HAS-QA: **H**ierarchical **A**nswer **S**pans Model for Open-domain Question Answering

Liang Pang [†], Yanyan Lan [†*], Jiafeng Guo [†], Jun Xu [†], Lixin Su [†], Xueqi Cheng [†]

† CAS Key Lab of Network Data Science and Technology

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

† University of Chinese Academy of Sciences, Beijing, China

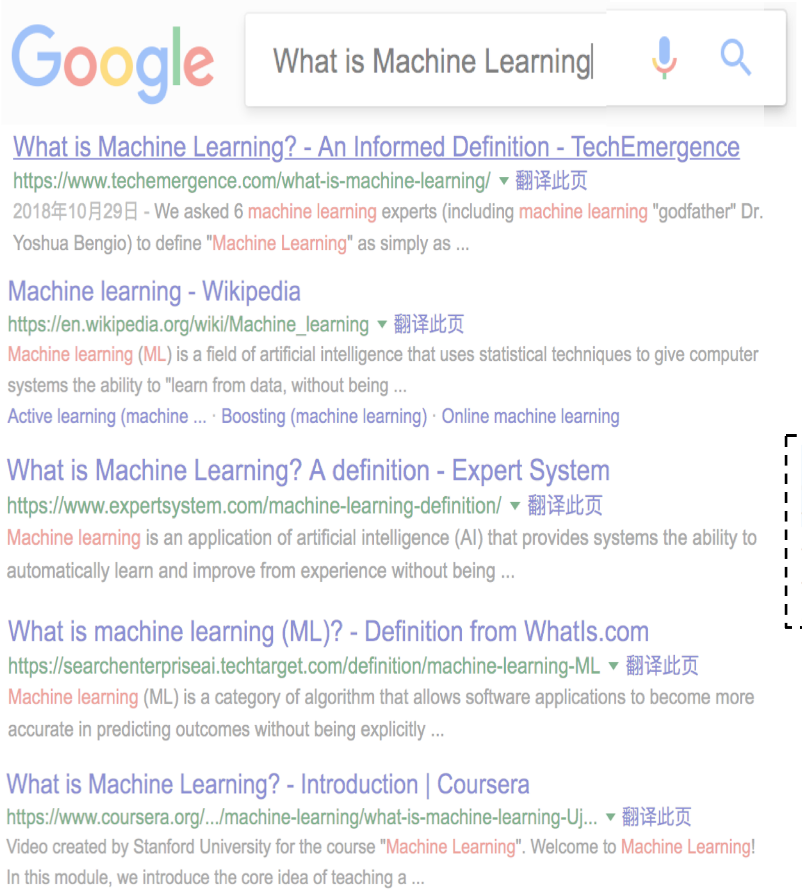
∗ Department of Statistics, University of California, Berkeley

中国科学院计算技术研究所

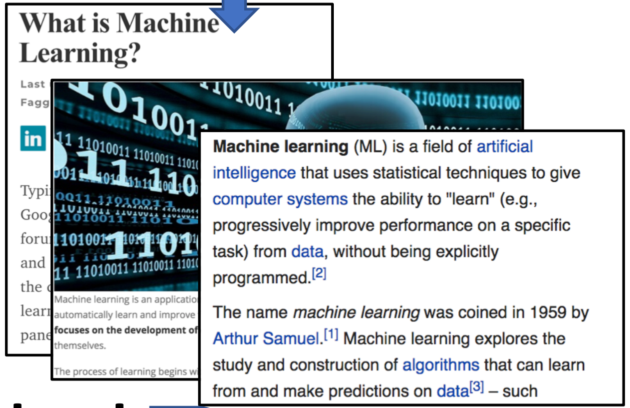
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

I Background

Search Engine



Search



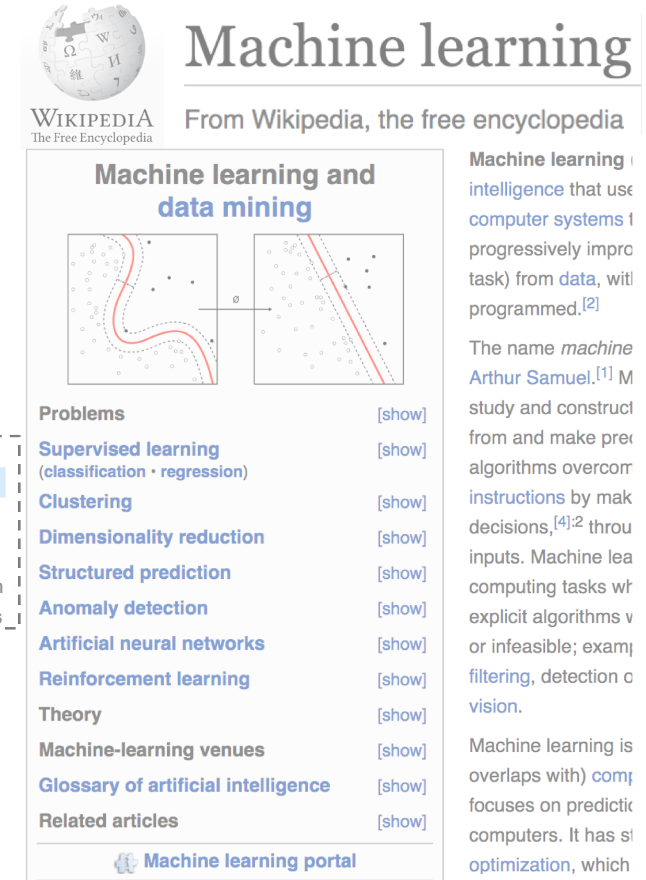
Extract

Machine learning (ML) is a field of artificial intelligence that uses statistical techniques to give computer systems the ability to "learn" (e.g., progressively improve performance on a specific task) from data, without being explicitly programmed.[2]

The name *machine learning* was coined in 1959 by Arthur Samuel.[1] Machine learning explores the study and construction of algorithms that can learn from and make predictions on data[3] – such algorithms overcome following strictly static program instructions by making data-driven predictions

Machine learning (ML) is a field of artificial intelligence that uses statistical techniques to give computer systems the ability to "learn" from data, without being explicitly programmed.

