A Fully Attention Based Information Retriever (FABIR)

WCCI-IJCNN - Rio 2018

Session 8a-1: Applications of deep networks

Alvaro H. C. Correia Jorge Luiz Moreira Silva Fabio G. Cozman

11th July 2018

Escola Politécnica da Universidade de São Paulo

Introduction

Problem Definition

To develop a Deep Learning model to solve the following task

Given a passage $\mathcal{P}$ and a query $\mathcal{Q}$ both written in English, to produce an answer $\mathcal{A}$ by selecting a continuous snippet from $\mathcal{P}$

Stanford Question-Answering Dataset

>100,000 Questions | >23,000 Passages | >500 Wikipedia Articles

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of the precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crsystal **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? **graupel**
- Where do water droplets collide with ice crystal to form precipitation? **within a cloud**

Goal

Neural Machine Translation

	Traditional RNN Approach	**Google's Transformer Approach**
# of Sequential Operations	$\mathcal{O}(n)$	$\mathcal{O}(1)$
Maximum path between words	$\mathcal{O}(n)$	$\mathcal{O}(1)$

Question-Answering

Why not to apply Google's Transformer approach in other NLP tasks?

- Develop new deep learning architecture
- Explore the potential of new attention mechanisms

FABIR

Fully

Attention

Based

Information

Retriever

Model Description

Model Input - Preprocessing

Raw Text

Obi-Wan was a Padawan learner to which Jedi Master?

Tokenized Text (10 Tokens)

Obi-Wan	was	a	Padawan	learner	to	which	Jedi	Master	?

Embedding Matrix

$$\begin{bmatrix} \omega_{1,1} & \omega_{2,1} & \omega_{3,1} & \omega_{4,1} & \omega_{5,1} & \omega_{6,1} & \omega_{7,1} & \omega_{8,1} & \omega_{9,1} & \omega_{10,1} \\ \omega_{1,2} & \omega_{2,2} & \omega_{3,2} & \omega_{4,2} & \omega_{5,2} & \omega_{6,2} & \omega_{7,2} & \omega_{8,2} & \omega_{9,2} & \omega_{10,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_{1,d} & \omega_{2,d} & \omega_{3,d} & \omega_{4,d} & \omega_{5,d} & \omega_{6,d} & \omega_{7,d} & \omega_{8,d} & \omega_{9,d} & \omega_{10,d} \end{bmatrix} =$$

$$\begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 & \Omega_5 & \Omega_6 & \Omega_7 & \Omega_8 & \Omega_9 & \Omega_{10} \end{bmatrix} \in \mathbb{R}^{10 \times 200}$$

