# PHRASE-BASED ATTENTIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Most state-of-the-art neural machine translation systems, despite being different in architectural skeletons (*e.g.,* recurrence, convolutional), share an indispensable feature: the Attention. However, most existing attention methods are token-based and ignore the importance of phrasal alignments, the key ingredient for the success of phrase-based statistical machine translation. In this paper, we propose novel phrase-based attention methods to model n-grams of tokens as attention entities. We incorporate our phrase-based attentions into the recently proposed Transformer network, and demonstrate that our approach yields improvements of 1.3 BLEU for English-to-German and 0.5 BLEU for German-to-English translation tasks on WMT newstest2014 using WMT'16 training data.

## 1 INTRODUCTION

**Neural Machine Translation (NMT)** has established breakthroughs in many different translation tasks, and has quickly become the standard approach to machine translation. NMT offers a simple encoder-decoder architecture that is trained end-to-end. Most NMT models (except a few like (Kaiser & Bengio, 2016) and (Huang et al., 2018)) possess attention mechanisms to perform alignments of the target tokens to the source tokens. The attention module plays a role analogous to the word alignment model in Statistical Machine Translation or SMT (Koehn, 2010). In fact, the transformer network introduced recently by Vaswani et al. (2017) achieves state-of-the-art performance in both speed and BLEU scores (Papineni et al., 2002) by using only attention modules.

On the other hand, phrasal interpretation is an important aspect for many language processing tasks, and forms the basis of Phrase-Based Machine Translation (Koehn, 2010). Phrasal alignments (Koehn et al., 2003) can model one-to-one, one-to-many, many-to-one, and many-to-many relations between source and target tokens, and use local context for translation. They are also robust to non-compositional phrases. Despite the advantages, the concept of **phrasal attentions** has largely been neglected in NMT, as most NMT models generate translations token-by-token autoregressively, and use the token-based attention method which is order invariant. Therefore, the intuition of phrase-based translation is vague in existing NMT systems that solely depend on the underlying neural architectures (recurrent, convolutional, or self-attention) to incorporate compositional information.

In this paper, we propose phrase-based attention methods for phrase-level alignments in NMT. Specifically, we propose two novel phrase-based attentions, namely **CONVKV** and **QUERYK**, designed to assign attention scores directly to phrases in the source and compute phrase-level attention vector for the target. We also introduce three new attention structures, which apply these methods to conduct phrasal alignments. Our **homogeneous** and **heterogeneous** attention structures perform *token-to-token* and *token-to-phrase* mappings, while the **interleaved heterogeneous** attention structure models all *token-to-token, token-to-phrase, phrase-to-token*, and *phrase-to-phrase* alignments.

To show the effectiveness of our approach, we apply our phrase-based attentions to all multi-head attention layers in the Transformer network. Our experiments on WMT'14 translation tasks between English and German show 1.3 BLEU improvement for English-to-German and 0.5 BLEU for German-to-English, compared to the Transformer network trained in identical settings.

## 2 BACKGROUND

Most NMT models adopt an encoder-decoder framework, where the encoder network first transforms an input sequence of symbols $\boldsymbol{x} = (x_1, x_2 \dots x_n)$ to a sequence of continuous represen-

tations $\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots \boldsymbol{z}_n)$, from which the decoder generates a target sequence of symbols $\boldsymbol{y} = (y_1, y_2, \ldots y_n)$ autoregressively, one element at a time. Recurrent seq2seq models with diverse structures and complexity (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Wu et al., 2016) are the first to yield state-of-the-art results. Convolutional seq2seq models (Kalchbrenner et al., 2016; Gehring et al., 2017; Kaiser et al., 2018) alleviate the drawback of sequential computation of recurrent models and leverage parallel computation to reduce training time.

The recently proposed **Transformer** network (Vaswani et al., 2017) structures the encoder and the decoder entirely with stacked self-attentions and cross-attentions (only in the decoder). In particular, it uses a multi-headed, scaled multiplicative attention defined as follows:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v) = \mathcal{S}(\frac{(\boldsymbol{Q}\boldsymbol{W}_q)(\boldsymbol{K}\boldsymbol{W}_k)^T}{\sqrt{d_k}})(\boldsymbol{V}\boldsymbol{W}_v) \quad (1)$$

$$\text{Head}^i = \text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{W}_q^i, \boldsymbol{W}_k^i, \boldsymbol{W}_v^i) \text{ for } i = 1 \ldots h \quad (2)$$

$$\text{AttentionOutput}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{W}) = \text{concat}(\text{Head}^1, \text{Head}^2, \ldots, \text{Head}^h)\boldsymbol{W} \quad (3)$$

where $\mathcal{S}$ is the *softmax* function, $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$ are the matrices with query, key, and value vectors, respectively, $d_k$ is the dimension of the query/key vectors; $\boldsymbol{W}_q^i$, $\boldsymbol{W}_k^i$, $\boldsymbol{W}_v^i$ are the head-specific weights for query, key, and value vectors, respectively; and $\boldsymbol{W}$ is the weight matrix that combines the outputs of the heads. The attentions in the encoder and decoder are based on **self-attention**, where all of $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ come from the output of the previous layer. The decoder also has **cross-attention**, where $\boldsymbol{Q}$ comes from the previous decoder layer, and the $\boldsymbol{K}$-$\boldsymbol{V}$ pairs come from the encoder. We refer readers to (Vaswani et al., 2017) for further details of the network design.

