

COMP4913 Capstone Project (final)

Application of Intra-ensemble Network on Machine Reading Comprehension (MRC)

Student Name: DENG Xindi

Student ID: 18081072d

Supervisor: Prof LI Wenjie

Outline

What:

What is MRC?

What is intra-ensemble?

Why:

Why MRC?

Why span-extraction MRC?

Why intra-ensemble?

How:

How to design and implement intra-ensemble models?

- Methodologies
- Implementation

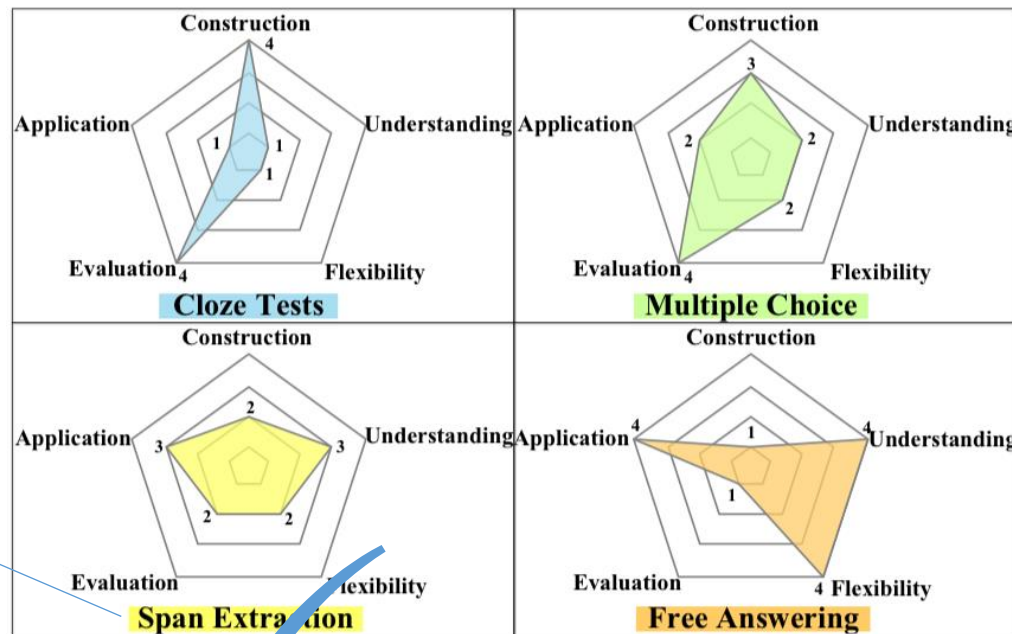
1 What

1.1 What is MRC?

- **Definition:**

asking the machine to answer questions based on the given context (Li et al. 2019)

- **Traditional categories** – based on answer forms (C. Zeng et al. 2020)



Database: Squad

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **grau-pel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?

gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?

grau-pel

Where do water droplets collide with ice crystals to form precipitation?

within a cloud

1.1 What is MRC?

- **Model development**

Time node 1: the use of neural network in MRC

※ General modules: **embedding -> reasoning -> answer prediction**

(Li et al. 2019), (Razieh et al. 2020)

1. Embedding

- **Feature extraction techniques:** RNN, CNN, Transformer
- **Different level pre-training:**
 - no pre-training ->
 - semantic and syntactic information ->
 - contextualized information

2. Reasoning

- Different **attention mechanism** to determine which parts of the context are relating to the query

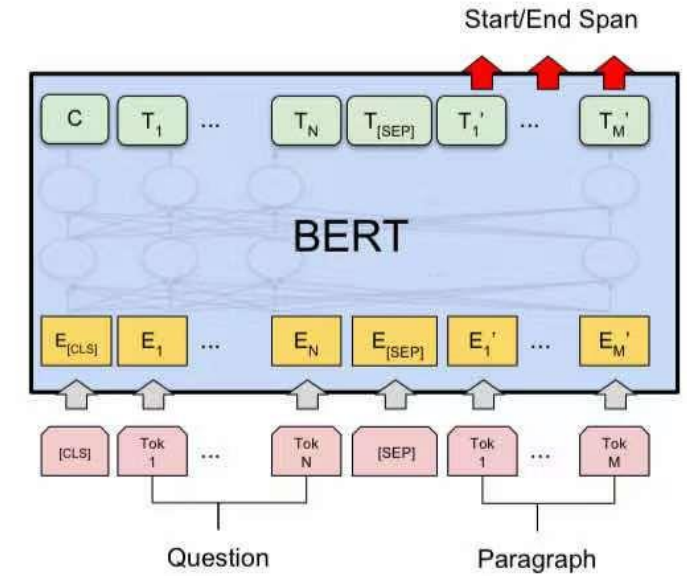
3. Answer prediction

- Add a layer to predict the **probability to be the start and end** for each word in the context

1.1 What is MRC?

- **Model development**

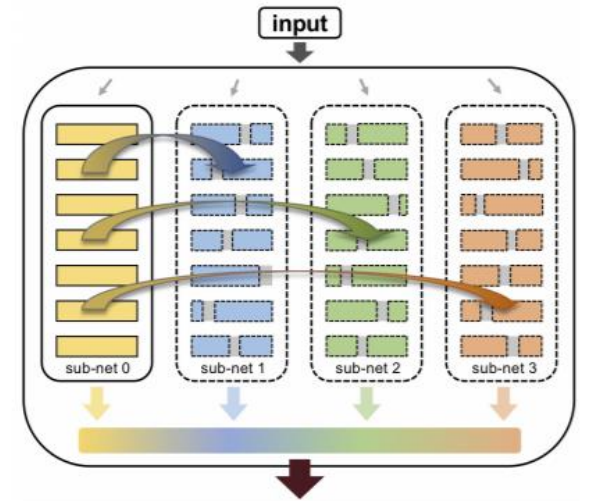
Time node 2: the emergence of Bert (Jacob Devlin et al. 2019)



1. Two-stage learning architecture: pre-train + fine-tune
2. Development of base models: Bert, Bert-based models (Albert, Roberta...), etc.
3. Read + Verify system

1.2 What is intra-ensemble?

- **Structure** - end-to-end ensemble (Yuan Gao et al. 2020)
Train several sub-networks simultaneously within **one neural network**
- **Main Idea:**
 1. resource sharing
 2. sub-networks generation



Main objective:

use **intra-ensemble** method to ensemble the **Bert-related models**,
to further improve the performance in **MRC task** with limited resources

2 Why

2.1 Why MRC?

- **Purpose of MRC:** test the degree of which a machine understands the text (Changchang et al. 2020)
V.S.
- **Main goal of NLP:** human-like comprehension

2.2 Why span-extraction MRC & SQuAD?

- **Research contribution**

× dataset

√ model structure



traditional database and task

YEAR	MODEL STRUCTURE	DATASET	OTHER TASKS	EVALUATION MEASURE
2016	50%	50%	21%	7%
2017	54%	14%	23%	6%
2018	71%	31%	14%	5%
2019	68%	20%	24%	11%
2020	57%	20%	29%	31%
All (249)	61%	23%	21%	12%

Statistics of different research contributions
to MRC task in the reviewed paper (Razieh et al. 2020)

2 Why

2.3 Why intra-ensemble?

1. Limited room for base model development
2. Time-consuming and cost-consuming for inter-ensemble
3. May achieve better performance (Yuan Gao et al. 2020)

3 How

3.1 Methodologies

- **Primary difficulty:** subnetwork generation
- **Proposed methodologies:**
 1. Methodology inspired by one-shot model ← generate subnetwork **manually**
 2. Methodology with Multi-Input-Multi-Output configuration ← generate subnetwork **automatically**

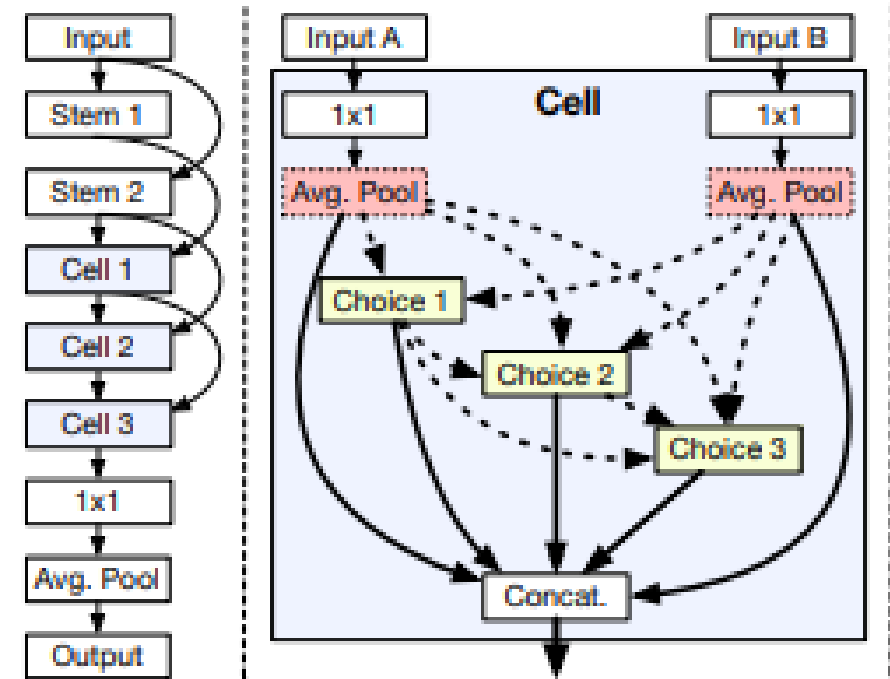
3.1.1 Methodology inspired by one-shot model

- **One-shot model** (Gabriel et al., 2018)

take different operations at different position

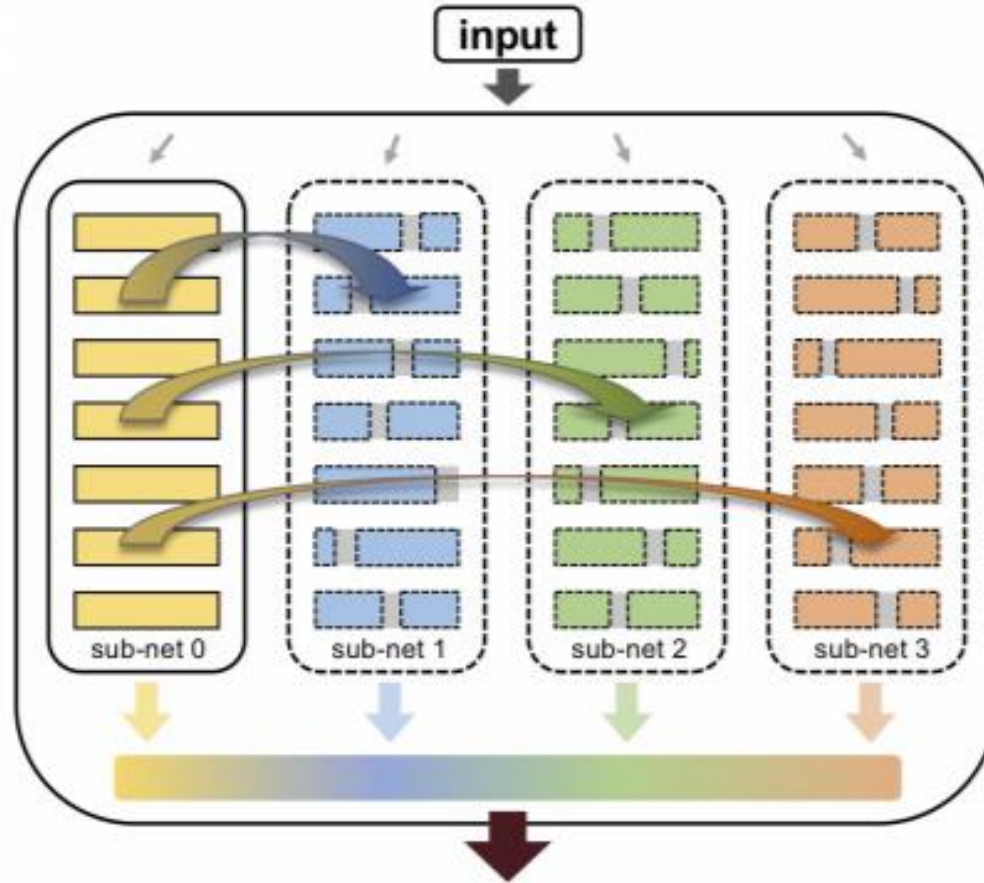


Different sub-networks



3.1.1 Methodology inspired by one-shot model

- **Design of intra-ensemble model**

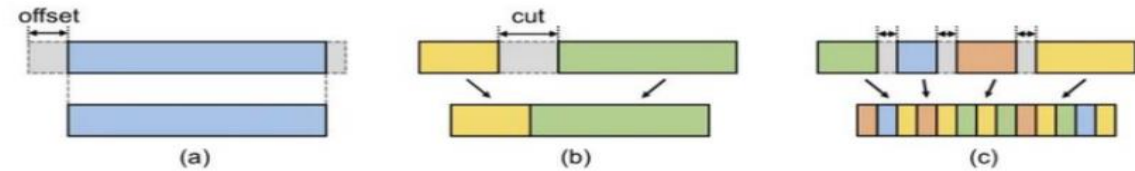


3.1.1 Methodology inspired by one-shot model

- **Design of intra-ensemble model**

Step 1: sub-network generation – two-dimension channel recombination (Yuan Gao et al. 2020)

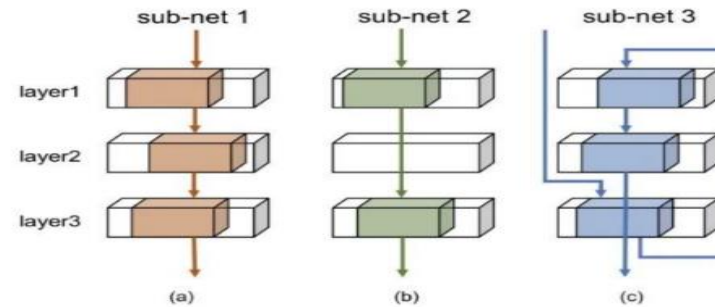
1. Width recombination:



Methods for width recombination.

(a) **Random offset.** (b) **Random cut.** (c) **Shuffle channel.**

2. Depth recombination:



Methods for depth recombination.

(a) Original. (b) **Random skip.** (c) **Shuffle layer.**

Step 2: sub-network combination

similar to inter-ensemble -> bagging / boosting / stacking

3.1.2 Methodology with MIMO configuration

- **MIMO (Multiple-Input Multiple-Output)** (Havasi et al., 2020)

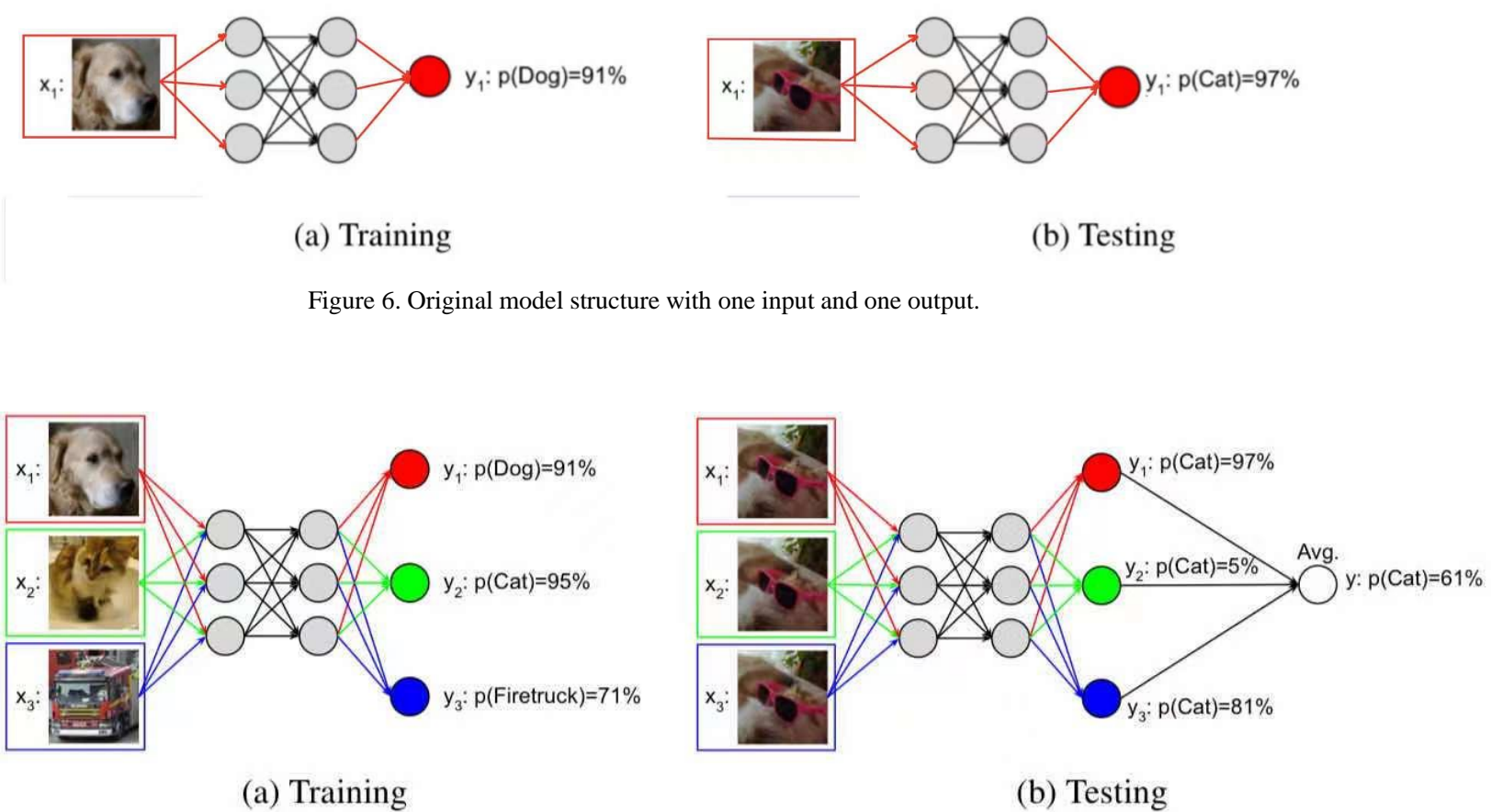


Figure 7. Modified model structure with $M=3$ input and M output. [35]

3.1.2 Methodologies with MIMO configuration

- **MIMO for intra-ensemble**

Theoretical support: lottery ticket hypothesis – 70% to 80% redundant neural network connections
(Jonathan et al., 2018)

Method of sub-network generation:

multiple independent input + multiple independent output



learn to ignore other inputs



make predictions independently



disjoint parts of the network are used for different predictions



diversified subnetwork

3 How

3.2 Implementation

- Implementation of base models
- Design and implementation of inter-ensemble model
- Design and implementation of intra-ensemble model

3.2.1 Implementation of base models

- **Database:** SQuAD (Pranav et al. 2016)
- **Source of the code:** transformer provided by Huggingface
- **Implementation step:** pre-trained model ->
fine-tune and retrain the base model by SQuAD

3.2.1 Implementation of base models

- Running results**

model_type	model_name_or_path	num_train_epochs	training time	HasAns_exact	HasAns_f1	NoAns_exact	NoAns_f1	exact	f1
bert	bert-base-uncased	1	1:32:23	68.74156	74.84500	72.78385	72.78385	70.76560	73.81295
		2	3:10:56	71.79487	78.01184	73.54079	73.54079	72.66908	75.77311
		4	6:10:00	72.43589	79.77064	72.56518	72.56518	72.50063	76.16275
albert	albert-base-v2	1	1:29:47	73.70107	80.09517	83.22960	83.22960	78.47216	81.66463
roberta	roberta-base	1	3:23:31	72.03103	78.75095	77.47687	77.47687	74.75785	78.11300
distilbert	distilbert-base-uncased	1	47:00	61.28542	67.99244	65.07989	65.07989	63.18537	66.53408
xlnet	xlnet-base-cased	1	2:54:56	0.21929	1.24681	67.43481	67.43481	33.87517	34.38820

3.2.2 Implementation of inter-ensemble models

- **Implementation step**

1. Model construction:

- download transformer package from **Huggingface** code ->
- add one class to define the ensemble model for each ensemble method ->
- invoke ensemble models to fine-tune

2. Ensemble strategies:

- Ensemble **sub-dataset**
- Ensemble **different type of models**
- Ensemble **the same model with different parameters**
- Ensemble with **the verifier**

3.2.2 Implementation of inter-ensemble models

- **Ensemble method selection**

1. Bagging (M. A. Ganaie et al. 1992)

- Pros: typical and simple
- Cons: not effective enough

2. Boosting (M. A. Ganaie et al. 1992)

- Pros: reduce the bias
- Cons: sophisticated base models always have **low bias and high variance**;
time-consuming because of sequential running

3. Stacking (David H Wolpert et al. 1992)

- Pros: **reduce the variance**
- Cons: generally used for heterogenous models

3.2.2 Implementation of inter-ensemble models

- Ensemble method selection

4. Retrospective reader (read + verify system) (Zhang et al., 2020)

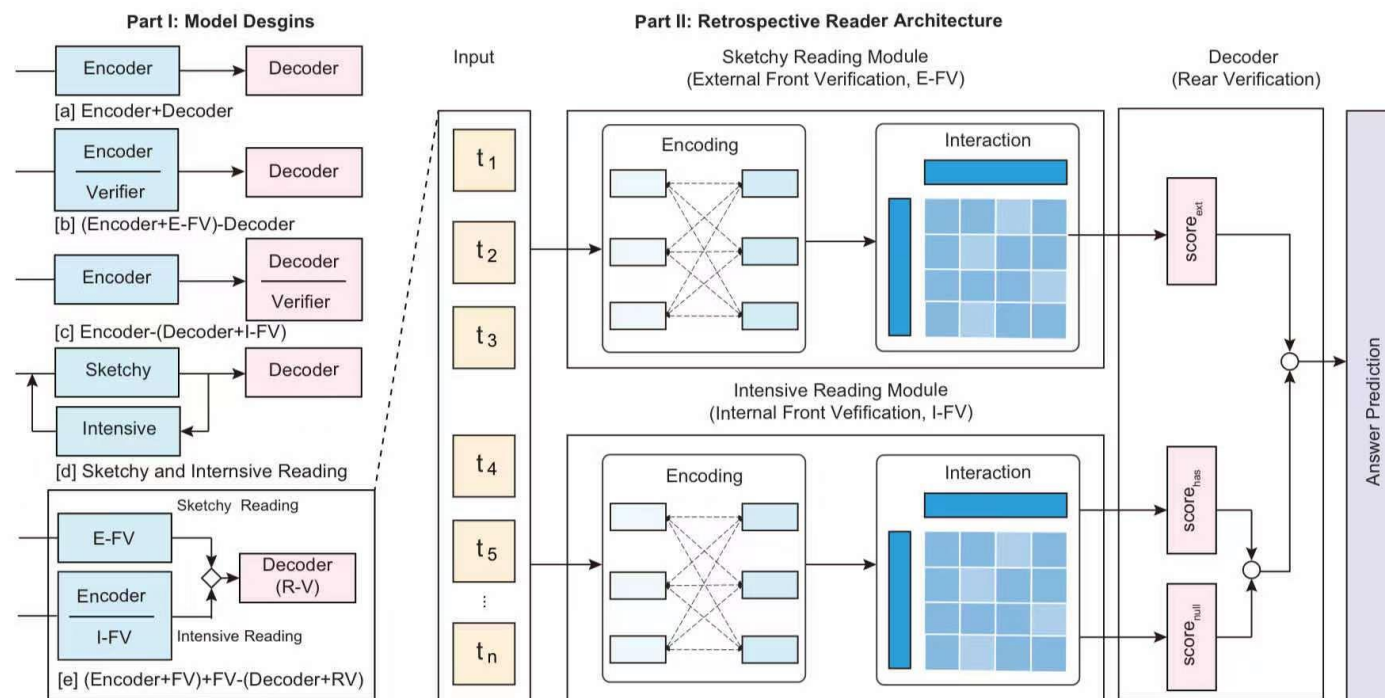


Figure 1: Reader overview. For the left part, models [a-c] summarize the instances in previous work, and model [d] is ours, with the implemented version [e]. In the names of models [a-e], “(·)” represents a module, “+” means the parallel module and “-” is the pipeline. The right part is the detailed architecture of our proposed Retro-Reader.

