Paper Reading: Phrase-Based Attentions

Phi Xuan Nguyen, Shafiq Joty

Nanyang Technological University, Singapore

Motivation

- Success of phrase-based statistical machine translation.
- In existing NMT systems. most of them use token based attention and ignore the importance of phrasal alignments.

Phase-based Attentions

- Phrase-based Attention methods (How to achieve phrasal attention):
  1. Key-Value Convolution
  2. Quesy-as-Kernel Convolution

- Multi-Headed Phrasal Attention (How to use in multi-head attention framework):
  1. Homogeneous n-gram Attention
  2. Heterogeneous n-gram Attention
  3. Interleaved Phrases to Phrase Heterogeneous Attention

One-dimensional Convolutional operation

The convolutional operator applied to each token x_t with corresponding vector representation $x_t \in R^{d_1}$ as:

$$o_t = \mathbf{w} \oplus_{k=0}^{n} \mathbf{x}_{t \pm k} \tag{4}$$

where $\oplus$ denotes vector concatenation, $\mathbf{w} \in \mathbb{R}^{n \times d_1}$ is the weight vector (*a.k.a.* kernel), and n is the window size. We repeat this process with d_2 different weight vectors to get a d_2-dimensional latent representation for each token x_t. We will use the notation $\mathrm{Conv}_n(\boldsymbol{X}, \boldsymbol{W})$ to denote the convolution operation over an input sequence $\boldsymbol{X}$ with window size n and kernel weights $\boldsymbol{W} \in \mathbb{R}^{n \times d_1 \times d_2}$.

Key-Value Convolution

Use trainable kernel parameters W_k and W_v to compute the latent representation of n-gram sequence using convolution operation over key and value vectors.

$$\textsc{ConvKV}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathcal{S}\left(\frac{(\boldsymbol{Q}\boldsymbol{W}_q)\mathrm{Conv}_n(\boldsymbol{K}, \boldsymbol{W}_k)^T}{\sqrt{d_k}}\right) \mathrm{Conv}_n(\boldsymbol{V}, \boldsymbol{W}_v)$$

Where S is the softmax function, $W_q \in R^{d_q * d_k}$, $W_k \in R^{n * d_k * d_k}$, $W_v \in R^{n * d_v * d_v}$, are the respective kernel weights for Q, K and V. The queries do not interact directly with the keys to learn the attention weights, instead the model relies on the kernel weights to learn n-gram patterns.

Query-as-Kernel Convolution

In order to allow the queries to directly and dynamically influence the word order of phrasal keys and values, the Query-as-Kernel Convolution is proposed.

$$\textsc{QueryK}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathcal{S}\left(\frac{\mathrm{Conv}_n(\boldsymbol{K}\boldsymbol{W}_k, \boldsymbol{Q}\boldsymbol{W}_q)}{\sqrt{d_k * n}}\right) \mathrm{Conv}_n(\boldsymbol{V}, \boldsymbol{W}_v)$$

Where $W_q \in R^{n*d_q*d_k}$, $W_k \in R^{d_k*d_k}$, $W_v \in R^{n*d_v*d_v}$ are trainable weights.

Multi-Headed Phrasal Attention: Homogeneous n-gram attention

Each head attends to one particular n-gram type (n=1, 2, . . . , N). For instance, figure shows a homogeneous structure, where the first four heads attend to unigrams, and the last four attend to bigrams.

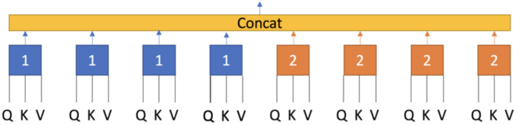

Figure 1: Homogeneous multi-head attention, where each attention head features one n-gram type. In this example, there are eight heads, which are distributed equally between unigrams and bigrams.

Heterogeneous n-gram attention

Heterogeneous n-gram attention allows the query to freely attend to all types of n-grams simultaneously.

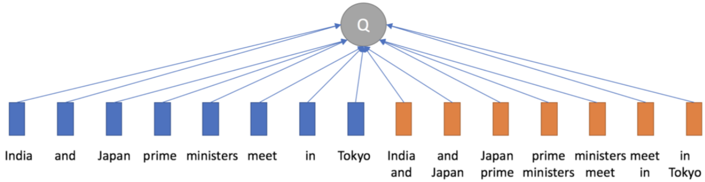

Figure 2: Heterogeneous n-gram attention for each attention head. Attention weights and vectors are computed from all n-gram types simultaneously.