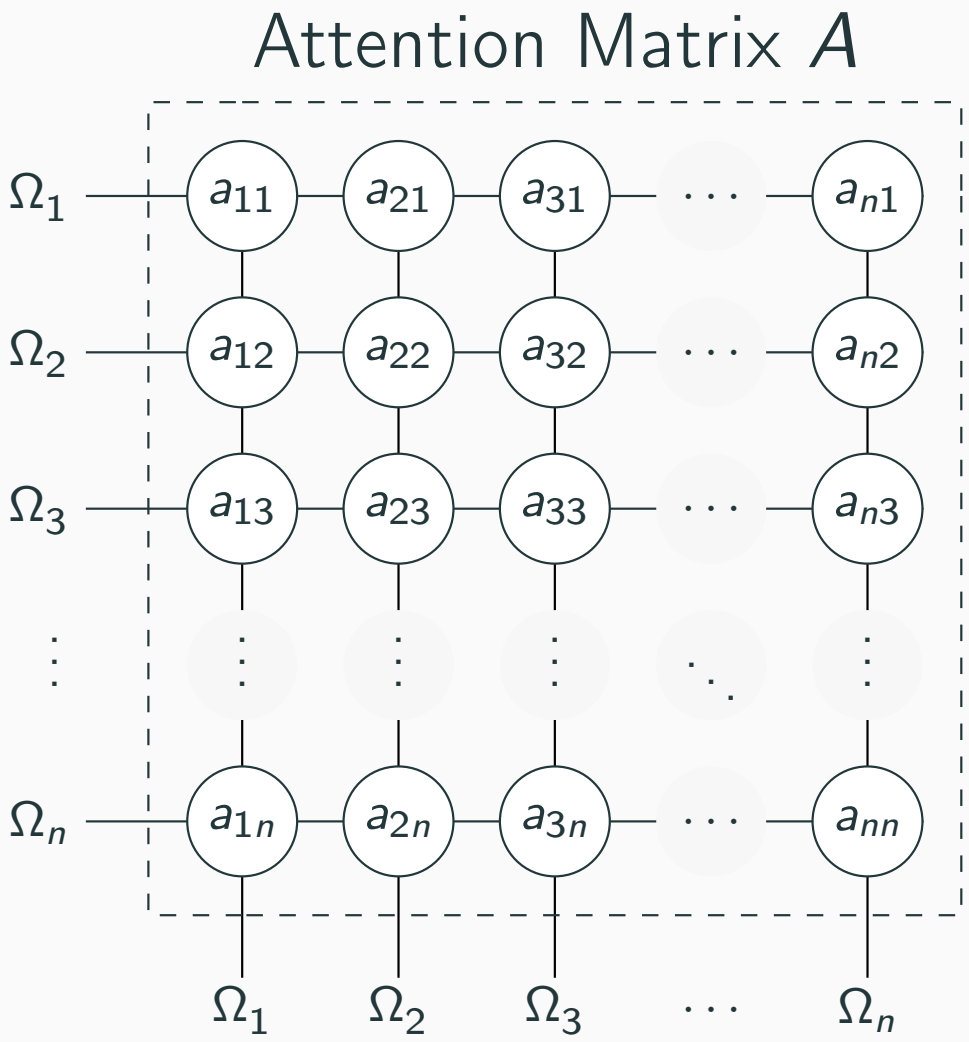
FABIR - Attention Mechanism

Input

n tokens piece of Text

Attention

1) $a_{ij} = f\left(\Omega_i, \Omega_j\right)$
2) $\overline{A} = \mathrm{softmax}(A)$
3) $O = \Omega \ * \overline{A}$