One crucial issue with the attention mechanisms employed in the Transformer network as well as other NMT architectures (Luong et al., 2015; Gehring et al., 2017) is that they are order invariant locally and globally. If this problem is not tackled properly, the model may not learn the sequential characteristics of the data. RNN-based models (Bahdanau et al., 2015; Luong et al., 2015) tackle this issue with a recurrent encoder and decoder, CNN-based models like (Gehring et al., 2017) use position embeddings, while the Transformer uses positional encoding. Another limitation is that these attention methods attend to tokens, and play a role analogous to word alignment models in SMT. It is, however, well admitted in SMT that phrases are better than words as translation units (Koehn, 2010). Without specific attention to phrases, a particular attention function has to depend entirely on the token-level *softmax* scores of a phrase for phrasal alignment, which is not robust and reliable, thus making it more difficult to learn the mappings.

There exists some research on phrase-based decoding in NMT framework. For example, Huang et al. (2018) proposed a phrase-based decoding approach based on a soft reordering layer and a Sleep-WAke Network (SWAN), a segmentation-based sequence model proposed by Wang et al. (2017a). Their decoder uses a recurrent architecture without any attention on the source. Tang et al. (2016) and Wang et al. (2017b) used an external phrase memory to decode phrases for a Chinese-to-English translation task. In addition, hybrid search and PBMT were introduced to perform phrasal translation in (Dahlmann et al., 2017). Nevertheless, to the best of our knowledge, our work is the first to embed phrases into attention modules, which thus propagate the information throughout the entire end-to-end Transformer network, including the encoder, decoder, and the cross-attention.

## 3 Multi-head Phrase-based Attention

In this section, we present our proposed methods to compute attention weights and vectors based on n-grams of queries, keys, and values. We compare and discuss the pros and cons of these methods. For simplicity, we describe them in the context of the Transformer network; however, it is straightforward to apply them to other architectures such as RNN-based or CNN-based seq2seq models.

### 3.1 Phrase-Based Attention Methods

In this subsection, we present two novel methods to achieve phrasal attention. In Subsection 3.2, we present our methods for combining different types of n-gram attentions. The key element in our methods is a temporal (or one-dimensional) convolutional operation that is applied to a sequence

of vectors representing tokens. Formally, we can define the **convolutional** operator applied to each token $x_t$ with corresponding vector representation $\mathbf{x}_t \in \mathbb{R}^{d_1}$ 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 $\text{Conv}_n(\boldsymbol{X}, \boldsymbol{W})$ to denote the convolution operation over an input sequence $\boldsymbol{X}$ with window size $n$ and kernel weights $\boldsymbol{W} \in \mathbb{R}^{n \times d_1 \times d_2}$.

#### 3.1.1 KEY-VALUE CONVOLUTION

The intuition behind **key-value convolution** technique is to use trainable kernel parameters $\boldsymbol{W}_k$ and $\boldsymbol{W}_v$ to compute the latent representation of n-gram sequences using convolution operation over key and value vectors. The attention function with key-value convolution is defined as:

$$\text{CONVKV}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathcal{S}(\frac{(\boldsymbol{Q}\boldsymbol{W}_q)\text{Conv}_n(\boldsymbol{K}, \boldsymbol{W}_k)^T}{\sqrt{d_k}})\ \text{Conv}_n(\boldsymbol{V}, \boldsymbol{W}_v) \tag{5}$$

where $\mathcal{S}$ is the *softmax* function, $\boldsymbol{W}_q \in \mathbb{R}^{d_q \times d_k}, \boldsymbol{W}_k \in \mathbb{R}^{n \times d_k \times d_k}, \boldsymbol{W}_v \in \mathbb{R}^{n \times d_v \times d_v}$ are the respective kernel weights for $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$. Throughout this paper, we will use $\mathcal{S}$ to denote the *softmax* function. Note that in this convolution, the key and value sequences are left zero-padded so that the sequence length is preserved after the convolution (*i.e.,* one latent representation per token).

The CONVKV method can be interpreted as *indirect* query-key attention, in contrast to the *direct* query-key approach to be described next. This means that the queries do not interact directly with the keys to learn the attention weights; instead the model relies on the kernel weights ($\boldsymbol{W}_k$) to learn n-gram patterns.

#### 3.1.2 QUERY-AS-KERNEL CONVOLUTION

In order to allow the queries to *directly* and *dynamically* influence the word order of phrasal keys and values, we introduce **Query-as-Kernel** attention method. In this approach, when computing the attention weights, we use the query as kernel parameters in the convolution applied to the series of keys. The attention output in this approach is given by:

$$\text{QUERYK}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathcal{S}(\frac{\text{Conv}_n(\boldsymbol{K}\boldsymbol{W}_k, \boldsymbol{Q}\boldsymbol{W}_q)}{\sqrt{d_k * n}})\ \text{Conv}_n(\boldsymbol{V}, \boldsymbol{W}_v) \tag{6}$$

where $\boldsymbol{W}_q \in \mathbb{R}^{n \times d_q \times d_k}, \boldsymbol{W}_k \in \mathbb{R}^{d_k \times d_k}, \boldsymbol{W}_v \in \mathbb{R}^{n \times d_v \times d_v}$ are trainable weights. Notice that we include the window size $n$ (phrase length) in the scaling factor to counteract the fact that there are $n$ times more multiplicative operations in the convolution than the traditional matrix multiplication.

### 3.2 MULTI-HEADED PHRASAL ATTENTION

We now present our extensions to the multi-headed attention framework of the Transformer to enable it to pay attention not only to tokens but also to phrases across many sub-spaces and locations.

