Neurocomputing 454 (2021) 14–24

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

On the diversity of multi-head attention

Jian Li [a], Xing Wang [b], Zhaopeng Tu [b,*], Michael R. Lyu [a]

[a] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China
[b] Tencent AI Lab, China

ARTICLE INFO

Article history:
Received 11 June 2019
Revised 19 October 2020
Accepted 12 April 2021
Available online 17 April 2021
Communicated by Zidong Wang

2020 MSC:
00-01
99-00

Keywords:
Natural language processing
Multi-head attention
Diversity
Routing-by-agreement
Neural machine translation
Sentence encoding

ABSTRACT

Multi-head attention is appealing for the ability to jointly attend to information from different representation subspaces at different positions. In this work, we propose two approaches to better exploit such diversity for multi-head attention, which are complementary to each other. First, we introduce a disagreement regularization to explicitly encourage the diversity among multiple attention heads. Specifically, we propose three types of disagreement regularization, which respectively encourage the subspace, the attended positions, and the output representation associated with each attention head to be different from other heads. Second, we propose to better capture the diverse information distributed in the extracted partial-representations with the *routing-by-agreement* algorithm. 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 the machine translation, sentence encoding and logical inference tasks demonstrate the effectiveness and universality of the proposed approaches, which indicate the necessity of better exploiting the diversity for multi-head attention. While the two strategies individually boost performance, combining them together can further improve the model performance.

1. Introduction

Attention model becomes a standard component of the deep learning networks, contributing to impressive results in machine translation [1,2], image captioning [3], speech recognition [4], among many other applications. Recently, the performance of attention is further improved by multi-head mechanism [5], which concurrently performs the attention functions on different representation subspaces of the input sequence. Consequently, different attention heads are able to capture distinct properties of the input, which are embedded in different subspaces [6]. Subsequently, a linear transformation is generally employed to aggregate the partial representations extracted by different attention heads [5,7], producing the final output representation.

However, the conventional multi-head mechanism may not fully exploit the *diversity* among attention heads. First, one strong point of multi-head attention is the ability to jointly attend to information from *different* representation subspaces at *different* positions. But currently there is no mechanism to guarantee that different attention heads indeed capture distinct information. Second, we believe that information extraction and information aggregation are both important to produce an informative representation. We argue that the straightforward linear transformation are not expressive enough to fully capture the rich information distributed in the extracted partial-representations. In this work, we propose two strategies to better exploit the diversity of multi-head attention, namely *disagreement regularization* and *advanced information aggregation*.

In response to the first problem, we introduce a disagreement regularization term to explicitly encourage the diversity among multiple attention heads. The disagreement regularization serves as an auxiliary objective to guide the training of the related attention component. Specifically, we propose three types of disagreement regularization, which are applied to the three key components that refer to the calculation of information vector using multi-head attention. Two regularization terms are respectively to maximize cosine distances of the input subspaces and output representations, while the last one is to disperse the positions attended by multiple heads with element-wise multiplication of the corresponding attention matrices. The three regularization terms can be either used individually or in combination.

* Corresponding author.
E-mail addresses: jianli@cse.cuhk.edu.hk (J. Li), brightxwang@tencent.com (X. Wang), zptu@tencent.com (Z. Tu), lyu@cse.cuhk.edu.hk (M.R. Lyu).

https://doi.org/10.1016/j.neucom.2021.04.038
0925-2312/

To address the second problem, we replace the standard linear transformation in conventional multi-head attention [5] with an advanced routing-by-agreement algorithm, to better aggregate the diverse information distributed in the extracted partial-representations. Specifically, we cast information aggregation as the *assigning-parts-to-wholes* problem [8], and investigate the effectiveness of the routing-by-agreement algorithm, which is an appealing alternative to solving this problem [9,10]. 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.

In addition, it is natural to combine the two types of approaches and apply them simultaneously, since the former focuses on extracting more diverse information while the latter aims to better aggregate the extracted information. We apply them simultaneously by modifying both the training objective and network architecture.

We evaluate the performance of the proposed approaches on three representative NLP tasks: machine translation, sentence encoding, and logical inference tasks. For machine translation, we validate our approaches on top of the advanced TRANSFORMER model [5] on both WMT14 English⇒German and WMT17 Chinese⇒English data. Experimental results show that our approaches consistently improve the translation performance across language pairs while keeping the computational efficiency. For sentence encoding, we evaluate with the linguistic probing tasks [11], which consist of 10 classification problems to study what linguistic properties are captured by input encoding representations. Probing analysis shows that our approaches indeed produce more informative representation, which embeds more syntactic and semantic information. Experiments on logical inference further demonstrate the ability of modeling hierarchical structure. Precisely, our study reveals that:

- Directly applying disagreement regularization on the output representations of multiple attention heads is most effective.
- The EM routing algorithm shows its superiority on information aggregation over the standard linear transformation and other aggregation algorithms.
- Disagreement regularization and advanced information aggregation are complementary to each other, as indicated from analyses in machine translation and sentence encoding.

This paper combines and extends results presented at the 2018 Conference on Empirical Methods in Natural Language Processing (entitled "Multi-Head Attention with Disagreement Regularization" [12]) and at the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (entitled "Information Aggregation for Multi-Head Attention with Routing-by-Agreement" [13]). The extensions include

1. We further refine our proposed model by combining the two sorts of strategies and exploiting the advantages of simultaneously applying them (Section 3.3). We demonstrate the effectiveness of the combined method in experiments (Table 5).
2. We carry out more experiments and in-depth analyses to validate the effectiveness of our approaches on more tasks, including linguistic probing tasks (Section 4.2) and and logical inference tasks (Section 4.3). Results on linguistic probing tasks prove the superiority of our approach on capturing surface, syntactic and semantic information. Results on logical inference tasks show that the proposed approach performs better at modeling hierarchical structure.
3. We present a more comprehensive description of the proposed models and algorithms (Section 3).
4. For reproducibility, we release the source code, preprocessed data, and trained models, which make it easy to reproduce the experiments in this work.[1]

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. [5] found it is beneficial to capture different context features with multiple individual attention functions, namely multi-head attention. Fig. 1 shows an example of a two–head attention model. For the query word "Bush", green and red heads pay attention to different positions of "talk" and "Sharon".

