Information Aggregation for Multi-Head Attention with Routing-by-Agreement

Jian Li[1] **Baosong Yang**[2] **Zi-Yi Dou**[3] **Xing Wang**[4] **Michael R. Lyu**[1] **Zhaopeng Tu**[4*]
[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong
{jianli,lyu}@cse.cuhk.edu.hk
[2]University of Macau
nlp2ct.baosong@gmail.com
[3]Carnegie Mellon University
zdou@andrew.cmu.edu
[4]Tencent AI Lab
{brightxwang,zptu}@tencent.com

Abstract

Multi-head attention is appealing for its ability to jointly extract different types of information from multiple representation subspaces. Concerning the information aggregation, a common practice is to use a concatenation followed by a linear transformation, which may not fully exploit the expressiveness of multi-head attention. In this work, we propose to improve the information aggregation for multi-head attention with a more powerful *routing-by-agreement* algorithm. Specifically, the routing algorithm iteratively updates the proportion of how much a part (i.e. the distinct information learned from a specific subspace) should be assigned to a whole (i.e. the final output representation), based on the agreement between parts and wholes. Experimental results on linguistic probing tasks and machine translation tasks prove the superiority of the advanced information aggregation over the standard linear transformation.

1 Introduction

Attention model becomes a standard component of the deep learning networks, contributing to impressive results in machine translation (Bahdanau et al., 2015; Luong et al., 2015), image captioning (Xu et al., 2015), speech recognition (Chorowski et al., 2015), among many other applications. Its superiority lies in the ability of modeling the dependencies between representations without regard to their distance. Recently, the performance of attention is further improved by multi-head mechanism (Vaswani et al., 2017), which parallelly performs attention functions on different representation subspaces of the input sequence. Consequently, different attention heads are able to capture distinct linguistic properties of the input, which are embedded in different subspaces (Raganato and Tiedemann, 2018). Subsequently, a linear transformation is generally employed to aggregate the partial representations extracted by different attention heads (Vaswani et al., 2017; Ahmed et al., 2018).

Most existing work focus on extracting informative or distinct partial-representations from different subspaces (e.g. Lin et al., 2017; Li et al., 2018), while few studies have paid attention to the aggregation of the extracted partial-representations. Arguably, information extraction and aggregation are both crucial for multi-head attention to generate an informative representation. Recent studies in multimodal learning show that a straightforward linear transformation for fusing features in different sets of representations usually limits the extent of abstraction (Fukui et al., 2016; Ben-Younes et al., 2017). A natural question arises: *whether the straightforward linear transformation is expressive enough to fully capture the rich information distributed in the extracted partial-representations?*

In this work, we provide the first answer to this question. We propose to empirically validate the importance of information aggregation in multi-head attention, by comparing the performance of the standard linear function and advanced aggregation functions on various tasks. Specifically, we cast information aggregation as the *assigning-parts-to-wholes* problem (Hinton et al., 2011), and investigate the effectiveness of the routing-by-agreement algorithm – an appealing alternative to solving this problem (Sabour et al., 2017; Hinton et al., 2018). The routing algorithm iteratively updates the proportion of how much a part should be assigned to a whole, based on the agreement between parts and wholes. We leverage the routing algorithm to aggregate the information distributed in the extracted partial-representations.

* Zhaopeng Tu is the corresponding author of the paper. This work was mainly conducted when Jian Li, Baosong Yang, and Zi-Yi Dou were interning at Tencent AI Lab.

We evaluate the performance of the aggregated representations on both linguistic probing tasks as well as machine translation tasks. The probing tasks (Conneau et al., 2018) consists of 10 classification problems to study what linguistic properties are captured by input representations. Probing analysis show that our approach indeed produces more informative representation, which embeds more syntactic and semantic information. For translation tasks, we validate our approach on top of the advanced TRANSFORMER model (Vaswani et al., 2017) on both WMT14 English⇒German and WMT17 Chinese⇒English data. Experimental results show that our approach consistently improves translation performance across languages while keeps the computational efficiency.

The main contributions of this paper can be summarized as follows:

- To our best knowledge, this is the first work to demonstrate the necessity and effectiveness of advanced information aggregation for multi-head attention.
- Our work is among the few studies (*cf.* (Gong et al., 2018; Zhao et al., 2018; Dou et al., 2019)) which prove that the idea of capsule networks can have promising applications on natural language processing tasks.