3.2.2 Implementation of inter-ensemble models

- **Ensemble method selection**

4. Retrospective reader (read + verify system)

model_name	epoch	training time	HasAns_exact	HasAns_f1	NoAns_exact	NoAns_f1	exact	f1
sketchy Reader	2	3:33:08	acc = 0.80645					
intensive Reader	1	1:48:15	71.7949	73.9372	82.2372	82.2372	78.0932	81.3998
rear verification	NA	NA	72.7227	79.2707	84.3230	84.3230	78.5311	81.8005

3.2.3 Implementation of intra-ensemble models

- **Methodologies comparison**

Intra-ensemble inspired by the one-shot model:





- Pros: **high performance** ← **sophisticated subnetworks**.
- Cons: difficult to implement and has **low reusability** ← constructed by modifying the **hidden layers**.

Intra-ensemble with MIMO configuration:

- Pros: **easy to implement** ← constructed by only modifying the **input and output layers**.
- Cons: **limited performance improvement** ← the independence of subnetworks is hard to guarantee in MRC task

3.2.3 Implementation of intra-ensemble models

- **Methodologies comparison**

	Performance	Complexity	Reusability
Methodologies inspired by one-shot model			
Methodologies with MIMO configuration 			

3.2.3 Implementation of intra-ensemble models

- **Intra-ensemble model with MIMO configuration**

Step 1: Choose the base model: BERT

Step 2: Set the value of M: $3 \leftarrow 70\%$ to 80% redundant connections

Step 3: Modify the input layer

Step 4: Modify the output layer

Step 5: Debug and retrain the model

3.1.1 Methodology inspired by one-shot model

- **Design of intra-ensemble model**

Step 1: base model selection

1. **Albert** (Zhenzhong et al. 2020)

Pros: best performance among tested base models

Cons: **cross-layer parameter sharing** (1 layer of transformer block)

+
weight sharing among sub-networks
↓
complicated

2. **Bert** (Jacob Devlin et al. 2019)

Pros: simple and general neural network structure (12 layers of transformer block)

Cons: moderate performance

3.1.1 Methodology inspired by one-shot model

Step 2: Set the value of M

70% to 80% redundant connections



20% to 30% for each subnetwork



$$\text{round} \left(\frac{100\%}{20\% \text{ to } 30\%} \right) = 3 \text{ to } 4$$



$$M = 3$$

3.2.3 Implementation of intra-ensemble models

Step 3: Modify the input layer

1. Feature reconstruction

Original input format:

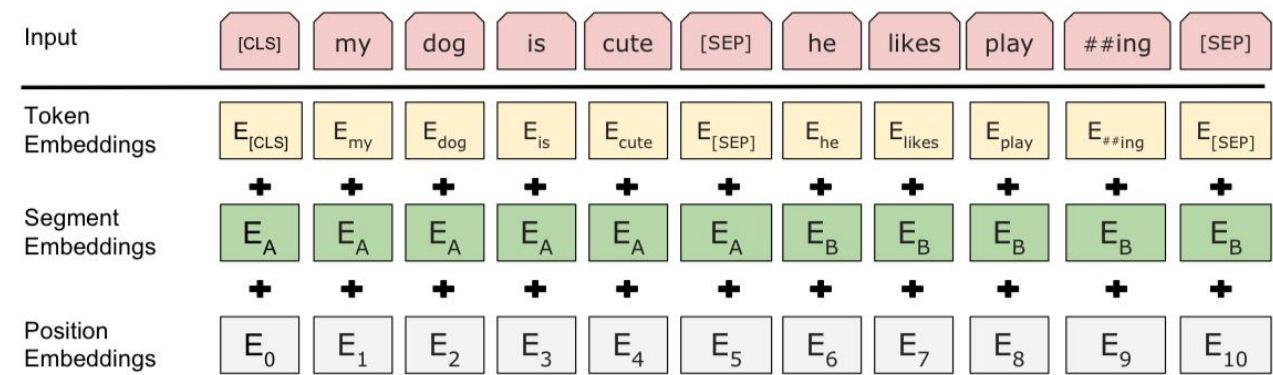


Figure 8. Input format after Bert embedding. [20]