Formally, attention function maps a sequence of query $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ 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 $\mathbf{Q} \in \mathbb{R}^{n\times d}, \{\mathbf{K}, \mathbf{V}\} \in \mathbb{R}^{m\times d}$. More specifically, multi-head attention model first transforms $\mathbf{Q}, \mathbf{K}$, and $\mathbf{V}$ into H subspaces with different, learnable linear projections:

$$\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h = \mathbf{Q}\mathbf{W}_h^Q, \mathbf{K}\mathbf{W}_h^K, \mathbf{V}\mathbf{W}_h^V, \tag{1}$$

where $\{\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h\}$ are respective the query, key, and value representations of the h-th head. $\left\{\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V\right\} \in \mathbb{R}^{d\times\frac{d}{H}}$ 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:

$$\mathbf{O}_h = Att(\mathbf{Q}_h, \mathbf{K}_h)\mathbf{V}_h, \tag{2}$$

where $\mathbf{O}_h \in \mathbb{R}^{n\times\frac{d}{H}}, Att(\cdot)$ is an attention model. In this work, we use scaled dot-product attention [2], which achieves similar performance with its additive counterpart [1] while is much faster and more space-efficient in practice [5].

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

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

$$\text{Linear}: \mathbf{O} = \widehat{\mathbf{O}}\mathbf{W}^O, \tag{4}$$

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

3. Approach

In this work, we propose to better exploit the diversity of multi-head attention from two perspectives:

- *Disagreement Regularization*: Conventional multi-head attention conducts multiple attention functions in parallel (Eq. 2), while there is no mechanism to guarantee that different attention heads indeed capture distinct information. In response to this problem, we introduce disagreement regularizations to explicitly encourage different attention heads to extract distinct information (Section 3.1);
- *Advanced Information aggregation*: As shown in Eqs. 3 and 4, the standard multi-head attention uses a straightforward concatenation and linear mapping to aggregate the partial-representations captured by multiple attention heads. We argue that this straightforward strategy may not fully exploit the expressiveness of multi-head attention, which can benefit from

[1] https://github.com/jack57lee/Diversify-MHA

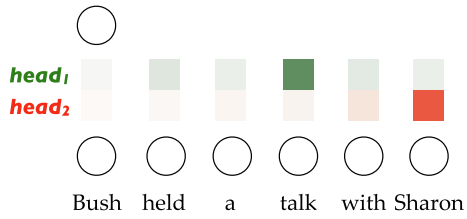

Fig. 1. Illustration of the multi-head attention, which jointly attends to different representation subspaces (colored boxes) at different positions (darker color denotes higher attention probability).

advanced information aggregation. In this study, we exploit more advanced routing-by-agreement method to aggregate the information extracted by different attention heads (Section 3.2).

The disagreement regularization encourages multiple attention functions to extract different information, and advanced information aggregation helps better aggregate the extracted information. Therefore, the two approaches are complementary to each other and can be employed simultaneously, which we will describe in Section 3.3.

3.1. Disagreement Regularization

Multi-head attention allows the model to jointly attend to information from *different* representation subspaces at *different* positions. To further guarantee the diversity, we enlarge the distances among multiple attention heads with disagreement regularization. To this end, we introduce an auxiliary regularization term in order to encourage the diversity among multiple attention heads. Taking the machine translation task as example, the training objective is revised as:

$$J(\theta) = \text{argmax}_\theta \left\{ \underbrace{L(\mathbf{y}|\mathbf{x};\theta)}_{likelihood} + \lambda * \underbrace{D(\mathbf{a}|\mathbf{x},\mathbf{y};\theta)}_{disagreement} \right\},$$

where $\mathbf{a}$ is the referred attention matrices, λ is a hyper-parameter and is empirically set to 1.0 in this paper. The auxiliary regularization term $D(\cdot)$ guides the related attention component to capture different features from the corresponding projected subspaces. Note that the introduced regularization term works like $L1$ and $L2$ terms, which do not introduce any new parameters and only influence the training of the standard model parameters.

Specifically, we propose three types of disagreement regularization to encourage each head vector $\mathbf{O}_h$ to be different from other heads:

- **Disagreement on Subspaces.** This disagreement is designed to maximize the cosine distance between the projected values. Specifically, we first calculate the cosine similarity $\cos(\cdot)$ between the vector pair V^i and V^j in different value subspaces, through the dot product of the normalized vectors[2], which measures the cosine of the angle between V^i and V^j. Thus, the cosine distance is defined as negative similarity, i.e, $-\cos(\cdot)$. Our training objective is to enlarge the average cosine distance among all head pairs. The regularization term is formally expressed as:

$$D_{subpace} = -\frac{1}{H^2}\sum_{i=1}^{H}\sum_{j=1}^{H}\frac{V^i \cdot V^j}{\|V^i\|\|V^j\|}. \quad (5)$$

- **Disagreement on Attended Positions.** Another strategy is to disperse the attended positions predicted by multiple heads. Inspired by the agreement regularization [14,15] which encourages multiple alignments to be similar, in this work, we deploy a variant of the original term by introducing an alignment disagreement regularization. Formally, we employ the sum of element-wise multiplication of corresponding matrix cells[3], to measure the similarity between two alignment matrices A^i and A^j ($Att(\cdot)$ in Eq. 2) of two heads:

$$D_{position} = -\frac{1}{H^2}\sum_{i=1}^{H} um_{j=1}^{H} |A^i \odot A^j|. \quad (6)$$

- **Disagreement on Outputs.** This disagreement directly applies regularization on the outputs of each attention head, by maximizing the difference among them. Similar to the *subspace* strategy, we employ negative cosine similarity to measure the distance:

$$D_{output} = -\frac{1}{H^2}\sum_{i=1}^{H}\sum_{j=1}^{H}\frac{O^i \cdot O^j}{\|O^i\|\|O^j\|}. \quad (7)$$

3.2. Advanced Information Aggregation

Information aggregation in multi-head attention (e.g. Eqs. 3 and 4) aims at composing the partial representations 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 [16–18]. In this work, we cast information aggregation in multi-head attention as the problem of *assigning-parts-to-wholes*, to which an appealing solution is the *routing-by-agreement* algorithm, as shown in Fig. 2.