#### 3.2.1 HOMOGENEOUS N-GRAM ATTENTION

In **homogeneous n-gram attention**, we distribute the heads to different n-gram types with each head attending to one particular n-gram type (n=$1, 2, \ldots, N$). For instance, Figure 1 shows a homogeneous structure, where the first four heads attend to unigrams, and the last four attend to bigrams. A head can apply one of the phrasal attention methods described in Subsection 3.1. The selection of which n-gram to assign to how many heads is arbitrary. Since all heads must have consistent sequence length, phrasal attention heads require left-padding of keys and values before convolution.

Since each head attends to a subspace resulting from one type of n-gram, homogeneous attention learns the mappings in a distributed way. However, the homogeneity restriction may limit the model

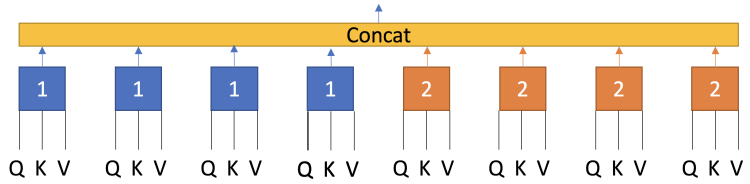



Figure 1: Homogeneous multi-head attention, where each attention head features one n-gram type. In this example, there are eight heads, which are distributed equally between unigrams and bigrams.

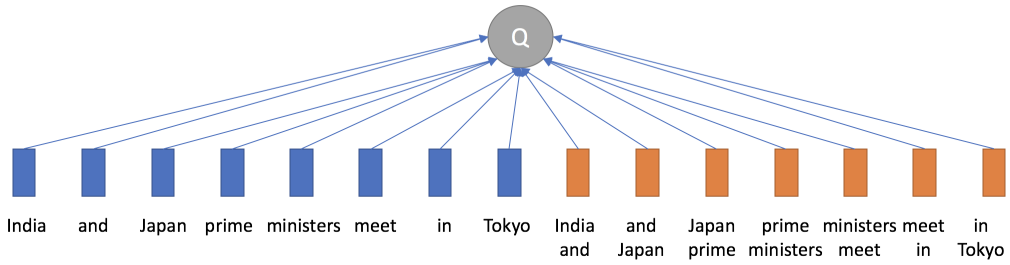



Figure 2: Heterogeneous n-gram attention for each attention head. Attention weights and vectors are computed from all n-gram types simultaneously.

to learn interactions between different n-gram types. Furthermore, the homogeneous heads force the model to assign each query with attentions on all n-gram types (*e.g.,* unigrams and bigrams) even when it does not need to do so, thus possibly inducing more noise into the model.

#### 3.2.2 Heterogeneous N-Gram Attention

The **heterogeneous n-gram attention** relaxes the constraint of the homogeneous approach. Instead of limiting each head's attention to a particular type of n-gram, it allows the query to freely attend to all types of n-grams simultaneously. To achieve this, we first compute the attention logit for each n-gram type separately (*i.e.,* for n $= 1, 2, \ldots, N$), then we concatenate all the logits before passing them through the *softmax* layer to compute the attention weights over all n-gram types. Similarly, the value vectors for the n-gram types are concatenated to produce the overall attention output. Figure 2 demonstrates the heterogeneous attention process for unigrams and bigrams.

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

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

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

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

Note that in heterogeneous attention, we do not need to pad the input sequences before the convolution operation to ensure identical sequence length. Also, the key/value sequences that are shorter than the window size do not have any valid phrasal component to be attended.

### 3.3 Interleaved Phrases to Phrase Heterogeneous Attention

All the methods presented above perform attention mappings from token-based queries to phrase-based key-value pairs. In other words, they adopt token-to-token and token-to-phrase structures. These types of attentions are beneficial when there exists a translation of a phrase in the source language (keys and values) to a single token in the target language (query). However, these methods are not explicitly designed to work in the reverse direction when a phrase or a token in the source language should be translated to a phrase in the target language. In this section, we present a novel approach to heterogeneous phrasal attention that allows phrases of queries to attend on tokens and phrases of keys and values (*i.e.,* phrase-to-token and phrase-to-phrase mappings).

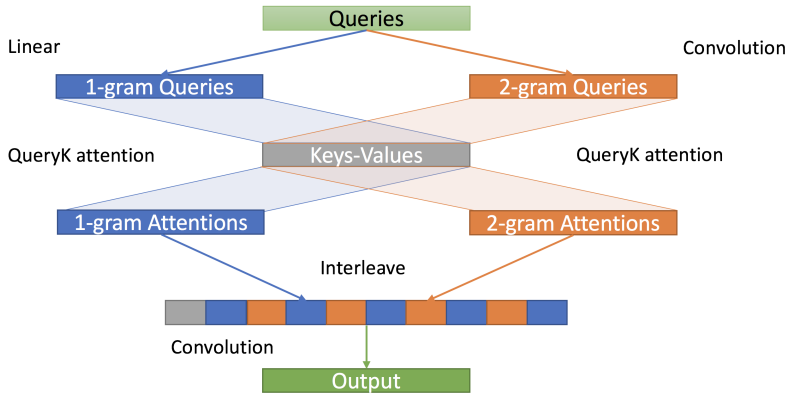



Figure 3: Interleaved phrase-to-phrase heterogeneous attention. The queries are first transformed into unigram and bigram representations, which in turn then attend independently on key-value pairs to produce unigram and bigram attention vectors. The attention vectors are then interleaved before passing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

We accomplish this with the QUERYK and CONVKV methods as follows. We first apply convolutions $\text{Conv}_n(\boldsymbol{Q}, \boldsymbol{W}_{q_n})$ on the query sequence with kernel weights $\boldsymbol{W}_{q_n}$ for window size $n$ to obtain the hidden representations for $n$-grams. Consider Figure 3, where we apply convolution on $\boldsymbol{Q}$ for $n = 1$ and $n = 2$ to generate the respective unigram and bigram queries.[1] These queries are then used to attend over unigram and bigram key-values to generate the heterogeneous attention vectors. The result of these operations is a sequence of unigram and bigram attention vectors $\boldsymbol{A}_1 = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N)$ and $\boldsymbol{A}_2 = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{N-1})$ respectively, where $N$ is the query length.

