A Gated Self-attention Memory Network for Answer Selection

Tuan Lai*[1,2], Quan Hung Tran*[2], Trung Bui[2], Daisuke Kihara[1]
[1]Purdue University, [2]Adobe Research

Introduction

- Answer selection is the task of identifying the correct answer to a question from a pool of candidate answers.
- Previous deep learning methods mainly adopt the Compare-Aggregate architecture.
  - Contextualized vector representations of small units are first *compared and aligned*. These comparison results are then *aggregated* to calculate a relevance score.
  - Limitation: The first few layers typically encode the question-candidate pair into sequences of contextualized vector representations separately.

    → These sequences are independent and completely ignore the information from the other sequence.

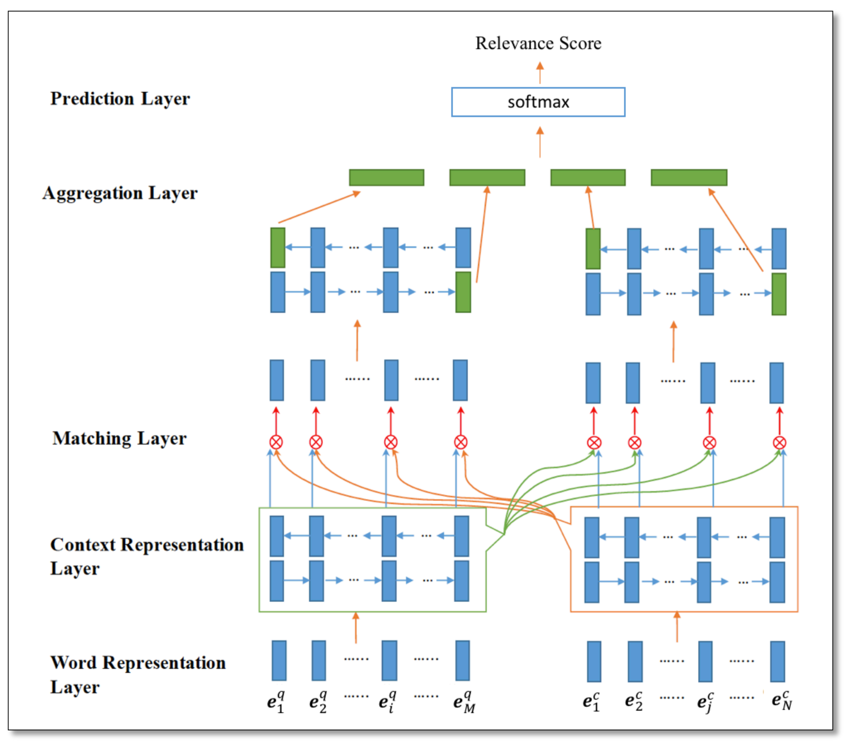
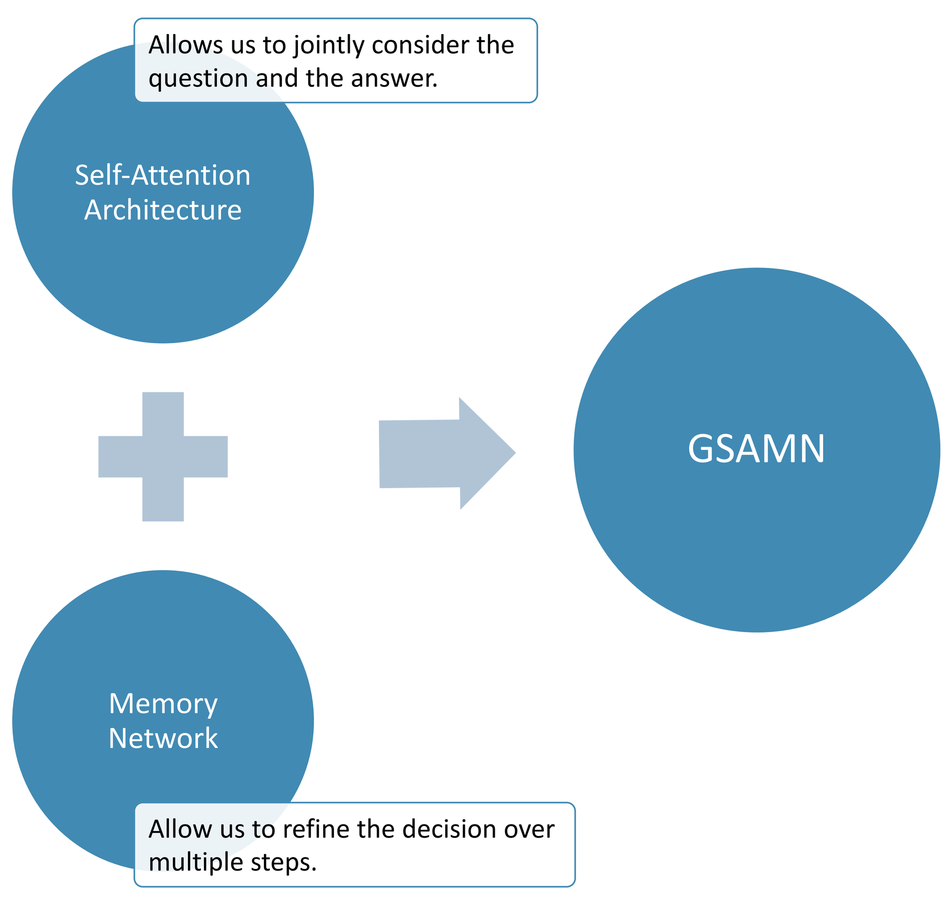


Figure 1: The architecture of the BiMPM model, a Compare-Aggregate model (figure adapted from (Wang et al., 2017)).

Abstract

- We take a departure from the popular Compare-Aggregate architecture.
- We propose a new gated self-attention memory network (GSAMN) for answer selection.



- We also propose a simple but effective transfer learning approach by utilizing the large amount of community question answering (CQA) data available online.
- We achieve new state-of-the-art results on the TrecQA and WikiQA datasets.

Gated Self-Attention Mechanism (GSAM)

- Given a context vector $\mathbf{c}$ and a sequence of input vectors $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$

Traditional Attention Mechanism

- Association score α_i is typically a scalar and calculated as a normalized dot product between $\mathbf{c}$ and $\mathbf{x}_i$.

$$\alpha_i = \frac{\exp(\mathbf{c}^T \mathbf{x}_i)}{\sum_{j \in [1..n]} \exp(\mathbf{c}^T \mathbf{x}_j)}$$

Gated Attention Mechanism

- The association between $\mathbf{c}$ and $\mathbf{x}_i$ is represented by a gate vector $\boldsymbol{g}_i$ (Dhingra et al., 2017).

$$\mathbf{g}_i = \sigma\big(f(\mathbf{c}, \mathbf{x}_i)\big)$$

- f is a parameterized function
  → More flexible in modelling the interaction between $\mathbf{c}$ and $\mathbf{x}_i$
- The gate vector depends only on a context vector and a single input vector.

Gated Self-Attention Mechanism (GSAM)

- We condition the gate vector not only on a context vector and a single input vector but also on the entire sequence of inputs using self-attention.

$$\mathbf{g}_i = f_i(c, X)$$

- Refer to our paper for the full equation.

Gated Self-Attention Memory Network (GSAMN)

- We combine GSAM with the memory network architecture to create GSAMN.
- At the k^th reasoning hop
  - Let $\mathbf{c}_k$ be the controlling context vector.
  - Let $X_k = [\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_n^k]$ be the memory values.
- Each memory cell update from the k^th hop to the next hop is calculated as follow:

$$\boxed{\mathbf{g}_i = f_i(\mathbf{c}_k, X_k)} \quad \text{GSAM}$$

$$\mathbf{x}_i^{k+1} = \mathbf{g}_i \odot \mathbf{x}_i^k$$

- The controller's update is calculated as follow:

$$\boxed{\mathbf{g}_c = f_c(\mathbf{c}_k, X_k)} \quad \text{GSAM}$$

$$\mathbf{c}_{k+1} = \mathbf{g}_c \odot \mathbf{c}_k + \frac{1}{n}\sum_i \mathbf{x}_i^{k+1}$$

- For answer selection, we first concatenate question Q and candidate answer A to a single input sequence. We then use BERT to initialize the memory values $X_0 = [\mathbf{x}_1^0, \mathbf{x}_2^0, \ldots, \mathbf{x}_n^0]$. The control vector $\mathbf{c}_0$ is a randomly initialized learnable vector.
- We use final controller state $\mathbf{c}_T$ as the final representation. The matching probability is:

$$P(A \mid Q) = \sigma\big(\mathbf{W}_c \mathbf{c}_T + \mathbf{b}_c\big)$$

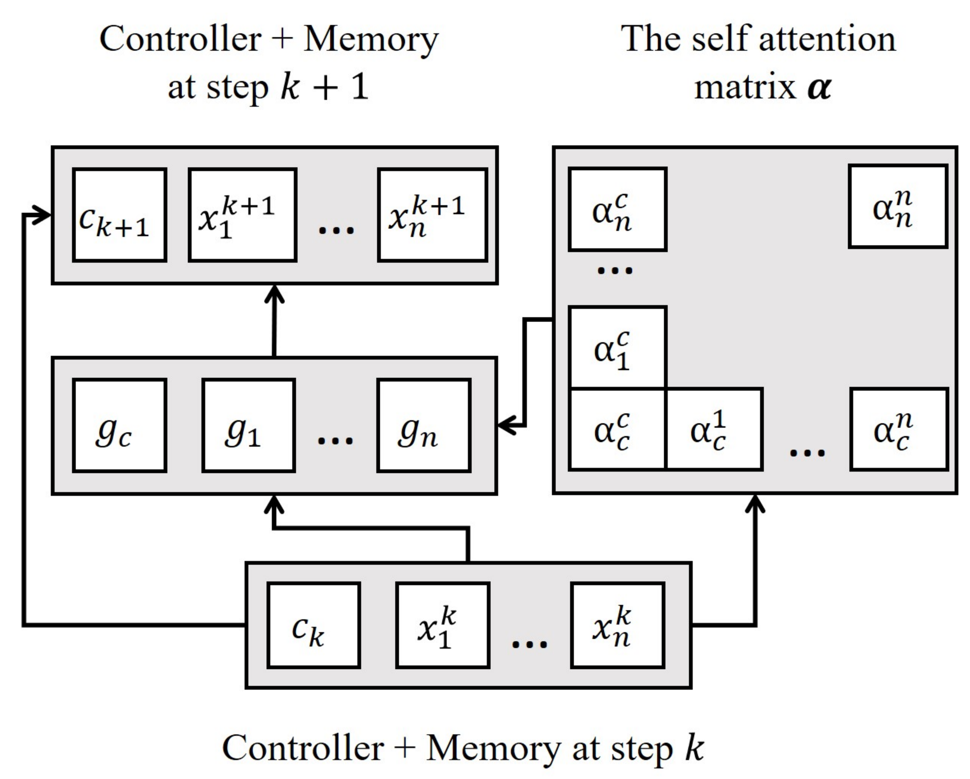
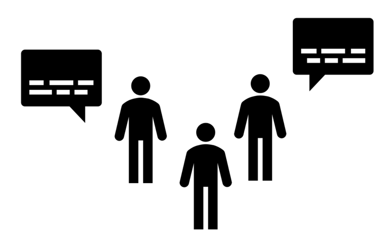


Figure 2: Simplified computation flow of GSAMN

Transfer Learning



Community Question Answering (CQA)

Domain	QA Pair
Academia	**Question**: Is it okay for a PhD student to go on holidays in breaks? **Answer**: Do you have an adviser? Have you talked to them about this? Most should be fine with you taking some time off to visit your family, but you should probably discuss longer breaks with them to work out all the details.
Apple Product	**Question**: What hidden features have you found in iOS 6? **Answer**: Newly downloaded apps have a "new" label on the home screen.
Chemistry	**Question**: Hydrochloric acid vs hydrogen chloride? **Answer**: Hydrochloric acid is an aqueous solution of hydrogen chloride.
Cooking	**Question**: How do I prevent tomatoes from falling in a green salad? **Answer**: I work around this by serving tomatoes on the top of the individual salads after they've been portioned out. I'm not sure of a way to keep them incorporated.

Many Question-Answer Pairs of Various Domains

- In this work, we employ a basic transfer learning technique consisting of two steps
  1. Pre-train our answer selection model on the question-answer pairs collected from CQA data.
  2. Fine-tune the same model on a target dataset of interest such as TrecQA or WikiQA.
- Different from previous works which use source datasets that were manually annotated, our source dataset required minimal effort to obtain and preprocess.

Results and Discussions

- Our full model [BERT + GSAMN + Transfer Learning] outperforms the previous state-of-the-art methods by a large margin.
- Both the variants [BERT + GSAMN] and [BERT + Transfer Learning] have better performance than the original BERT baseline. However, both of the partial variants still perform worse than the one with all the techniques.
- GSAMN outperforms the Transformer based variants, with or without the transfer learning component.
- Our model significantly outperforms the variant [ELMo + Compare-Aggregate].

	TrecQA		WikiQA	
Model	MAP	MRR	MAP	MRR
BERT + GSAMN+ Transfer	**0.914**	**0.957**	**0.857**	**0.872**
BERT + Transformers + Transfer	0.895	0.939	0.831	0.848
BERT + GSAMN	0.906	0.949	0.821	0.832
BERT + Transformers	0.886	0.926	0.813	0.828
ELMo + Compare-Aggregate	0.850	0.898	0.746	0.762
BERT + Transfer	0.902	0.949	0.832	0.849
BERT	0.877	0.922	0.810	0.827
QC + PR + MP CNN (2018)	0.865	0.904	—	—
IWAN + sCARNN (2018)	0.829	0.875	0.716	0.722
IWAN (2017)	0.822	0.889	0.733	0.750
Compare-Aggregate (2017)	0.821	0.899	0.748	0.758
BiMPM (2017)	0.802	0.875	0.718	0.731
HyperQA (2017a)	0.784	0.865	0.705	0.720
NCE-CNN (2016)	0.801	0.877	—	—
Attentive Pooling CNN (2016)	0.753	0.851	0.689	0.696
W&I (2015)	0.746	0.820	—	—