HIT-SCIR

Attention-over-Attention Neural Networks for Reading Comprehension

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu and Guoping Hu

Joint Laboratory of HIT and iFLYTEK Research (HFL), Beijing, China

2017-08-01

ACL2017, Vancouver, Canada

OUTLINE

- Introduction: Cloze-style Reading Comprehension
- Related Works
- Attention-over-Attention Reader (AoA Reader)
- N-best Re-ranking Strategy
- Experiments & Analysis
- Conclusions & Future Works

Introduction

- *Machine Reading Comprehension (MRC)* is to read and comprehend a given article and answer the questions based on it, which has become enormously popular in recent few years

- The related datasets and algorithms are mutually benefitted

  - From cloze-style to sentence-style

  - From simple model to complex model

- In this paper, we focus on solving the cloze-style RC problem

INTRODUCTION

- Key components in RC

→**Document**

- Query
- Candidates
- Answer

James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food. Other times he'd sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.
One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.
His aunt was waiting for him in his room. She told James that she loved him, but he would have to start acting like a well-behaved turtle.
After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

1) What is the name of the trouble making turtle?
A) Fries
B) Pudding
C) James
D) Jane

*Example is chosen from the MCTest dataset (Richardson et al., 2013)

Introduction

- Key components in RC
  - Document
  - → **Query**
  - Candidates
  - Answer

James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food. Other times he'd sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.
One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.
His aunt was waiting for him in his room. She told James that she loved him, but he would have to start acting like a well-behaved turtle.
After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

1) What is the name of the trouble making turtle?
A) Fries
B) Pudding
C) James
D) Jane

*Example is chosen from the MCTest dataset (Richardson et al., 2013)

INTRODUCTION

- Key components in RC
  - Document
  - Query
  - → **Candidates**
  - Answer

James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food. Other times he'd sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.
One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.
His aunt was waiting for him in his room. She told James that she loved him, but he would have to start acting like a well-behaved turtle.
After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

1) What is the name of the trouble making turtle?
A) Fries
B) Pudding
C) James
D) Jane

*Example is chosen from the MCTest dataset (Richardson et al., 2013)

Introduction

- Key components in RC
  - Document
  - Query
  - Candidates

→ **Answer**

James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food. Other times he'd sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.

One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.

His aunt was waiting for him in his room. She told James that she loved him, but he would have to start acting like a well-behaved turtle.

After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

1) What is the name of the trouble making turtle?

A) Fries

B) Pudding

C) James

D) Jane

*Example is chosen from the MCTest dataset (Richardson et al., 2013)

INTRODUCTION

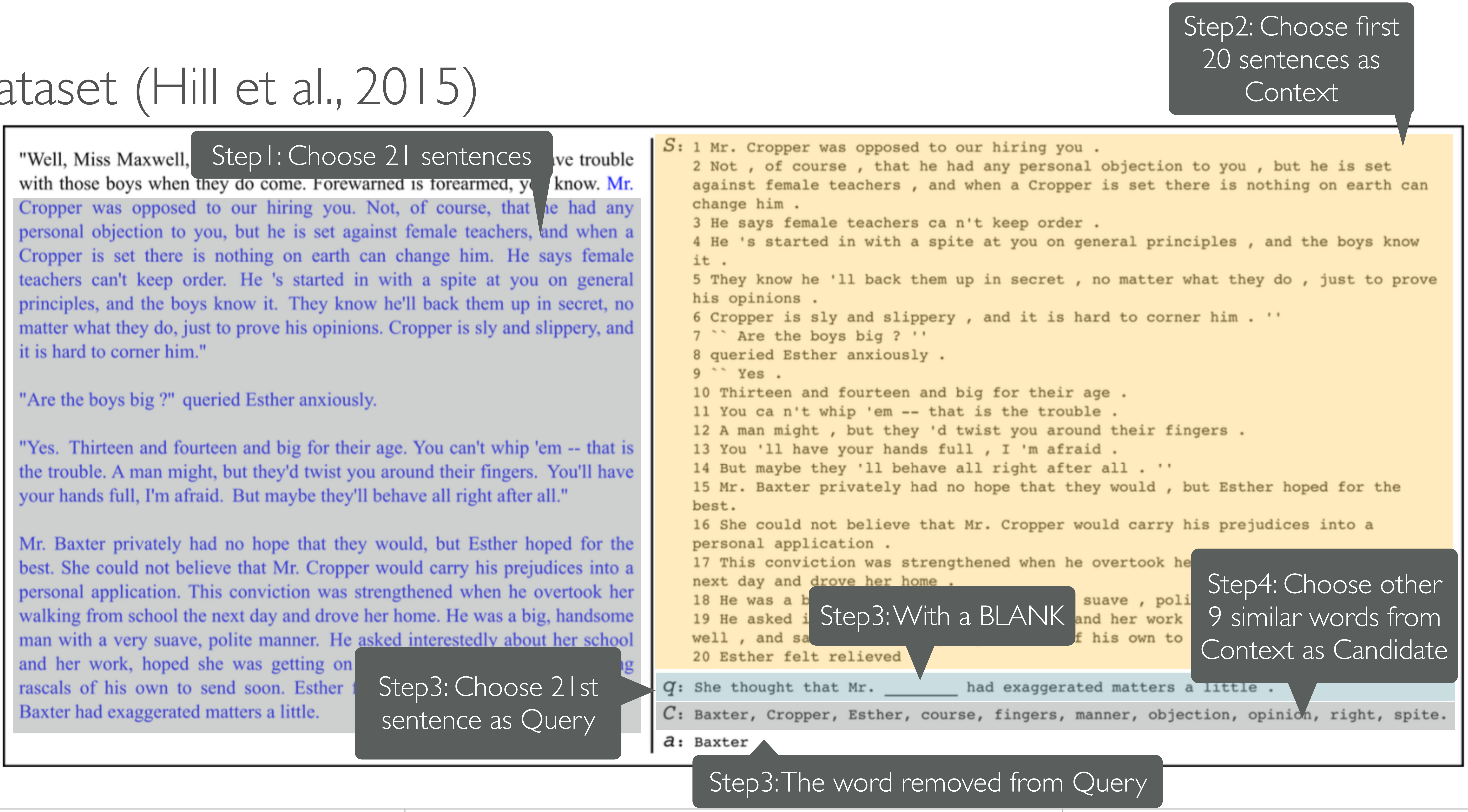
- Specifically, in cloze-style RC
  - Document: the same as the general RC
  - Query: a sentence with a blank
  - Candidate (optional): several candidates to fill in
  - Answer: a single word that exactly match the query (the answer word should appear in the document)

Original Version
Context
The BBC producer allegedly struck by Jeremy Clarkson will not press charges against the “Top Gear” host, his lawyer said Friday. Clarkson, who hosted one of the most-watched television shows in the world, was dropped by the BBC Wednesday after an internal investigation by the British broadcaster found he had subjected producer Oisin Tymon “to an unprovoked physical and verbal attack.” …
Query
Producer **X** will not press charges against Jeremy Clarkson, his lawyer says.
Answer
Oisin Tymon

*Example is chosen from the CNN dataset (Hermann et al., 2015)

INTRODUCTION

- CBT dataset (Hill et al., 2015)