$$\boldsymbol{A}_{1,\text{ConvKV}} = \mathcal{S}\left(\frac{(\boldsymbol{Q}\boldsymbol{W}_{q_1})[(\boldsymbol{K}\boldsymbol{W}_{k,1})^T; \text{Conv}_2(\boldsymbol{K}, \boldsymbol{W}_{k,2})^T]}{\sqrt{d_k}}\right)[(\boldsymbol{V}\boldsymbol{W}_{v,1}); \text{Conv}_2(\boldsymbol{V}, \boldsymbol{W}_{v,2})] \tag{9}$$

$$\boldsymbol{A}_{2,\text{ConvKV}} = \mathcal{S}\left(\frac{\text{Conv}_2(\boldsymbol{Q}, \boldsymbol{W}_{q_2})[(\boldsymbol{K}\boldsymbol{W}_{k,1})^T; \text{Conv}_2(\boldsymbol{K}, \boldsymbol{W}_{k,2})^T]}{\sqrt{d_k}}\right)[(\boldsymbol{V}\boldsymbol{W}_{v,1}); \text{Conv}_2(\boldsymbol{V}, \boldsymbol{W}_{v,2})] \tag{10}$$

$$\boldsymbol{A}_{1,\text{QueryK}} = \mathcal{S}\left(\left[\frac{(\boldsymbol{Q}\boldsymbol{W}_{q_1,1})(\boldsymbol{K}\boldsymbol{W}_{k,1})^T}{\sqrt{d}}; \frac{\text{Conv}_2(\boldsymbol{K}\boldsymbol{W}_{k,2}, \boldsymbol{Q}\boldsymbol{W}_{q_1,2})}{\sqrt{d * n_2}}\right]\right)[(\boldsymbol{V}\boldsymbol{W}_{v,1}); \text{Conv}_2(\boldsymbol{V}, \boldsymbol{W}_{v,2})] \tag{11}$$

$$\boldsymbol{A}_{2,\text{QueryK}} = \mathcal{S}\left(\left[\frac{\text{Conv}_2(\boldsymbol{Q}, \boldsymbol{W}_{q_2,1})(\boldsymbol{K}\boldsymbol{W}_{k,1})^T}{\sqrt{d}}; \frac{\text{Conv}_2(\boldsymbol{K}\boldsymbol{W}_{k,2}, \text{Conv}_2(\boldsymbol{Q}, \boldsymbol{W}_{q_2,2}))}{\sqrt{d * n_2}}\right]\right)[(\boldsymbol{V}\boldsymbol{W}_{v,1}); \text{Conv}_2(\boldsymbol{V}, \boldsymbol{W}_{v,2})] \tag{12}$$

Notice that each $\mathbf{b}_i \in \boldsymbol{A}_2$ represents the attention vector for $(\boldsymbol{Q}_i\text{-}\boldsymbol{Q}_{i+1})$ bigram queries. In the next step, the phrase-level attention states in $\boldsymbol{A}_2$ are *interleaved* with the unigram attentions in $\boldsymbol{A}_1$ to form an interleaved attention sequence $\boldsymbol{I}$. For unigram and bigram queries, the interleaved vector sequences at the encoder and decoder are formed as

$$\boldsymbol{I}_{\text{enc}} = (\mathbf{0}, \mathbf{u}_1, \mathbf{b}_1, \mathbf{u}_2, \mathbf{b}_2, \mathbf{u}_3, \ldots, \mathbf{b}_{N-1}, \mathbf{u}_N, \mathbf{0}) \tag{13}$$

$$\{\boldsymbol{I}_{\text{dec}}, \boldsymbol{I}_{\text{cross}}\} = (\mathbf{0}, \mathbf{u}_1, \mathbf{b}_1, \mathbf{u}_2, \mathbf{b}_2, \mathbf{u}_3, \ldots, \mathbf{b}_{N-1}, \mathbf{u}_N) \tag{14}$$

where $\boldsymbol{I}_{\text{enc}}$ and $\boldsymbol{I}_{\text{dec}}$ denote the interleaved sequence for self-attention at the encoder and decoder respectively, and $\boldsymbol{I}_{\text{cross}}$ denotes the interleaved sequence for cross-attention between the encoder and the decoder. Note that, to prevent information flow from the future in the decoder, the right connections are masked out in $\boldsymbol{I}_{\text{dec}}$ and $\boldsymbol{I}_{\text{cross}}$ (similar to the original Transformer). The interleaving operation places the phrase- and token-based representations of a token next to each other. The interleaved vectors are passed through a convolution layer (as opposed to a point-wise feed-forward layer in the Transformer) to compute the overall representation for each token. By doing so, each query is intertwined with the n-gram representations of the phrases containing itself, which enables the model to learn the query's correlation with neighboring tokens. For unigram and bigram queries, the encoder uses a convolution layer with a window size of 3 and stride of 2 to allow the token to intertwine with its past and future phrase representations, while the ones in the decoder (self-attention and cross-attention) use a window size of 2 and stride of 2 to incorporate only the past phrase representations to preserve the autoregressive property. More formally,

[1] $\text{Conv}_1(\boldsymbol{Q}, \boldsymbol{W}_{q_1})$ is equivalent to a feed-forward connection.

$$\boldsymbol{O}_{\text{enc}} = \text{Conv}_{\text{window=3,stride=2}}(\boldsymbol{I}_{\text{enc}}, \boldsymbol{W}_{\text{enc}}) \quad (15)$$

$$\boldsymbol{O}_{\text{cross}} = \text{Conv}_{\text{window=2,stride=2}}(\boldsymbol{I}_{\text{cross}}, \boldsymbol{W}_{\text{cross}}) \quad (16)$$

$$\boldsymbol{O}_{\text{dec}} = \text{Conv}_{\text{window=2,stride=2}}(\boldsymbol{I}_{\text{dec}}, \boldsymbol{W}_{\text{dec}}) \quad (17)$$

# 4 EXPERIMENTS

In this section, we present the training settings, experimental results and analysis of our models.