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.

Mathematically, the input capsules $\Omega^{in} = \left\{\Omega_1^{in}, \ldots, \Omega_H^{in}\right\}$ with $\Omega^{in} \in \mathbb{R}^{n\times d}$ are constructed from the outputs of multi-head attention:

$$\Omega_h^{in} = f_h\left(\widehat{\mathbf{O}}\right), \quad (8)$$

where $f_h(\cdot)$ is a distinct non-linear transformation function associated with the input capsule Ω_h^{in}. Given N output capsules, each input capsule Ω_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

$$\mathbf{V}_{h\to n} = \Omega_h^{in}\mathbf{W}_{h\to n}, \quad (9)$$

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

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

[2] We did not employ the Euler Distance between vectors since we do not care the absolute value in each vector.

[3] We also used the squared element-wise subtraction of two matrices in our preliminary experiments, and found it underperforms its multiplication counterpart, which is consistent with the results in [15].

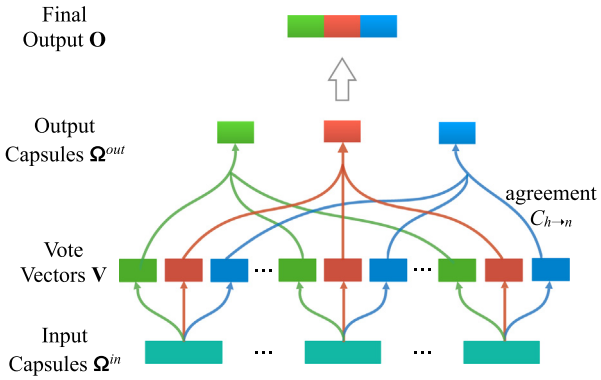

Fig. 2. Illustration of routing-by-agreement.

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 Ω_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, we concatenate the N output capsules to form the final representation: $\mathbf{O}=[\Omega_1^{out},\ldots,\Omega_N^{out}]$. 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 $\frac{d}{N}$.

In this work, we explore two representative routing mechanisms, namely *simple routing* [9] (Section 3.2.1) and *EM routing* [10] (Section 3.2.2), which differ at how the agreement weights $C_{h\to n}$ are calculated.

3.2.1. Simple Routing

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: $\quad\quad\forall\Omega_n^{out}$: compute Ω_n^{out} by Eq. 10
6: $\quad\quad\forall\mathbf{V}_{h\to *}$: $B_{h\to n}+=\Omega_n^{out}\cdot\mathbf{V}_{h\to n}$
$\quad$**return** Ω

Algorithm 1 lists a straightforward implementation of routing. $B_{h\to n}$ measures the degree that the input capsule Ω_h^{in} should be coupled to the output capsule Ω_h^{in}, which is initialized as all 0 (Line 2). The agreement weights $C_{h\to n}$ are then iteratively refined by measuring the agreement between the vote vector $\mathbf{V}_{h\to n}$ and the output capsule Ω_n^{out} (Lines 4–6), which is implemented as a simple scalar product $\Omega_n^{out}\cdot\mathbf{V}_{h\to n}$ (Line 5).

To represent the probability that the output capsule Ω_n^{out} is activated, we follow Sabour et al. [9] use a non-linear "squashing" function:

$$\Omega_n^{out}=\frac{||\Omega_n^{out}||^2}{1+||\Omega_n^{out}||^2}\frac{\Omega_n^{out}}{||\Omega_n^{out}||},\tag{11}$$

The scalar product $\Omega_n^{out}\cdot\mathbf{V}_{h\to n}$ 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. [10] propose a novel Expectation–Maximization (EM) routing algorithm.

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 (Eq. 11). Second, it casts the routing process as fitting a mixture of Gaussians using EM, 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.

3.2.2. EM Routing

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: $\quad\quad\forall\Omega_n^{out}$: 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: $\quad\forall\Omega_n^{out}$: $\Omega_n^{out}=A_n\ *\boldsymbol{\mu}_n$
$\quad$**return** Ω

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 Ω_n^{out} is calculated as

$$\Omega_n^{out}=A_n*\boldsymbol{\mu}_n=A_n*\frac{\sum_{h=1}^{H}C_{h\to n}\mathbf{V}_{h\to n}}{\sum_{h=1}^{H}C_{h\to n}},\tag{12}$$

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. Ω_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:

$$\boldsymbol{\mu}_n=\frac{\sum_{h=1}^{H}C_{h\to n}\mathbf{V}_{h\to n}}{\sum_{h=1}^{H}C_{h\to n}},\tag{13}$$

$$(\boldsymbol{\sigma}_n)^2=\frac{\sum_{h=1}^{H}C_{h\to n}(\mathbf{V}_{h\to n}-\boldsymbol{\mu}_n)^2}{\sum_{h=1}^{H}C_{h\to n}}.\tag{14}$$

The incremental cost of using an active capsule Ω_n^{out} is

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

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

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

where β_A is a fixed cost for coding the mean and variance of Ω_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 [10] for more details.

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

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

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

$$C_{h\to n}=\frac{A_n P_{h\to n}}{\sum_{n'=1}^{N} A_{n'} P_{h\to n'}}. \tag{15}$$

3.3. Combining disagreement regularization and information aggregation

Coupling different representations with diversity is a well-known technique to improve the performance [19]. While disagreement regularization focuses on adjusting the training objective, i.e. the loss function, advanced information aggregation aims at modifying the network architecture. In terms of functionality, they are potentially complementary to each other as one improves information extraction and the other benefits information aggregation. Therefore, it is natural to combine the two approaches and apply them simultaneously. In consideration of computation cost, we first respectively choose the best strategy from the two kinds of approach, and then apply them simultaneously by modifying both the training objective and network architecture.