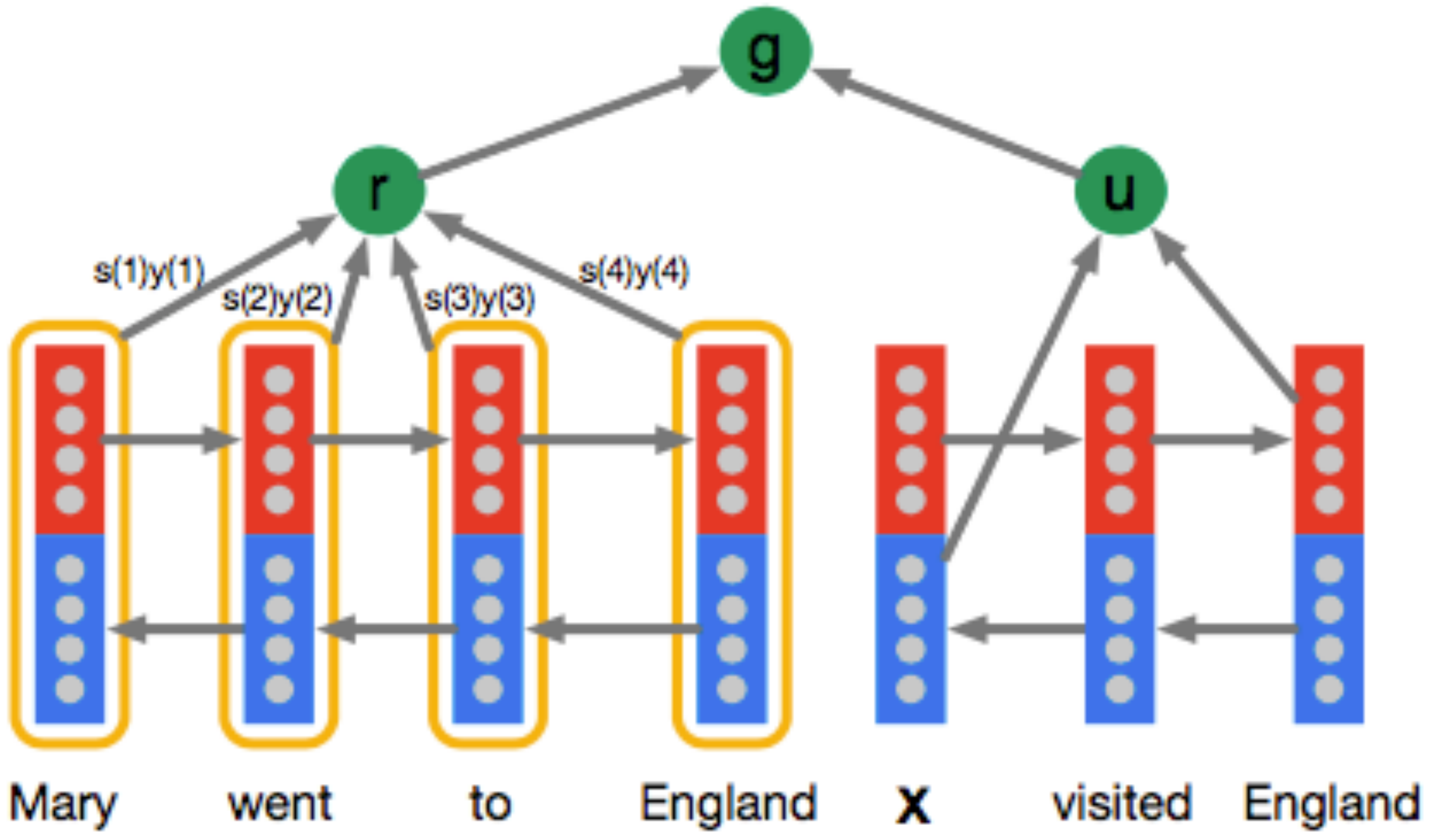


RELATED WORKS

- Predictions on full vocabulary
  - Attentive Reader (Hermann et al., 2015)
  - Stanford AR (Chen et al., 2016)
- Pointer-wise predictions (Vinyals et al., 2015)
  - Attention Sum Reader (Kadlec et al., 2016)
  - Consensus Attention Reader (Cui et al., 2016)
  - Gated-attention Reader (Dhingra et al., 2017)

Attentive Reader

- Teaching Machines to Read and Comprehend (Hermann et al., 2015)

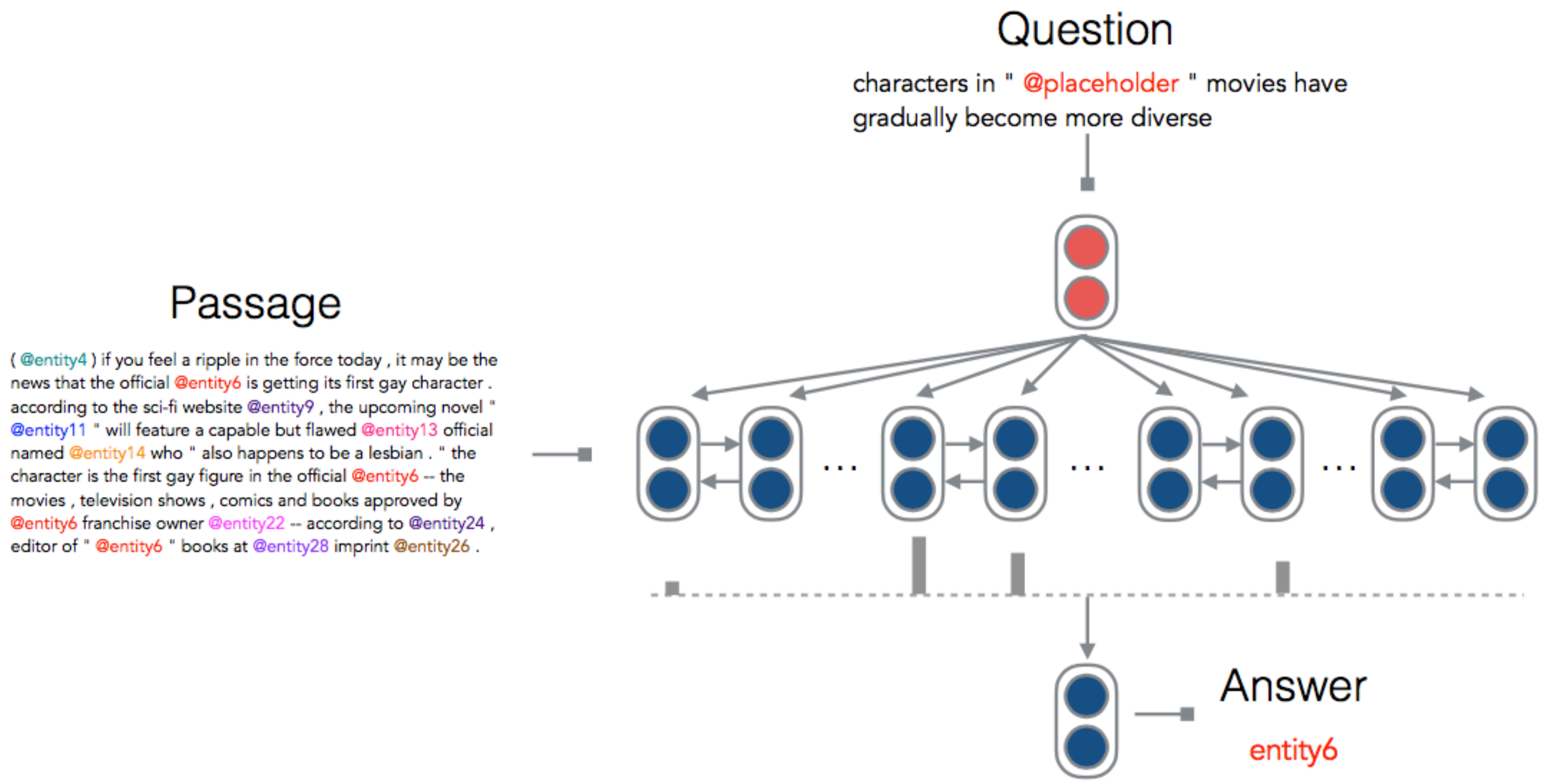


$$m(t) = \tanh\left(W_{ym} y_d(t) + W_{um} u\right),$$
$$s(t) \propto \exp\left(\mathrm{w}_{ms}^{\mathsf{T}} m(t)\right),$$
$$r = y_d s,$$

$$g^{\mathrm{AR}}(d, q) = \tanh\left(W_{rg} r + W_{ug} u\right).$$

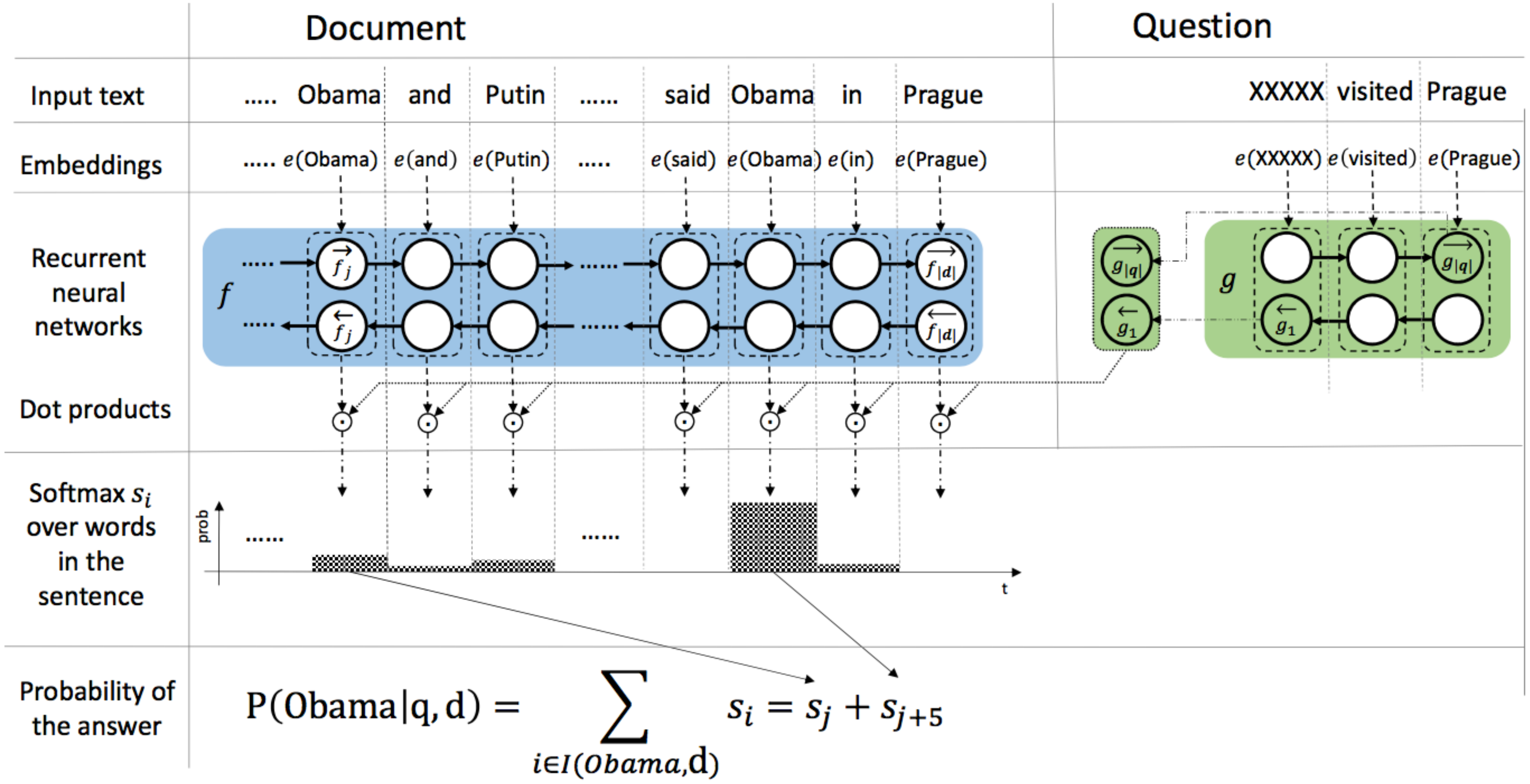
STANFORD AR

- A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task (Chen et al., 2016)