## 4.1 TRAINING SETTINGS

We preserve most of the training settings from Vaswani et al. (2017) to enable a fair comparison with the original Transformer. Specifically, we use the Adam optimizer (Kingma & Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. We follow a similar learning rate schedule with $warmup_steps$ of 16000: $LearningRate = 2 * d^{-0.5} * \min(step_num^{-0.5}, step_num * warmup_steps^{-1.5})$.

We trained our models and the baseline on a single GPU for 500,000 steps.[2] The batches were formed by sentence pairs containing approximately 4096 source and 4096 target tokens. Similar to Vaswani et al. (2017), we also applied residual dropout with 0.1 probability and label smoothing with $\epsilon_{ls} = 0.1$. Our models are implemented in the *tensor2tensor*[3] library (Vaswani et al., 2018), on top of the original Transformer codebase. We trained our models on the standard WMT'16 English-German dataset constaining about 4.5 million sentence pairs, using WMT **newstest2013** as our development set and **newstest2014** as our test set. We used byte-pair encoding (Sennrich et al., 2016) with combined source and target vocabulary of 37,000 sub-words for English-German. We took the average of the last 5 checkpoints (saved at 10,000-iteration intervals) for evaluation, and used a beam search size of 5 and length penalty of 0.6 (Wu et al., 2016).

## 4.2 RESULTS

Table 1 compares our model variants with the Transformer base model of Vaswani et al. (2017) on the **newstest2014** test set. All models were trained on identical settings. We notice that all of our models achieve higher BLEU scores than the baseline, showing the effectiveness of our approach.

On the **English-to-German (En-De)** translation task, our **homogeneous** structure using CONVKV phrasal attention with 44 head distribution already outperforms the Transformer base by more than 0.5 BLEU, while the one using QUERYK performs even better with a BLEU of 26.78. When we compare our **heterogeneous** models with homogeneous ones, we notice even higher scores for the heterogeneous models – the CONVKV achieves a score of 27.04, and the QUERYK achieves a score of 26.95. This shows the effectiveness of relaxing the 'same n-gram type' attention constraint in the heterogeneous approach, which allows it to attend to all n-gram types simultaneously within a single head, avoiding any forced attention to a particular n-gram type. In fact, our experiments show that the models perform worse than the baseline if we remove token-level (unigram) attentions entirely.

Finally, we notice that our **interleaved** heterogeneous models surpass all aforementioned scores achieving up to 27.40 BLEU and establishing a 1.33 BLEU improvement over the Transformer. This demonstrates the existence of phrase-to-token and phrase-to-phrase mappings from target to source language within their mutual latent space. This is especially necessary when the target language is morphologically rich, like German, whose words are usually compounded with sub-words expressing different meanings and grammatical structures. Phrase-to-phrase mapping helps model local agreement, *e.g.,* between an adjective and a noun (in terms of gender, number and case) or between subject and verb (in terms of person and number). Our interleaved models with BPE (Sennrich et al., 2016) tackle all four possible types of alignments (token-to-token, token-to-phrase, phrase-to-token, phrase-to-phase) to counteract such linguistic differences in morphology and syntax.

On **German-to-English (De-En)** translation, likewise, all of our models achieve improvement compared to the Transformer base, but the gain is not as high as in the English-to-German task. Specif-

[2]We were unable to replicate state-of-the-art results in (Vaswani et al., 2017) because of limited GPUs. Hence, we conducted all the experiments including Transformer base in an identical setup for fair comparisons.

[3]https://github.com/tensorflow/tensor2tensor

| Model | Technique | N-grams | En-De | De-En |
|---|---|---|---|---|
| Transformer (Base, 1 GPU) | - | - | 26.07 | 29.82 |
| Transformer (Base, 8 GPUs) | - | - | 27.30 | —- |
| Vaswani et al. (2017) | | | | |
| 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.

| 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.

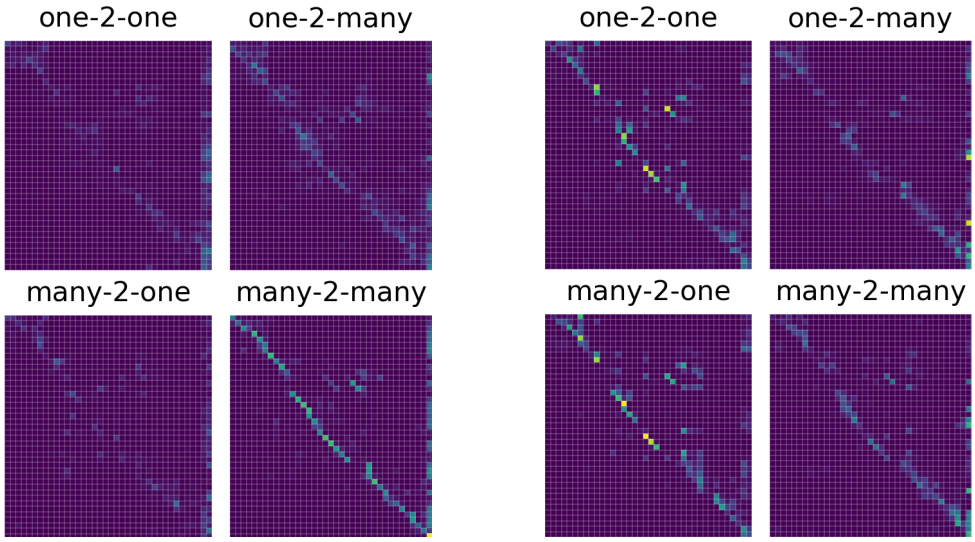
ically, homogeneous and heterogeneous attentions perform comparably, giving up to +0.38 BLEU improvement compared to the Transformer base. Our interleaved models show more improvements (up to 30.30 BLEU), outperforming the Transformer by about 0.5 points. This demonstrates the importance of phrase-level query representation in the target.