2 Background

Attention mechanism aims at modeling the relevance between representation pairs, thus a representation is allowed to build a direct relation with another representation. Instead of performing a single attention function, Vaswani et al. (2017) found it is beneficial to capture different context features with multiple individual attention functions, namely multi-head attention. Formally, attention function maps a sequence of query $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_J\}$ and a set of key-value pairs $\{\mathbf{K}, \mathbf{V}\} = \{(\mathbf{k}_1, \mathbf{v}_1), \ldots, (\mathbf{k}_M, \mathbf{v}_M)\}$ to outputs, where $\mathbf{Q} \in \mathbb{R}^{J\times d}$, $\{\mathbf{K}, \mathbf{V}\} \in \mathbb{R}^{M\times d}$. More specifically, multi-head attention model first transforms $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ into H subspaces with different, learnable linear projections:

$$\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h = \mathbf{Q}\mathbf{W}_h^Q, \mathbf{K}\mathbf{W}_h^K, \mathbf{V}\mathbf{W}_h^V, \quad (1)$$

where $\{\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h\}$ are respectively the query, key, and value representations of the h-th head. $\{\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V\} \in \mathbb{R}^{d\times\frac{d}{H}}$ denote parameter matrices associated with the h-th head, where d represents the dimensionality of the model hidden states. Furthermore, H attention functions are applied in parallel to produce the output states $\{\mathbf{O}_1, \ldots, \mathbf{O}_H\}$, among them:

$$\mathbf{O}_h = \text{ATT}(\mathbf{Q}_h, \mathbf{K}_h)\mathbf{V}_h, \quad (2)$$

where $\mathbf{O}_h \in \mathbb{R}^{J\times\frac{d}{H}}$, $\text{ATT}(\cdot)$ is an attention model. In this work, we use scaled dot-product attention (Luong et al., 2015), which achieves similar performance with its additive counterpart (Bahdanau et al., 2015) while is much faster and more space-efficient in practice (Vaswani et al., 2017).

Finally, the H output states are concatenated and linearly transformed to produce the final state:

$$\text{Concat:} \quad \widehat{\mathbf{O}} = [\mathbf{O}_1, \ldots, \mathbf{O}_H], \quad (3)$$

$$\text{Linear:} \quad \mathbf{O} = \widehat{\mathbf{O}}\mathbf{W}^O, \quad (4)$$

where $\mathbf{O} \in \mathbb{R}^{J\times d}$ denotes the final output states, $\mathbf{W}^O \in \mathbb{R}^{d\times d}$ is a trainable matrix.

As shown in Equations 3 and 4, the conventional multi-head attention uses a straightforward concatenation and linear mapping to aggregate the output representations of multiple attention heads. We argue that this straightforward strategy may not fully exploit the expressiveness of multi-head attention, which can benefit from advanced information aggregation by exploiting the intrinsic relationship among the learned representations.

3 Related Work

Our work synthesizes two strands of research work, namely *multi-head attention* and *information aggregation*.

3.1 Multi-Head Attention

Multi-head attention has shown promising empirical results in many NLP tasks, such as machine translation (Vaswani et al., 2017; Domhan, 2018), semantic role labeling (Strubell et al., 2018), and subject-verb agreement task (Tang et al., 2018). The strength of multi-head attention lies in the rich expressiveness by using multiple attention functions in different representation subspaces.

Previous work show that multi-head attention can be further enhanced by encouraging individual attention heads to extract distinct information. For example, Lin et al. (2017) introduce a penalization term to reduce the redundancy of attention weights among different attention heads. Li et al. (2018) propose disagreement regularizations

to encourage different attention heads to capture distinct features, and Yang et al. (2019) model the interactions among attention heads. Shen et al. (2018) explicitly use multiple attention heads to model different dependencies of the same word pair, and Strubell et al. (2018) employ different attention heads to capture different linguistic features. Our approach is complementary to theirs, since they focus on extracting distinct information while ours aims at effectively aggregating the extracted information. Our study shows that information aggregation is as important as information extraction for multi-head attention.

3.2 Information Aggregation

Information aggregation in multi-head attention (e.g. Equations 3 and 4) aims at composing the partial representations of the input captured by different attention heads to a final representation. Recent work shows that representation composition benefits greatly from advanced functions beyond simple concatenation or mean/max pooling. For example, Fukui et al. (2016) and Ben-Younes et al. (2017) succeed on fusing multi-modal features (e.g., visual features and textual features) more effectively via employing the higher-order bilinear pooling instead of vector concatenation or element-wise operations. In NLP tasks, Peters et al. (2018) aggregate layer representations with linear combination, and Dou et al. (2018) compose deep representations with layer aggregation and multi-layer attention mechanisms.

