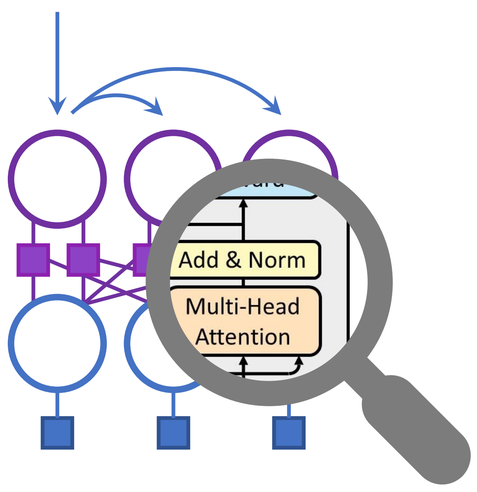
Probabilistic Transformer:

A Probabilistic Dependency Model for Contextual Word Representation

Haoyi Wu, Kewei Tu
School of Information Science and Technology, ShanghaiTech University



TL;DR: We build a (non-neural) probabilistic syntactic model and find striking similarity between its computation graph and the transformer!

Why it matters:

Syntactic structures are no longer deemed essential by many in modern neural NLP. Our work shows that syntactic structures may still have an important role to play. We hope our work could:

- benefit the analysis and extension of transformers
- inspire future research of linguistically more principled neural models
- bridge the gap between traditional statistical NLP (incl. decades of syntax research) and modern neural NLP

The Conditional Random Field

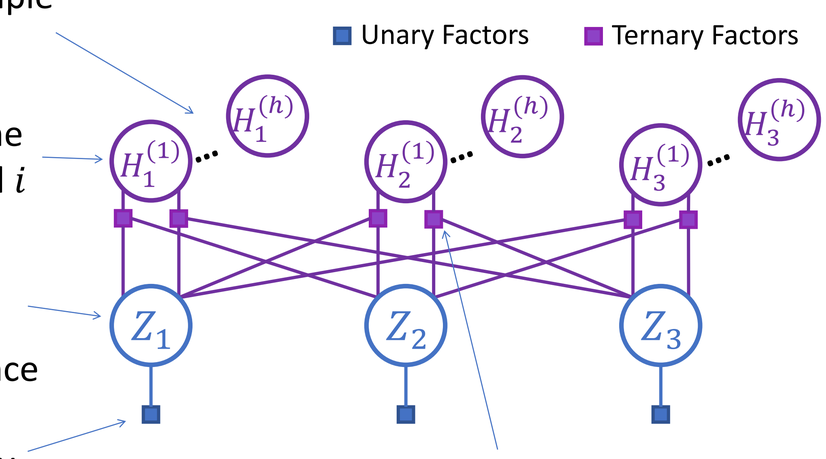
h channels, allowing multiple dependency structures

$H_i \in \{1, \dots, n\}$: 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

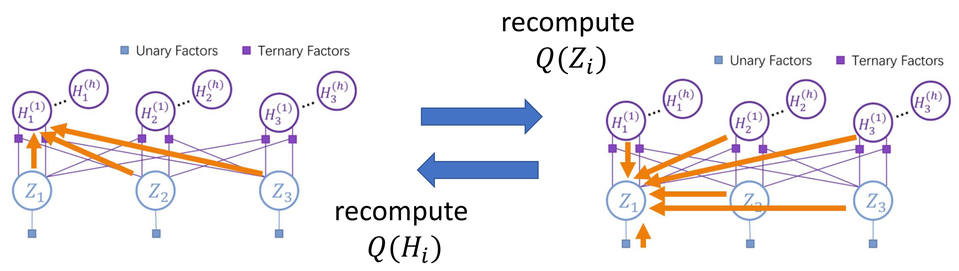
Factor Graph of Probabilistic Transformers