Considering CONVKV vs. QUERYK, both methods perform rather equivalently. QUERYK is better than CONVKV for homogeneous models but worse for their heterogeneous counterparts on the English-to-German task. However, the opposite trend can also be observed for German-to-English in one case. Meanwhile, QUERYK achieves higher BLEU in interleaved models in both tasks.

For further analysis, Table 2 shows how the performance differs if we include a trigram component in our homogeneous and heterogeneous models.[4] For homogeneous models, CONVKV leads to a deterioration in performance, while QUERYK leads to a minor improvement. By contrast, heterogeneous models achieve higher BLEU scores for both techniques. This supports the argument that some trigrams are not useful, and heterogeneous models allow the queries to select any type of phrasal keys and values, whereas homogeneous ones force them to attend to unnecessary n-grams, inducing more noise.

To interpret our phrasal attention models, we now discuss how they learn the alignments. Figure 4 shows the attention heat maps for an English-German sample in newstest2014; figure 4a displays the heat map in layer 3 (mid layer) while figure 4b shows the one in layer 6 (top layer) within a six-layer transformer model based on our interleaved attention. Each figure shows 4 quadrants representing *token-to-token, token-to-phrase, phrase-to-token, phrase-to-phrase* attentions, respectively. It can be seen that phrasal attentions are activated strongly in the mid-layers; particularly, the phrase-to-phrase attention is the most concentrated. The models learn phrasal alignments in the middle of the network. On the other hand, token-to-token attention is activated the most in the top layer, which is understandable because the top layer connects directly with the peripheral *softmax* layer to generate translations token-by-token autoregressively. In fact, we observed that phrasal attention intensity

[4]It is nontrivial to use unigram, bigram and trigram components in the interleaved model, because one needs to mix three sequences into one such that different n-gram types align properly. We leave it to future work.




(a) Layer 3 (b) Layer 6

Figure 4: Sample attention heat map of QUERYK interleaved heterogeneous model.

| Layer | token-to-token | token-to-phrase | phrase-to-token | phrase-to-phrase |
|---|---|---|---|---|
| 1 | 98.01 | 00.20 | 01.60 | 00.19 |
| 2 | 39.63 | 19.49 | 02.73 | 38.15 |
| 3 | 01.54 | 02.30 | 00.00 | 96.16 |
| 4 | 36.13 | 37.53 | 06.60 | 19.74 |
| 5 | 61.96 | 18.12 | 08.59 | 11.33 |
| 6 | 53.72 | 10.53 | 34.63 | 01.12 |

Table 3: Activation percentages for different attention types in each layer of the Interleaved model.

transits gradually from phrase-level attention in the lower layer to token-level attention in the upper layer, and this phenomenon occurs frequently. Heterogeneous models follow the same trend as well.

Table 3 presents quantitatively the activation percentage of each attention type per layer computed by averaging over all sentences in the English-to-German newstest2014. We notice that attentions are distributed unevenly across four alignment types and differently in each layer. Phrase-to-phrase attention interestingly accounts for 96.16% in layer 3, while layers 5 and 6 concentrate attention scores mostly in token-to-token mappings [5]. This indicates a particular combination of mapping types is learned distinctively by a specific layer of the network. We refer the readers to the Appendix (Section 6) for more figures and details about attention heat maps and statistics.

## 5 CONCLUSIONS

We have presented novel approaches to incorporating phrasal alignments into the attention mechanism of state-of-the-art neural machine translation models. Our methods assign attentions to all four possible mapping relations between target and source sequences: token-to-token, token-to-phrase, phrase-to-phrase and phrase-to-phrase. While we have applied our attention mechanisms to the Transformer network, they are generic and can be implemented in other architectures. On WMT'14 English-to-German and German-to-English translation tasks, all of our methods surpass the Transformer base model. Our model with interleaved heterogeneous attention, which tackles all the possible phrasal mappings in a unified framework, achieves improvements of 1.33 BLEU for English-German and 0.48 BLEU for German-English translation tasks over the baseline model. We are planning future extensions of our techniques to other tasks, such as summarization and question answering. We also plan to improve our models with a phrase-based decoding procedure.

[5]Note that the layer 1 also squeezes attention scores to the token-to-token quadrant. However, it does not learn the proper token mappings; instead it concentrates the weights to the last <*eos*> token.

## REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

Leonard Dahlmann, Evgeny Matusov, Pavel Petrushkov, and Shahram Khadivi. Neural machine translation leveraging phrase-based models in a hybrid search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1411–1420. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1148.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional Sequence to Sequence Learning. In *Proc. of ICML*, 2017.