Knowledge Base



Low Information Entropy & Question Diversity

Outline

- Open-domain Question Answering Challenges
- Formulation
- HAS-QA
  - Question Aware Context Encoder
  - Conditional Span Predictor
  - Multiple Spans Aggregator
  - Paragraph Quality Estimator
- Experiments
- Conclusion & Going Forward

I OpenQA

- Open-domain Question Answering

- **Input:**
  - Q: real world question.
  - $\mathbb{D}$: huge *unstructured* text dataset, such as **web page corpus** or **Wikipedia.**

- **Output:**
  - A: answer text from text dataset, which can answer the input question

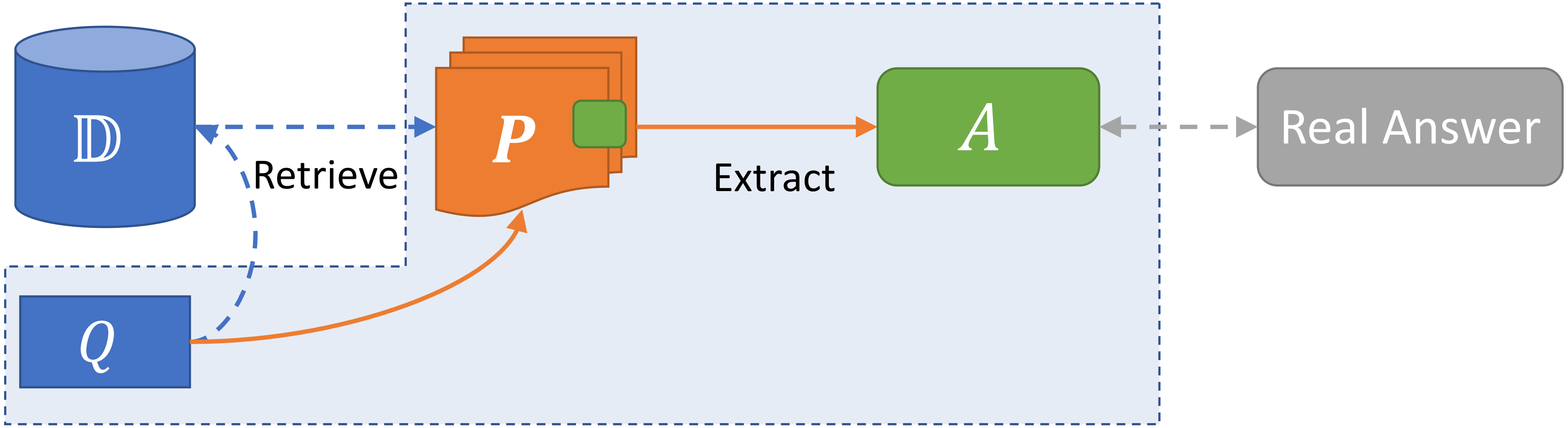
Task – Two Assumptions

- **Assumption 1:**

  Question relevant paragraphs $\boldsymbol{P}$, can be retrieved from unstructured text dataset $\mathbb{D}$, considering the question Q.

- **Assumption 2:**

  Answer A is a text span in the paragraphs $\boldsymbol{P}$.



Challenges

- **Example:**

Search

Question: What does a camel store in its hump?

Paragraph1*(multiple-answer-spans)*: The humps are reservoirs of fatty tissue: concentrating body **fat** in their humps minimizes the insulating effect **fat** would have if distributed over the rest of their bodies, helping camels survive in hot climates.

Paragraph2*(no-answer-span)*: Camels with one hump are called Arabian camels, or Dromedaries, and come from North Africa. Camels with two humps are from Asia, and are called Bactrian camels.

Answer: fat

- **Challenges**:
  - 1) Many paragraphs without the answer span are included in the data collection;
  - 2) Multiple answer spans may exist within one given paragraph;
  - 3) The end position of an answer span is dependent with the start position.

I Outline

- Open-domain Question Answering Challenges
- Formulation
- HAS-QA
  - Question Aware Context Encoder
  - Conditional Span Predictor
  - Multiple Spans Aggregator
  - Paragraph Quality Estimator
- Experiments
- Conclusion & Going Forward

Probabilistic Views – Target

- **Target:**

Answer Text

Question

Answer Probability: $P(A|Q, \mathbf{P})$

Retrieved Paragraphs

- **Interpret:**

  - The probability of the text A is the answer text, given the question Q and retrieved paragraphs $\boldsymbol{P}$.

Probabilistic Views – Challenge 1

- **Challenge 1:**

  Many paragraphs without the answer span are included in the data collection.

  Answer Probability: $P(A|Q,\mathbf{P}) = \sum_{i=1}^{K} \underbrace{P(P_i|Q,\mathbf{P})}_{\text{Paragraph Probability}} \underbrace{P(A|Q,P_i)}_{\text{Conditional Answer Probability}};$

- **Interpret:**

  - **Paragraph Probability**: The probability of the paragraph P_i that contains answer text, given the question Q and retrieved paragraphs $\boldsymbol{P}$.

  - **Conditional Answer Probability**: The probability of the text A is the answer text, given the question Q and paragraph P_i.

Probabilistic Views – Challenge 2

- **Challenge 2:**

  Multiple answer spans may exist within one given paragraph.

Conditional Answer Probability:

$$P(A|Q, P_i) := \mathcal{F}(\{\underbrace{P(L_j(A)|Q, P_i)}_{\text{Span Probability}}\}_j),$$
$$j \in [1, |\mathcal{L}(A, P_i)|];$$

- **Interpret:**

  - **Span Probability**: The probability of the text span $L_j(A)$ is the answer span, given the question Q and paragraph P_i.
  - There exist $|\mathcal{L}(A, P_i)|$ answer spans in paragraph P_i.

