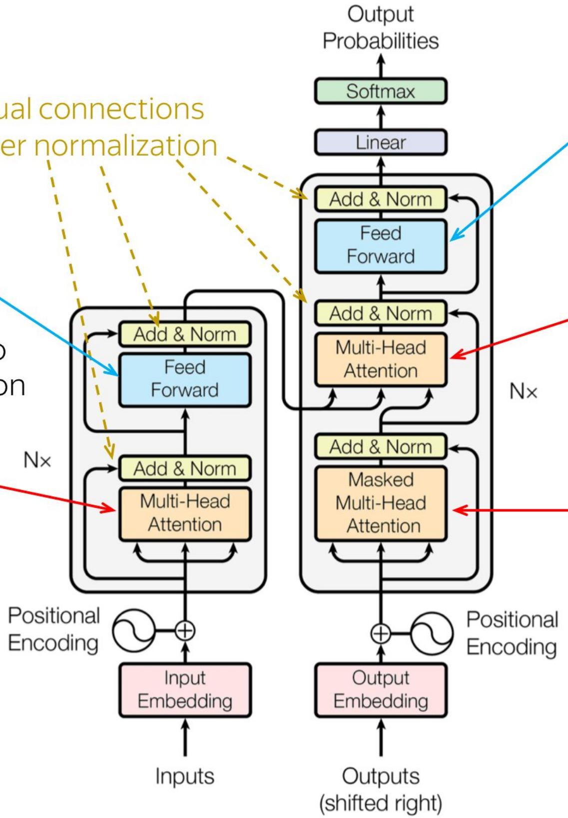
Residual connections and layer normalization

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

↑

Encoder self-attention: tokens look at each other

queries, keys, values are computed from encoder states



Feed-forward network: after taking information from other tokens, take a moment to think and process this information