Po-Sen Huang, Chong Wang, Sitao Huang, Dengyong Zhou, and Li Deng. Towards neural phrase-based machine translation. In *ICLR*, 2018.

Ł ukasz Kaiser and Samy Bengio. Can active memory replace attention? In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3781–3789. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6295-can-active-memory-replace-attention.pdf.

Lukasz Kaiser, Aidan N. Gomez, and Francois Chollet. Depthwise separable convolutions for neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1jBcueAb.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010. ISBN 0521874157, 9780521874151.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pp. 48–54, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073462. URL https://doi.org/10.3115/1073445.1073462.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1412–1421. ACL, 2015.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1162. URL http://www.aclweb.org/anthology/P16-1162.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3104–3112. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Yaohua Tang, Fandong Meng, Zhengdong Lu, Hang Li, and Philip LH Yu. Neural machine translation with external phrase memory. *arXiv preprint arXiv:1606.01792*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416, 2018. URL `http://arxiv.org/abs/1803.07416`.

Chong Wang, Yining Wang, Po-Sen Huang, Abdelrahman Mohamed, Dengyong Zhou, and Li Deng. Sequence modeling via segmentations. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 3674–3683, 2017a.

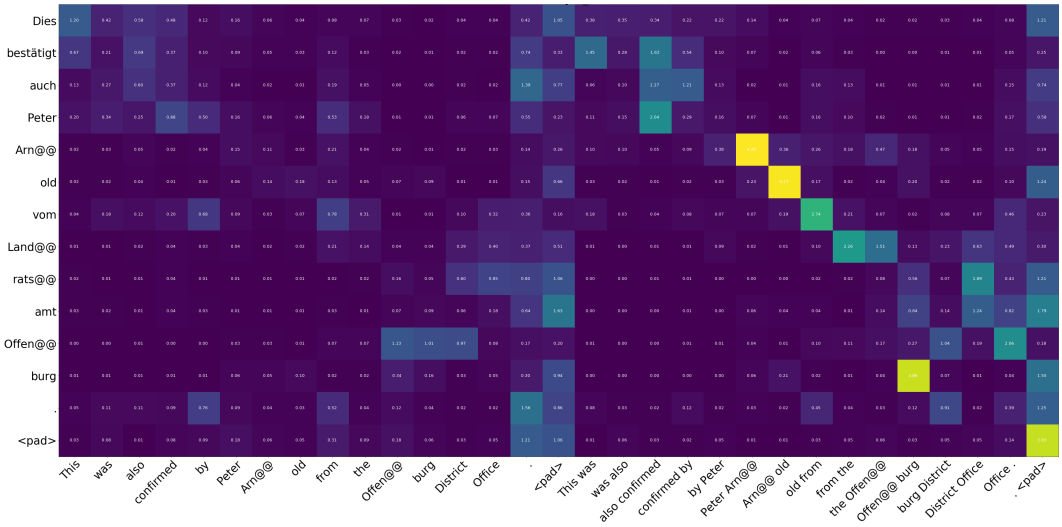
Xing Wang, Zhaopeng Tu, Deyi Xiong, and Min Zhang. Translating phrases in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1421–1431. Association for Computational Linguistics, 2017b. URL `http://aclweb.org/anthology/D17-1149`.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

# 6 APPENDIX

## 6.1 ATTENTION HEAT MAP OF QUERYK HETEROGENEOUS MODEL

(a) Layer 3 of QUERYK heterogeneous




(b) Layer 6 QUERYK heterogeneous

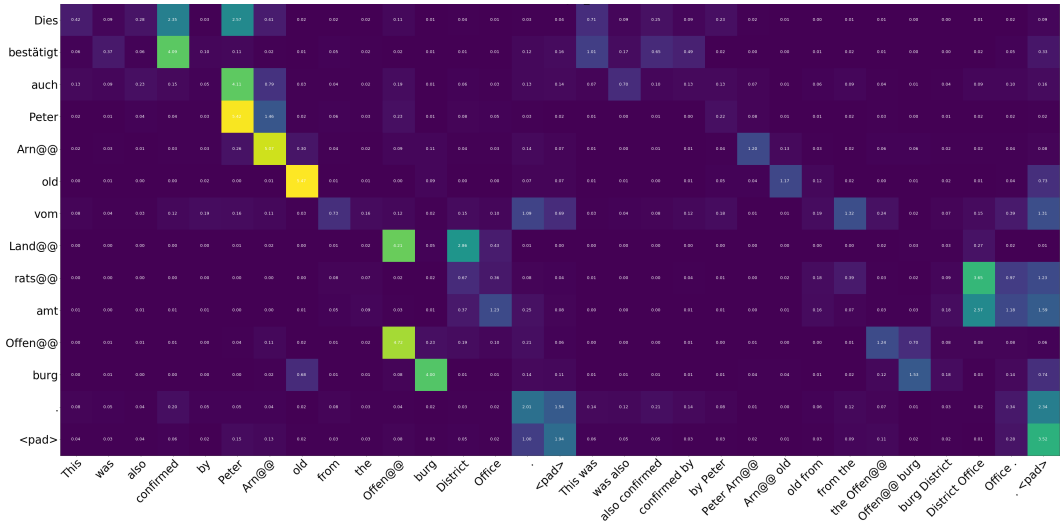



Figure 5: Attention heat maps at layer 3 and layer 6 of **QUERYK heterogeneous** model for a sample sentence pair in English-German newstest2014 test set. The left half in each figure indicates **token-to-token** mappings, while the right half indicates **token-to-phrase** mappings.