Modified input format:

[CLS] + <paragraph 1> + [SEP] + <question 1> + [SEP]
+ <paragraph 2> + [SEP]+ <question 2> + [SEP]+
<paragraph 3> + [SEP]+ <question 3> + [SEP]

3.2.3 Implementation of intra-ensemble models

Step 4: Modify the output layer

1. Number of nodes for output layers

```
(pooler): BertPooler(  
  (dense): Linear(in_features=768, out_features=768, bias=True)  
  (activation): Tanh()  
)  
(qa_outputs): Linear(in_features=768, out_features=6, bias=True)
```

2. Loss computation

cross-entropy → average of the loss of six positions

3. Output content for testing

three pair of predictions

→ average score for each positions that can be the start-> highest score → start

→ average score for each positions that can be the end-> highest score → end

→ answer span

3.2.3 Implementation of intra-ensemble models

Step 5: Debug and retrain the model

Reference

- Changchang Zeng and Shaobo Li and Qin Li and Jie Hu and Jianjun Hu. 2020. "A Survey on Machine Reading Comprehension: Tasks, Evaluation Metrics and Benchmark Datasets", arXiv: 2006.11880, [cs.CL].
- C. Zeng, S. Li, Q. Li, J. Hu, and J. Hu. Oct. 2020. "A Survey on Machine Reading Comprehension—Tasks, Evaluation Metrics and Benchmark Datasets," *Applied Sciences*, vol. 10, no. 21, p. 7640.
- David H Wolpert. 1992. "Stacked generalization", *Neural networks*, 5(2):241–259.
- Havasi, Marton and Jenatton, Rodolphe and Fort, Stanislav and Liu, Jeremiah Zhe and Snoek, Jasper and Lakshmi Narayanan, Balaji and Dai, Andrew M. and Tran, Dustin. 2020. "Training independent subnetworks for robust prediction", in ICLR.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "Bert: Pre-training of deep bidirectional transformers for language understanding", *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long and Short Papers), pages 4171–4186.
- Jonathan, Frankle and Michael Carbin. 2020. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." arXiv preprint arXiv:1803.03635, 2018.
- Yuan Gao and Zixiang Cai and Lei Yu. 2020. "Intra-Ensemble in Neural Networks", arXiv: 1904.04466, [cs.CL].
- Li, Kaixuan and Xian, Xiujuan and Wang, Jiafu and Yu, Niannian. Mar 2019. "First-principle study on honeycomb fluorated-InTe monolayer with large Rashba spin splitting and direct bandgap", *Applied Surface Science*, 471, Isevier BV, 8–22.
- Razieh Baradaran and Razieh Ghiasi and Hossein Amirkhani. 2020. "A Survey on Machine Reading Comprehension Systems", arXiv: 2001.01582, [cs.CL].
- M. A. Ganaie and Minghui Hu and M. Tanveer and P. N. Suganthan. 1992. "Ensemble deep learning: A review", arXiv: 2104.02395, [cs.CL], 2021.
- Pranav Rajpurkar and Jian Zhang and Konstantin Lopyrev and Percy Liang. 2016. "SQuAD: 100,000+ Questions for Machine Comprehension of Text", arXiv: 1606.05250, [cs.CL].
- Zhang, Zhuosheng and Yang, Junjie and Zhao, Hai. 2020. "Retrospective Reader for Machine Reading Comprehension", arXiv: 2001.09694, [cs.CL].
- Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. 2018. "Understanding and simplifying one-shot architecture search," *in ICML*, pp. 549–558.
- Zhenzhong Lan and Mingda Chen and Sebastian Goodman and Kevin Gimpel and Piyush Sharma and Radu Soricut. 2020. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations", arXiv: 1909.11942, [cs.CL].

THANKS!

Q&A