Recently, the routing-by-agreement algorithm, which origins from the capsule networks (Hinton et al., 2011), becomes an appealing alternative to representation composition. The majority of existing work on capsule networks has focused on computer vision tasks, such as MNIST tasks (Sabour et al., 2017; Hinton et al., 2018), CIFAR tasks (Xi et al., 2017), and object segmentation task (LaLonde and Bagci, 2018). The applications of capsule networks in NLP tasks, however, have not been widely investigated to date. Zhao et al. (2018) testify capsule networks on text classification tasks and Gong et al. (2018) propose to aggregate a sequence of vectors via dynamic routing for sequence encoding. Dou et al. (2019) use routing-by-agreement strategies to aggregate layer representations dynamically. Inspired by these successes, we apply the routing algorithms to multi-head attention on both linguistic probing and machine translation tasks, which demonstrates the necessity and effectiveness of advanced information aggregation for multi-head attention.

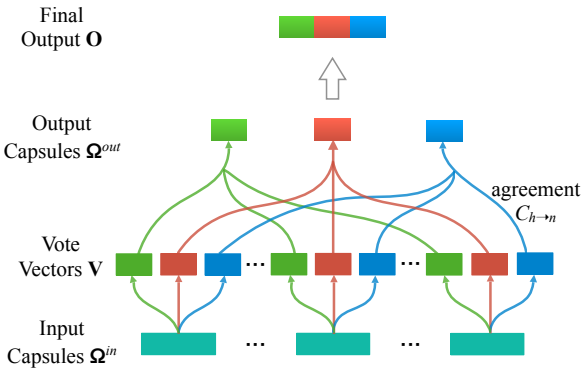

Figure 1: Illustration of routing-by-agreement.

4 Approach

In this work, we cast information aggregation in multi-head attention as the problem of *assigning-parts-to-wholes*. Specifically, each attention head extracts different linguistic properties of the same input (Raganato and Tiedemann, 2018), and the goal of information aggregation is to compose the partial representations extracted by different heads to a whole representation. An appealing solution to this problem is the *routing-by-agreement* algorithm, as shown in Figure 1.

The routing algorithm consists of two layers: *input capsules* and *output capsules*. The input capsules are constructed from the transformation of the partial representations extracted by different attention heads. For each output capsule, each input capsule proposes a distinct "voting vector", which represents the proportion of how much the information is transformed from this input capsule (i.e parts) to the corresponding output capsule (i.e. wholes). The proportion is iteratively updated based on the agreement between the voting vectors and the output capsule. Finally, all output capsules are concatenated to form the final representation.

4.1 Routing-by-Agreement

Mathematically, the input capsules $\mathbf{\Omega}^{in} = \{\mathbf{\Omega}^{in}_1, \ldots, \mathbf{\Omega}^{in}_H\}$ with $\mathbf{\Omega}^{in} \in \mathbb{R}^{n\times d}$ are constructed from the outputs of multi-head attention:

$$\mathbf{\Omega}^{in}_h = f_h(\widehat{\mathbf{O}}), \tag{5}$$

where $f_h(\cdot)$ is a distinct non-linear transformation function associated with the input capsule

Algorithm 1 Iterative Simple Routing.

1: **procedure** ROUTING($\mathbf{V}, T$):
2: $\quad\forall \mathbf{V}_{h\to *}$: $B_{h\to n} = 0$
3: $\quad$**for** T iterations **do**
4: $\quad\quad\forall \mathbf{V}_{h\to *}$: $C_{h\to n} = \frac{\exp(B_{h\to n})}{\sum_{n'=1}^{N}\exp(B_{h\to n'})}$
5: $\quad\quad\forall \mathbf{\Omega}_n^{out}$: compute $\mathbf{\Omega}_n^{out}$ by Eq. 7
6: $\quad\quad\forall \mathbf{V}_{h\to *}$: $B_{h\to n}$ += $\mathbf{\Omega}_n^{out} \cdot \mathbf{V}_{h\to n}$
$\quad$**return** $\mathbf{\Omega}$

Algorithm 2 Iterative EM Routing.

1: **procedure** EM ROUTING($\mathbf{V}, T$):
2: $\quad\forall \mathbf{V}_{h\to *}$: $C_{l\to n} = 1/N$
3: $\quad$**for** T iterations **do**
4: $\quad\quad\forall \mathbf{\Omega}_n^{out}$: M-STEP($\mathbf{V}$, C) ▷ *hold C constant, adjust $(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, A_n)$*
5: $\quad\quad\forall \mathbf{V}_{h\to *}$: E-STEP($\mathbf{V}$, $\boldsymbol{\mu}, \boldsymbol{\sigma}, A$) ▷ *hold $(\boldsymbol{\mu}, \boldsymbol{\sigma}, A)$ constant, adjust $C_{h\to *}$*
6: $\quad\forall \mathbf{\Omega}_n^{out}$: $\mathbf{\Omega}_n^{out} = A_n * \boldsymbol{\mu}_n$
$\quad$**return** $\mathbf{\Omega}$