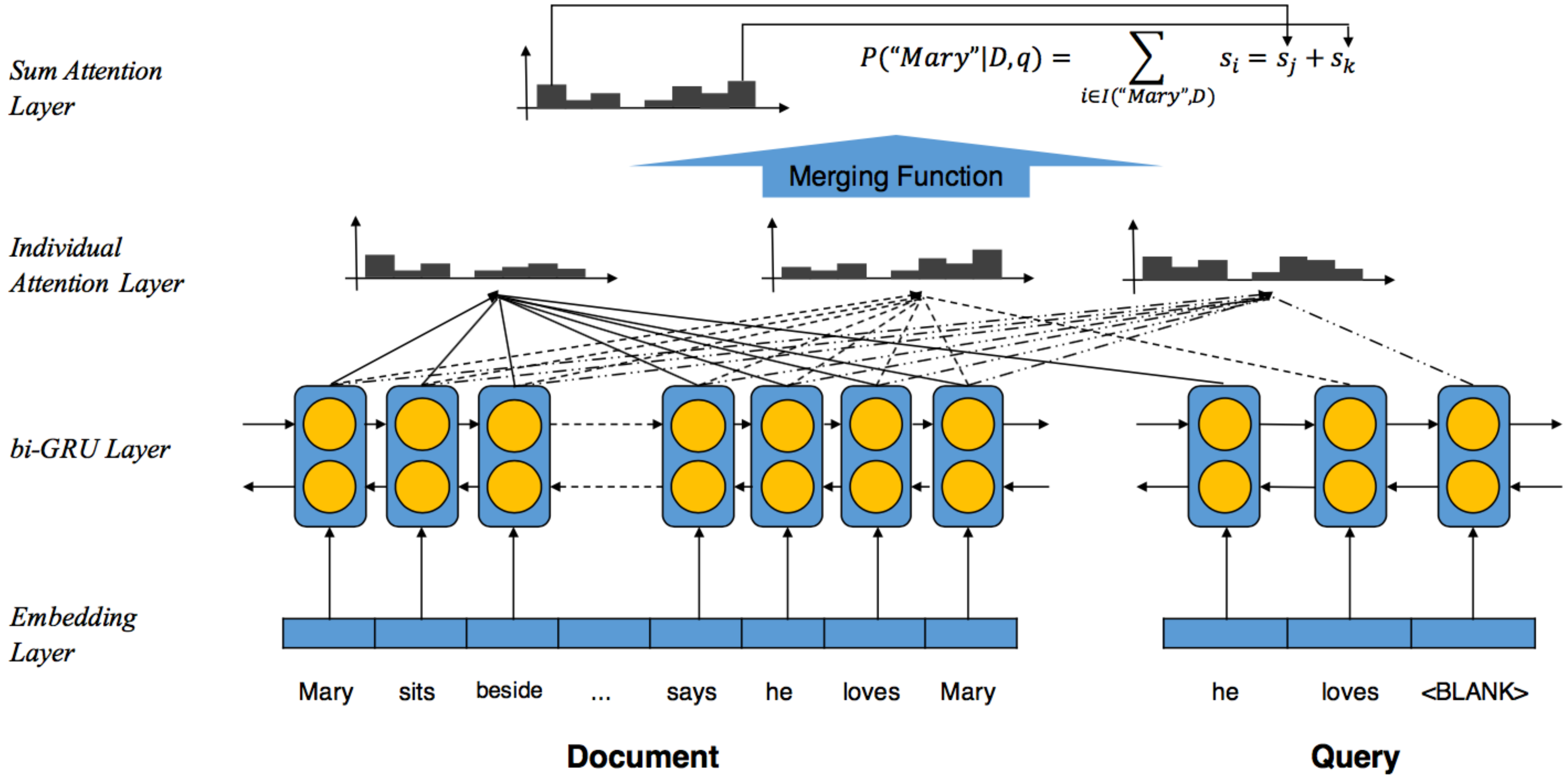
ATTENTION SUM READER

- Text Understanding with the Attention Sum Reader Network (Kadlec et al., 2016)



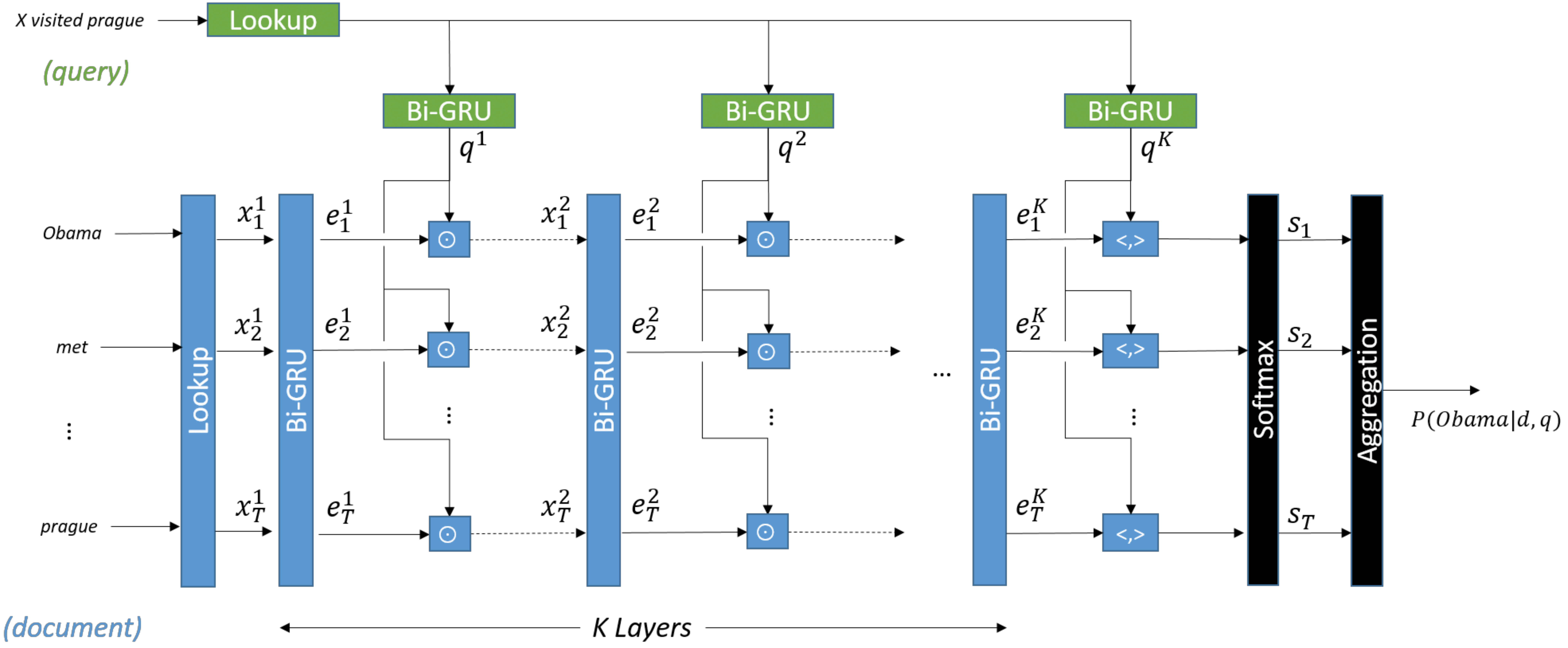
CONSENSUS ATTENTION READER

- Consensus Attention-based Neural Networks for Chinese Reading Comprehension (Cui et al., 2016)



Gated-Attention Reader

- Gated-Attention Reader for Text Comprehension (Dhingra et al., 2016)