FABIR - w/ Position Encoder

Input

n tokens piece of Text

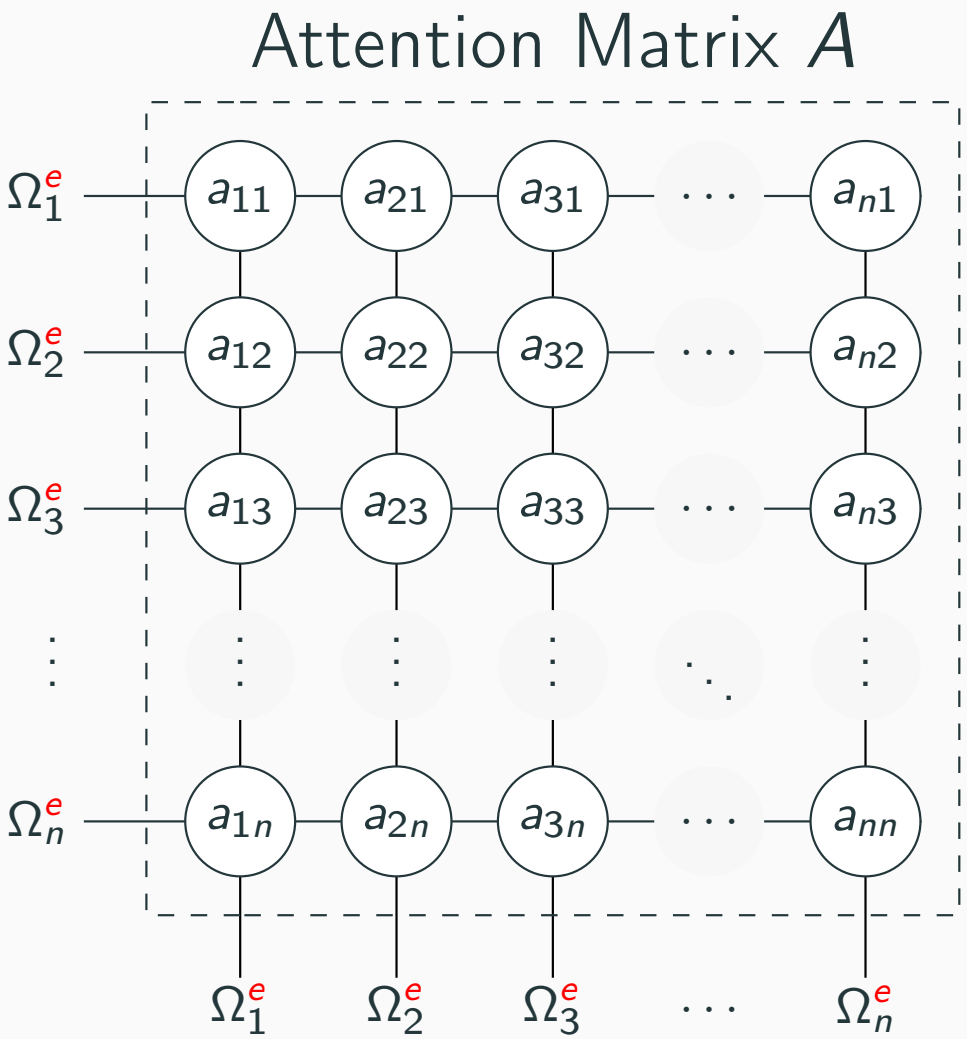
Attention

0) $\Omega_i^e = \Omega_i + E_i$

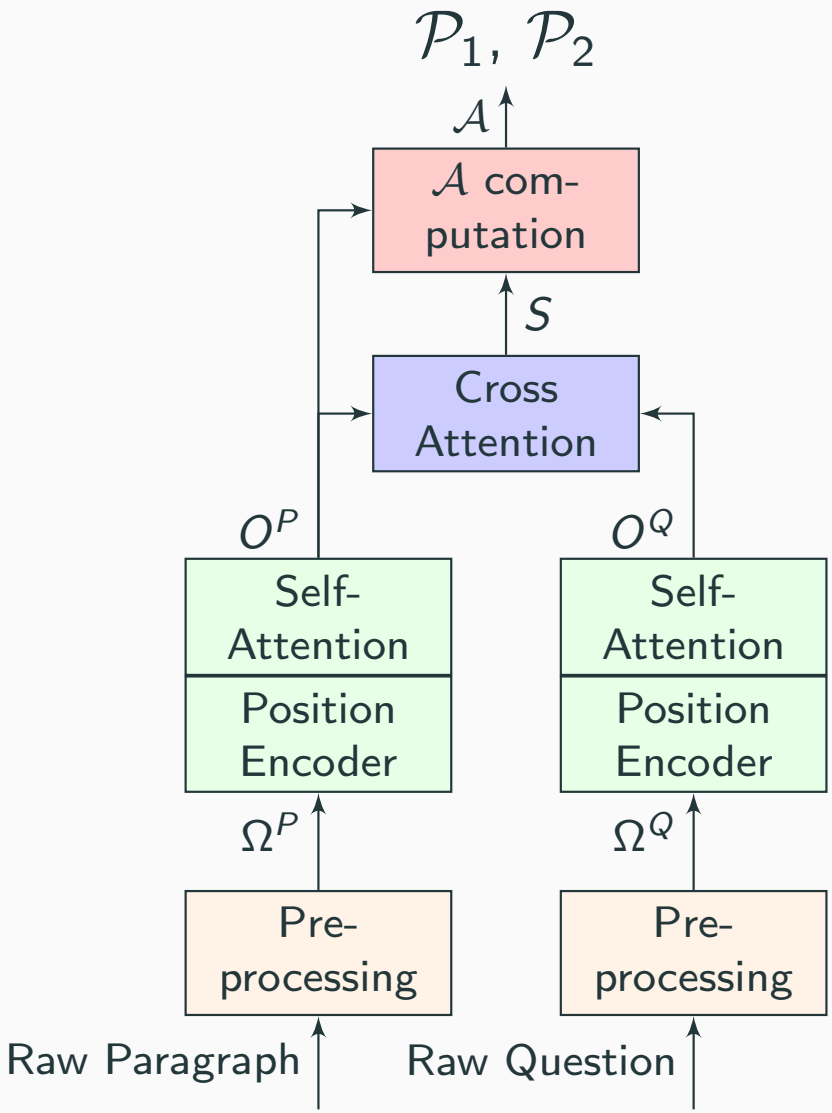
1) $a_{ij} = f\left(\Omega_i^e, \Omega_j^e\right)$