Note that there are many possible ways to implement the general idea of combining disagreement regularization and information aggregation. The aim of this paper is not to explore this whole space but simply to show that one fairly straightforward implementation works well and the two methods are complementary to each other.

4. Experiments

In this section, we validate the effectiveness of our approaches on machine translation tasks (Section 4.1), sentence encoding tasks (Section 4.2), and logical inference tasks(Section 4.3). We conduct ablation study of the proposed approaches on the benchmark machine translation tasks, and carry out final evaluation on all the other tasks.

4.1. Machine translation tasks

4.1.1. Setting

Data. We conduct experiments on the widely-used WMT2014 English⇒German (En⇒De) and WMT2017 Chinese⇒English (Zh⇒En) translation tasks. For the En⇒De task, the dataset consists of 4.6 M 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 with maximum length limited to 50, consisting of about 20.6 M sentence pairs. We use newsdev2017 as the development set and newstest2017 as the test set. We employ byte pair encoding (BPE) [20] with 32 K merge operations for both language pairs. We use the case-sensitive 4-gram NIST BLEU score [21] as evaluation metric, and bootstrap resampling [22] for statistical significance test.

Models. We implement the proposed approaches on top of the advanced TRANSFORMER model [5]. We follow Vaswani et al. [5] to set the configurations and have reproduced their reported results on the En⇒De task with both *Base* and *Big* models. The embedding size of and hidden size of *Base* model are 512, the filter size is 2048 and the number of attention head is 8. The *Big* model has embedding size and hidden size of 1024, filter size of 4096 and attention heads of 16. For both *Base* and *Big* models, the number of encoder and decoder layers is 6, all types of dropout rate is set to 0.1. The Adam optimizer [23] is employed with $\beta_1=0.9,\beta_2=0.98$ and $\epsilon=10^{-9}$. The learning rate is initially 1.0 and linearly warms up over the first 4,000 steps, then decreases proportionally to the inverse square root of the step number [5]. Label smoothing is set to 0.1 during training [24]. All the models are trained on eight NVIDIA P40 GPUs where each is allocated with a batch size of 4096 tokens. All the results we reported are based on the individual models without using the averaging or ensemble model.

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.

4.1.2. Ablation study on disagreement regularization

4.1.2.1. Effect of regularization terms. In this section, we evaluate the impact of different regularization terms on the Zh⇒En task using TRANSFORMER-BASE. For simplicity and efficiency, here we only apply regularizations on the encoder side. As shown in Table 1, all the models with the proposed disagreement regularizations (Rows 2–4) consistently outperform the vanilla TRANSFORMER (Row 1). Among them, the *Output* term performs best which is + 0.65 BLEU score better than the baseline model, the *Position* term is less effective than the other two. In terms of training speed, we do not observe obvious decrease, which in turn demonstrates the advantage of our disagreement regularizations.

However, the combinations of different disagreement regularizations fail to further improve translation performance (Rows 5–7). One possible reason is that different regularization terms have overlapped guidance, and thus combining them does not introduce too much new information while makes training more difficult.

4.1.2.2. Effect on attention components. The TRANSFORMER consists of three attention networks, including encoder self-attention, decoder self-attention, and encoder-decoder attention. In this experiment, we investigate how each attention network benefits from the disagreement regularization. As seen from Table 2, all models consistently improve upon the baseline model. When applying disagreement regularization to all three attention networks, we achieve the best performance, which is + 0.72 BLEU score better than the baseline model. The training speed decreases by 12%, which is acceptable considering the performance improvement.

Table 1
Effect of regularization terms, which are applied to the encoder self-attention only. "Speed" denotes the training speed (steps/s). Results are reported on the WMT17 Zh⇒En translation task using TRANSFORMER-BASE.

#	Regularization			Speed	BLEU
	Subspace	*Position*	*Output*		
1	×	×	×	1.21	24.13
2	✓	×	×	1.15	24.64
3	×	✓	×	1.14	24.42
4	×	×	✓	1.15	**24.78**
5	✓	×	✓	1.12	24.73
6	✓	✓	×	1.11	24.38
7	✓	✓	✓	1.05	24.60

Table 2
Effect of regularization on different attention networks, i.e., encoder self-attention ("*Encoder*"), encoder-decoder attention ("*Encoder-Decoder*"), and decoder self-attention ("*Decoder*"). We use *Output Disagreement* as the regularization term. Results are reported on the WMT17 Zh⇒En translation task using Transformer-Base.

Applying to			Speed	BLEU
Encoder	*Encoder-Decoder*	*Decoder*		
×	×	×	1.21	24.13
✓	×	×	1.15	24.78
✓	✓	×	1.10	24.67
✓	×	✓	1.11	24.69
✓	✓	✓	1.06	**24.85**

In the following sections, we apply the *Output Disagreement* to all the three attention networks, which we term "Disagreement".

4.1.2.3. Visualization of attention heads. To more directly show the effectiveness of three disagreement regularizations, we visualize different attention heads in a two-dimensional embedding space via the t-SNE technique [25]. Specifically, we run Transformer-Base with different disagreements on the WMT17 Zh⇒En development set and record all the head representations on the encoder side. Then we average the representations over all the words and all the layers, obtaining 8 representation vectors for the 8 attention heads which are subsequently fed to t-SNE projection. We plot the projected results in Fig. 3, where different colors denote different disagreements. We can see that the *Output Disagreement* is most effective at dispersing the attention heads while the *Position Disagreement* almost does not have such function. This finding is consistent with our results in Table 1.

4.1.3. Ablation study on information aggregation

In this section, we evaluate the impact of different information aggregation functions on the En⇒De task using Transformer-Base. As the results shown in Table 3, the proposed routing mechanisms outperform the standard aggregation in all cases, demonstrating the necessity of advanced aggregation functions for multi-head attention.

4.1.3.1. 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.