↑

Decoder-encoder attention: target token looks at the source

queries – from decoder states; keys and values from encoder states

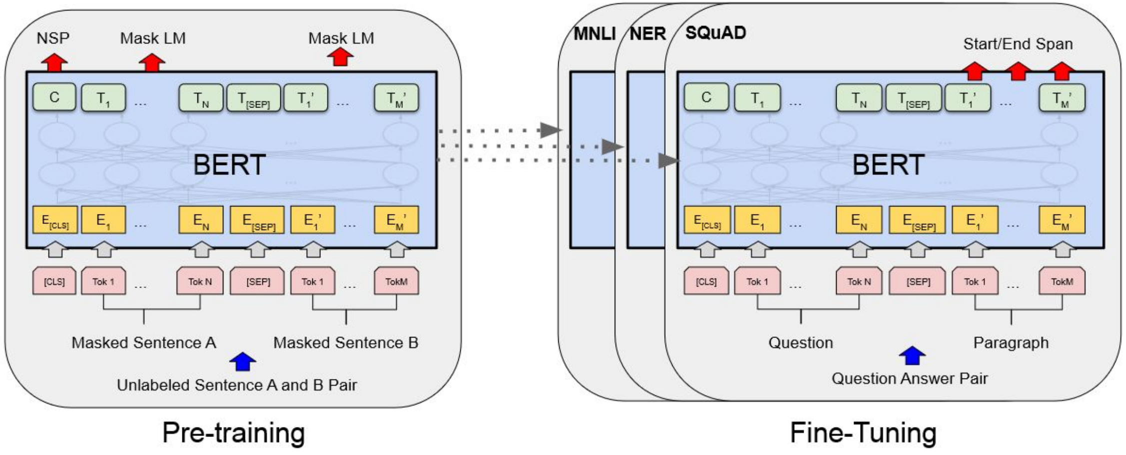
↑

Decoder self-attention (masked): tokens look at the previous tokens

queries, keys, values are computed from decoder states

Image courtesy: Internet

BERT

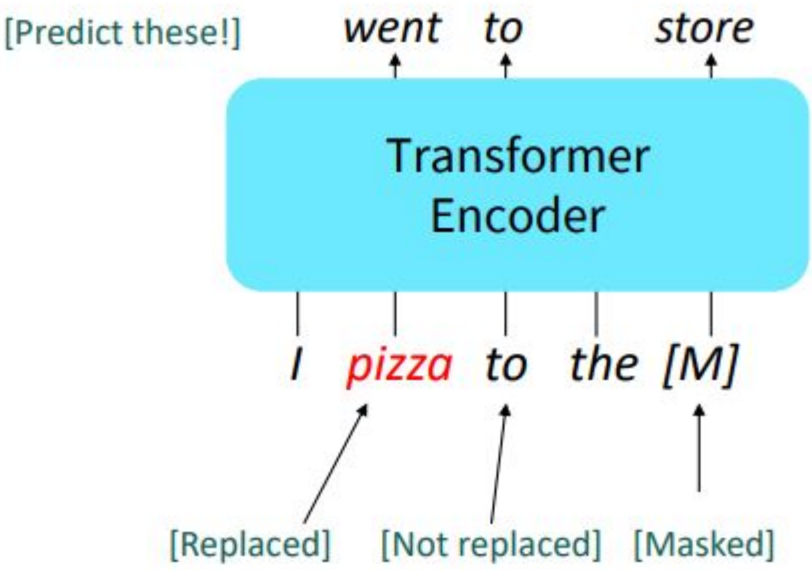


With self supervision

With task-specific supervision

- Input representation:
  - Single sentence
  - Pair of sentences
- With special tokens
  - [CLS], [SEP]
- Self-supervised objectives
  - Masked Language Model
  - Next Sentence Prediction

BERT: Bidirectional Encoder Representations from Transformers



Predict original text from corrupted text
(denoising objective)

Image Source: http://web.stanford.edu/class/cs224n/slides/cs224n-2021-lecture10-pretraining.pdf
Paper reference: Devlin, Jacob et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL 2019*

BERT Training Data Source: Used the Books Corpus (800M words) and English Wikipedia (2,500M words).

Language Support: English

Model Architecture: Encoder of Transformer

BERT Objective Function:

- Masked Language Model (MLM): Predict a random 15% of (sub)word tokens. • Replace input word with [MASK] 80% of the time
  - Replace input word with a random token 10% of the time
  - Leave input word unchanged 10% of the time (but still predict it!)
- Next Sentence Prediction (NSP)

Source: http://web.stanford.edu/class/cs224n/slides/cs224n-2021-lecture10-pretraining.pdf
Paper Link: https://arxiv.org/pdf/1810.04805.pdf&usg=ALkJrhhzxlCL6yTht2BRmH9atgvKFxHsxQ

mBERT: Multilingual Bidirectional Encoder Representations from Transformers

Language Support: 104 languages, No English and chinese languages

mBERT Training Data: Entire Wikipedia dump for each language excluding user and talk pages

Model Architecture: Encoder of Transformer, same as BERT

mBERT Objective Function: Same as BERT (only MLM objective)

1. 110k shared Word-Piece vocabulary
2. Exponentially smoothed weighting of the data during pre-training data creation
3. Use: [BERT-Base, Multilingual Cased (New, recommended)](#) : 104 languages, 12-layer, 768-hidden, 12-heads, 110M parameters

Source: https://github.com/google-research/bert/blob/master/multilingual.md
Paper Link: https://arxiv.org/pdf/1810.04805.pdf&usg=ALkJrhhzxlCL6yTht2BRmH9atgvKFxHsxQ

MuRIL: Multilingual Representations for Indian Languages

Language Support: A BERT model pre-trained on 17 Indian languages, and their transliterated counterparts.

mBERT Training Data: Wikipedia, Common Crawl, PMINDIA and Dakshina

Model Architecture: Encoder of Transformer, same as BERT

mBERT Objective Function: Same as BERT (only MLM objective)