$\mathbf{\Omega}_h^{in}$. Given N output capsules, each input capsule $\mathbf{\Omega}_h^{in}$ propose N "vote vectors" $\mathbf{V}_{h\to *} = \{\mathbf{V}_{h\to 1}, \ldots, \mathbf{V}_{h\to N}\}$, which is calculated by

$$\mathbf{V}_{h\to n} = \mathbf{\Omega}_h^{in}\mathbf{W}_{h\to n}, \tag{6}$$

Each output capsule $\mathbf{\Omega}_n^{out}$ is calculated as the normalization of its total input, which is a weighted sum over all "vote vectors" $\mathbf{V}_{*\to n}$:

$$\mathbf{\Omega}_n^{out} = \frac{\sum_{h=1}^{H} C_{h\to n}\mathbf{V}_{h\to n}}{\sum_{h=1}^{H} C_{h\to n}}, \tag{7}$$

The weight $C_{h\to n}$ with $\sum_n C_{h\to n} = 1$ measures the agreement between vote vector $\mathbf{V}_{h\to n}$ and output capsule $\mathbf{\Omega}_n^{out}$, which is determined by the iterative routing as described in the next section. Note that $\sum_{h=1}^{H} C_{h\to n}$ is not necessarily equal to 1. After the routing process, following Gong et al. (2018), we concatenate the N output capsules to form the final representation: $\mathbf{O} = [\mathbf{\Omega}_1^{out}, \ldots, \mathbf{\Omega}_N^{out}]$. To make the dimensionality of the final output be consistent with that of hidden layer (i.e. d), we set the dimensionality of each output capsule be $\frac{d}{N}$.

4.2 Routing Mechanisms

In this work, we explore two representative routing mechanisms, namely *simple routing* (Sabour et al., 2017) and *EM routing* (Hinton et al., 2018), which differ at how the agreement weights $C_{h\to n}$ are calculated.

4.2.1 Simple Routing

Algorithm 1 lists a straightforward implementation of routing mechanism. $B_{h\to n}$ measures the degree that the input capsule $\mathbf{\Omega}_h^{in}$ should be coupled to the output capsule $\mathbf{\Omega}_n^{out}$, which is initialized as all 0 (Line 2). The agreement weights $C_{h\to n}$ are then iteratively refined by measuring the agreement between vote vector $\mathbf{V}_{h\to n}$ and output capsule $\mathbf{\Omega}_n^{out}$ (Lines 4-6), which is implemented as a simple scalar product $\mathbf{\Omega}_n^{out} \cdot \mathbf{V}_{h\to n}$ (Line 6).

To represent the probability that the output capsule $\mathbf{\Omega}_n^{out}$ is activated, Sabour et al. (2017) use a non-linear "squashing" function:

$$\mathbf{\Omega}_n^{out} = \frac{||\mathbf{\Omega}_n^{out}||^2}{1+||\mathbf{\Omega}_n^{out}||^2}\frac{\mathbf{\Omega}_n^{out}}{||\mathbf{\Omega}_n^{out}||}, \tag{8}$$

The scalar product $\mathbf{\Omega}_n^{out} \cdot \mathbf{V}_{h\to n}$ saturates at 1, which makes it insensitive to the difference between a quite good agreement and a very good agreement. In response to this problem, Hinton et al. (2018) propose a novel Expectation-Maximization (EM) routing algorithm.

4.2.2 EM Routing

Comparing with simple routing, EM routing has two modifications. First, it explicitly assigns an activation probability A to represent the probability of whether each output capsule is activated, rather than the length of vector calculated by a squashing function (Equation 8). Second, it casts the routing process as fitting a mixture of Gaussians using EM algorithm, where the output capsules play the role of Gaussians and the means of the input capsules play the role of the datapoints. Accordingly, EM routing can better estimate the agreement by allowing activated output capsules to receive a cluster of similar votes.

Algorithm 2 lists the EM routing, which iteratively adjusts the means, variances, and activation probabilities $(\boldsymbol{\mu}, \boldsymbol{\sigma}, A)$ of the output capsules, as well as the agreement weights C of the input capsules (Lines 4-5). The representation of output capsule $\mathbf{\Omega}_n^{out}$ is calculated as

$$\mathbf{\Omega}_n^{out} = A_n * \boldsymbol{\mu}_n = A_n * \frac{\sum_{h=1}^{H} C_{h\to n}\mathbf{V}_{h\to n}}{\sum_{h=1}^{H} C_{h\to n}}, \tag{9}$$

The EM algorithm alternates between an E-step and an M-step. The E-step determines, for

each datapoint (i.e. input capsule), the probability of agreement (i.e. C) between it and each of the Gaussians (i.e. output capsules). The M-step holds the agreement weights constant, and for each Gaussian (i.e. output capsule) consists of finding the mean of these weighted datapoints (i.e. input capsules) and the variance about that mean.