Probabilistic Views – Challenge 3

- **Challenge 3:**

  The end position of an answer span is dependent with the start position.

Span Probability: $$P(L_j(A)|Q,P_i) = P(L_j^s(A)|Q,P_i) \cdot P(L_j^e(A)|Q,P_i,L_j^s(A)).$$

Location Start Probability

Location End Probability

- **Interpret:**

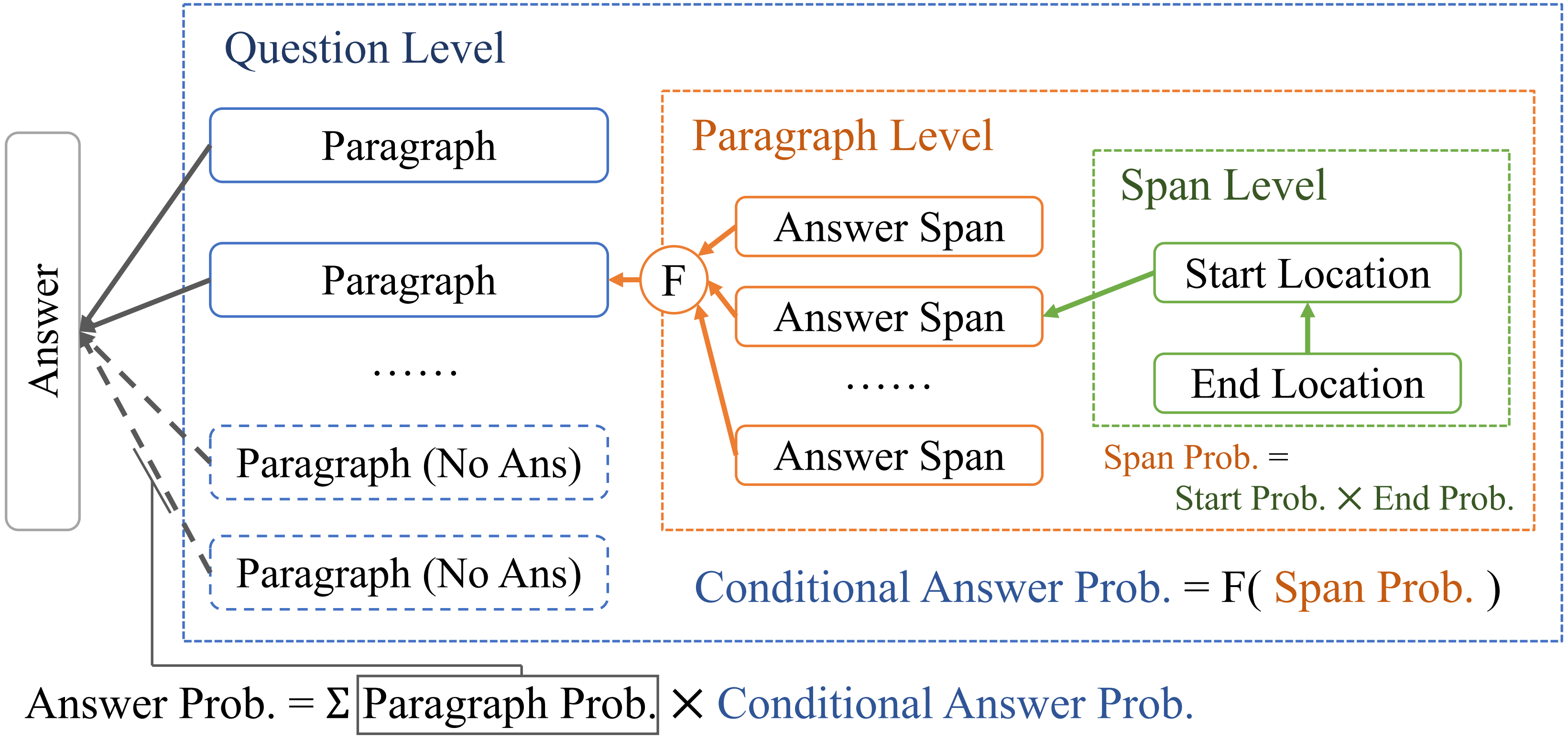
  - **Location Start Probability**: The probability of the text span start location $L_j^s(A)$ is the answer span location, given the question Q and paragraph P_i.
  - **Location End Probability**: The probability of the text span end location $L_j^e(A)$ is the answer span location, given the question Q , paragraph P_i and start location $L_j^s(A)$.

Probabilistic Views – OpenQA Task

- **Target:**

Answer Probability: $P(A|Q, \mathbf{P})$

- **Model:**



Probabilistic Views – RC Task

For a **Reading Comprehension Task**

- **Object:** $P(A|Q, P^{+})$
- **Model:** $P(A|Q, P^{+}) := P(L(A)|Q, P^{+});$

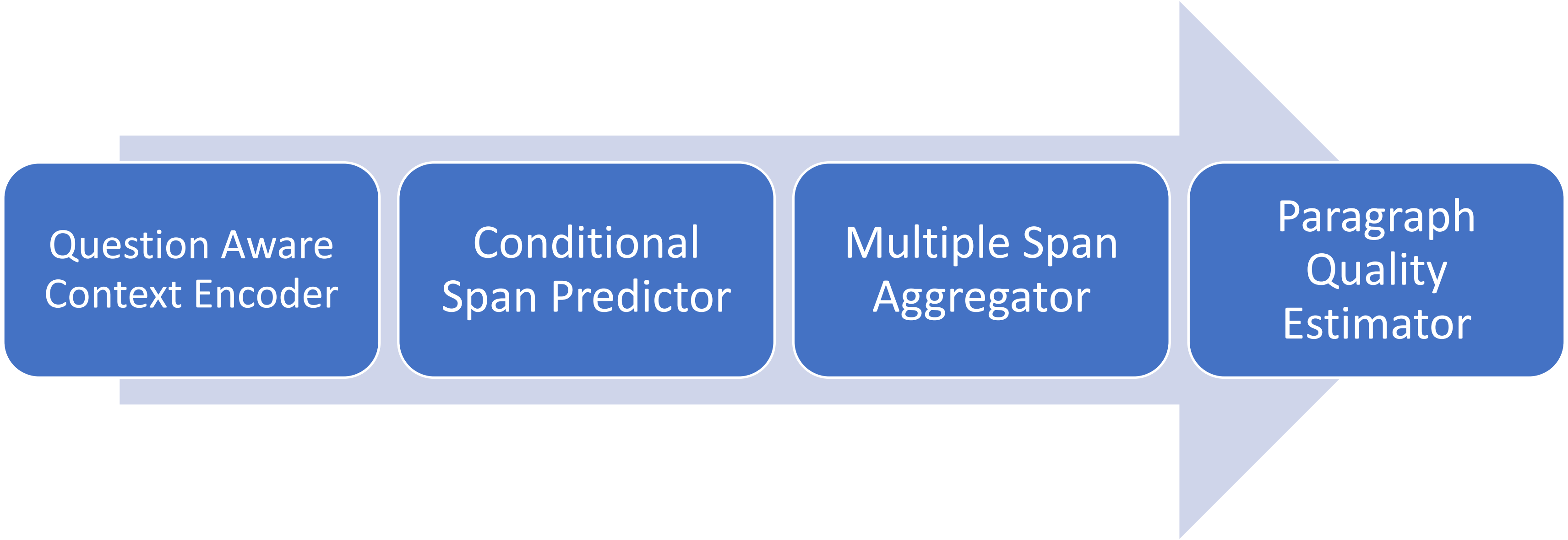
$$P(L(A)|Q, P^{+}) = P(L^{s}(A)|Q, P^{+}) \cdot P(L^{e}(A)|Q, P^{+}).$$

RC ⊆ OpenQA

1. One paragraph
2. One answer span
3. Independence start and end location

From above analysis,

we propose a model named HAS-QA.

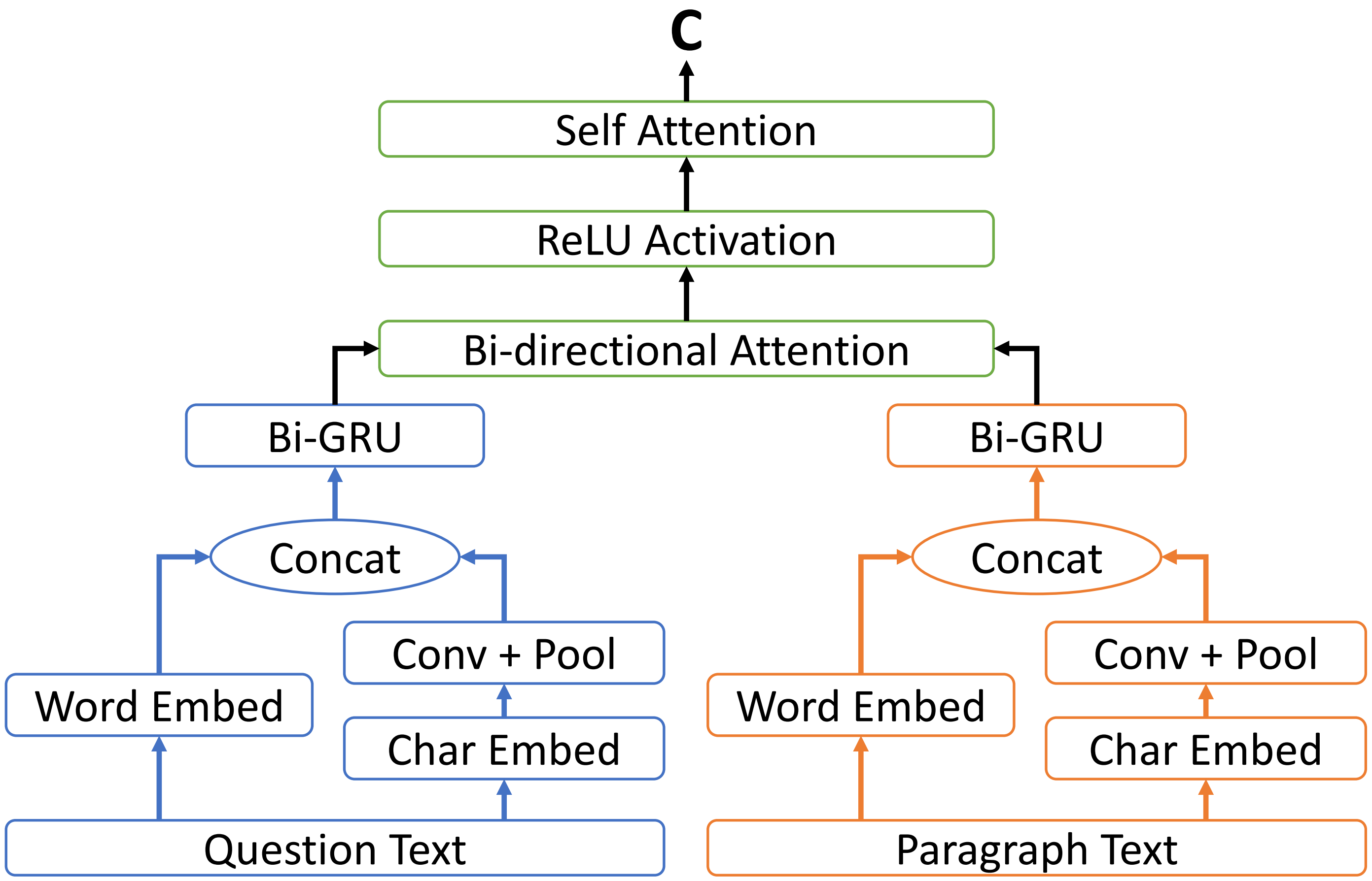


I Outline

- Open-domain Question Answering Challenges
- Formulation
- HAS-QA
  - Question Aware Context Encoder
  - Conditional Span Predictor
  - Multiple Spans Aggregator
  - Paragraph Quality Estimator
- Experiments
- Conclusion & Going Forward

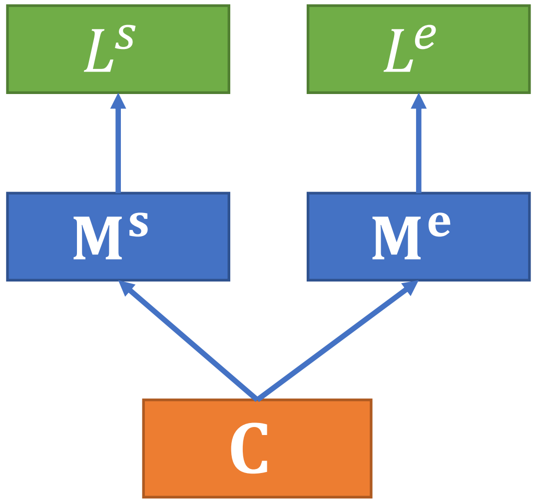
HAS-QA – Question Aware Context Encoder

- **Word Embeddings**: use size 300 pre-trained GloVe (Pennington et al., 2014) word embeddings.
- **Char Embeddings**: encode characters in size 20, which are learnable. Then processed by convolutional layer and max pooling layer to obtain the embedding of each word.
- **Context Embeddings**: concatenate word embeddings and char embeddings, and apply bidirectional GRU (Cho et al., 2014) to obtain the context embeddings. Both question and paragraph get their own context embeddings.
- **Question Aware Context Embeddings:** use bi-directional attention mechanism from the BiDAF (Seo et al., 2016) to build a question aware context embeddings.

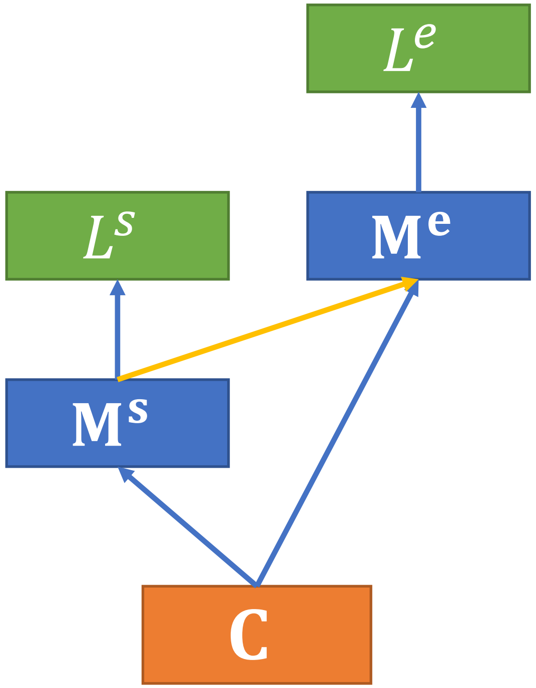


HAS-QA – Conditional Span Predictor

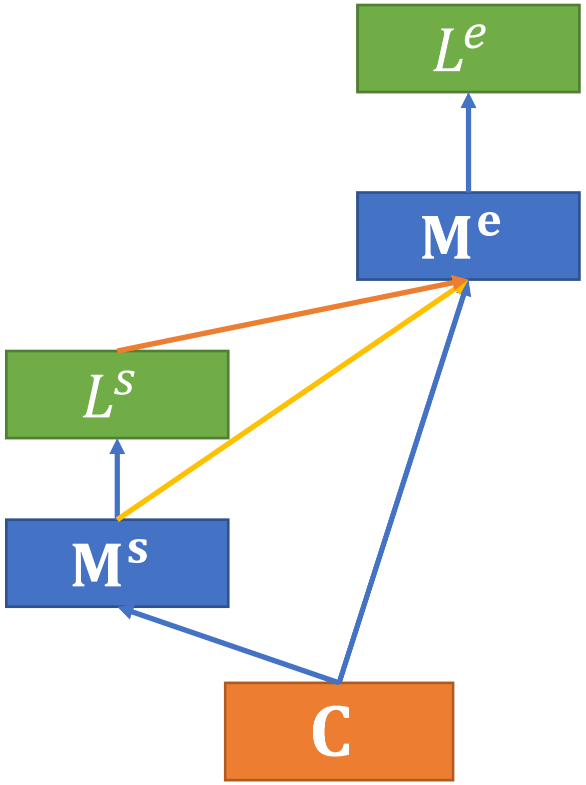
Two independently position classifiers (**IndCls**)



Pointer Networks (**PtrNet**)

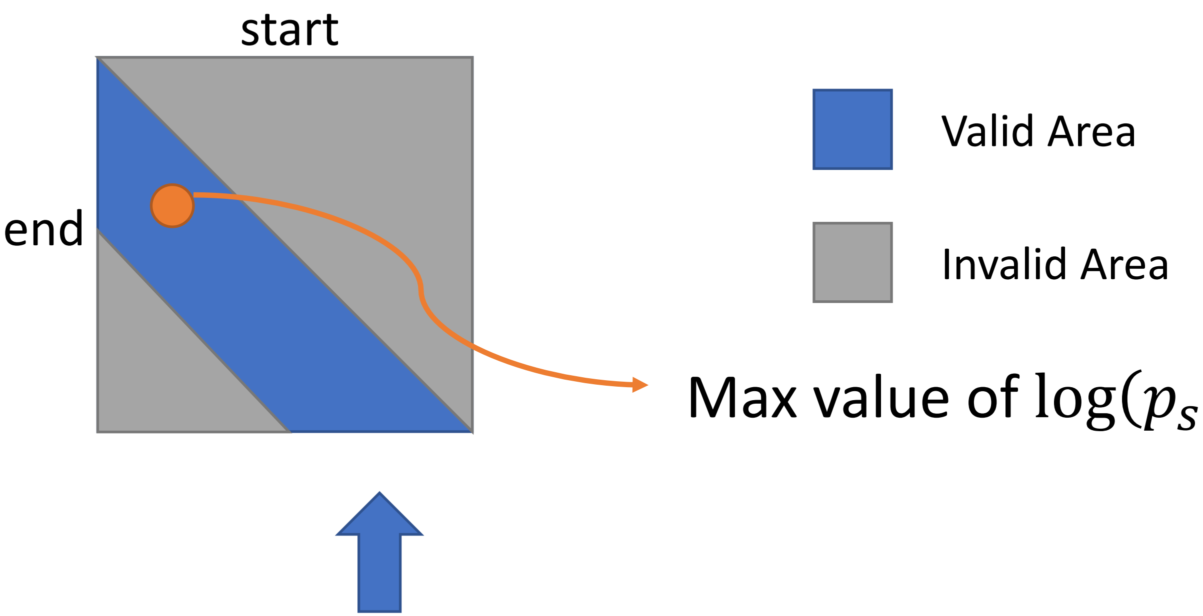


Conditional Pointer Networks (**CondPtrNet**)



HAS-QA – Conditional Span Predictor

Problem of Independence:



Max value of $\log(p_s(x_s)) + \log(p_e(x_e))$

Additional Answer Length Constrain

HAS-QA – Conditional Span Predictor

Span Probability:

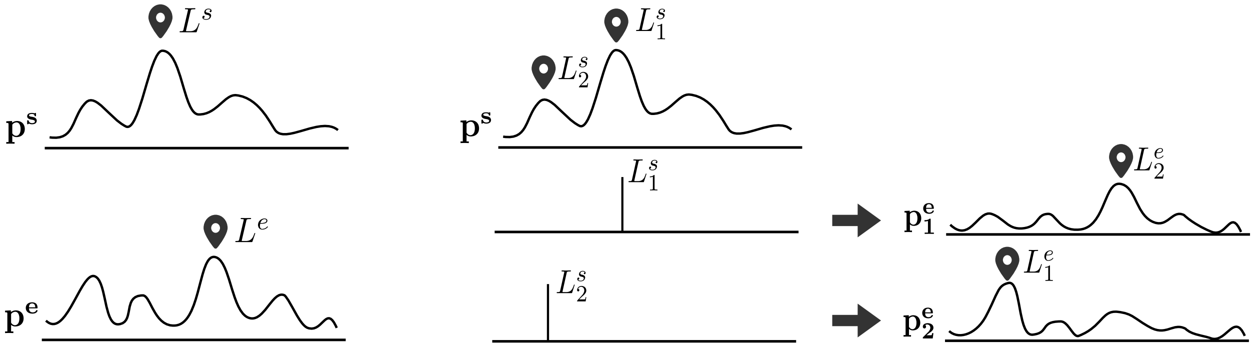
Location Start Hidden Representation: $\mathbf{M^s} = \mathrm{BiGRU}(\mathbf{C})$

Location Start Distribution: $\mathbf{p^s} = \mathrm{softmax}(\mathbf{M^s} w_s),$

Location End Hidden Representation: $\mathbf{M_j^e} = \mathrm{BiGRU}([\mathbf{C}, \mathbf{M^s}, \boxed{\mathrm{OneHot}(L_j^s)}]),$

Location End Distribution: $\mathbf{p_j^e} = \mathrm{softmax}(\mathbf{M_j^e} w_e),$

$$P(L_j(A)|Q, P_i) = s_j = \mathbf{p^s}[L_j^s] \cdot \mathbf{p_j^e}[L_j^e].$$



(A) Independent Positions (B) Conditional Positions

HAS-QA – Multiple Spans Aggregator

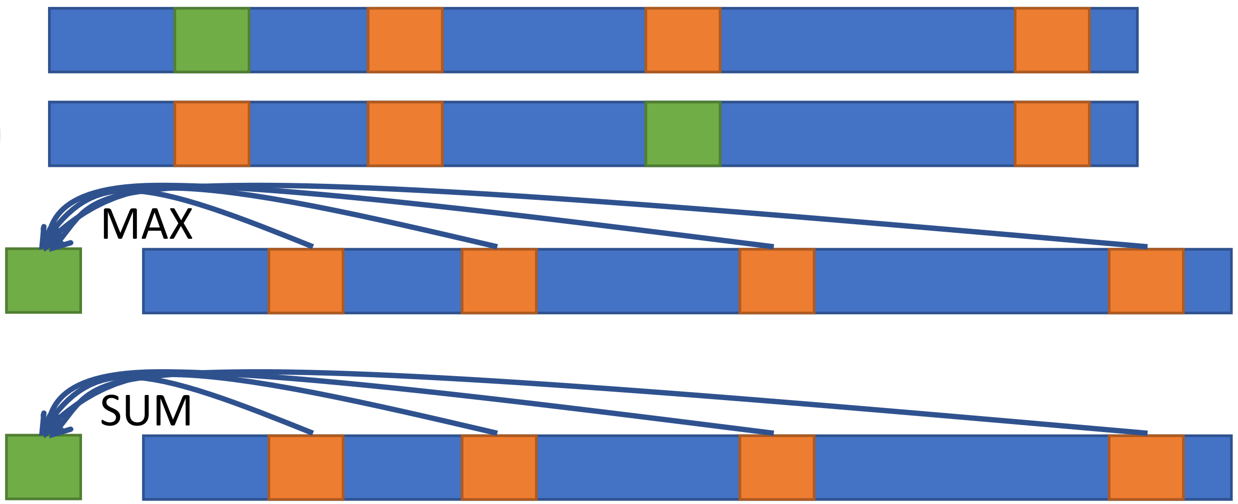
Conditional Answer Probability:

HEAD: $P(A|Q,P_i) = s_1$

RAND: $P(A|Q,P_i) = \mathrm{Random}(s_j)$

MAX: $P(A|Q,P_i) = \max_j(s_j)$

SUM: $P(A|Q,P_i) = \sum_j (s_j)$



HAS-QA – Paragraph Quality Estimator

Paragraph Probability:

Hidden Representation: $\mathbf{M^c} = \mathrm{BiGRU}(\mathbf{C}),$

Location Start Distribution Attention: $\hat{q}_i = (\mathbf{M^c}^\top \cdot \mathbf{p^s}) \cdot w_c.$

Apply Softmax across all the given paragraphs:

$$P(P_i|Q,\mathbf{P}) = q_i = \frac{\exp(\hat{q}_i)}{\sum_{P_j \in \mathbf{P}} \exp(\hat{q}_j)}.$$

HAS-QA – Algorithms

Algorithm 1 HAS-QA Model in Training Phase

Input: Q: question; A: answer string;
$\mathbf{P}$: retrieved paragraphs;
Output: $\mathcal{L}$: loss function

for P^+, P^- in $\mathbf{P}$ **do**: Negative Sampling
  Get answer locations $\mathbf{L^s}, \mathbf{L^e}$ for P^+;
  Get the context embedding $\mathbf{C}$;
  Compute $\mathbf{p^s}$; (Eq 9)
  for L^s_j, L^e_j in $\mathbf{L^s}, \mathbf{L^e}$ **do**:
    $p^s_j \leftarrow \mathbf{p^s}[L^s_j]$;
    Compute $\mathbf{p^e_j}$; (Eq 10)
    $p^e_j \leftarrow \mathbf{p^e_j}[L^e_j]$;
    $s_j \leftarrow p^s_j p^e_j$;
  Apply function: $p^+ \leftarrow \mathcal{F}(\{s_j\})$;
  Compute q^+ in $[P^+, P^-]$; (Eq 13, Eq 14)
  $\mathcal{L}_i \leftarrow -(\log(q^+) + \log(p^+))$;
$\mathcal{L} \leftarrow \mathrm{Avg}(\{\mathcal{L}_i\})$.

Algorithm 2 HAS-QA Model in Inference Phase

Input: Q: question; $\mathbf{P}$: retrieved paragraphs;
Output: A_{best}: answer string

for P_i in $\mathbf{P}$ **do**:
  Get the context embedding $\mathbf{C}$;
  Compute $\mathbf{p^s}$; (Eq 9)
  for L^s_j in Top-K_1 $\mathbf{p^s}$ **do**:
    $p^s_j \leftarrow \mathbf{p^s}[L^s_j]$;
    Compute $\mathbf{p^e_j}$; (Eq 10)
    for L^e_{jk} in Top-K_2 $\mathbf{p^e_j}$ **do**:
      $p^e_{jk} \leftarrow \mathbf{p^e_j}[L^e_{jk}]$;
      $s_{jk} \leftarrow p^s_j p^e_{jk}$; Beam Search
  Group s_{jk} by extracted answer string A_t;
  Apply function: $p_i^{A_t} \leftarrow \mathcal{F}(\{s_{jk}\}_{A_t})$;
  Compute $\hat{q}_i$; (Eq 13)
Normalize $\{\hat{q}_i\}$ get $\{q_i\}$; (Eq 14)
$S(A_t) \leftarrow \sum_i q_i \cdot p_i^{A_t}$;
$A_{best} \leftarrow \arg\max(S(A_t))$.

I Outline

- Open-domain Question Answering Challenges
- Formulation
- HAS-QA
  - Question Aware Context Encoder
  - Conditional Span Predictor
  - Multiple Spans Aggregator
  - Paragraph Quality Estimator
- Experiments
- Conclusion & Going Forward

I Dataset

Three OpenQA Dataset

	QuasarT	TriviaQA	SearchQA
Question Count	43K	95K	140K
Negative Paragraph Ratio	Neg, 1.21 Pos, 98.79	Neg, 37.24 Pos, 62.76	Neg, 25.06 Pos, 74.94
Average Answer Span Count	SPAN SPAN SPAN SPAN SPAN	SPAN SPAN SPAN SPAN	SPAN SPAN SPAN SPAN SPAN SPAN

Large Dataset

Noisy Paragraphs

Multiple Answer Spans

Experiments

Model	QuasarT		TriviaQA		SearchQA	
	EM	F1	EM	F1	EM	F1
GA (Dhingra et al., 2017a)	0.264	0.264	-	-	-	-
BiDAF (Seo et al., 2016)	0.259	0.285	0.411	0.474	0.286	0.346
AQA (Buck et al., 2017)	-	-	-	-	0.387	0.456
DrQA (Chen et al., 2017)	0.377	0.445	0.323	0.383	0.419	0.487
R^3 (Wang et al., 2017a)	0.353	0.417	0.473	0.537	0.490	0.553
Shared-Norm (Clark and Gardner, 2017)	0.386	0.454	0.613	0.672	0.598	0.671
HAS-QA (MAX Ans. Span)	**0.432**	**0.489**	**0.636**	**0.689**	**0.627**	**0.687**

1) HAS-QA **outperforms** traditional RC baselines with a large gap, such as GA, BiDAF, AQA listed in the first part.