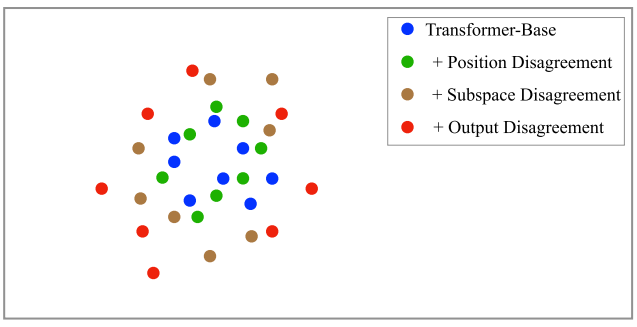

Fig. 3. t-SNE visualization of attention heads with different disagreement regularizations.

Table 3
Effect of information aggregation on different attention components, i.e., encoder self-attention ("*Enc*"), encoder-decoder attention ("*E-D*"), and decoder self-attention ("*Dec*"). "Para." denotes the number of parameters, and "Speed" denotes the training speed (steps/second). Results are reported on the WMT14 En⇒De translation task using Transformer-Base.

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

4.1.3.2. 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 [26], 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.

4.1.3.3. Encoder layers. As shown in Row 4 of Table 3, 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 [27,28]. Therefore, an investigation to study whether EM routing works for multi-head attention at different layers is highly desirable.

As shown in Table 4, 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 drops while parameters are fewer and training speeds are quicker.

Table 4
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. Results are reported on the WMT14 En⇒De translation task using Transformer-Base.

#	Layers	Para.	Train	BLEU
1	None	88.0M	1.92	27.31
2	[1-6]	+12.6M	1.20	28.28
3	[4-6]	+6.3M	1.54	28.26
4	[1-3]	+6.3M	1.54	28.27
5	[1,2]	+4.2M	1.67	28.26
6	[6]	+2.1M	1.88	27.68
7	[1]	90.1M	1.88	27.75

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 "Aggregation" in the following sections.

4.1.4. Combining together and main results

Finally, we validate the proposed disagreement regularization and advanced information aggregation for multi-head attention on both WMT14 En⇒De and WMT17 Zh⇒En translation tasks. The results are concluded in Table 5. Our baseline models, both TRANSFORMER-Base and TRANSFORMER-Big, outperform all existing NMT systems on the same data, and match the results of TRANSFORMER reported in previous works, which we believe make the evaluation convincing.

As seen, incorporating disagreement regularization and advanced information aggregation consistently improve translation performance for both base and big TRANSFORMER models across language pairs, demonstrating the efficiency and universality of the proposed approaches. Combining them together further improves translation performances, which confirms our conjecture that the two approaches are complementary to each other as one improves information extraction and the other benefits information aggregation. It is encouraging to see that TRANSFORMER-Base with both approaches even achieves comparable performance to TRANSFORMER-Big, with about two thirds fewer parameters, which further demonstrates that our performance gains are not simply brought by additional parameters.

Less Improvements on Transformer-Big From the last 4 rows of Table 5, we can see that while the two proposed methods individually boost the model performance for big Transformer models, combining them together improves translation performance little, for instance, the improvement on the En⇒De dataset over baseline is only 0.51 BLEU. One possible reason is that, TRANSFORMER-big has more attention heads than the base model (16 vs. 8), which potentially alleviate the diversity problem when augmented with individual approach (e.g., disagreement regularization or information aggregation). To verify this hypothesis, we conduct another experiment on TRANSFORMER-big with only 8 attention heads. As the results shown in Table 6, the final combination model outperforms baseline with 0.84 BLEU score, which is more significant than the TRANSFORMER-big with 16 heads (i.e., 0.51 BLEU).

Table 6
Evaluation on TRANSFORMER-big with 8 attention heads on the WMT14 En⇒De translation task.

Model	Para.	BLEU	△
TRANSFORMER-Big (8 heads)	264 M	28.10	–
+ Disagreement	264 M	28.63	+0.53
+ Aggregation	288 M	28.65	+0.55
+ Both	288 M	28.94	+0.84

4.2. Linguistic probing tasks

Although we have shown that our proposed disagreement regularization and advanced information aggregation can improve NMT systems with respect to the translation quality, we still have a poor understanding of what they are capturing and changing from the linguistic perspective. Recently, Conneau et al. [11] designed 10 probing tasks to study what linguistic properties are captured by input encoding representations. We conduct these probing tasks here to study whether our proposed approaches can benefit multi-head attention to produce more informative representations.

4.2.1. Task Description

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

Table 5
Comparing with existing NMT systems on WMT14 English⇒German ("En⇒De") and WMT17 Chinese⇒English ("Zh⇒En") tasks. "↑ / ⇑": significantly better than the baseline counterpart ($p < 0.05/0.01$), tested by bootstrap resampling.

Architecture	En⇒De		Zh⇒En	
	# Para.	BLEU	# Para.	BLEU
Existing NMT systems				
RNN with 8 layers [29]	n/a	26.30	n/a	n/a
CNN with 15 layers [30]	n/a	26.36	n/a	n/a
TRANSFORMER-Base [5]	65 M	27.3	n/a	n/a
TRANSFORMER-Big [5]	213 M	28.4	n/a	n/a
TRANSFORMER-Big [31]	n/a	n/a	n/a	24.2
Our NMT systems				
TRANSFORMER-Base	88 M	27.31	108 M	24.13
+ Disagreement	88 M	28.20⇑	108 M	24.85⇑
+ Aggregation	92 M	28.26⇑	112 M	24.68⇑
+ Both	92 M	28.41⇑	112 M	24.90⇑
TRANSFORMER-Big	264 M	28.58	304 M	24.56
+ Disagreement	264 M	28.96↑	304 M	25.08⇑
+ Aggregation	297 M	28.96↑	337 M	25.00↑
+ Both	297 M	29.09⇑	337 M	25.12⇑

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.