1. Kept an exponent value of 0.3 and not 0.7 for upsampling, shown to enhance low-resource performance
2. More data details:
    A. Monolingual Data
    B. Parallel Data: Translated Data and Transliterated Data

Source: https://tfhub.dev/google/MuRIL/1

BART Training Data: Not specified

Language Support: English

Model Architecture: Transformer large sequence to sequence model (12 layers both sides)

BART Objective Function: Set of Noise functions to corrupt the document. Noise functions are:

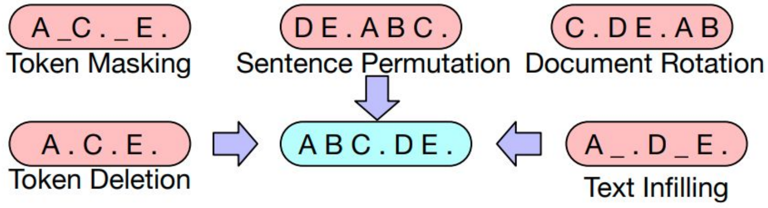


Figure 2: Transformations for noising the input that we experiment with. These transformations can be composed.

Reference: Lewis, Mike, et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." *arXiv preprint arXiv:1910.13461* (2019).
Paper link: https://arxiv.org/pdf/1910.13461.pdf

mBART: Multilingual Denoising Pre-training for Neural Machine Translation

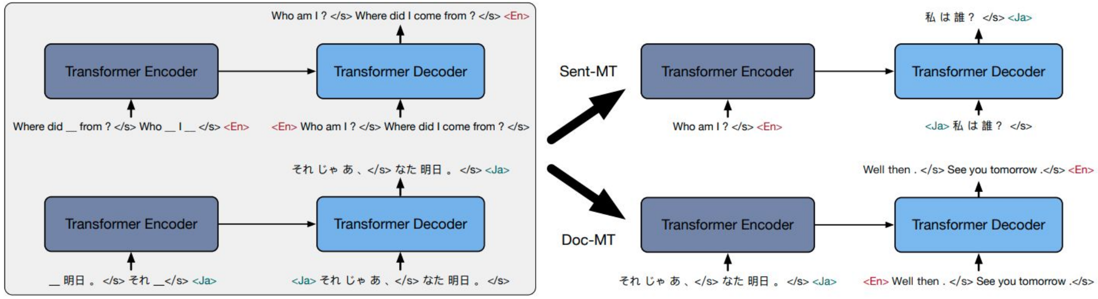
mBART Training Data: Common Crawl

Language Support: 25 (mBART25)and 50 (mBART50) languages

Model Architecture: Transformer large seq-to-seq model (12 layers), same as BART

mBART Objective Function:

1. Sentence permutation
2. word-span masking



Multilingual Denoising Pre-Training (mBART)   Fine-tuning on Machine Translation

Reference: Lewis, Mike, et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." *arXiv preprint arXiv:1910.13461* (2019).
Paper link: https://arxiv.org/pdf/1910.13461.pdf

Contextual Representations

- We have seen so many:
  - BERT, mBERT, BART, mBART, XLMR, MuRIL, ..
- How are they different?
  - Noising strategy, Objective functions, Architecture used
- Are these enough?
  - New models are still coming up!
- What next?
  - See how they cater to the needs of different low-resource languages
- What next?
  - Use these pre-trained embeddings for downstream tasks

Conclusion

- Context is key
- Contextual models are essential for NLU and NLG Tasks
- Different models are there
- They differ in architectures/task/objective-function
- Can be used as pretrained/seed models
- Pretrained models provide reasonable performance on variety of tasks
- Fine-tuning on downstream tasks are necessary
- What next?
    - Compressed models?
    - Support to more languages?
    - More natural-looking objectives?

Thank you!!

Dr. Maunendra Sankar Desarkar

Homepage: https://iith.ac.in/~maunendra/
Twitter: https://twitter.com/msades
LinkedIn: https://www.linkedin.com/in/maunendra-sankar-desarkar-6a89907/

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad