Extracurricular Materials

Probabilistic Transformer

Deciphering transformers with a probabilistic syntactic model

Kewei Tu (joint work with **Haoyi Wu**)

ShanghaiTech University

上海科技大学
ShanghaiTech University

NLP: the past and the present

- Once upon a time…
  - NLP ≈ probabilistic modeling of explicit linguistic structures (e.g., syntactic structures)
  - ✓ Mathematically well-founded, interpretable (white-box)
  - ✓ Linguistically principled

- Since the deep learning revolution…
  - NLP → pretrained transformers
  - ✓ Great performance!!
  - ✗ Black-box!
  - ☹ Linguistically murky

This work

- We propose probabilistic transformers
  - A (non-neural) probabilistic syntactic model
  - Yet, its computation graph is strikingly similar to a transformer!
- Goal?
  - A white-box transformer, which may…
  - …benefit the analysis and extension of transformers
  - …inspire future research of more interpretable & linguistically more principled neural models
  - …bridge the gap between traditional statistical NLP (incl. decades of syntax research) and modern neural NLP

Outline

- Preliminary
  - CRF, MFVI, unfolding as GNN
- Probabilistic transformers
  - Model
  - Inference
  - Extensions
- Similarities to transformers
- Experiments

Outline

- Preliminary
  - CRF, MFVI, unfolding as GNN
- Probabilistic transformers
  - Model
  - Inference
  - Extensions
- Similarities to transformers
- Experiments

Markov Random Fields (MRF)

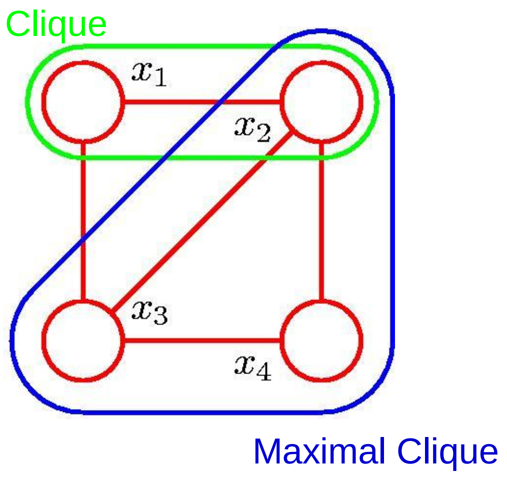
- MRF = undirected graph + potential functions
  - For each clique (or max clique), define a potential function
  - A joint probability is proportional to the product of potentials

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C)$$

where $\psi_C(\mathbf{x}_C)$ is the potential over clique C and

$$Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$$

is the normalization coefficient (aka. partition function).



Conditional Random Fields (CRF)

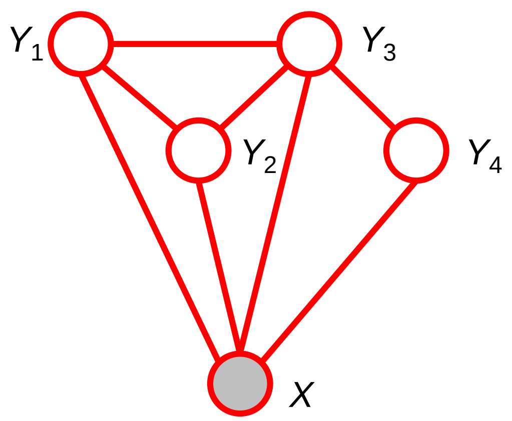
- An extension of MRF where everything is conditioned on an input

$$P(\mathbf{y}|x) = \frac{1}{Z(x)} \prod_C \psi_C(\mathbf{y}_C, x)$$

where $\psi_C(y_C, x)$ is the potential over clique C and

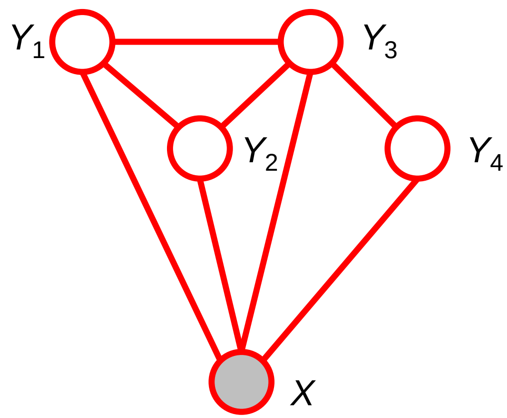
$$Z(x) = \sum_{\mathbf{y}} \prod_C \psi_C(\mathbf{y}_C, x)$$

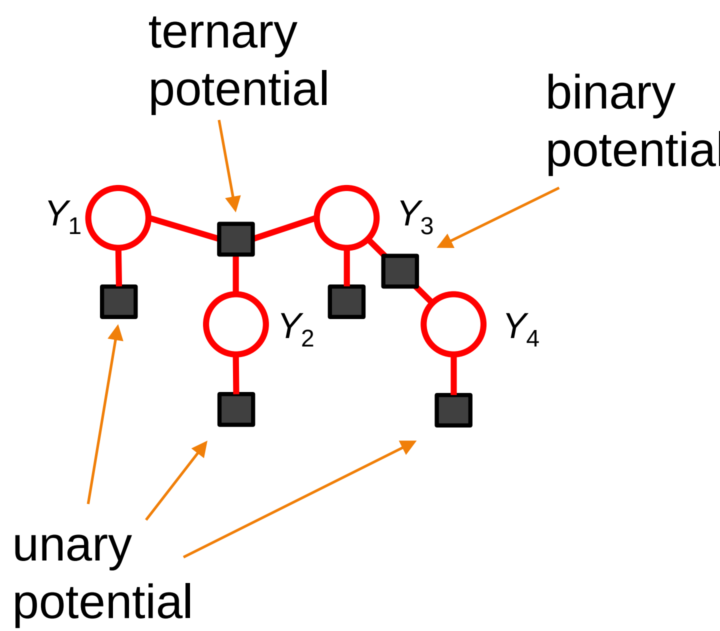
is the normalization coefficient.



Factor Graph

- A factor graph explicitly shows the potential functions (aka factors) in an MRF/CRF





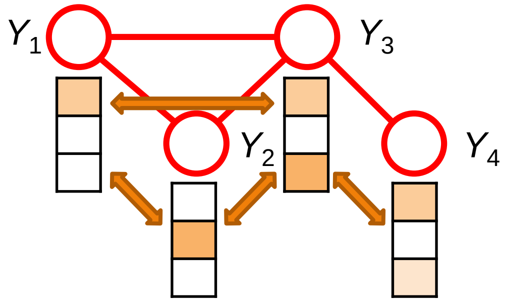
Inference over MRF/CRF

- Inference
  - Some variables are known (evidence)
  - Some variables are latent (we want to marginalize them)
  - Some variables are what we care about (query)

- Exact inference is hard or even intractable in general
- Iterative algorithms for approximate inference
  - Mean-field Variational Inference
  - Loopy Belief Propagation
  - …

Inference over MRF/CRF

- Iterative algorithms for approximate inference
- At each iteration:
  - Compute an intermediate vector (e.g., a discrete distribution) for each random variable…
  - …based on the vectors from the previous iteration
  - …following a fixed graph structure
  - …using fixed model parameters
  - …in a fully differentiable way



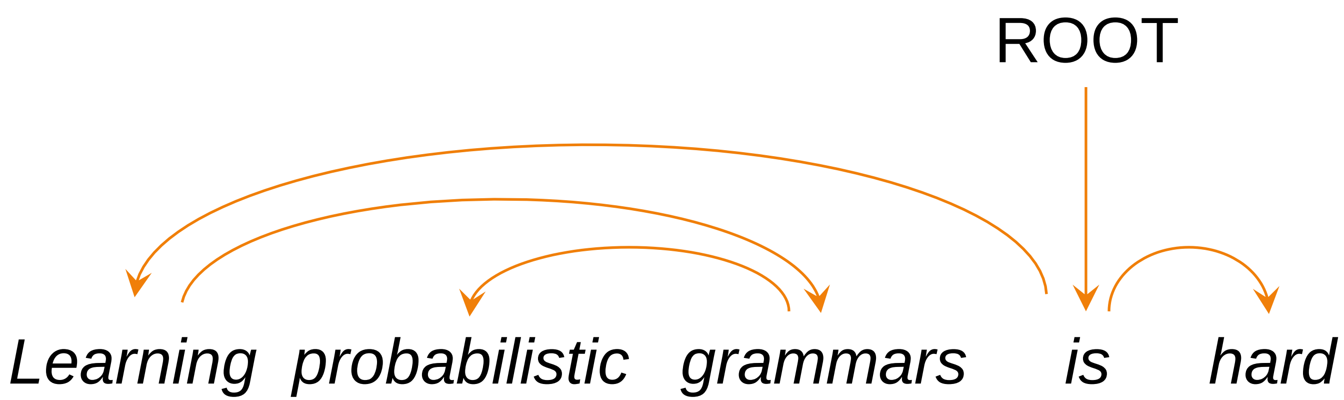
Inference can be unfolded as a Graph Neural Network!

Outline

- Preliminary
  - CRF, MFVI, unfolding as GNN
- Probabilistic transformers
  - Model
  - Inference
  - Extensions
- Similarities to transformers
- Experiments

Dependency parsing

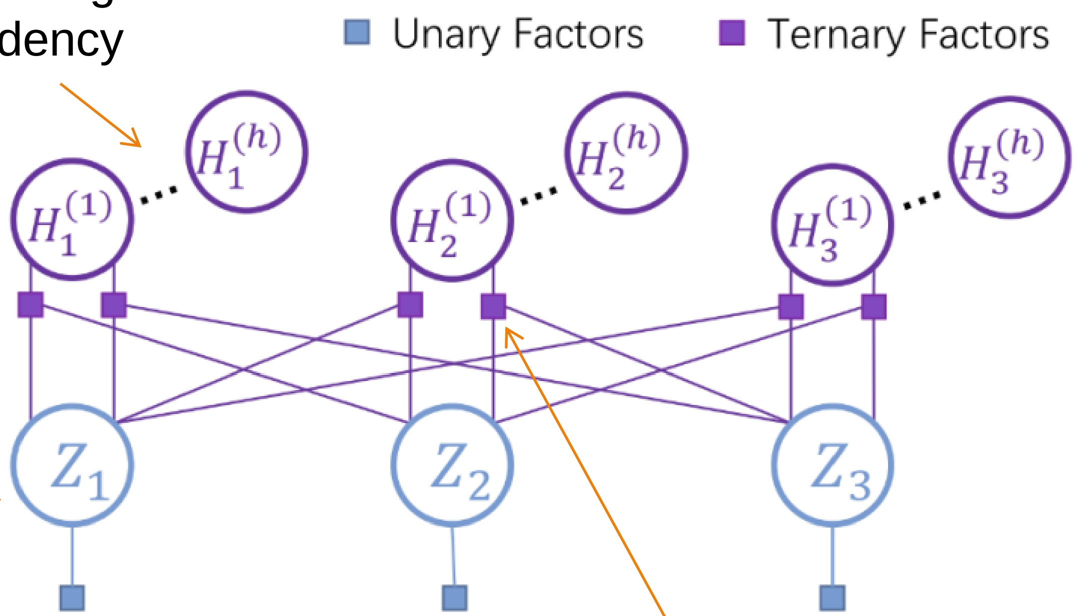
- Identify binary relations (i.e., dependencies) between words that form a tree



- Head-selection: a simplification of dependency parsing
  - Identify the parent word (i.e., dependency head) of each word
  - No tree constraint

Our CRF: head selection over latent word representation

h channels, allowing multiple dependency structures



$H_i \in \{1, \ldots, n\}$: index of the dependency head of word i

Z_i: a discrete variable, representing property of word i in the input sentence

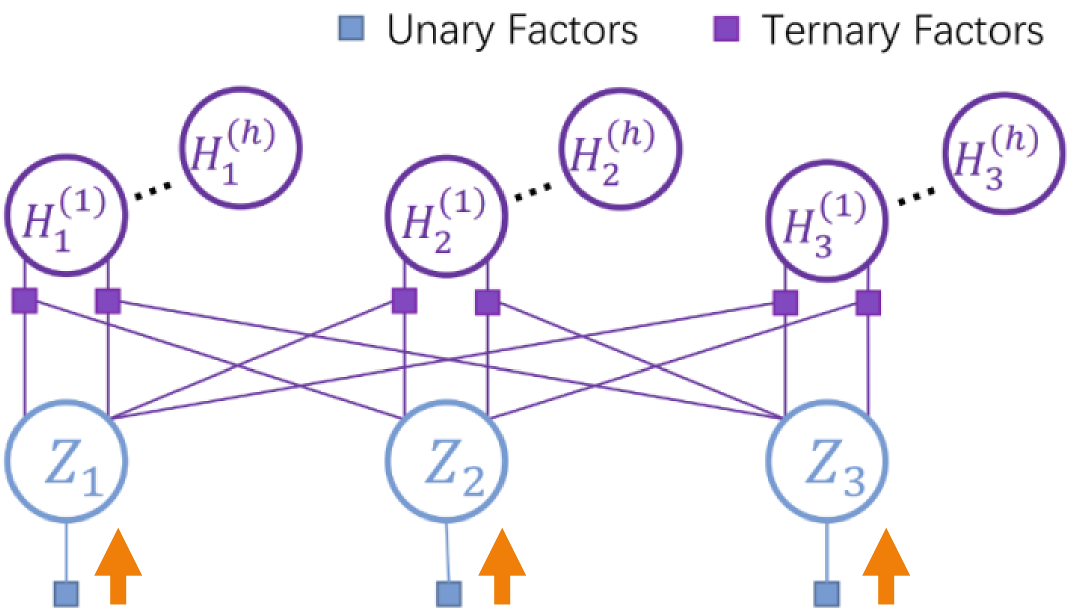
Unary factor: compatibility of Z_i and word i

Ternary factor: compatibility between Z_i and Z_j if word j is the dependency head of word i ($H_i = j$)

Mean Field Variational Inference (MFVI)

- Iteratively recompute marginal distribution $Q(\cdot)$ of each variable

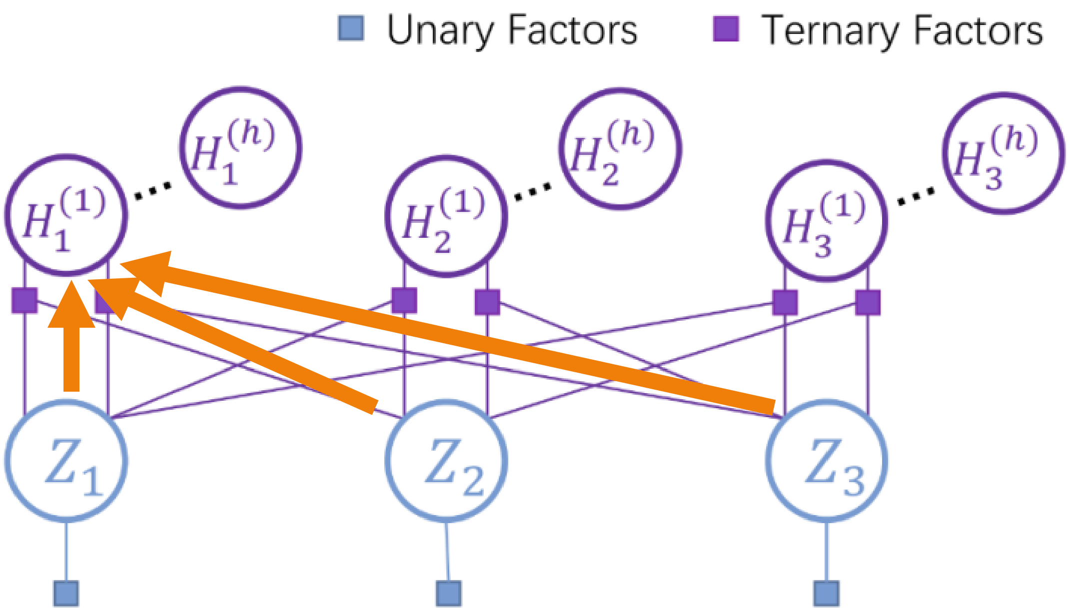
- Initialize $Q(Z_i)$



Mean Field Variational Inference (MFVI)

- Iteratively recompute marginal distribution $Q(\cdot)$ of each variable

- Initialize $Q(Z_i)$
- Repeat
  - Recompute $Q(H_i)$

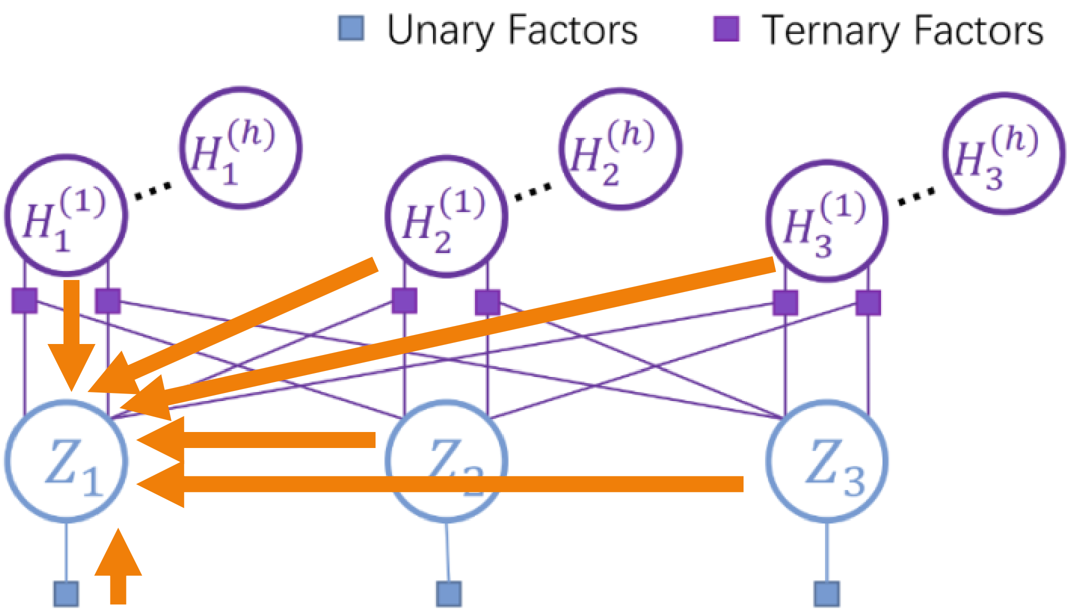


Mean Field Variational Inference (MFVI)

- Iteratively recompute marginal distribution $Q(\cdot)$ of each variable

See paper for all the math

- Initialize $Q(Z_i)$
- Repeat
  - Recompute $Q(H_i)$
  - Recompute $Q(Z_i)$

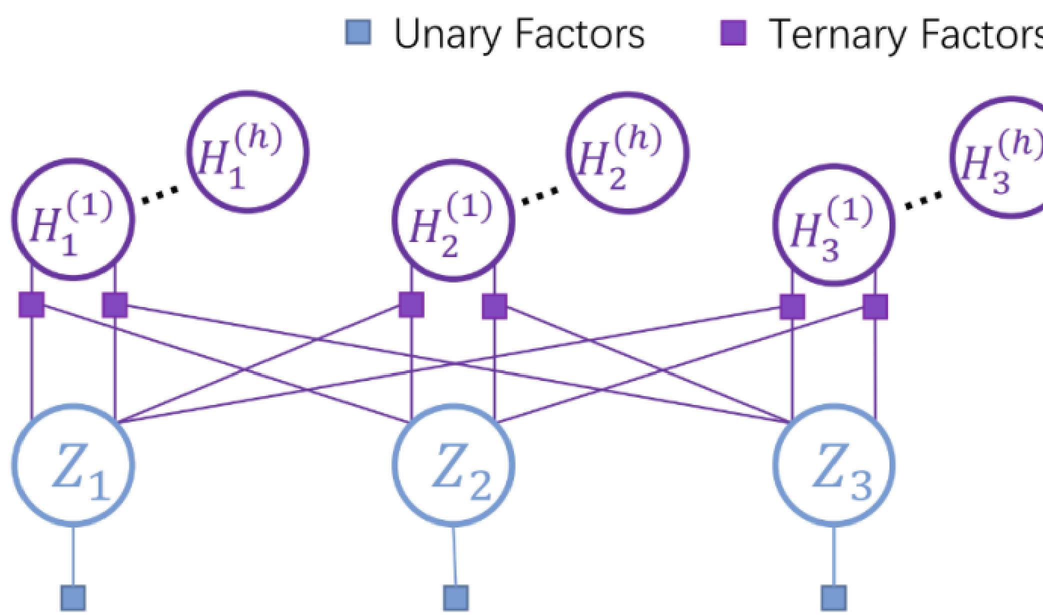


Mean Field Variational Inference (MFVI)

- Iteratively recompute marginal distribution $Q(\cdot)$ of each variable

- Initialize $Q(Z_i)$
- Repeat
  - Recompute $Q(H_i)$
  - Recompute $Q(Z_i)$

- $Q(Z_i)$ can be seen as a contextual representation of word i



Further refinements

- Entropic Frank-Wolfe algorithm
  - Generalization of MFVI
- Rank decomposition of ternary factor
  - $T\left(Z_i, Z_j\right) = \sum_r U(Z_i, r) \times V(Z_j, r)$
- Dependency root
- Incorporating word distance in ternary factors
- …

Learning

- Inference can be unfolded as a Graph Neural Network
- Learning can be done by back-propagation
  - Model parameters: unary & ternary factors
  - Objective function: MLM, downstream tasks, …

Outline

- Preliminary
  - CRF, MFVI, unfolding as GNN
- Probabilistic transformers
  - Model
  - Inference
  - Extensions
- Similarities to transformers
- Experiments

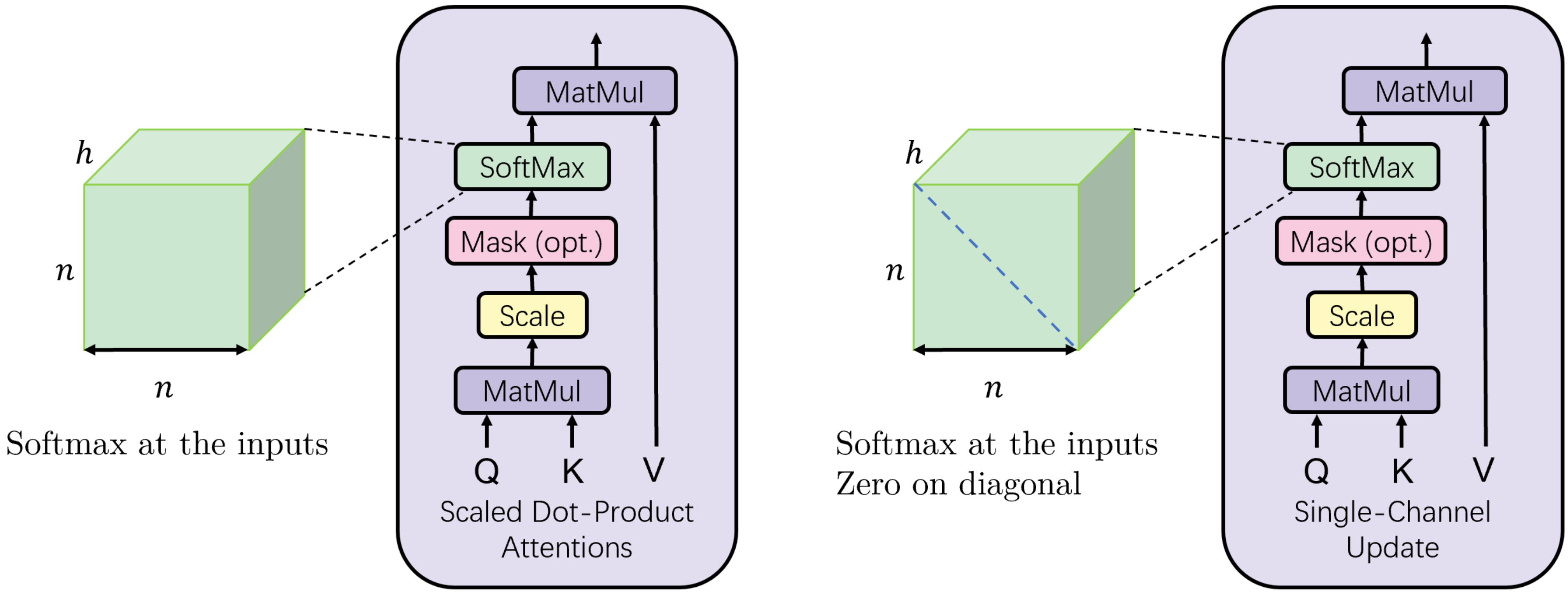
Similarities to transformers

- We compare the computation graph of MFVI on our CRF with transformers
  - Assumption: symmetric ternary factors
- Roughly speaking:

Our intermediate distributions $Q(H_i)$ over dependency heads	$\approx$	Self-attention scores in a transformer
Our intermediate distributions $Q(Z_i)$ over latent word representations	$\approx$	Intermediate word embeddings in a transformer

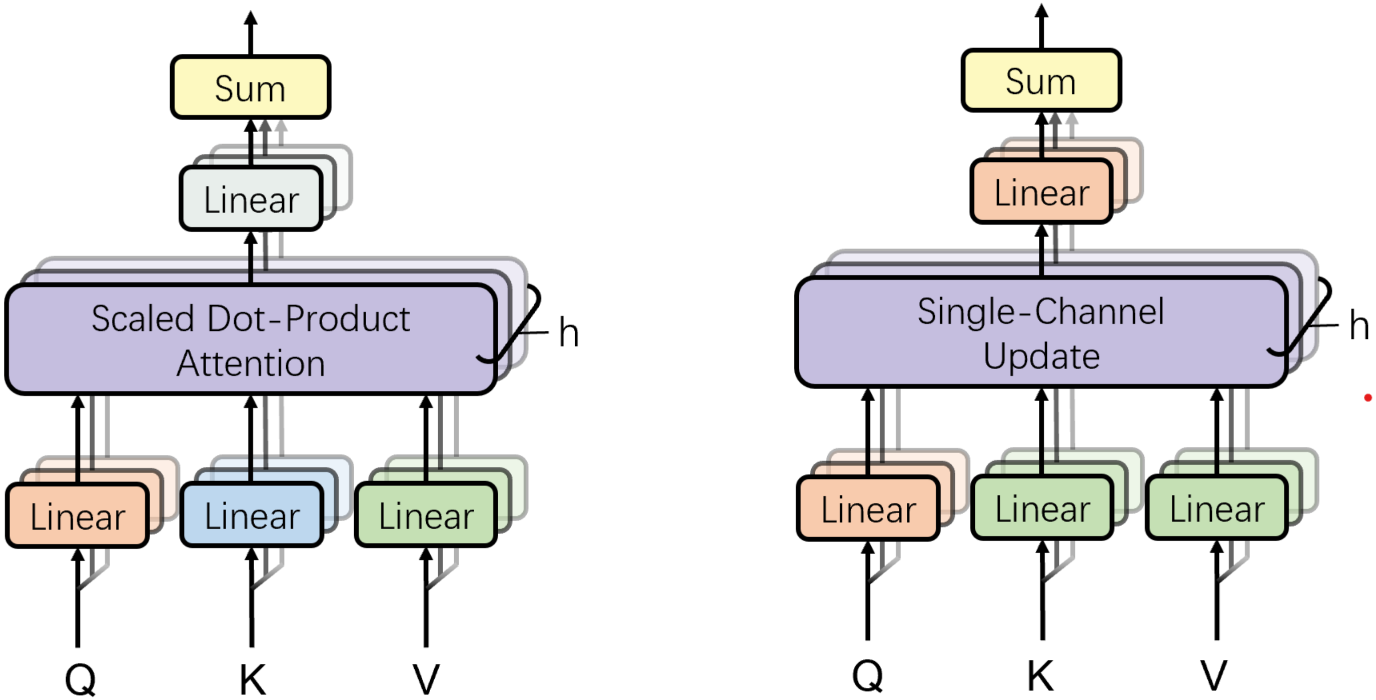
Similarities to transformers

- Single-Channel Update vs. Scaled Dot-Product Attention



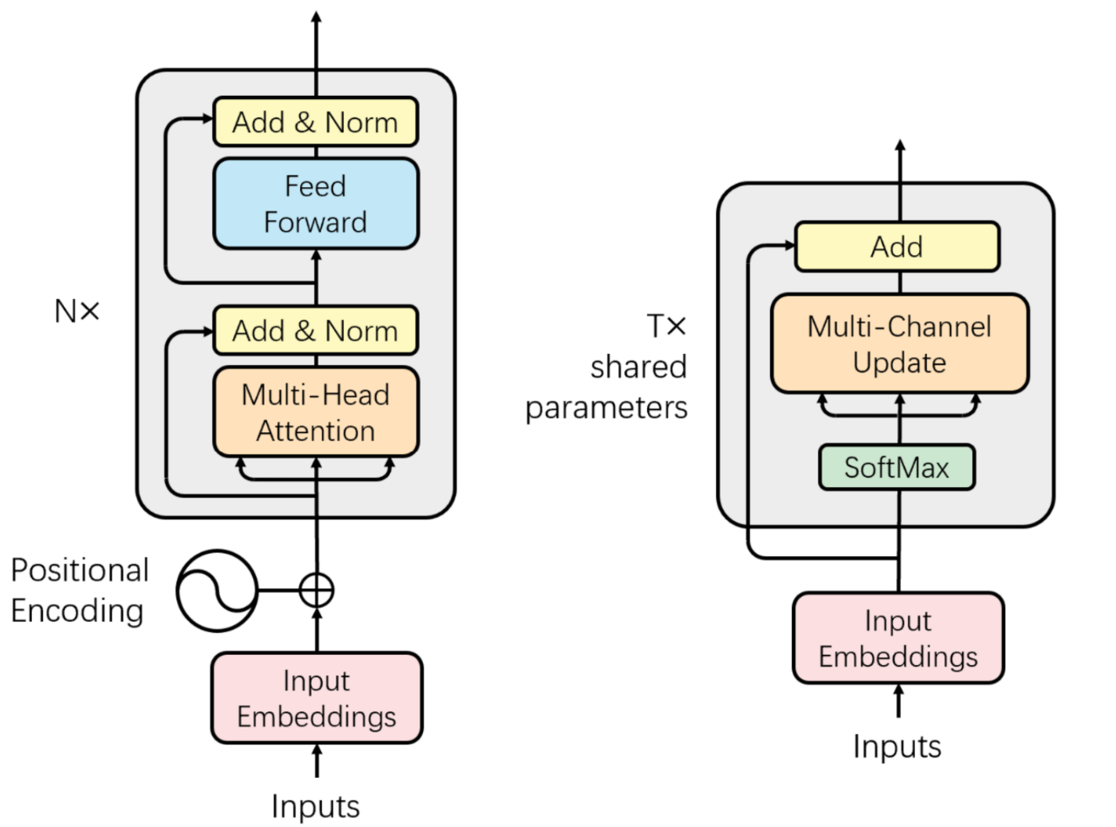
Similarities to transformers

- Multi-Channel Update vs. Multi-Head Attention



Similarities to transformers

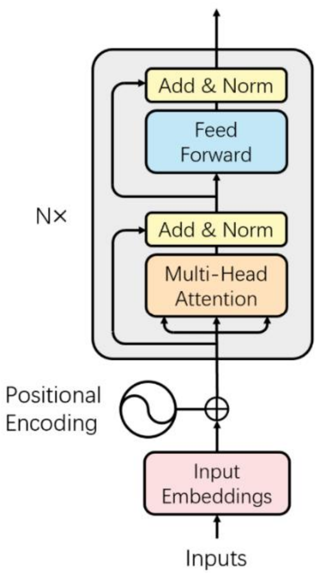
- Full Model Comparison

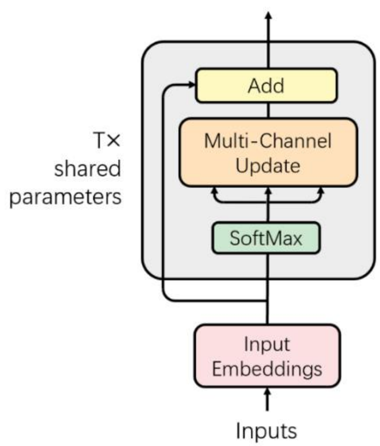


(a) Transformer (b) Probabilistic Transformer

Differences

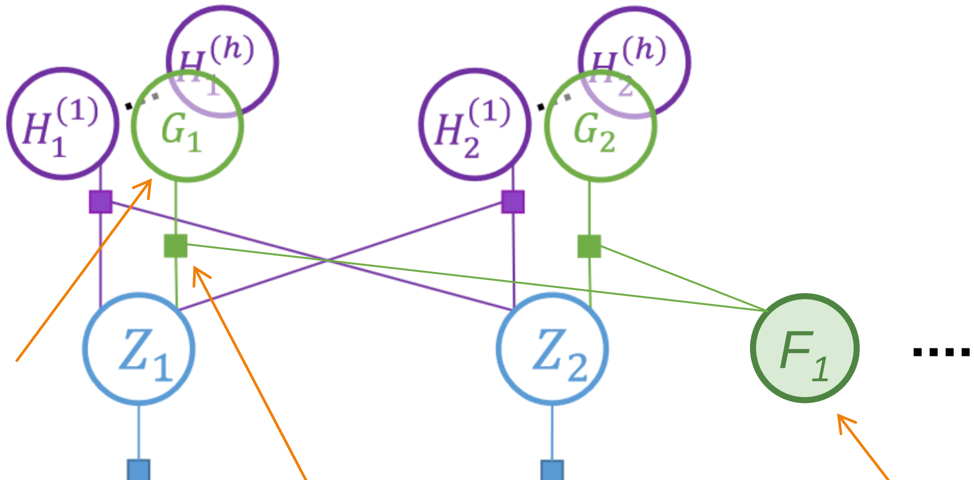
- Feed-forward vs. No feed-forward
- Residual connection vs. Adding input
- Post layer norm vs. Softmax before each layer
  - Similar to pre-LN
- No parameter sharing vs. Layer-wise parameter sharing
  - Similar to Universal Transformer, ALBERT, …





Feed-forward layer

- Adding m global topic variables in our CRF



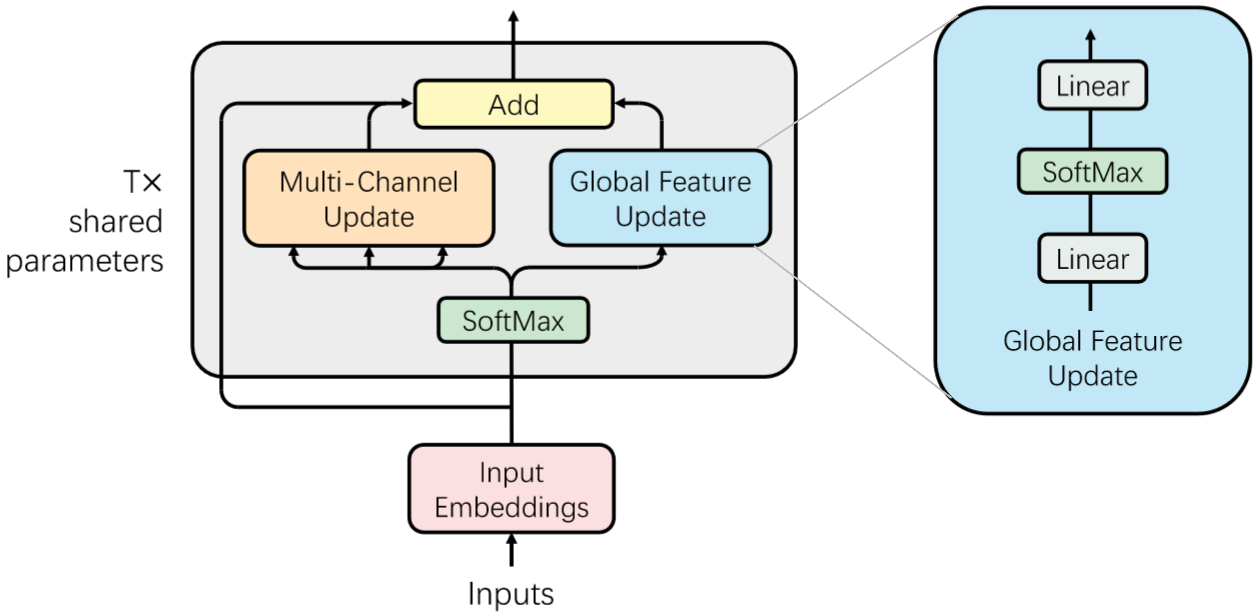
$G_i \in \{1, \dots, m\}$: index of the topic of word i

Ternary factor: compatibility between Z_i and F_j if word i belongs to topic j ($G_i = j$)

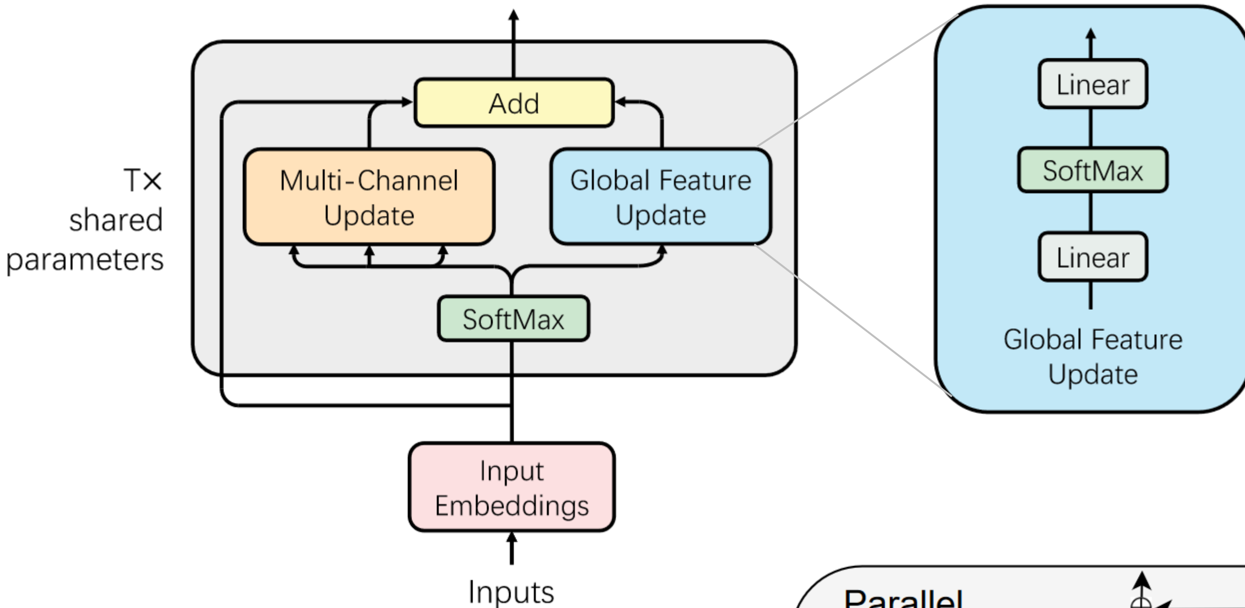
F_i: a discrete global topic variable

Feed-forward layer

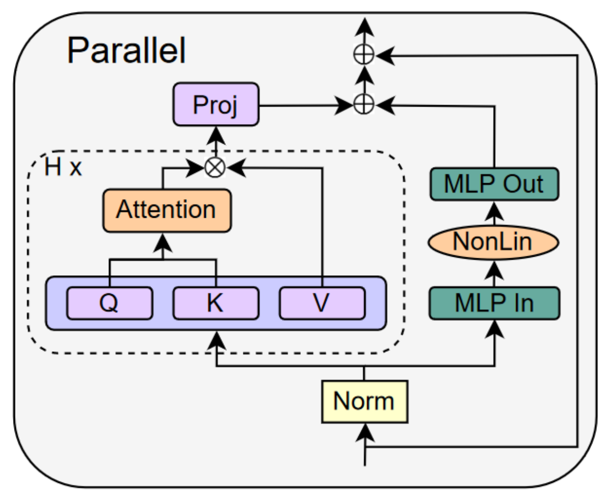
- MFVI computation graph



Feed-forward layer



Similar to transformer parallel block
(e.g., GPT-J-6B)



Outline

- Preliminary
  - CRF, MFVI, unfolding as GNN
- Probabilistic transformers
  - Model
  - Inference
  - Extensions
- Similarities to transformers
- Empirical evaluation