For CONVKV technique in Equation 5, the attention output is given by:

$$\mathcal{S}\left(\frac{(\boldsymbol{Q}\boldsymbol{W}_q)[(\boldsymbol{K}\boldsymbol{W}_{k,1})^T; \mathrm{Conv}_2(\boldsymbol{K}, \boldsymbol{W}_{k,2})^T; ...]}{\sqrt{d_k}}\right)[(\boldsymbol{V}\boldsymbol{W}_{v,1}); \mathrm{Conv}_2(\boldsymbol{V}, \boldsymbol{W}_{v,2}); ...] \tag{7}$$

For QUERYK technique (Equation 6), the attention output is given as follows:

$$\mathcal{S}\left(\left[\frac{(\boldsymbol{Q}\boldsymbol{W}_{q,1})(\boldsymbol{K}\boldsymbol{W}_{k,1})^T}{\sqrt{d}}; \frac{\mathrm{Conv}_2(\boldsymbol{K}\boldsymbol{W}_{k,2}, \boldsymbol{Q}\boldsymbol{W}_{q,2})}{\sqrt{d * n_2}}; ...\right]\right)[(\boldsymbol{V}\boldsymbol{W}_{v,1}); \mathrm{Conv}_2(\boldsymbol{V}, \boldsymbol{W}_{v,2}); ...] \tag{8}$$

Interleaved Phrases to Phrase Heterogeneous Attention

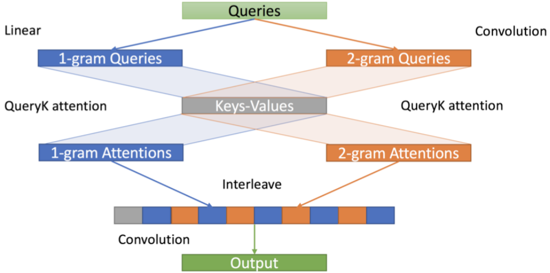

Figure 3: Interleaved phrase-to-phrase heterogeneous attention. The queries are first transformed into unigram and bigram representations, which in turn then attend independently on key-value pairs to produce unigram and bigram attention vectors. The attention vectors are then interleaved before passing through another convolutional layer.

Experiment results

Model	Technique	N-grams	En-De	De-En
Transformer (Base, 1 GPU)	-	-	26.07	29.82
Transformer (Base, 8 GPUs) Vaswani et al. (2017)	-	-	27.30	—-
Homogeneous	ConvKV	44	26.60 (+0.53)	30.17 (+0.36)
Homogeneous	QueryK	44	26.78 (+0.71)	30.03 (+0.21)
Heterogeneous	ConvKV	12	27.04 (+0.97)	30.09 (+0.27)
Heterogeneous	QueryK	12	26.95 (+0.88)	30.20 (+0.38)
Interleaved	ConvKV	12	27.33 (+1.26)	30.17 (+0.36)
Interleaved	QueryK	12	**27.40 (+1.33)**	**30.30 (+0.48)**

Table 1: BLEU (cased) scores on WMT'14 testset for English-German and German-English. For homogeneous models, the **N-grams** column denotes how we distribute the 8 heads to different n-gram types; *e.g.*, 323 means 3 unigram heads, 2 bigram heads and 3 trigram heads. For heterogeneous, the numbers indicate the phrase lengths of the collection of n-gram components jointly attended by each head; *e.g.*, 12 means attention scores are computed across unigram and bigram logits.

Experiment results

Model	Technique	Uni-bi-grams		Uni-bi-tri-grams	
		Head/N-gram	BLEU	Head/N-gram	BLEU
Homogeneous	ConvKV	44	26.60	323	26.55
Homogeneous	QueryK	44	26.78	323	26.86
Heterogeneous	ConvKV	12	27.04	123	27.15
Heterogeneous	QueryK	12	26.95	123	27.09

Table 2: BLEU scores for models that use only uni-bi-grams vs. the ones that use uni-bi-tri-grams.

Conclusion

- Embed phrases into attention modules
- n-gram information can be used.