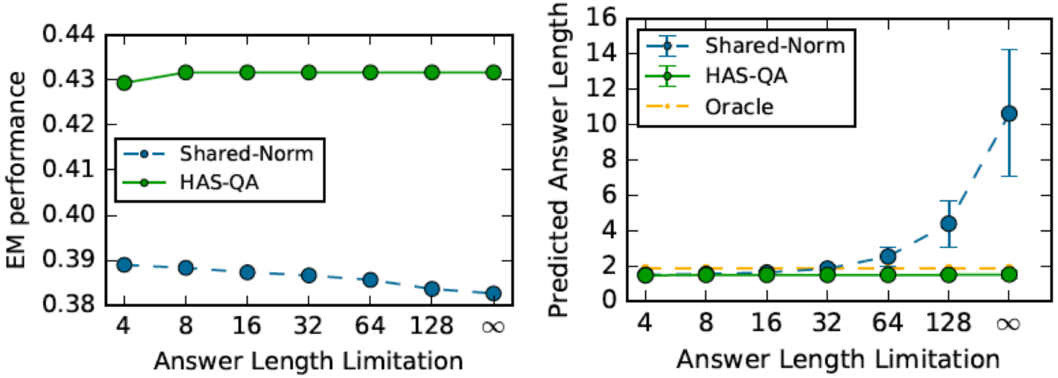
2) HAS-QA **outperforms** recent OpenQA baselines, such as DrQA, R3 and Shared-Norm listed in the second part.

Experimental Analysis

○ Effects of Conditional Pointer Networks

1. The performance of Shared-Norm **decreases** when removing the answer length limitation, while the performance of HAS-QA first **increases** then becomes stable.

2. The average predicted answer length **increases** in Shared-Norm when removing the answer length limitation. However, HAS-QA stably **keeps** the about 1.8 average words, where the oracle average answer length is about 1.9 words.



Example:
About Celebrating the contributions of Louis Braille January 5th , 2009 On the 200th anniversary of Louis Braille ' s birth , people around the world are saluting a man whose tactile alphabet has provided a lifeline to people with impaired vision . Shared-Norm HAS-QA

Experimental Analysis

- Effects of Multiple Spans Aggregation

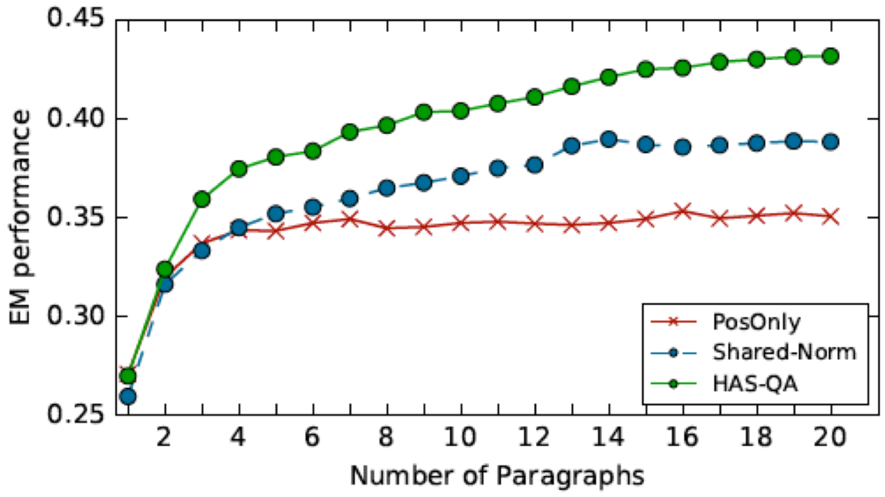
Model	EM	F1
HAS-QA (HEAD Ans. Span)	0.372	0.425
HAS-QA (RAND Ans. Span)	0.341	0.394
HAS-QA (SUM Ans. Span)	0.423	0.484
HAS-QA (MAX Ans. Span)	0.432	0.489

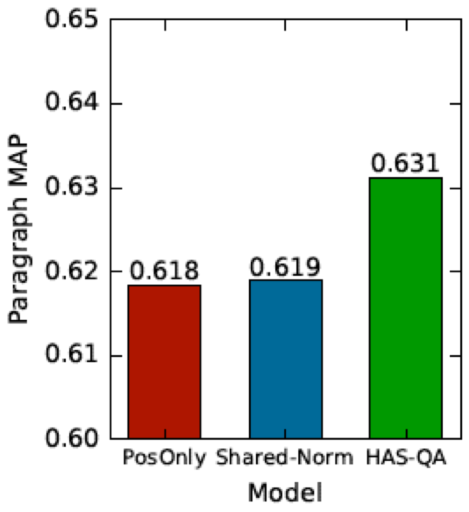
1. **SUM and MAX operations**. They take advantages of using multiple answer spans for training and improve about 6% - 10% in EM comparing to the **HEAD operation**.

2. The failure of **RAND operation**, mainly comes down to the conflicting training samples.

Experimental Analysis

- Effects of Paragraph Quality





1. With the increasing number of given paragraphs which ordered by the rank of search engine, EM performance of HAS-QA **sustainably grows**.

2. The Mean Average Precision (MAP) score between the predicted scores and the label whether a paragraph contains answer spans, shows that HAS-QA can rank **the high quality paragraphs** in the front of the given paragraph list.

I Outline

- Open-domain Question Answering Challenges
- Formulation
- HAS-QA
  - Question Aware Context Encoder
  - Conditional Span Predictor
  - Multiple Spans Aggregator
  - Paragraph Quality Estimator
- Experiments
- Conclusion & Going Forward

I Conclusion

- A new probabilistic formulation of **OpenQA**, based on a three-level hierarchical structure, i.e., the question level, the paragraph level and the answer span level.

- HAS-QA Model
  - 1) a paragraph quality estimator makes it robust for the paragraphs without answer spans
  - 2) a multiple span aggregator points out that it is necessary to combine the contributions of multiple answer spans in a paragraph
  - 3) a conditional span predictor is proposed to model the dependence between the start and end positions of each answer span.

I Going Forward

- Single Answer Spans Extraction
- Multiple Answer Spans Extraction
- Well-formed Answers Generation

Thanks Q & A

Name: Liang Pang | Email: pl8787@gmail.com