M-Step for each Gaussian (i.e. $\mathbf{\Omega}_n^{out}$) consists of finding the mean $\boldsymbol{\mu}_n$ of the votes from input capsules and the variance $\boldsymbol{\sigma}_n$ about that mean:

$$\boldsymbol{\mu}_n = \frac{\sum_{h=1}^{H} C_{h\to n}\mathbf{V}_{h\to n}}{\sum_{h=1}^{H} C_{h\to n}}, \quad (10)$$

$$(\boldsymbol{\sigma}_n)^2 = \frac{\sum_{h=1}^{H} C_{h\to n}(\mathbf{V}_{h\to n}-\boldsymbol{\mu}_n)^2}{\sum_{h=1}^{H} C_{h\to n}} \quad (11)$$

The incremental cost of using an active capsule $\mathbf{\Omega}_n^{out}$ is

$$\chi_n = \sum_i \big(\log(\boldsymbol{\sigma}_n^i) + \frac{1+\log(2\pi)}{2}\big)\sum_{h=1}^{H} C_{h\to n},$$

where $\boldsymbol{\sigma}_n^i$ denotes the i-th dimension of the variance vector $\boldsymbol{\sigma}_n$. The activation probability of capsule $\mathbf{\Omega}_n^{out}$ is calculated by

$$A_n = logistic\big(\lambda(\beta_A - \beta_\mu \sum_{h=1}^{H} C_{h\to n} - \chi_n)\big),$$

where β_A is a fixed cost for coding the mean and variance of $\mathbf{\Omega}_n^{out}$ when activating it, β_μ is another fixed cost per input capsule when not activating it, and λ is an inverse temperature parameter set with a fixed schedule. We refer the readers to (Hinton et al., 2018) for more details.

E-Step adjusts the assignment probabilities $C_{h\to *}$ for each input $\mathbf{\Omega}_h^{in}$. First, we compute the negative log probability density of the vote $\mathbf{V}_{h\to n}$ from $\mathbf{\Omega}_h^{in}$ under the Gaussian distribution fitted by the output capsule $\mathbf{\Omega}_n^{out}$ it gets assigned to:

$$P_{h\to n} = \sum_i \frac{1}{\sqrt{2\pi(\boldsymbol{\sigma}_n^i)^2}} \exp\Big(-\frac{(\mathbf{V}_{h\to n}^i - \boldsymbol{\mu}_n^i)^2}{2(\boldsymbol{\sigma}_n^i)^2}\Big).$$

Again, i denotes the i-th dimension of the vectors $\{\mathbf{V}_{h\to n}, \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n\}$. Accordingly, the agreement weight is re-normalized by

$$C_{h\to n} = \frac{A_n P_{h\to n}}{\sum_{n'=1}^{N} A_{n'} P_{h\to n'}}. \quad (12)$$

5 Experiments

In this section, we evaluate the performance of our proposed models on both linguistic probing tasks and machine translation tasks.

5.1 Linguistic Probing Tasks

5.1.1 Setup

Tasks Recently, Conneau et al. (2018) designed 10 probing tasks to study what linguistic properties are captured by input representations. A probing task is a classification problem that focuses on simple linguistic properties of sentences. 'SeLen' is to predict the length of sentences in terms of number of words. 'WC' tests whether it is possible to recover information about the original words given its sentence embedding. 'TrDep' checks whether an encoder infers the hierarchical structure of sentences. In 'ToCo' task, sentences should be classified in terms of the sequence of top constituents immediately below the sentence node. 'Bshif' tests whether two consecutive tokens within the sentence have been inverted. 'Tense' asks for the tense of the main-clause verb. 'SubNm' focuses on the number of the subject of the main clause. 'ObjNm' tests for the number of the direct object of the main clause. In 'SOMO', some sentences are modified by replacing a random noun or verb with another noun or verb and the classifier should tell whether a sentence has been modified. 'CoIn' benchmark contains sentences made of two coordinate clauses. Half of the sentences are inverted the order of the clauses and the task is to tell whether a sentence is intact or modified. We conduct probing tasks to study whether the routing-based aggregation benefits multi-head attention to produce more informative representation.