4.2.2. Data and models

The models on each classification task are trained and examined using the open-source dataset provided by Conneau et al. [11], where each task is assigned 100 k sentences for training and 10 k sentences for validating and testing. Each of our probing model consists of 6 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 [32,33]. The mean of the top encoding layer is served as the sentence representation passed to the classifier. The difference between the compared models merely lies in the disagreement or aggregation mechanism of multiple attention heads. As we have conduct ablation study on translation task, here we merely evaluate the representative models in each category. "Disagreement" and "Aggregation" are assigned output disagreement regularization and EM routing algorithm respectively, while "Combine" denotes employing the two mechanisms simultaneously. The learning rate is set to 0.0005 with the Adam optimizer and the models are trained for 250 epochs.

4.2.3. Experimental results

Table 7 lists the classification accuracies of the three models on the 10 probing tasks. We highlight the best accuracies under each category (i.e., "Surface", "Syntactic", and "Semantic") in bold. Obviously, the proposed models outperform the baseline system on almost all the probing tasks, verifying that more informative representations are produced by enhancing multi-head attention networks with disagreement regularization and advanced information aggregation. Besides, several interesting observations can be made here.

First, disagreement regularization gains better results on surface and syntactic tasks than advanced information aggregation, as indicated with the italic numbers in Table 7. Advanced information aggregation, on the contrary, performs better on semantic tasks, especially on "SubNm" and "ObjNm" tasks which are the benchmarks for examining the semantic consistency of the model. This empirical result also is consistent with the conclusion in [11]: as a model captures deeper linguistic properties, it will tend to forget about some superficial features.

Table 7
Classification accuracies on 10 probing tasks of evaluating the linguistic properties ("Surface", "Syntactic", and "Semantic") embedded in the encoding representation produced by each model. "Ave." denotes the averaged accuracy in each type of linguistic tasks. "Disagreement" denotes the disagreement regularization, "Aggregation" denotes the advanced information aggregation, and "Combine" is the combination of the two mechanisms.

Task		Baseline	Disagreement	Aggregation	Combine
Surface	SeLen	95.35	96.47	96.02	96.55
	WC	98.03	98.65	98.31	98.87
	Ave.	96.69	*97.56*	97.15	**97.71**
Syntactic	TrDep	44.40	46.54	45.77	46.93
	ToCo	83.48	84.24	84.05	84.17
	BShif	51.45	53.54	50.97	54.26
	Ave.	59.77	*61.44*	60.26	**61.78**
Semantic	Tense	84.57	85.03	85.56	86.07
	SubNm	82.80	83.15	85.47	85.84
	ObjNm	80.31	80.49	82.46	83.38
	SOMO	49.87	49.58	50.09	50.13
	CoIn	69.39	68.48	70.21	69.99
	Ave.	73.38	73.34	*74.76*	**75.08**

Second, the combination of the two mechanisms achieves the best accuracies on almost all tasks, which is on par with the results in machine translation task. Concerning the three main categories, the relative improvements over the baseline are respectively 1.05%, 3.36%, and 2.31%. Together with the first observation, we can conclude that the two types of approaches are complementary to each other concerning extracting linguistic information of the input sentence.

Note that the improvement on *syntactic* tasks is most significant, we further conduct another experiment to evaluate the ability of modeling syntactic structures in next section.

4.3. Logical Inference Tasks

4.3.1. Task description

We finally verify the model's performance in the logical inference task proposed by Bowman et al. [34]. This task is well suited to evaluate the ability of modeling hierarchical structure. Models need to learn the hierarchical and nested structures of language in order to predict accurate logical relations between sentences [34–36].

The task has six types of words {*a, b, c, d, e, f*} in the vocabulary and three logical operators {*or*, *and*, *not*}. The goal of the task is to predict one of seven logical relations between two given sentences. These seven relations are: two entailment types ($\sqsubset, \sqsupset$), equivalence ($\equiv$), exhaustive and non-exhaustive contradiction ($\wedge, |$), and semantic independence ($\#, \smile$). Below is a sample from the data:

$$(\text{not } ((\text{a}(\text{and a}))(\text{or}(\text{not e})))) \# (\text{not d})$$

4.3.2. Data and models

We use the data described in Bowman et al. [34][4]. The train/dev/test dataset ratios are set to 0.8/0.1/0.1 with the number of logical operations range from 1 to 12. We follow Tran et al. [35,37] to implement the architectures: premise and hypothesis sentences are encoded in fixed-size vectors, which are concatenated and fed to a three layer feed-forward network for classification of the logical relation.

We evaluate the baseline Transformer, Transformer with output disagreement, Transformer with EM routing aggregation, and Transformer with the two mechanisms combined together. We follow Tran et al. [35] to use two hidden layers with residual connection [38] in all models and employ two trainable queries to obtain the fixed-size representation. Both word embedding size and hidden size are set to 256. All models have two layers, a dropout rate of 0.2, a learning rate of 0.0001 with Adam optimizer, and were trained for 100 epochs.

4.3.3. Experimental results

As shown in Fig. 4, the combined model outperforms the baseline Transformer and the two individual models on all cases. It is interesting to see that Disagreement performs better at short sequences while Aggregation performs better at long sequences, verifying our hypothesis that they are complementary to each other. Consistent with Shen et al. [36], on the longer sequences ($\geqslant 7$) that were not observed in training, the combined model also obtains the best performance and has a larger gap compared with other models than on the shorter sequences ($\leqslant 6$), which verifies that the combined model is better at modeling more complex hierarchical structure in sequence. It also indicates that the combined model has stronger generalization ability.

[4] https://github.com/sleepinyourhat/vector-entailment

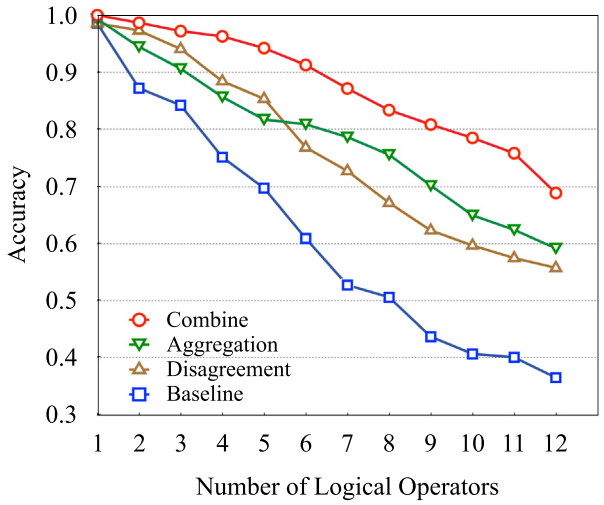

Fig. 4. The results of logical inference. "Disagreement" and "Aggregation" denote the output disagreement regularization and EM routing aggregation respectively, while "Combine" denotes employing the two mechanisms simultaneously.

5. Related work

5.1. Multi-head attention

Multi-head attention has shown promising empirical results in many NLP tasks, such as machine translation [5,39], semantic role labeling [40], dialog [41], subject-verb agreement task [26]. 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. [42] introduce a penalization term to reduce the redundancy of attention weights among different attention heads, and Yang et al. [43] model the interactions among attention heads. Shen et al. [44] explicitly use multiple attention heads to model different dependencies of the same word pair, and Strubell et al. [40] 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.

5.2. Agreement learning

The regularization on attended positions is inspired by agreement learning in prior works, which encourages alignments or hidden variables of multiple models to be similar. Liang et al. [14] assign agreement terms for jointly training word alignment in phrase-based statistic machine translation. The idea is further extended into other natural language processing tasks such as grammar induction [45]. Levinboim et al. [46] extend the agreement for general bidirectional sequence alignment models with model inevitability regularization. Cheng et la. [15] further explore the agreement on modeling the source-target and target-source alignments in neural machine translation model. In contrast to the mentioned approaches which assign agreement terms into loss function, we deploy an alignment disagreement regularization by maximizing the distance among multiple attention heads.

5.3. Information aggregation

Information aggregation in multi-head attention (e.g. Eqs. 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. [16] and Ben et al. [17] 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. [28] aggregate layer representations with linear combination, and Dou et al. [18] compose deep representations with layer aggregation and multi-layer attention mechanisms. Li et al. [47] exploit neuron interaction to aggregate different layers for neural machine translation.

Recently, the routing-by-agreement algorithm, which origins from the capsule networks [8], 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 [9,10], CIFAR tasks [48], and object segmentation task [49]. The applications of capsule networks in NLP tasks, however, have not been widely investigated to date. Zhao et al. [50] testify capsule networks on text classification tasks and Gong et al. [51] propose to aggregate a sequence of vectors via dynamic routing for sequence encoding. Dou et al. [52] 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 machine translation and linguistic probing tasks, which demonstrates the necessity and effectiveness of advanced information aggregation for multi-head attention.

6. Conclusion

In this work, we propose to better exploit the diversity of multi-head attention by incorporating disagreement regularization and employing advanced information aggregation. To this end, we propose several effective and efficient strategies to implement the disagreement regularization and advanced information aggregation. We find that the output disagreement term and EM routing algorithm yield the best performances, and are complementary to each other. Experimental results on machine translation, linguistic probing and logical inference tasks demonstrate the effectiveness and universality of the proposed approaches, suggesting that our models produce more informative representation of the input sentence.

The models also suggest a wide range of potential advantages and extensions, from being able to improve the performance of multi-head attention in other tasks such as reading comprehension and language inference, to being able to combine with other techniques [44,18,53] to further improve the performance of multi-head attention.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the anonymous reviewers for their insightful comments and suggestions.