2) $\overline{A} = \mathrm{softmax}(A)$

3) $O = \Omega^e * \overline{A}$



FABIR - Simplified Pipeline



Major Contributions

Convolutional Attention

Reduction Layer

Column-wise Cross Attention

Major Contributions - Convolutional Attention

Input

n tokens piece of Text

Attention

$\Omega_i^e = \Omega_i + E_i$

$a_{ij} = f\left(\Omega_i^e, \Omega_j^e\right)$

$a_{ij}^c = g\left(a_{i-s_1:i+s_1, j-s_2:j+s_2}\right)$

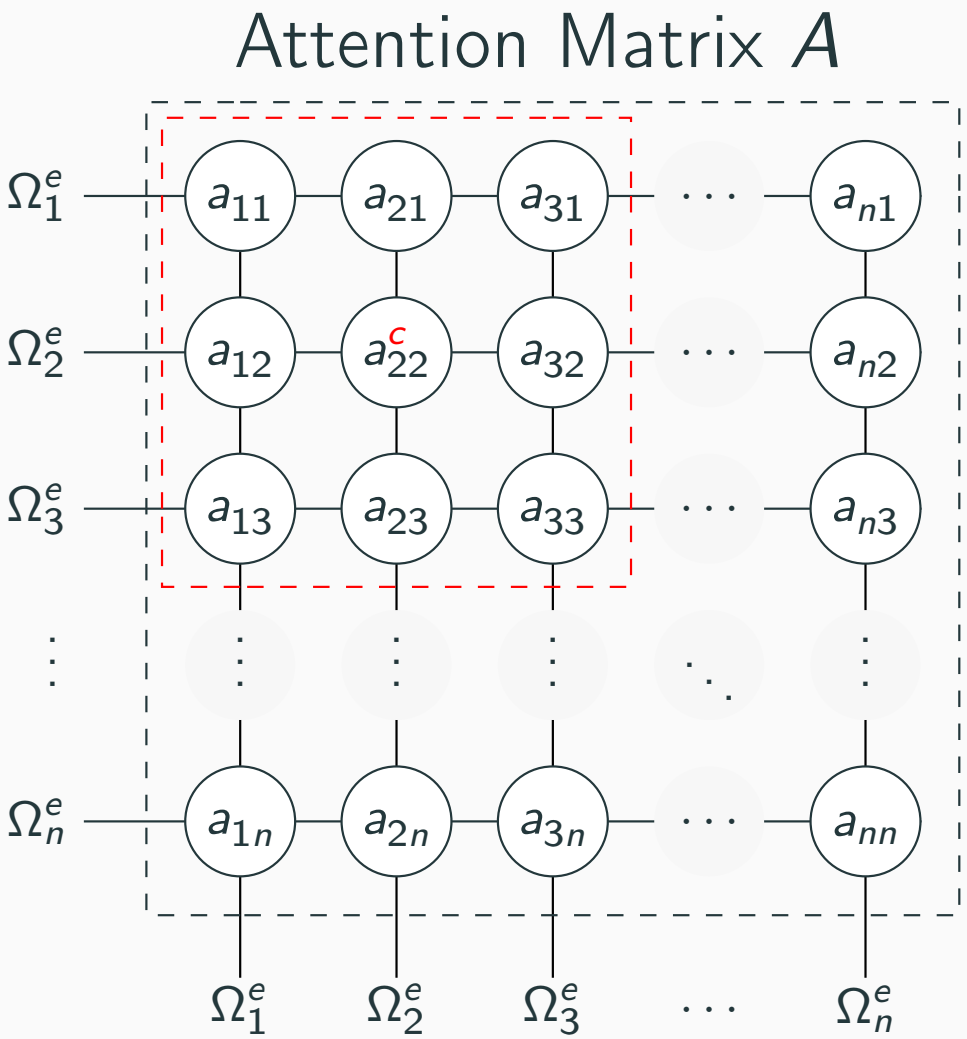
$\overline{A^c} = \mathrm{softmax}(A^c)$