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

Inference by Mean Field Variational Inference

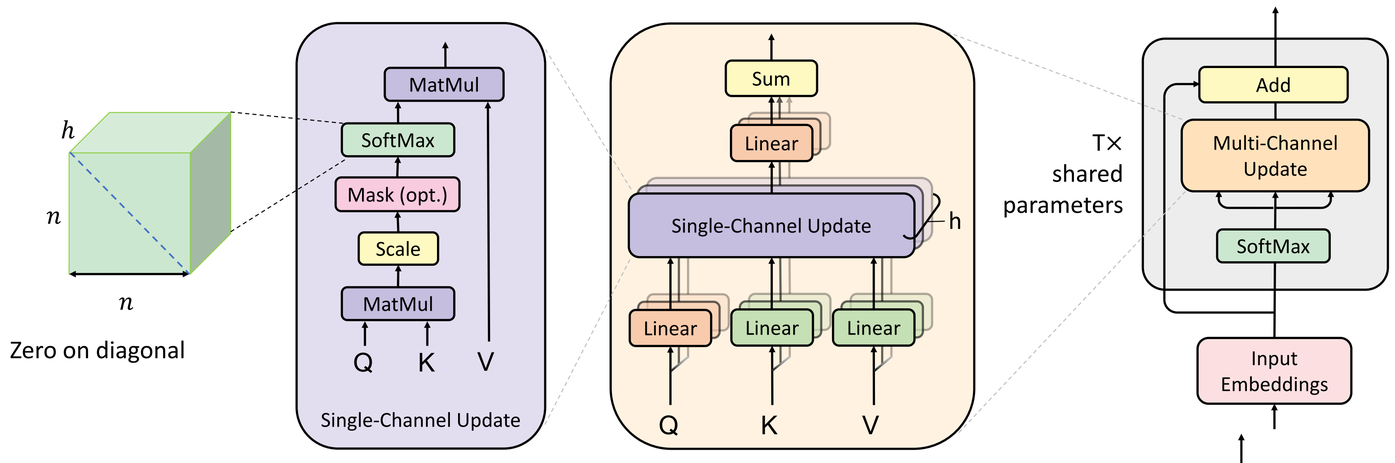
Iteratively recompute marginal distribution $Q(\cdot)$ of each variable



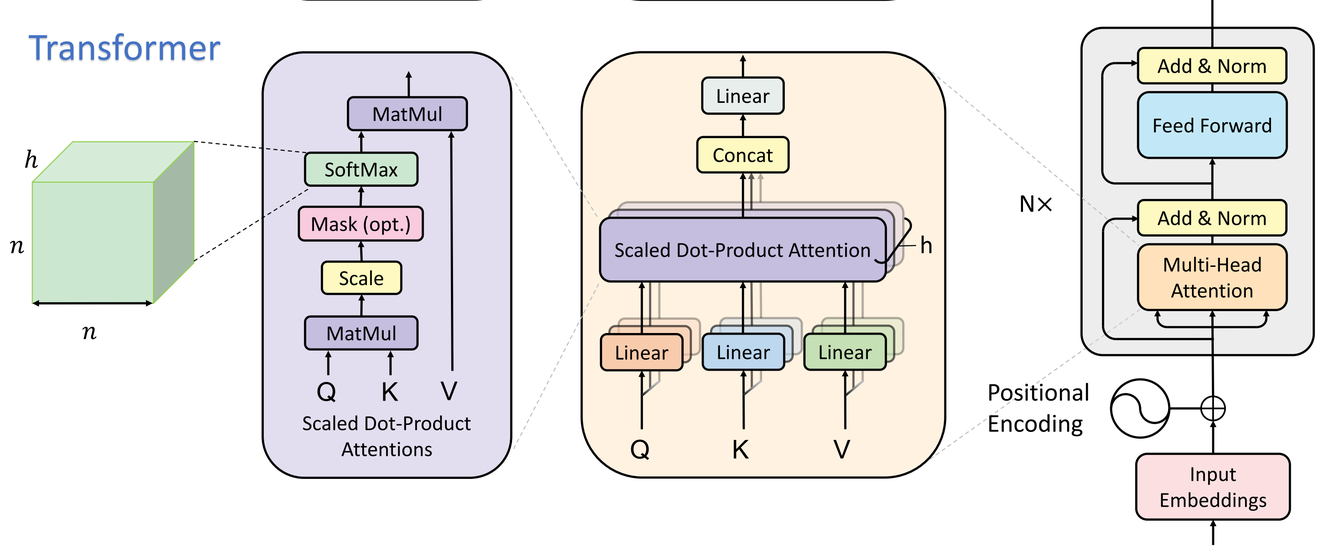
After a few iterations, $Q(Z_i)$ can be seen as a contextual representation of word i.

The computation is fully differentiable and can be seen as a graph neural network. Learning can be done by back-propagation (e.g., using a MLM objective).

Probabilistic Transformer



Transformer



Similarities to Transformers

Assumption: symmetric ternary factors

Roughly speaking: $Q(H_i)$ corresponds to self-attention scores and $Q(Z_i)$ corresponds to intermediate word embeddings in a transformer.

Single-channel update ≈ scaled dot-product attention
The only difference is that the diagonal of the tensor after softmax is zero in our model because the head of a word cannot be itself.

Multi-channel update ≈ multi-head attention
The main difference is that in our model, some of the weight tensors are tied.

The full computation graphs
Very similar with a few interesting differences: FF layer, residual connection, post layer norm, layer-wise parameter sharing.

Experiment Results

- MLM: Masked Language Modeling (lower is better)
- POS: Part-of-Speech Tagging
- NER: Named Entity Recognition
- CLS: Classification

Probabilistic transformers have competitive performance with transformers.

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

Code

Paper

wuhy1@shanghaitech.edu.cn

https://github.com/whyNLP/ Probabilistic-Transformer

上海科技大学
ShanghaiTech University