References

[1] D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, in: ICLR, 2015..
[2] M.-T. Luong, H. Pham, C. D. Manning, Effective Approaches to Attention-based Neural Machine Translation, in: EMNLP, 2015..
[3] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, Y. Bengio, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention., in: ICML, 2015..
[4] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, Attention-based Models for Speech Recognition, in: NIPS, 2015..
[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need, in: NIPS, 2017..
[6] A. Raganato, J. Tiedemann, An Analysis of Encoder Representations in Transformer-Based Machine Translation, in: EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, 2018..
[7] K. Ahmed, N. S. Keskar, R. Socher, Weighted Transformer Network for Machine Translation, in: arXiv preprint arXiv:1711.02132, 2018..
[8] G. E. Hinton, A. Krizhevsky, S. D. Wang, Transforming Auto-encoders, in: ICANN, 2011..
[9] S. Sabour, N. Frosst, G. E. Hinton, Dynamic Routing Between Capsules, in: NIPS, 2017..
[10] G. E. Hinton, S. Sabour, N. Frosst, Matrix Capsules with EM Routing, in: ICLR, 2018..
[11] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, M. Baroni, What You Can Cram into A Single $&!#* Vector: Probing Sentence Embeddings for Linguistic Properties, in: ACL, 2018..
[12] J. Li, Z. Tu, B. Yang, M. R. Lyu, T. Zhang, Multi-Head Attention with Disagreement Regularization, in: EMNLP, 2018..
[13] J. Li, B. Yang, Z.-Y. Dou, X. Wang, M. R. Lyu, Z. Tu, Information aggregation for multi-head attention with routing-by-agreement, in: NAACL, 2019..
[14] P. Liang, B. Taskar, D. Klein, Alignment by agreement, in: NAACL, 2006..
[15] Y. Cheng, S. Shen, Z. He, W. He, H. Wu, M. Sun, Y. Liu, Agreement-based joint training for bidirectional attention-based neural machine translation, in: IJCAI, 2016..
[16] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, M. Rohrbach, Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding, in: EMNLP, 2016..
[17] H. Ben-Younes, R. Cadene, M. Cord, N. Thome, Mutan: Multimodal Tucker Fusion for Visual Question Answering, in: ICCV, 2017..
[18] Z. Dou, Z. Tu, X. Wang, S. Shi, T. Zhang, Exploiting deep representations for neural machine translation, in: EMNLP, 2018..
[19] W. Jiang, L. Ma, Y.-G. Jiang, W. Liu, T. Zhang, Recurrent fusion network for image captioning, arXiv preprint arXiv:1807.09986..
[20] R. Sennrich, B. Haddow, A. Birch, Neural Machine Translation of Rare Words with Subword Units, in: ACL, 2016..
[21] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: A method for Automatic Evaluation of Machine Translation, in: ACL, 2002..
[22] P. Koehn, Statistical Significance Tests for Machine Translation Evaluation, in: EMNLP, 2004..
[23] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, ICLR..
[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[25] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605..
[26] G. Tang, M. Müller, A. Rios, R. Sennrich, Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures, in: EMNLP, 2018..
[27] X. Shi, I. Padhi, K. Knight, Does String-based Neural MT Learn Source Syntax?, in: EMNLP, 2016..
[28] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep Contextualized Word Representations, in: NAACL, 2018..
[29] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, arXiv preprint arXiv:1609.08144..
[30] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin, Convolutional sequence to sequence learning, in: ICML, 2017..
[31] H. Hassan, A. Aue, C. Chen, V. Chowdhary, J. Clark, C. Federmann, X. Huang, M. Junczys-Dowmunt, W. Lewis, M. Li, et al., Achieving Human Parity on Automatic Chinese to English News Translation, arXiv preprint arXiv:1803.05567..
[32] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, L. Kaiser, Universal Transformers, arXiv preprint arXiv:1807.03819..
[33] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805..
[34] S. R. Bowman, C. D. Manning, C. Potts, Tree-structured composition in neural networks without tree-structured architectures, NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches..
[35] K. Tran, A. Bisazza, C. Monz, The importance of being recurrent for modeling hierarchical structure, in: EMNLP, 2018..
[36] Y. Shen, S. Tan, A. Sordoni, A. Courville, Ordered neurons: Integrating tree structures into recurrent neural networks, in: ICLR, 2019..
[37] J. Hao, X. Wang, S. Shi, J. Zhang, Z. Tu, Towards better modeling hierarchical structure for self-attention with ordered neurons, in: EMNLP, 2019..
[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016..
[39] T. Domhan, How Much Attention Do You Need? A Granular Analysis of Neural Machine Translation Architectures, in: ACL, 2018..
[40] E. Strubell, P. Verga, D. Andor, D. Weiss, A. McCallum, Linguistically-Informed Self-Attention for Semantic Role Labeling, in: EMNLP, 2018..
[41] C. Tao, S. Gao, M. Shang, W. Wu, D. Zhao, R. Yan, Get the point of my utterance! learning towards effective responses with multi-head attention mechanism., in: IJCAI, 2018..
[42] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A Structured Self-attentive Sentence Embedding, in: ICLR, 2017..
[43] B. Yang, J. Li, D. F. Wong, L. S. Chao, X. Wang, Z. Tu, Context-aware self-attention networks, in: AAAI, 2019..
[44] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, C. Zhang, DiSAN: Directional Self-Attention Network for RNN/CNN-free Language Understanding, in: AAAI, 2018..
[45] P. S. Liang, D. Klein, M. I. Jordan, Agreement-based Learning, in: NIPS, 2008..
[46] T. Levinboim, A. Vaswani, D. Chiang, Model invertibility regularization: Sequence alignment with or without parallel data, in: NAACL, 2015..
[47] J. Li, X. Wang, B. Yang, S. Shi, M. R. Lyu, Z. Tu, Neuron interaction based representation composition for neural machine translation, arXiv preprint arXiv:1911.09877..
[48] E. Xi, S. Bing, Y. Jin, Capsule network performance on complex data, arXiv preprint arXiv:1712.03480..
[49] R. LaLonde, U. Bagci, Capsules for object segmentation, arXiv preprint arXiv:1804.04241..
[50] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, Z. Zhao, Investigating Capsule Networks with Dynamic Routing for Text Classification, in: EMNLP, 2018..
[51] J. Gong, X. Qiu, S. Wang, X. Huang, Information Aggregation via Dynamic Routing for Sequence Encoding, in: COLING, 2018..
[52] Z. Dou, Z. Tu, X. Wang, L. Wang, S. Shi, T. Zhang, Dynamic layer aggregation for neural machine translation, in: AAAI, 2019..
[53] B. Yang, Z. Tu, D. F. Wong, F. Meng, L. S. Chao, T. Zhang, Modeling localness for self-attention networks, in: EMNLP, 2018..

Jian Li is a Ph.D. candidate at The Chinese University of Hong Kong. He received his bachelor degree at University of Electronic Science and Technology of China, Chengdu, China, in 2015. His research interests include machine translation, question answering, and information retrieval.

Xing Wang is a researcher with the Tencent AI Lab, Shenzhen, China. He received his Ph.D. degree from Soochow University in 2018. His research interests include statistical machine translation and neural machine translation.

Zhaopeng Tu is a Principal Researcher with the Tencent AI Lab, Shenzhen, China. He received his Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences in 2013. He was a Postdoctoral Researcher at University of California at Davis from 2013 to 2014. He was a researcher at Huawei Noahs Ark Lab, Hong Kong from 2014 to 2017. His research focuses on deep learning for natural language processing.

Michael R. Lyu received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, the M.S. degree in computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, and the Ph.D. degree in computer engineering from the University of California at Los Angeles, Los Angeles, CA, USA. He was with the Jet Propulsion Laboratory, Pasadena, CA, USA, Telcordia Technologies, Piscataway, NJ, USA, and the Bell Laboratory, Murray Hill, NJ, USA, and taught at The University of Iowa, Iowa City, IA, USA. He is currently a Professor with the Computer Science and Engineering Department, The Chinese University of Hong Kong, Hong Kong. He has participated in more than 30 industrial projects and authored more than 500 papers. Dr. Lyu is a fellow of The Institute of Electrical and Electronics Engineers (IEEE) the The Association for Computing Machinery (ACM). His current research interests include software engineering, distributed systems, multimedia technologies, machine learning, social computing.