$O = \Omega^e * \overline{A^c}$

Normalized Weights

$\overline{a}_{t,i}^c \geq 0$

$\sum_{i=1}^{n} \overline{a}_{it}^c = 1$



Major Contributions - Convolutional Attention

Input

n tokens piece of Text

Attention

$\Omega_i^e = \Omega_i + E_i$

$a_{ij} = f\left(\Omega_i^e, \Omega_j^e\right)$

$a_{ij}^c = g\left(a_{i-s_1:i+s_1, j-s_2:j+s_2}\right)$

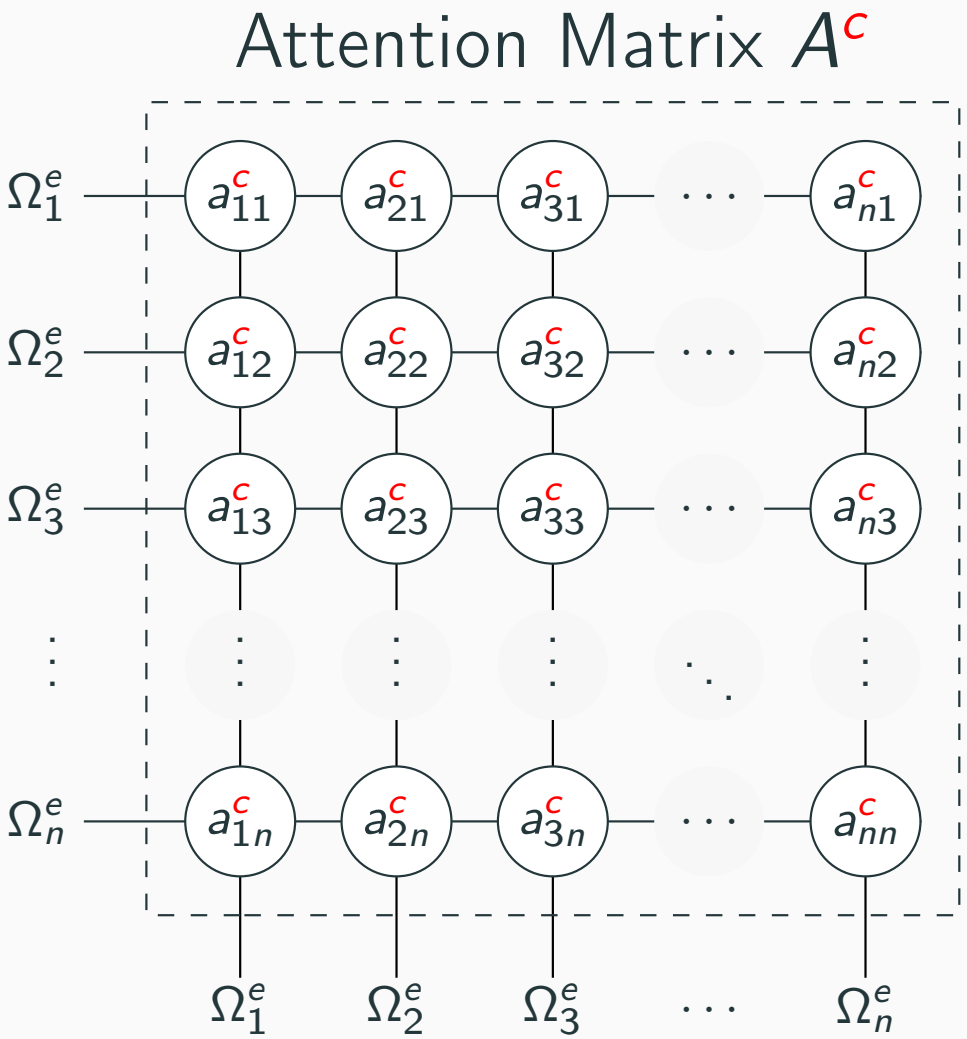
$\overline{A^c} = \mathrm{softmax}(A^c)$