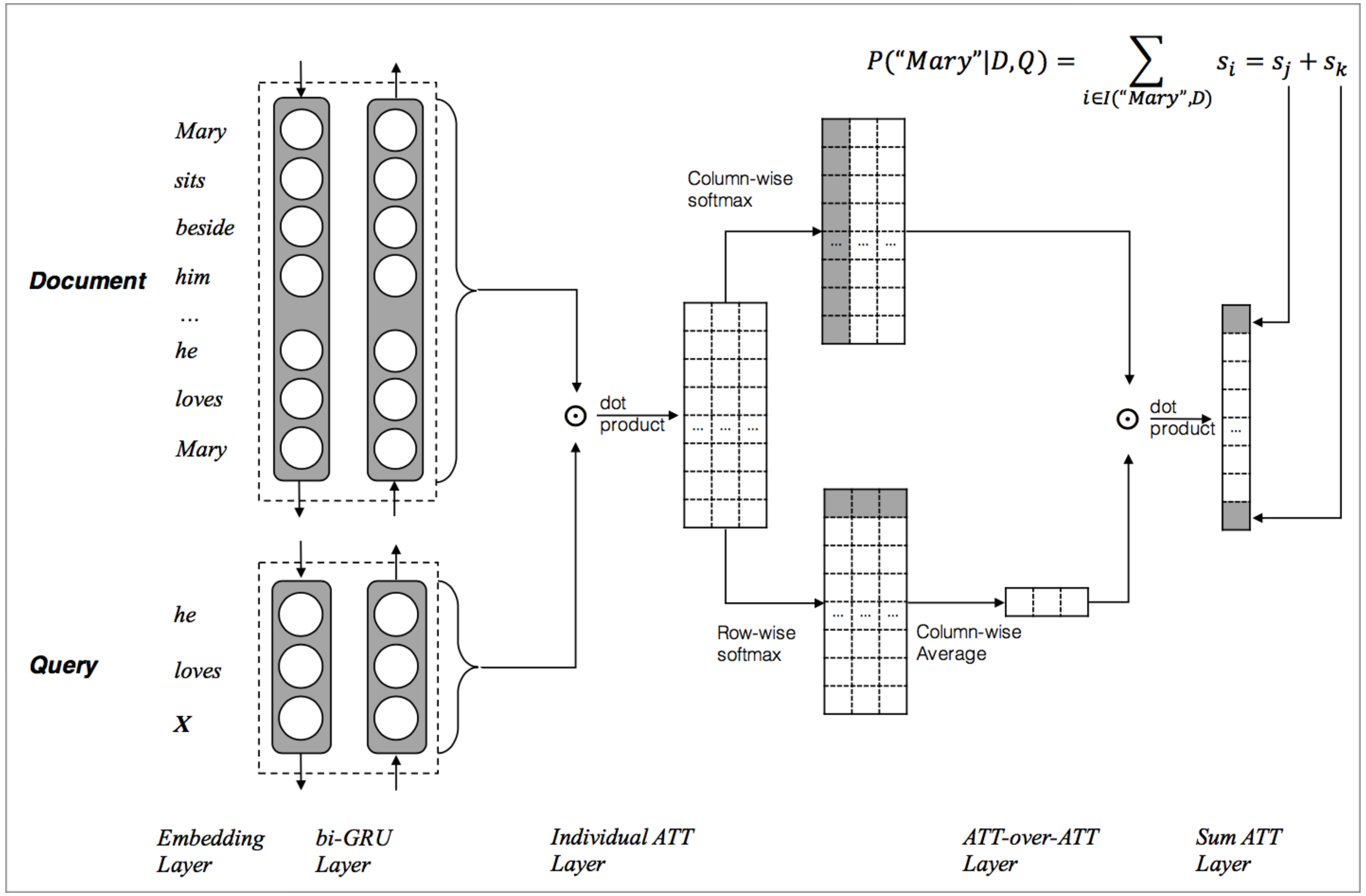


AoA Reader

- Primarily motivated by AS Reader (Kadlec et al., 2016) and CAS Reader (Cui et al., 2016)
  - Introduce matching matrix for indicating doc-query relationships
  - Mutual attention: doc-to-query and query-to-doc
  - Instead of using heuristics to combine individual attentions, we place another attention to dynamically assign weights to the individual ones
- Some of the ideas in our work has already been adopted in the follow-up works not only in cloze-style RC but also other types of RC (such as SQuAD).

AoA Reader

- Model architecture at a glance



AoA Reader

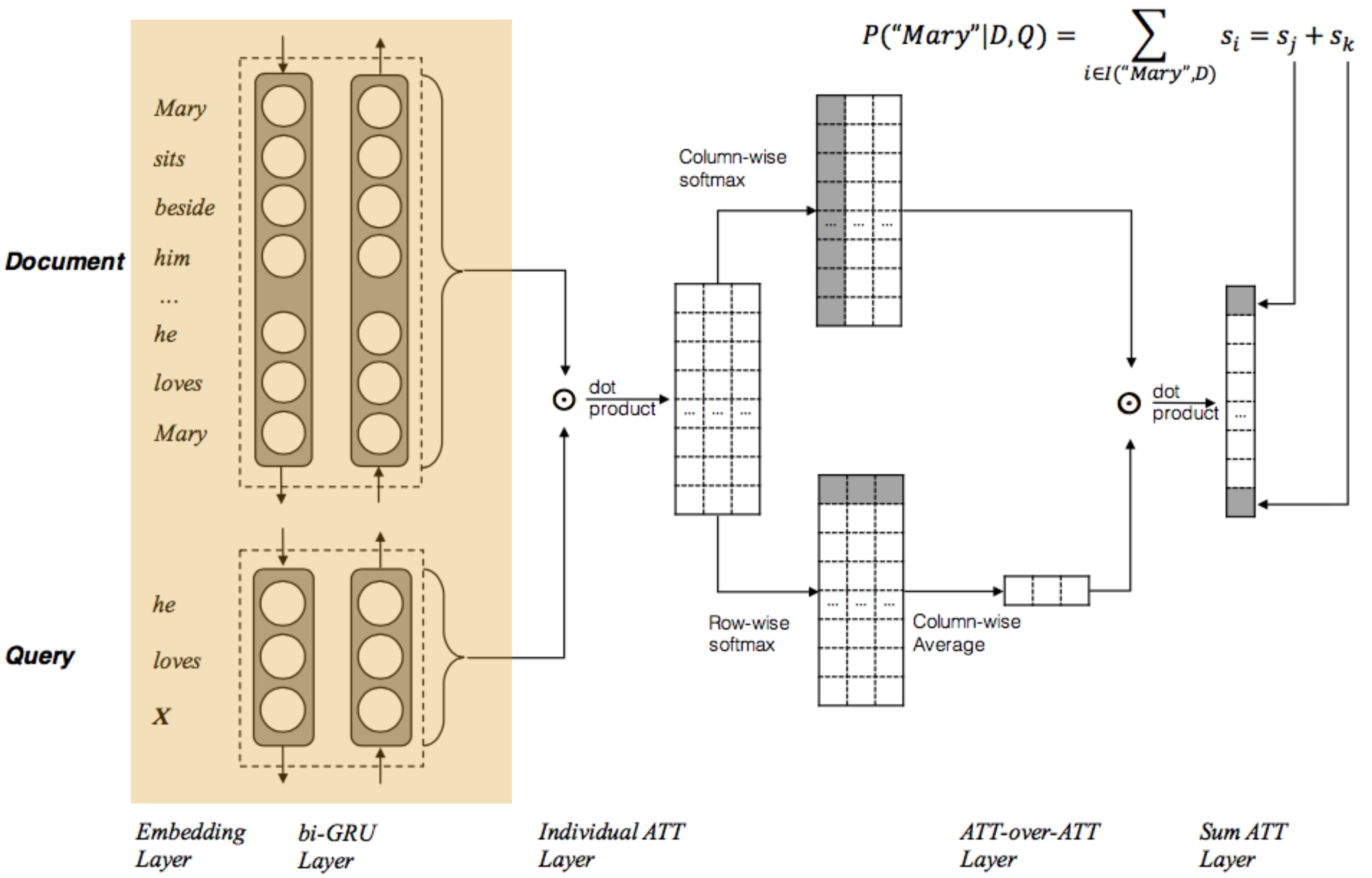
- **Contextual Embedding**
  - Transform document and query into contextual representations using word-embeddings and bi-GRU units

$$e(x) = W_e \cdot x, \ where \ x \in \mathcal{D}, \mathcal{Q} \quad (1)$$

$$\overrightarrow{h_s(x)} = \overrightarrow{GRU}(e(x)) \quad (2)$$

$$\overleftarrow{h_s(x)} = \overleftarrow{GRU}(e(x)) \quad (3)$$

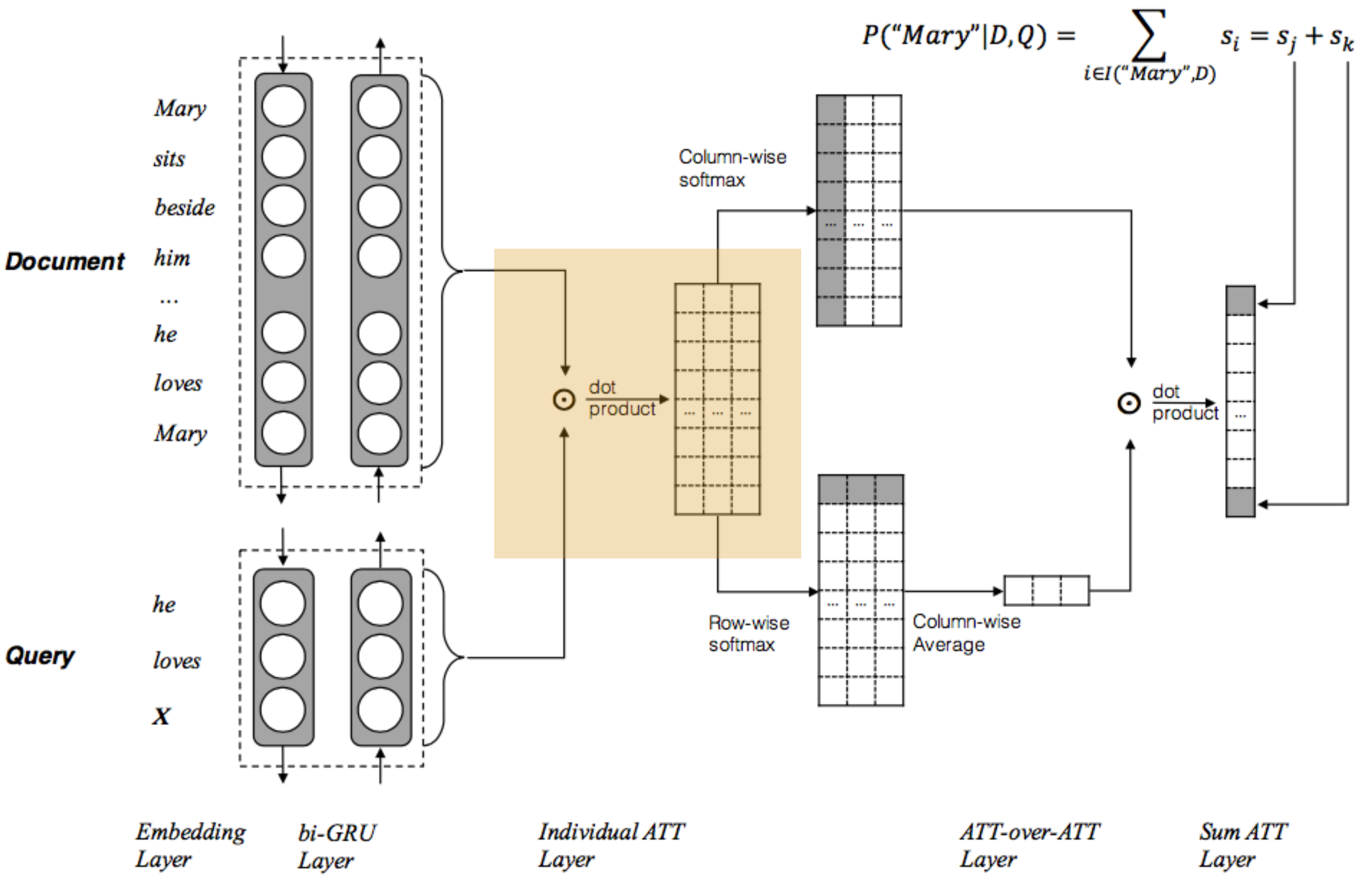
$$h_s(x) = [\overrightarrow{h_s(x)}; \overleftarrow{h_s(x)}] \quad (4)$$



AoA Reader

- **Pair-wise Matching Score**
  - Calculate similarity between each document word and query word
  - For simplicity, we just calculate dot product between document and query word

$$M(i,j) = h_{doc}(i)^T \cdot h_{query}(j) \qquad (5)$$

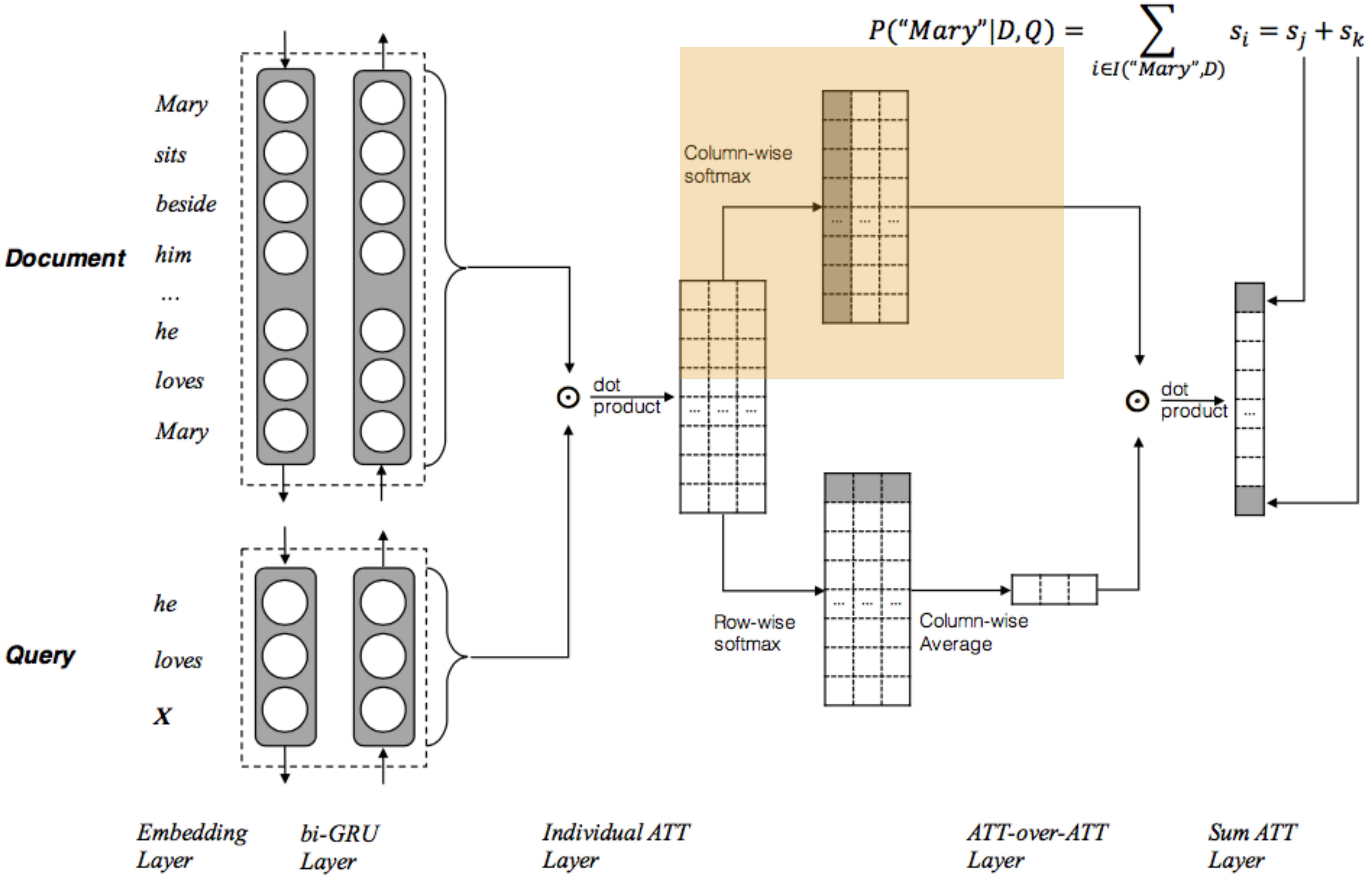


AoA Reader

- **Individual Attentions**
  - Calculate document-level attention with respect to each query word

$$\alpha(t) = softmax(M(1,t), ..., M(|\mathcal{D}|,t)) \quad (6)$$

$$\alpha = [\alpha(1), \alpha(2), ..., \alpha(|\mathcal{Q}|)] \quad (7)$$



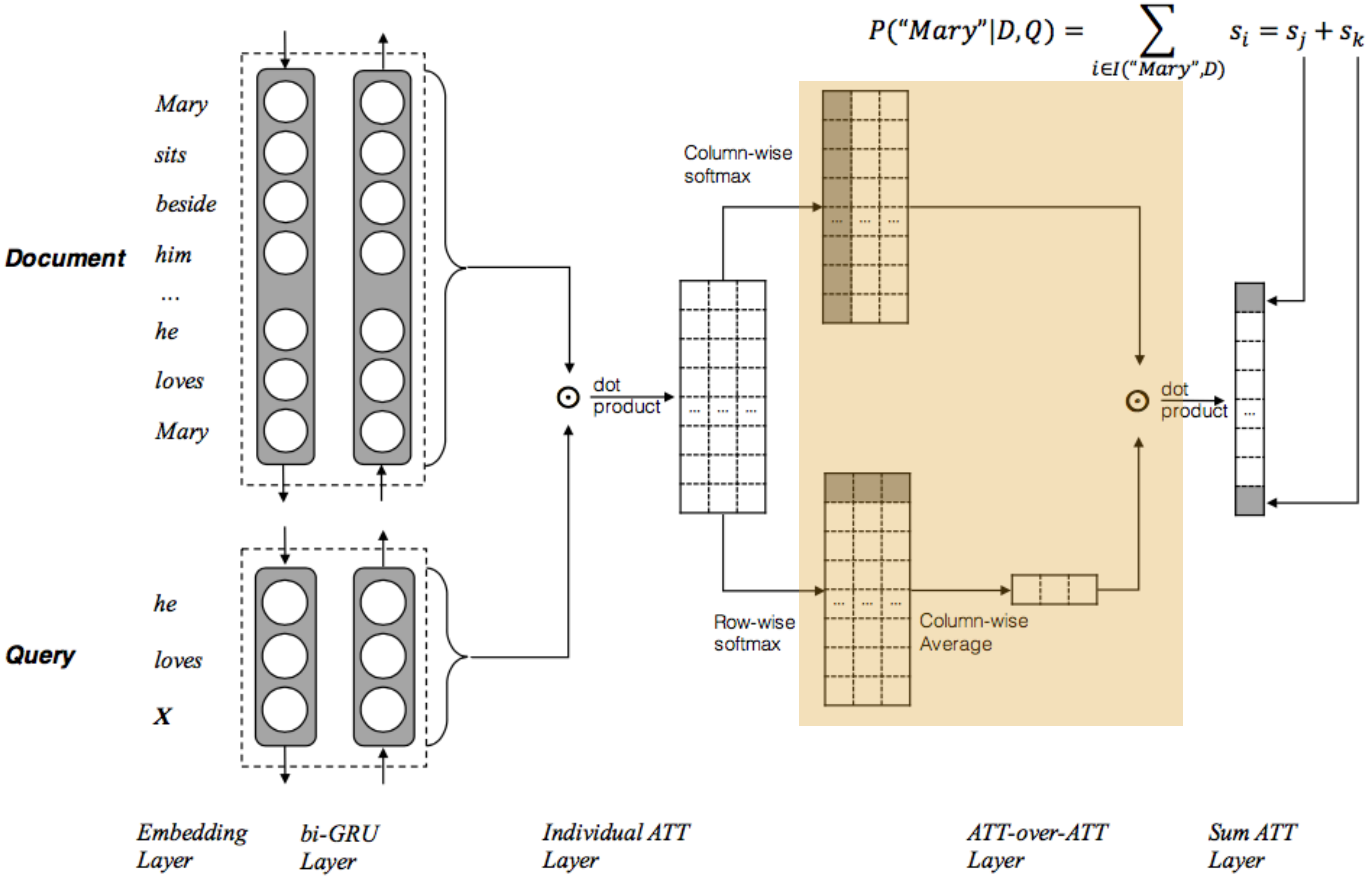
AoA Reader

- **Attention-over-Attention**
  - Dynamically assign weights to individual doc-level attentions

$$\beta(t) = softmax(M(t,1), ..., M(t,|\mathcal{Q}|)) \quad (8)$$

$$\beta = \frac{1}{n}\sum_{t=1}^{|\mathcal{D}|}\beta(t) \quad (9)$$

$$s = \alpha^T\beta \quad (10)$$

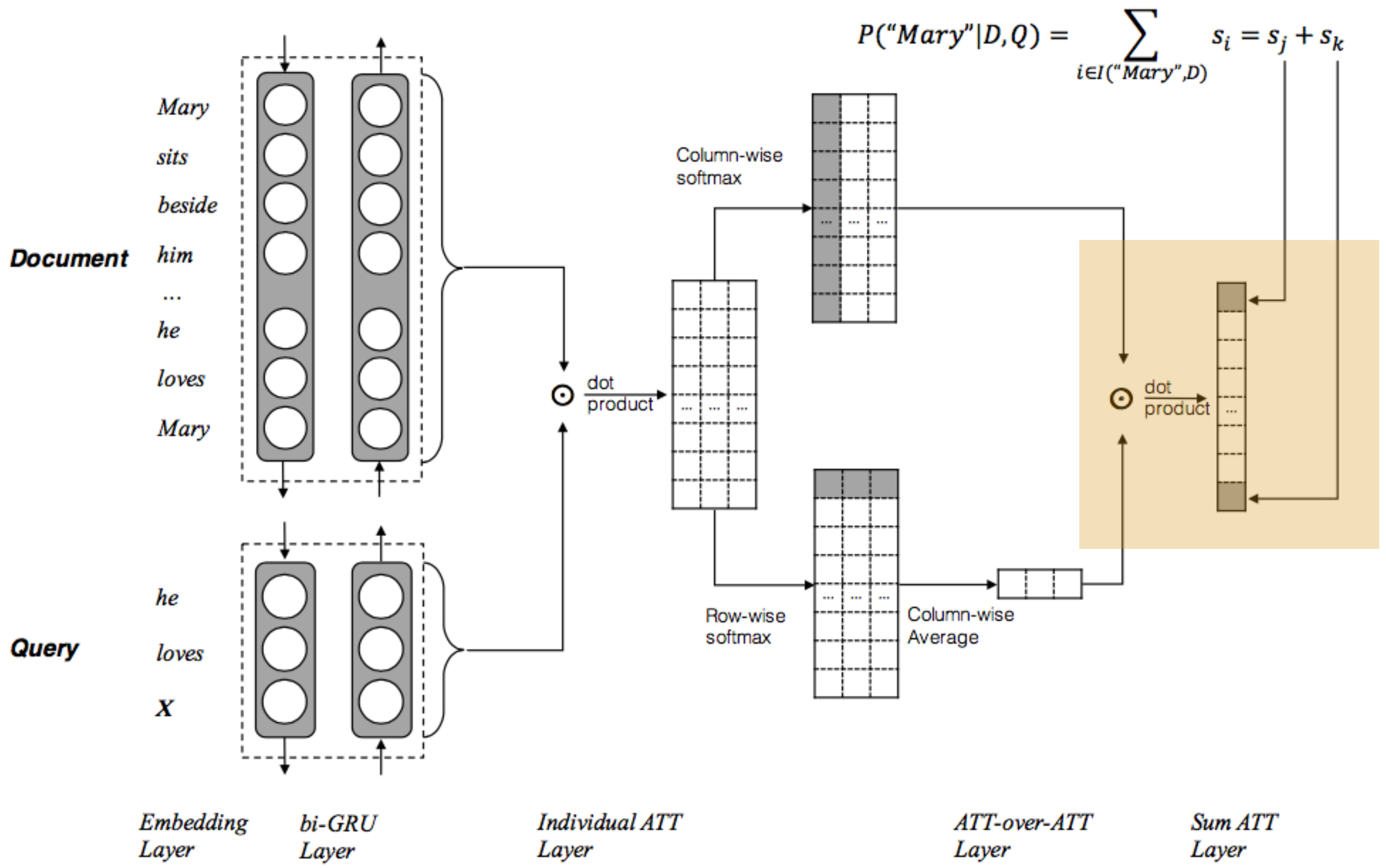


AoA Reader

- **Final Predictions**
  - Pointer Network (Vinyals et al., 2015)
  - Apply sum-attention mechanism (Kadlec et al., 2016) to get the final probability of the answer

$$P(w|\mathcal{D},\mathcal{Q}) = \sum_{i \in I(w,\mathcal{D})} s_i, \quad w \in V \qquad (11)$$

$$\mathcal{L} = \sum_i \log(p(x)) \quad , x \in \mathcal{A} \qquad (12)$$



AoA Reader

- **An intuitive example:** Let say this is a story about `Tom bought a diamond ring for his beloved girl friend…`

	Tom	loves	<blank>	.
Query-level Attention	0.5	0.3	0.15	0.05
Candidate Answers	Mary = 0.6 diamond = 0.3 beside = 0.1	Mary = 0.3 diamond = 0.5 beside = 0.2	Mary = 0.4 diamond = 0.4 beside = 0.2	Mary = 0.2 diamond = 0.4 beside = 0.4
Average Score (Cui et al., 2016)	Mary = (0.6+0.3+0.4+0.2) / 4 = 0.375 **diamond = (0.3+0.5+0.4+0.4) / 4 = 0.400** beside = (0.1+0.2+0.2+0.4) / 4 = 0.225			
Weighted Score (This work)	**Mary = 0.6*0.5+0.3*0.3+0.4*0.15+0.2*0.05 = 0.460** diamond = 0.3*0.5+0.5*0.3+0.4*0.15+0.4*0.05 = 0.380 beside = 0.1*0.5+0.2*0.3+0.2*0.15+0.4*0.05 = 0.160			

RE-RANKING

- **N-best re-ranking strategy for cloze-style RC**
  - Mimic the process of double-checking, in terms of fluency, grammatical correctness etc.
  - Main idea: Re-fill the candidate answer into the blank of query to form a complete sentence and using additional features to score the sentences

RE-RANKING

- **Procedure of re-ranking**
  - Generate candidate answers: N-best decoding
  - Refill the candidate into query
  - Scoring with additional features: mainly LM features
  - Feature weight tuning: using K-Best MIRA algorithm (Cherry and Foster, 2012)
  - Re-scoring and Re-ranking

RE-RANKING

- **Features that used in re-ranking**
  - Global LM: trained on document part of training data
    - Word LM: 8-gram LM using SRILM tool (Stolcke, 2002)
    - Word-class LM: 1,000 word classes using `mkcls` tool (Josef Och, 1999)
  - Local LM: trained on document part of test-time data **sample-by-sample**

EXPERIMENTS

- **Dataset**
  - CNN(Hermann et al., 2015), CBT-NE/CN (Hill et al., 2015)
- **Hyper-parameters**
  - Embedding: uniform distribution with l2-regularization, dropout 0.1
  - Hidden Layer: bi-GRU
  - Optimization: Adam(lr=0.001), gradient clipping 5, batch 32
- **Implementation**: Keras (Chollet, 2015) + Theano (Theano Development Team, 2016)

Experimental Results

- **Single model performance**
  - Significantly outperform previous works
  - Re-ranking strategy could substantially improve performance

	CNN News		CBTest NE		CBTest CN	
	Valid	Test	Valid	Test	Valid	Test
Deep LSTM Reader (Hermann et al., 2015)	55.0	57.0	-	-	-	-
Attentive Reader (Hermann et al., 2015)	61.6	63.0	-	-	-	-
Human (context+query) (Hill et al., 2015)	-	-	-	*81.6*	-	*81.6*
MemNN (window + self-sup.) (Hill et al., 2015)	63.4	66.8	70.4	66.6	64.2	63.0
AS Reader (Kadlec et al., 2016)	68.6	69.5	73.8	68.6	68.8	63.4
CAS Reader (Cui et al., 2016)	68.2	70.0	74.2	69.2	68.2	65.7
Stanford AR (Chen et al., 2016)	72.4	72.4	-	-	-	-
GA Reader (Dhingra et al., 2016)	73.0	73.8	74.9	69.0	69.0	63.9
Iterative Attention (Sordoni et al., 2016)	72.6	73.3	75.2	68.6	72.1	69.2
EpiReader (Trischler et al., 2016)	**73.4**	74.0	75.3	69.7	71.5	67.4
AoA Reader	73.1	**74.4**	**77.8**	**72.0**	**72.2**	**69.4**
AoA Reader + Reranking	-	-	**79.6**	**74.0**	**75.7**	**73.1**
MemNN (Ensemble)	66.2	69.4	-	-	-	-
AS Reader (Ensemble)	73.9	75.4	74.5	70.6	71.1	68.9
GA Reader (Ensemble)	76.4	77.4	75.5	71.9	72.1	69.4
EpiReader (Ensemble)	-	-	76.6	71.8	73.6	70.6
Iterative Attention (Ensemble)	74.5	75.7	76.9	72.0	74.1	**71.0**
AoA Reader (Ensemble)	-	-	**78.9**	**74.5**	**74.7**	70.8
AoA Reader (Ensemble + Reranking)	-	-	**80.3**	**75.6**	**77.0**	**74.1**

Experimental Results

- **Single model performance**
  - Introducing attention-over-attention mechanism instead of using heuristic merging function (Cui et al., 2016) may bring significant improvements

	CNN News		CBTest NE		CBTest CN	
	Valid	Test	Valid	Test	Valid	Test
Deep LSTM Reader (Hermann et al., 2015)	55.0	57.0	-	-	-	-
Attentive Reader (Hermann et al., 2015)	61.6	63.0	-	-	-	-
Human (context+query) (Hill et al., 2015)	-	-	-	*81.6*	-	*81.6*
MemNN (window + self-sup.) (Hill et al., 2015)	63.4	66.8	70.4	66.6	64.2	63.0
AS Reader (Kadlec et al., 2016)	68.6	69.5	73.8	68.6	68.8	63.4
CAS Reader (Cui et al., 2016)	68.2	70.0	74.2	69.2	68.2	65.7
Stanford AR (Chen et al., 2016)	72.4	72.4	-	-	-	-
GA Reader (Dhingra et al., 2016)	73.0	73.8	74.9	69.0	69.0	63.9
Iterative Attention (Sordoni et al., 2016)	72.6	73.3	75.2	68.6	72.1	69.2
EpiReader (Trischler et al., 2016)	**73.4**	74.0	75.3	69.7	71.5	67.4
AoA Reader	73.1	**74.4**	**77.8**	**72.0**	**72.2**	**69.4**
AoA Reader + Reranking	-	-	**79.6**	**74.0**	**75.7**	**73.1**
MemNN (Ensemble)	66.2	69.4	-	-	-	-
AS Reader (Ensemble)	73.9	75.4	74.5	70.6	71.1	68.9
GA Reader (Ensemble)	76.4	77.4	75.5	71.9	72.1	69.4
EpiReader (Ensemble)	-	-	76.6	71.8	73.6	70.6
Iterative Attention (Ensemble)	74.5	75.7	76.9	72.0	74.1	**71.0**
AoA Reader (Ensemble)	-	-	**78.9**	**74.5**	**74.7**	70.8
AoA Reader (Ensemble + Reranking)	-	-	**80.3**	**75.6**	**77.0**	**74.1**

EXPERIMENTAL RESULTS

- **Ensemble performance**
  - We use greedy ensemble approach of 4 models trained on different random seed
  - Significant improvements over state-of-the-art systems

	CNN News		CBTest NE		CBTest CN	
	Valid	Test	Valid	Test	Valid	Test
Deep LSTM Reader (Hermann et al., 2015)	55.0	57.0	-	-	-	-
Attentive Reader (Hermann et al., 2015)	61.6	63.0	-	-	-	-
Human (context+query) (Hill et al., 2015)	-	-	-	*81.6*	-	*81.6*
MemNN (window + self-sup.) (Hill et al., 2015)	63.4	66.8	70.4	66.6	64.2	63.0
AS Reader (Kadlec et al., 2016)	68.6	69.5	73.8	68.6	68.8	63.4
CAS Reader (Cui et al., 2016)	68.2	70.0	74.2	69.2	68.2	65.7
Stanford AR (Chen et al., 2016)	72.4	72.4	-	-	-	-
GA Reader (Dhingra et al., 2016)	73.0	73.8	74.9	69.0	69.0	63.9
Iterative Attention (Sordoni et al., 2016)	72.6	73.3	75.2	68.6	72.1	69.2
EpiReader (Trischler et al., 2016)	**73.4**	74.0	75.3	69.7	71.5	67.4
AoA Reader	73.1	**74.4**	**77.8**	**72.0**	**72.2**	**69.4**
AoA Reader + Reranking	-	-	**79.6**	**74.0**	**75.7**	**73.1**
MemNN (Ensemble)	66.2	69.4	-	-	-	-
AS Reader (Ensemble)	73.9	75.4	74.5	70.6	71.1	68.9
GA Reader (Ensemble)	76.4	77.4	75.5	71.9	72.1	69.4
EpiReader (Ensemble)	-	-	76.6	71.8	73.6	70.6
Iterative Attention (Ensemble)	74.5	75.7	76.9	72.0	74.1	**71.0**
AoA Reader (Ensemble)	-	-	**78.9**	**74.5**	**74.7**	70.8
AoA Reader (Ensemble + Reranking)	-	-	**80.3**	**75.6**	**77.0**	**74.1**

RE-RANKING ABLATIONS

- Calculate weight proportion between global and local LMs

$$\eta = \frac{LM_{global} + LM_{wc}}{LM_{local}}$$

- Observations
  - NE category seems to be more dependent on local LM
  - CN category seems to be more dependent on global LM

	CBTest NE		CBTest CN	
	Valid	Test	Valid	Test
AoA Reader	77.8	72.0	72.2	69.4
+Global LM	78.3	72.6	73.9	71.2
+Local LM	79.4	73.8	74.7	71.7
+Word-class LM	79.6	74.0	75.7	73.1

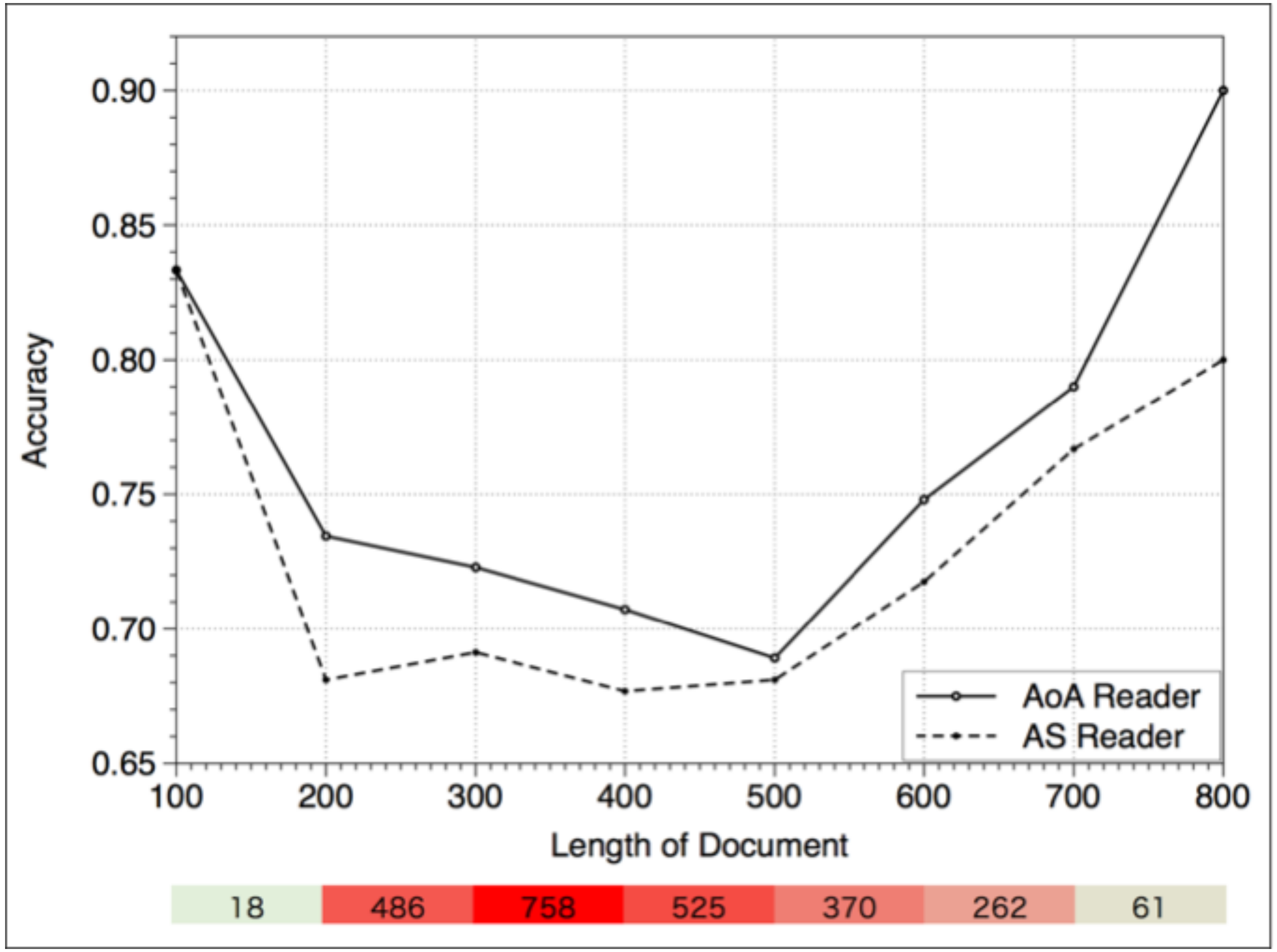
	CBTest NE	CBTest CN
NN	0.64	0.20
Global LM	0.16	0.10
Word-class LM	0.04	0.39
Local LM	0.16	0.31
RATIO η	1.25	1.58