Empirical evaluation

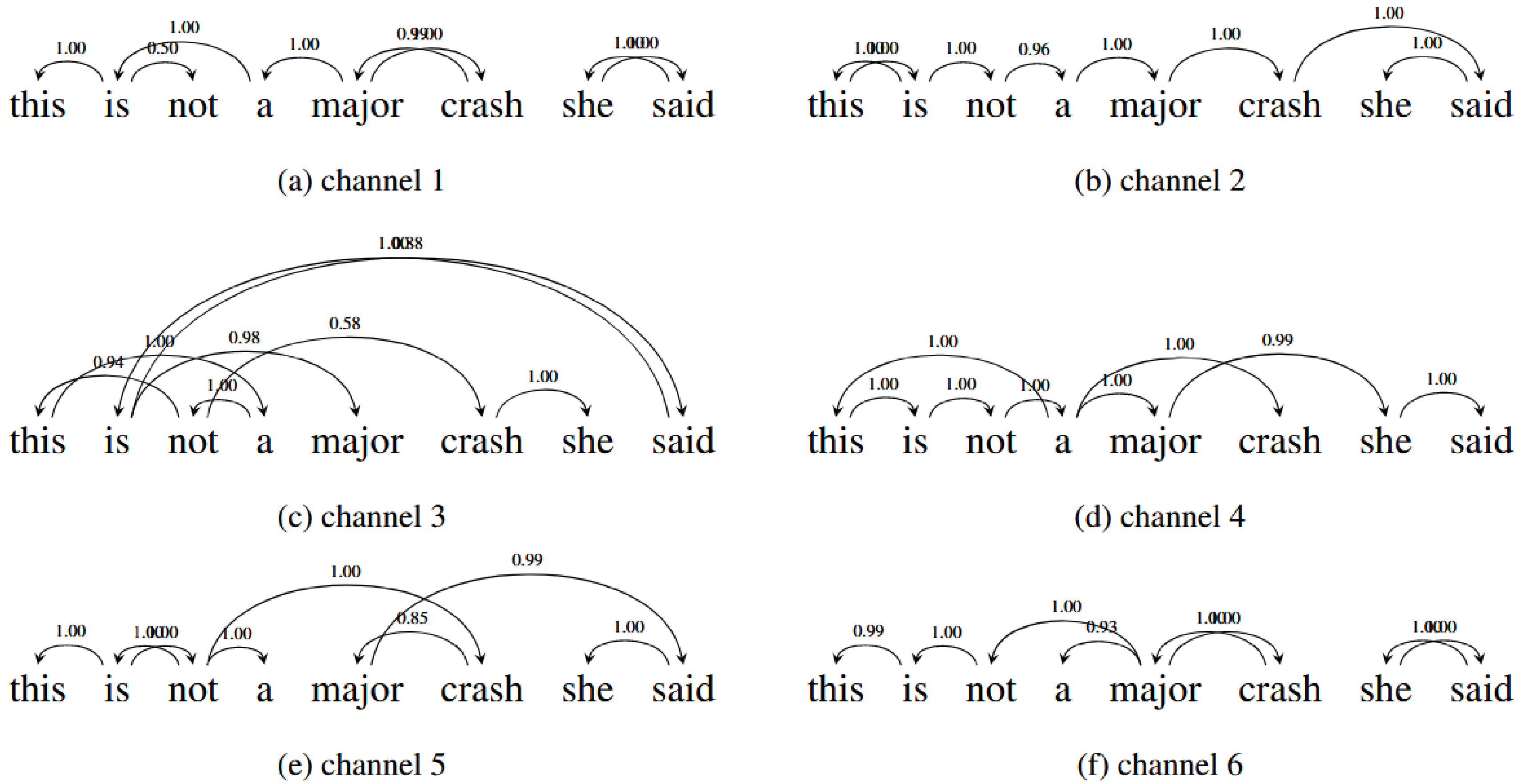
- Masked Language Modeling (MLM)
- Part-of-Speech Tagging (POS)
- Named Entity Recognition (NER)
- Classification (CLS)
- Syntactic Test

Empirical evaluation

Task	Dataset	Metric	Transformer	Probabilistic Transformer
MLM	PTB BLLIP	Perplexity	58.43 ± 0.58 101.91 ± 1.40	62.86 ± 0.40 123.18 ± 1.50
POS	PTB UD	Accuracy	96.44 ± 0.04 91.17 ± 0.11	96.29 ± 0.03 90.96 ± 0.10
NER	CoNLL-2003	F1	74.02 ± 1.11	75.47 ± 0.35
CLS	SST-2 SST-5	Accuracy	82.51 ± 0.26 40.13 ± 1.09	82.04 ± 0.88 42.77 ± 1.18
Syntactic Test	COGS	Sentence-level Accuracy	82.05 ± 2.18	84.60 ± 2.06

- In most cases, our best model is about 1/5~1/2 in size of the best transformer.
- For larger datasets, our models clearly underperform transformers.

Inferred dependency structures (MLM)



(a) channel 1
(b) channel 2
(c) channel 3
(d) channel 4
(e) channel 5
(f) channel 6

Outline

- Preliminary
  - CRF, MFVI, unfolding as GNN
- Probabilistic transformers
  - Model
  - Inference
  - Extensions
- Similarities to transformers
- Empirical evaluation
- Summary

Summary

- Probabilistic transformers: a white-box transformer
  - A purely probabilistic syntactic model
  - Approximate inference using mean field variational inference
  - Its computation graph is very similar to a transformer

- We hope our work could:
  - benefit the analysis and extension of transformers
  - inspire future research of more interpretable & linguistically more principled neural models
  - bridge the gap between traditional statistical NLP (incl. decades of syntax research) and modern neural NLP

Summary

- Paper
  - https://aclanthology.org/2023.findings-acl.482/
- Code
  - https://github.com/whyNLP/Probabilistic-Transformer

Thank you!

Q&A