$O = \Omega^e * \overline{A^c}$

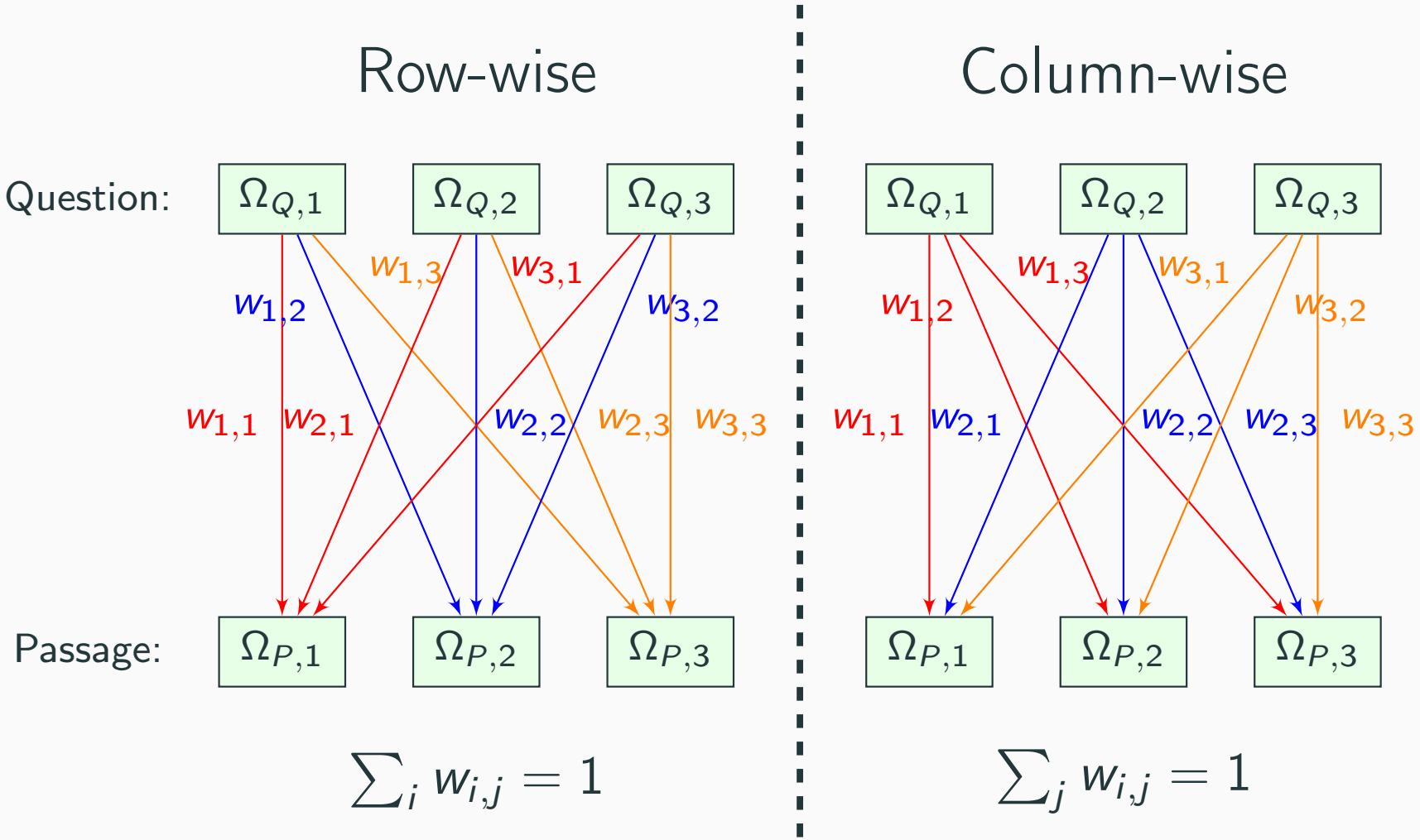
Normalized Weights

$\overline{a}_{t,i}^c \geq 0$

$\sum_{i=1}^{n} \overline{a}_{it}^c = 1$



Major Contributions - Column-wise Cross-Attention



Major Contributions - Reduction Layer

Problem

Attention-based architectures are susceptible to overfitting

Solution

Reduction Layer

- Compress word representations while maintaining position encoding
- Reduce number of parameters - less overfitting
  Number of parameters is quadratic on the embedding size
- Richer word representations

Results

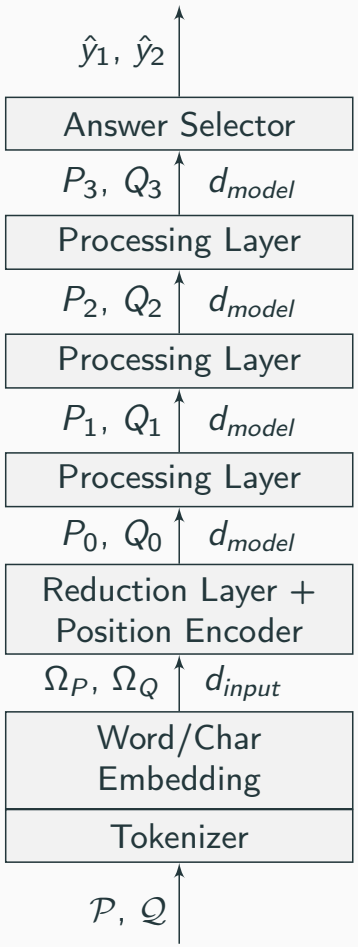
FABIR - Ablation Study over 18-epoch runs

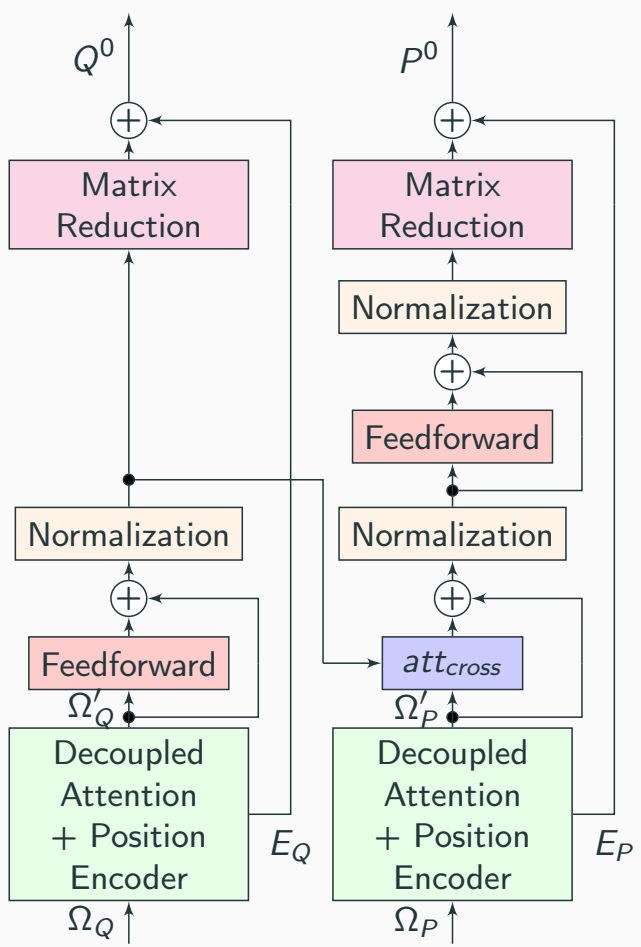
Architecture	**F1(%)**	**EM(%)**	**Training Time**
FABIR	75.6	65.1	2h14m
without convolutional attention	-2.4	-2.5	-25m
without column-wise cross attention	-2.0	-1.9	-6m
without the reduction layer	-2.1	-2.5	-15m