Data and Models The models on each classification task are trained and examined using the open-source dataset provided by Conneau et al. (2018), where each task is assigned 100k sentences for training and 10k sentences for validating and testing. Each of our probing model consists of 3 encoding layers followed by a MLP classifier. For each encoding layer, we employ a multi-head self-attention block and a feed-forward block as in TRANSFORMER-BASE, which have achieved promising results on several NLP tasks (Dehghani et al., 2018; Devlin et al., 2018). The mean of the top encoding layer is served as the sentence

Model	Surface		Syntactic			Semantic				
	SeLen	WC	TrDep	ToCo	BShif	Tense	SubNm	ObjNm	SOMO	CoIn
BASE	**97.22**	97.92	44.48	84.44	49.30	84.20	87.66	82.94	50.24	68.77
SIMPLE	97.10	**98.85**	43.37	86.15	49.87	**88.22**	87.25	85.07	48.77	69.12
EM	96.26	98.75	**47.72**	**87.00**	**51.82**	88.17	**89.97**	**86.40**	**51.55**	**69.86**

Table 1: Classification accuracies on 10 probing tasks of evaluating the linguistic properties ("Surface", "Syntectic", and "Semantic") learned by sentence encoder. "BASE" denotes the standard linear transformation, "SIMPLE" is the simple routing algorithm, and "EM" is the EM routing algorithm.

representation passed to the classifier. The difference between the compared models merely lies in the aggregation mechanism of multiple attention heads: "BASE" uses a standard concatenation and linear transformation, "SIMPLE" and "EM" are assigned simple routing and EM routing algorithms, respectively. For routing algorithms, the number of output capsules and routing iterations are empirically set to 512 and 3.

5.1.2 Results

Table 1 lists the classification accuracies of the three models on the 10 probing tasks. We highlight the best accuracies in bold. Several observations can be made here.

First, *routing-based models produce more informative representation.* The representation produced by encoders with routing-based aggregation outperforms that by the baseline in most tasks, proving that routing mechanisms indeed aggregate attention heads more effectively. The only exception is the sentence length classification task ('SeLen'), which is consistent with the conclusion in (Conneau et al., 2018): as a model captures deeper linguistic properties, it will tend to forget about this superficial feature.

Second, *EM routing outperforms simple routing by embedding more syntactic and semantic information.* As shown in the last row, EM routing for multi-head aggregation consistently achieves best performances on most syntactic and semantic tasks. Especially on task 'TrDep', 'Tense' and 'ObjNm', EM routing-based model surpasses the baseline more than 3 points, demonstrating that EM routing benefits multi-head attention to capture more syntax structure and sentence meaning. Simple routing, however, underperforms the baseline model in some cases such as 'TrDep' and 'SubNm'. We attribute the superiority of EM routing to generating more accurate agreement weights with the Gaussian estimation.

5.2 Machine Translation Tasks

5.2.1 Setup

Data We conduct experiments on the widely-used WMT2014 English⇒German (En⇒De) and WMT2017 Chinese⇒English (Zh⇒En) machine translation tasks. For the En⇒De task, the dataset consists of 4.6M sentence pairs. We use newstest2013 as the development set and newstest2014 as the test set. For the Zh⇒En task, we use all of the available parallel data, consisting of about 20.6M sentence pairs. We use newsdev2017 as the development set and newstest2017 as the test set. We employ byte-pair encoding (BPE) (Sennrich et al., 2016) with 32K merge operations for both language pairs. We use the case-sensitive 4-gram NIST BLEU score (Papineni et al., 2002) as evaluation metric, and bootstrap resampling (Koehn, 2004) for statistical significance test.

Models We implement the proposed approaches on top of the advanced TRANSFORMER model (Vaswani et al., 2017). We follow Vaswani et al. (2017) to set the configurations and have reproduced their reported results on the En⇒De task. The *Base* and *Big* models differ at hidden size (512 vs. 1024) and number of attention heads (8 vs. 16). All the models are trained on eight NVIDIA P40 GPUs where each is allocated with a batch size of 4096 tokens.

TRANSFORMER consists of three attention components: encoder-side self-attention, decoder-side self-attention and encoder-decoder attention, all of which are implemented as multi-head attention. For the information aggregation in multi-head attention, we replace the standard linear transformation with the proposed routing mechanisms. We experimentally set the number of iterations to 3 and the number of output capsules as model hidden size, which outperform other configurations during our investigation.

#	Applying Aggregation to ...			Routing	# Para.	Speed		BLEU	△
1	*Enc-Self*	*Enc-Dec*	*Dec-Self*			Train	Decode		
2	×	×	×	n/a	88.0M	1.92	1.67	27.31	–
3	✓	×	×	Simple	+12.6M	1.23	1.66	27.98	+0.67
4	✓	×	×	EM	+12.6M	1.20	1.65	28.28	+0.97
5	×	✓	×	EM	+12.6M	1.20	1.21	27.94	+0.63
6	×	×	✓	EM	+12.6M	1.21	1.21	28.15	+0.84
7	✓	✓	×	EM	+25.2M	0.87	1.20	28.45	+1.14
8	✓	✓	✓	EM	+37.8M	0.66	0.89	28.47	+1.16

Table 2: Effect of information aggregation on different attention components, i.e., encoder self-attention ("*Enc-Self*"), encoder-decoder attention ("*Enc-Dec*"), and decoder self-attention ("*Dec-Self*"). "# Para." denotes the number of parameters, and "Train" and "Decode" respectively denote the training speed (steps/second) and decoding speed (sentences/second).