QUANTITATIVE ANALYSIS

- Accuracy vs. Length of Document

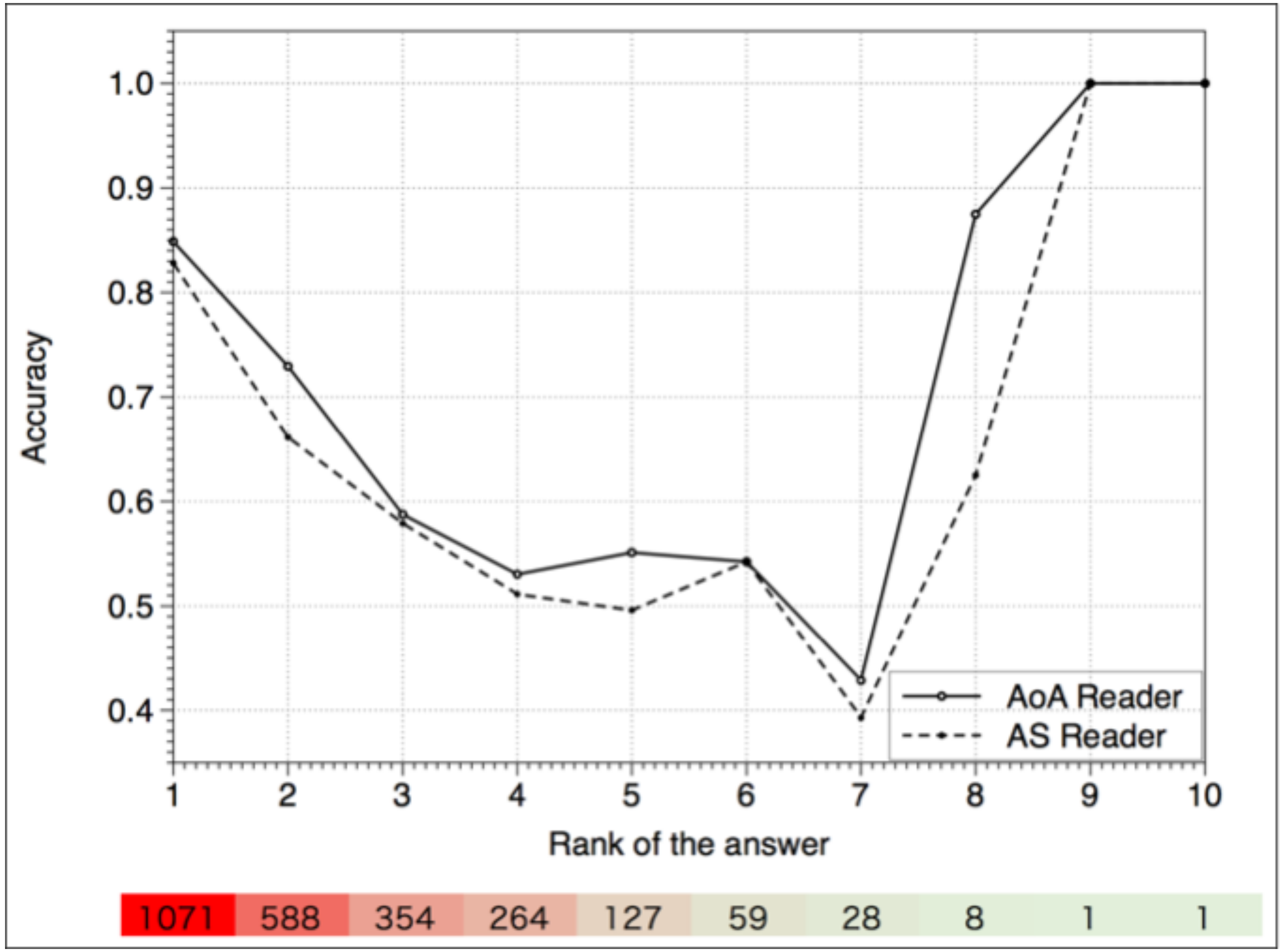
  - AoA Reader shows consistent improvements over AS Reader on different length of document

  - Improvements become larger when the length of document increases, indicating that our model could better handle the long documents



Quantitative Analysis

- Accuracy vs. Frequency of answer
  - Most of the answers are the top frequent word among candidates
  - Tend to choose either high or low frequency word



CONCLUSIONS & FUTURE WORK

- **Conclusion**
  - Propose a novel mechanism called "**Attention-over-Attention**" to dynamically assign weights to the individual attentions
  - Two-way attention: adopt both **doc-to-query** and **query-to-doc** attentions for final predictions
  - Experimental results show significant improvements over various state-of-the-art systems
- **Future Work**
  - Investigate more complex attention mechanism via adopting external knowledge
  - Look into the questions that need comprehensive reasoning over several sentences

EXTENSION: INTERACTIVE AOA READER

- **Interactive AoA Reader**
  - As a step further of our work, we've refined our model as 'interactive', which dynamically and progressively filter the context for question answering
  - Shows state-of-the-art performance and ranked **No.1** in Stanford SQuAD Task (Rajpurkar et al., 2016)

Leaderboard

Since the release of our dataset, the community has made rapid progress! Here are the ExactMatch (EM) and F1 scores of the best models evaluated on the test and development sets of v1.1. Will your model outperform humans on the QA task?

Rank	Model	EM	F1
1 Jul 2017	Interactive AoA Reader (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	77.845	85.297

*As of August 1, 2017. http://stanford-qa.com

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 .
- Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In Association for Computational Linguistics (ACL).
- Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Montre´al, Canada, pages 427–436. http://www.aclweb.org/anthology/N12-1047.
- Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, pages 1724–1734. http://aclweb.org/anthology/D14-1179.
- François Chollet. 2015. Keras. https://github. com/fchollet/keras.

References

- Yiming Cui, Ting Liu, Zhipeng Chen, Shijin Wang, and Guoping Hu. 2016. Consensus attention-based neural networks for chinese reading comprehension. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. Osaka, Japan, pages 1777–1786.

- Bhuwan Dhingra, Hanxiao Liu, William W Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension. arXiv preprint arXiv:1606.01549 .

- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Advances in Neural Information Processing Systems. pages 1684–1692.

- Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. arXiv preprint arXiv:1511.02301 .

- Franz Josef Och. 1999. An efficient method for determining bilingual word classes. In Ninth Conference of the European Chapter of the Association for Computational Linguistics. http://aclweb.org/anthology/E99-1010.

REFERENCES

- Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. arXiv preprint arXiv:1603.01547 .
- Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .
- Ting Liu, Yiming Cui, Qingyu Yin, Shijin Wang, Weinan Zhang, and Guoping Hu. 2016. Generating and exploiting large-scale pseudo training data for zero pronoun resolution. arXiv preprint arXiv:1606.01603 . In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL-2017).
- Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. ICML (3) 28:1310–1318.
- Andrew M Saxe, James L McClelland, and Surya Ganguli. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv preprint arXiv:1312.6120 .
- Alessandro Sordoni, Phillip Bachman, and Yoshua Bengio. 2016. Iterative alternating neural attention for machine reading. arXiv preprint arXiv:1606.02245

REFERENCES

- Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1):1929–1958.
- Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002). pages 901–904.
- Wilson L Taylor. 1953. Cloze procedure: a new tool for measuring readability. Journalism and Mass Communication Quarterly 30(4):415.
- Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints abs/1605.02688. http://arxiv.org/abs/1605.02688.
- Adam Trischler, Zheng Ye, Xingdi Yuan, and Kaheer Suleman. 2016. Natural language comprehension with the epireader. arXiv preprint arXiv:1606.02270 .
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In Advances in Neural Information Processing Systems. pages 2692–2700.

Thank you !

Enjoy your time in Vancouver !

Slides download

Contact: me [at] ymcui [dot] com