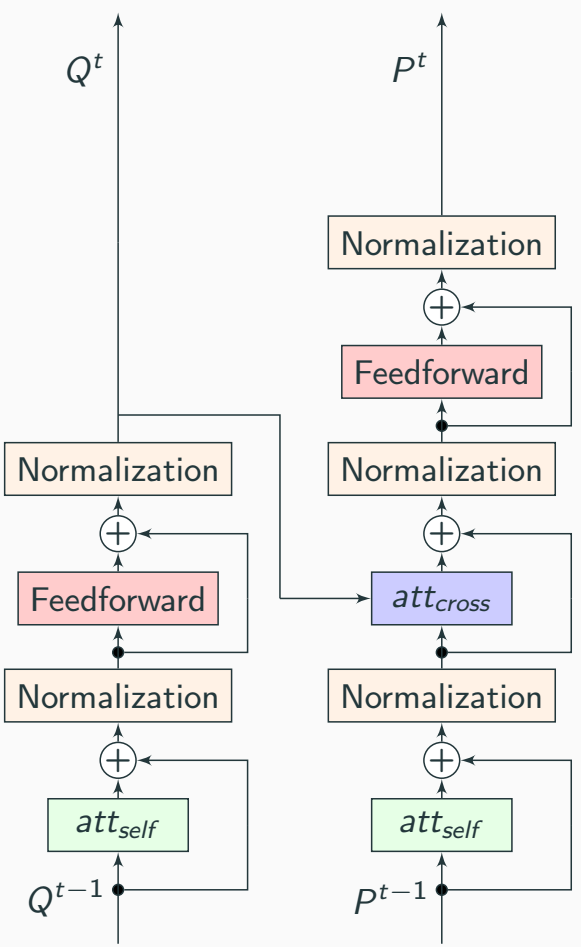
FABIR vs RNN (BiDAF) - General Results

	FABIR	**BiDAF**
F1 (%)	**77.6**	77.0 (77.3)
Exact Match (%)	67.6	67.3 (**68.0**)
Training Time	6h30m	6h30m
Training Epochs	54	**12**
Training Time/Epoch	**7m15s**	37m30s
# of Training Variables	**1,385,198**	2,695,851
Inference Time (full dev)	**24s**	135s