#	Layers	# Para.	Train	BLEU
1	None	88.0M	1.92	27.31
2	[1-6]	100.6M	1.20	28.28
3	[4-6]	94.3M	1.54	28.26
4	[1-3]	94.3M	1.54	28.27
5	[1,2]	92.2M	1.67	28.26
6	[6]	90.1M	1.88	27.68
7	[1]	90.1M	1.88	27.75

Table 3: Evaluation of different layers in the encoder, which are implemented as multi-head self-attention with the EM routing based information aggregation. "1" denotes the bottom layer, and "6" the top layer.

5.2.2 Component Analysis

Table 2 lists the results on the En⇒De translation task with TRANSFORMER-BASE. As seen, the proposed routing mechanism outperforms the standard aggregation in all cases, demonstrating the necessity of advanced aggregation functions for multi-head attention.

Routing Mechanisms (Rows 3-4) We first apply simple routing and EM routing to encoder self-attention. Both strategies perform better than the standard multi-head aggregation (Row 1), verifying the effectiveness of the non-linear aggregation mechanisms. Specifically, the two strategies require comparable parameters and computational speed, but EM routing achieves better performance on translation qualities. Considering the training speed and performance, *EM routing* is used as the default multi-head aggregation method in subsequent experiments.

Effect on Attention Components (Rows 4-8) Concerning the individual attention components (Rows 4-6), we found that the encoder and decoder self-attention benefit more from the routing-based information aggregation than the encoder-decoder attention. This is consistent with the finding in (Tang et al., 2018), which shows that self-attention is a strong semantic feature extractor. Encouragingly, applying EM routing in the encoder (Row 4) significantly improve the translation quality with almost no decrease in decoding speed, which matches the requirement of online MT systems. We find that this is due to the auto-regressive generation schema, modifications on the decoder influence the decoding speed more than the encoder.

Compared with individual attention components, applying routing to multiple components (Rows 7-8) marginally improves translation performance, at the cost of a significant decrease of the training and decoding speeds. Possible reasons include that the added complexity makes the model harder to train, and the benefits enjoyed by different attention components are overlapping to some extent. To balance translation performance and efficiency, we only apply EM routing to aggregate multi-head self-attention at the *encoder* in subsequent experiments.

Encoder Layers As shown in Row 4 of Table 2, applying EM routing to all encoder layers significantly decreases the training speed by 37.5%, which is not acceptable since TRANSFORMER is best known for both good performance and quick training. We expect applying to fewer layers can alleviate the training burden. Recent studies show that different layers of NMT encoder can capture different levels of syntax and semantic features (Shi et al., 2016; Peters et al., 2018). There-

System	Architecture	En⇒De		Zh⇒En	
		# Para.	BLEU	# Para.	BLEU
Existing NMT systems					
(Wu et al., 2016)	GNMT	n/a	26.30	n/a	n/a
(Gehring et al., 2017)	ConvS2S	n/a	26.36	n/a	n/a
(Vaswani et al., 2017)	Transformer-Base	65M	27.3	n/a	n/a
	Transformer-Big	213M	28.4	n/a	n/a
(Hassan et al., 2018)	Transformer-Big	n/a	n/a	n/a	24.2
Our NMT systems					
this work	Transformer-Base	88M	27.31	108M	24.13
	+ Effective Aggregation	92M	28.26$^{\Uparrow}$	112M	24.68$^{\Uparrow}$
	Transformer-Big	264M	28.58	304M	24.56
	+ Effective Aggregation	297M	28.96$^{\uparrow}$	337M	25.00$^{\uparrow}$

Table 4: Comparing with existing NMT systems on WMT14 English⇒German ("En⇒De") and WMT17 Chinese⇒English ("Zh⇒En") tasks. "$\uparrow$ / $\Uparrow$": significantly better than the baseline counterpart ($p < 0.05/0.01$).

fore, an investigation to study whether EM routing works for multi-head attention at different layers is highly desirable.

As shown in Table 3, we respectively employ EM routing for multi-head attention at the high-level three layers (Row 3) and low-level three layers (Row 4). The translation quality marginally drop while parameters are fewer and training speeds are quicker. This phenomena verifies that it is unnecessary to apply the proposed model to all layers. We further reduce the applied layers to low-level two (Row 5), the above phenomena still holds. However, a big drop on translation quality occurs when the number of layer is reduced to 1 (Rows 6-7).