## 6.2 ATTENTION STATISTICS FOR QUERYK HETEROGENEOUS MODEL

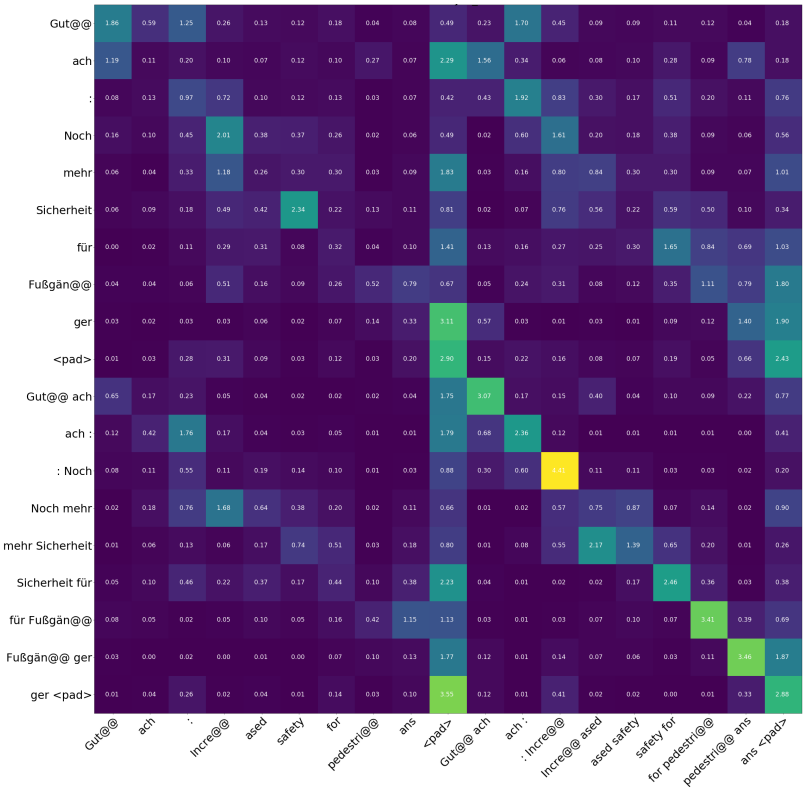
Table 4 presents quantitatively the activation percentage of each attention type per layer for QUERYK heterogeneous model computed by averaging over all sentences in the English-to-German newstest2014. The mid layers (2, 3, 4, 5) activate mostly (if not entirely) phrasal attentions while only the first and last layer emphasize token-level attentions.

| Layer | token-to-token | token-to-phrase |
|---|---|---|
| 1 | 100.00 | 00.00 |
| 2 | 00.24 | 99.76 |
| 3 | 01.10 | 98.90 |
| 4 | 05.83 | 94.17 |
| 5 | 23.84 | 76.16 |
| 6 | 86.30 | 13.70 |

Table 4: Activation percentages for different attention types in each layer of the **QUERYK heterogeneous attention** model. These numbers are computed by averaging over all sentences in newstest2014 for En-De translation task. The mid layers (2, 3, 4, 5) activate mostly (if not entirely) phrasal attentions while only the first and last layer emphasize token-level attentions.

### 6.3 ATTENTION HEAT MAP OF QUERYK INTERLEAVED MODEL

(a) Layer 3 of QUERYK interleaved




(b) Layer 6 QUERYK interleaved

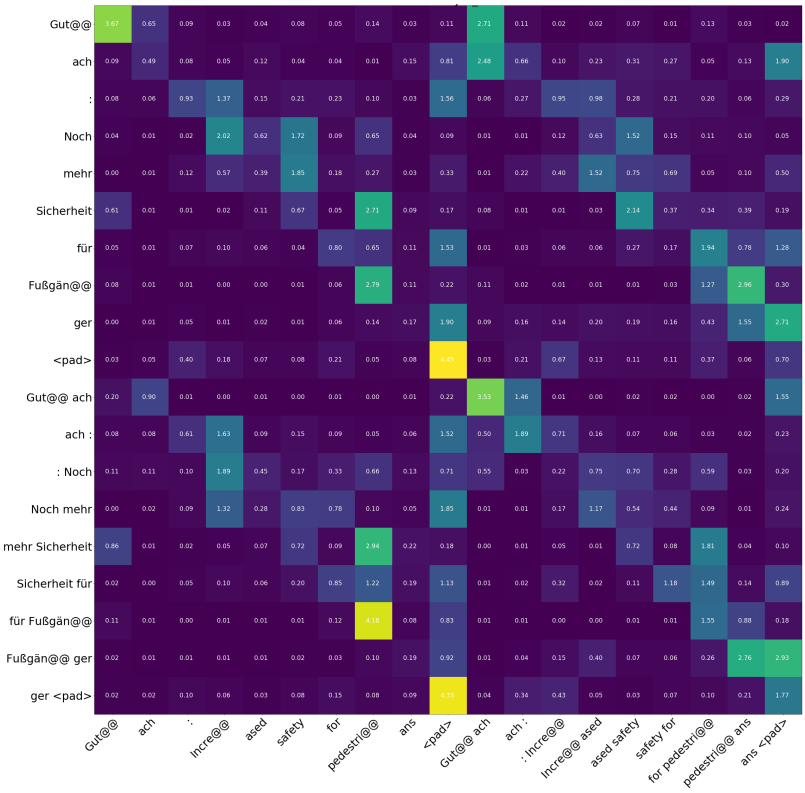



Figure 6: Attention heat maps at layer 3 and layer 6 of **QUERYK interleaved heterogeneous** model for another sample from English-German newstest2014 test set. Upper-left, upper-right, lower-left, lower-right quadrants of each figure show token-to-token, token-to-phase, phrase-to-token, phrase-to-phrase alignments respectively.