Accordingly, to balance translation performance and efficiency, we only apply EM routing for multi-head aggregation at the *low-level two layers of the encoder*, which we term "Effective Aggregation" in the following sections.

5.2.3 Main Results

In this section, we validate the proposed "Effective Aggregation" for multi-head attenion on both WMT17 Zh⇒En and WMT14 En⇒De translation tasks. The results are listed in Table 4. Our implementations of both Transformer-Base and Transformer-Big outperform the reported NMT systems on the same data and match the strong results of Transformer reported in previous works, which we believe make the evaluation convincing. Incorporating the effective aggregation consistently and significantly improves translation performance for both base and big Transformer models across language pairs, demonstrating the efficiency and universality of our proposed multi-head aggregation mechanism.

Moreover, it is encouraging to see that Transformer-Base with effective aggregation strategy even achieves comparable performances to that of Transformer-Big, with about two thirds fewer parameters, which further demonstrates that our performance gains are not simply brought by additional parameters.

6 Conclusion

In this work, we provide first empirical validation on the importance of information aggregation for multi-head attention. Instead of the conventional linear transformation, we propose to aggregate the partial representations learned by multiple attention heads via *routing-by-agreement*. The routing algorithm iteratively updates the proportion of how much a partial representation should be assigned to the final output representation, based on the agreement between parts and wholes.

Experimental results across 10 linguistic probing tasks reveal that our EM routing-based model indeed produces more informative representation, which benefits multi-head attention to capture more syntactic and semantic information. In addition, our approach on various machine translation tasks consistently and significantly outperforms the strong Transformer baseline. Extensive analysis further suggests that only applying EM routing to low-level two layers of the encoder can best balance the translation performance and computational efficiency.

Future work includes combining our information aggregation techniques together with other advanced information extraction models for multi-head attention (Li et al., 2018). We expect that the two kinds of approaches can complement each other to further improve the expressiveness of multi-head attention.

Acknowledgments

Jian Li and Michael R. Lyu were supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14210717 of the General Research Fund), and Microsoft Research Asia (2018 Microsoft Research Asia Collaborative Research Award). We thank the anonymous reviewers for their insightful comments and suggestions.

References

Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. 2018. Weighted Transformer Network for Machine Translation. *arXiv preprint arXiv:1711.02132*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.

Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. 2017. Mutan: Multimodal Tucker Fusion for Visual Question Answering. In *ICCV*.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based Models for Speech Recognition. In *NIPS*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What You Can Cram into A Single $&!#* Vector: Probing Sentence Embeddings for Linguistic Properties. In *ACL*.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2018. Universal Transformers. *arXiv preprint arXiv:1807.03819*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Tobias Domhan. 2018. How Much Attention Do You Need? A Granular Analysis of Neural Machine Translation Architectures. In *ACL*.

Ziyi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *EMNLP*.

Ziyi Dou, Zhaopeng Tu, Xing Wang, Longyue Wang, Shuming Shi, and Tong Zhang. 2019. Dynamic layer aggregation for neural machine translation. In *AAAI*.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In *EMNLP*.

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

Jingjing Gong, Xipeng Qiu, Shaojing Wang, and Xuanjing Huang. 2018. Information Aggregation via Dynamic Routing for Sequence Encoding. In *COLING*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving Human Parity on Automatic Chinese to English News Translation. *arXiv preprint arXiv:1803.05567*.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. 2011. Transforming Auto-encoders. In *ICANN*.

Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. 2018. Matrix Capsules with EM Routing. In *ICLR*.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *EMNLP*.

Rodney LaLonde and Ulas Bagci. 2018. Capsules for object segmentation. *arXiv preprint arXiv:1804.04241*.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. Multi-Head Attention with Disagreement Regularization. In *EMNLP*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. In *ICLR*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for Automatic Evaluation of Machine Translation. In *ACL*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *NAACL*.

Alessandro Raganato and Jörg Tiedemann. 2018. An Analysis of Encoder Representations in Transformer-Based Machine Translation. In *EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic Routing Between Capsules. In *NIPS*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *ACL*.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. DiSAN: Directional Self-Attention Network for RNN/CNN-free Language Understanding. In *AAAI*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does String-based Neural MT Learn Source Syntax? In *EMNLP*.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. In *EMNLP*.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures. In *EMNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *NIPS*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv preprint arXiv:1609.08144*.

Edgar Xi, Selina Bing, and Yang Jin. 2017. Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*.

Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. 2019. Convolutional self-attention networks. In *NAACL*.

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. 2018. Investigating Capsule Networks with Dynamic Routing for Text Classification. In *EMNLP*.