FABIR - Model Overview







Conclusion

New architecture

- Competitive performance
- Five times faster at both learning and inference

Contributions to attention-based architectures

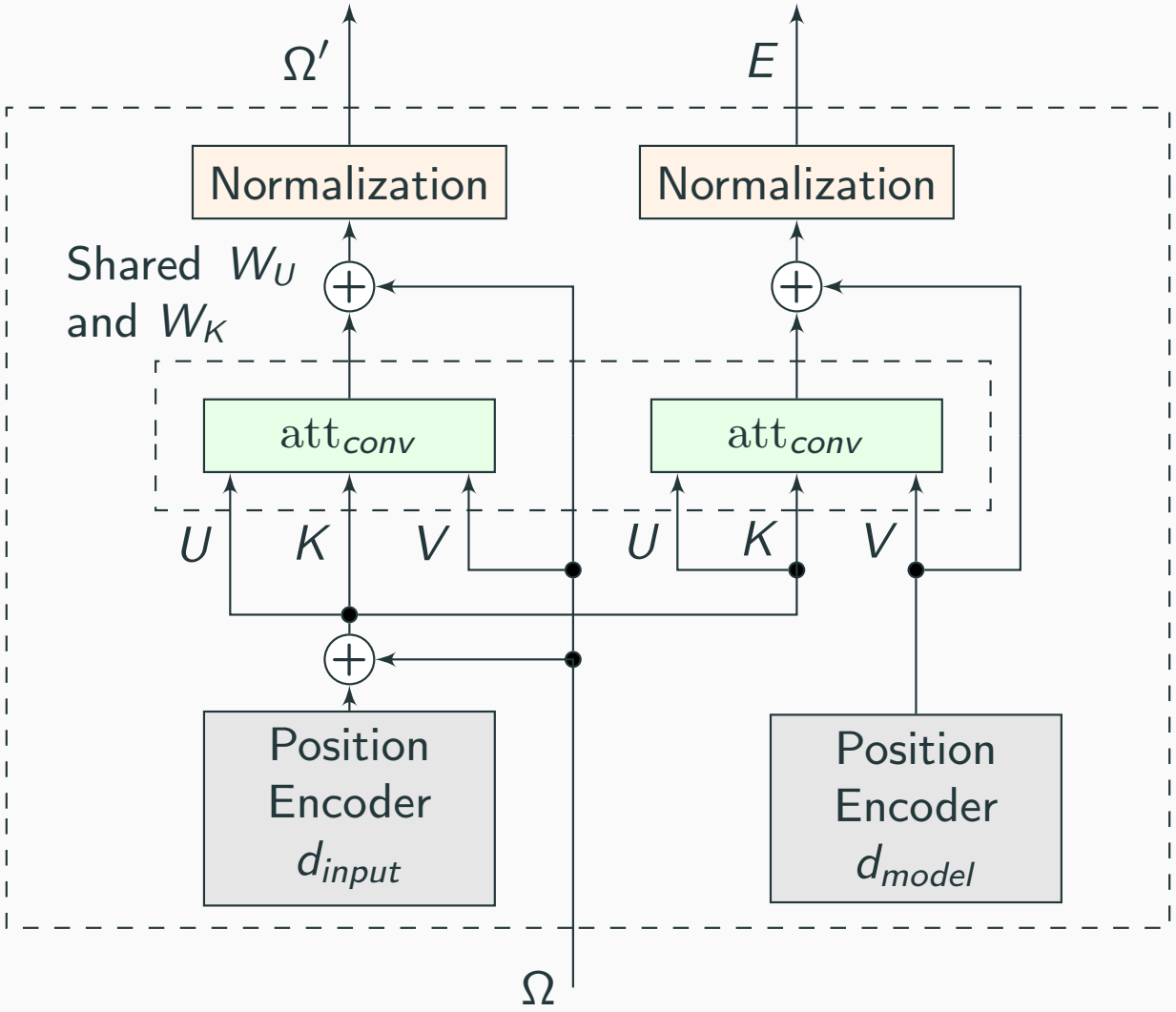
- Convolutional attention
- Column-wise cross-attention
- Reduction layer

Thank you for your attention!

Questions?

Questions

Layer Reduction - Decoupled Attention



Encoder

Construction

$$p_i = \left[sin\left(i * f_1\right), cos\left(i * f_1\right), ..., sin\left(i * f_{d/2}\right), cos\left(i * f_{d/2}\right)\right]^T$$

Properties

Given the ith and jth position encoder vector p_i and p_j, we define dot product between them as position attention $dot(p_i, p_j)$.

Commutativity $dot\left(p_i, p_j\right) = dot\left(p_j, p_i\right), \forall i, j \in \mathbb{N}$

Symmetry $dot\left(p_i, p_{i+k}\right) = dot\left(p_i, p_{i-k}\right), \forall i, k \in \mathbb{N}$

Linear Dependence $p_{i+1} = Rp_i, \forall i \in \mathbb{N}$

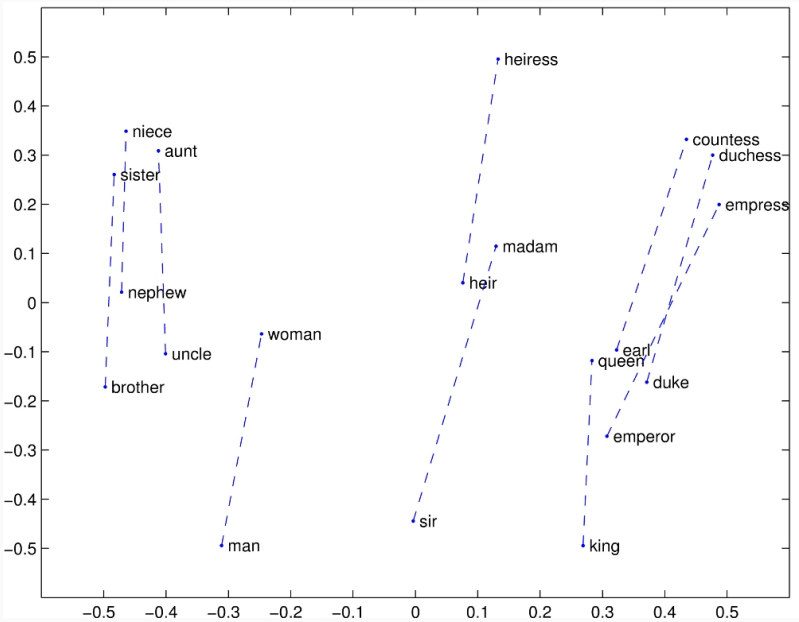
High Identity $dot\left(p_i, p_i\right) = \frac{d}{2} \geq dot\left(p_i, p_j\right), \forall i, j \in \mathbb{N}$

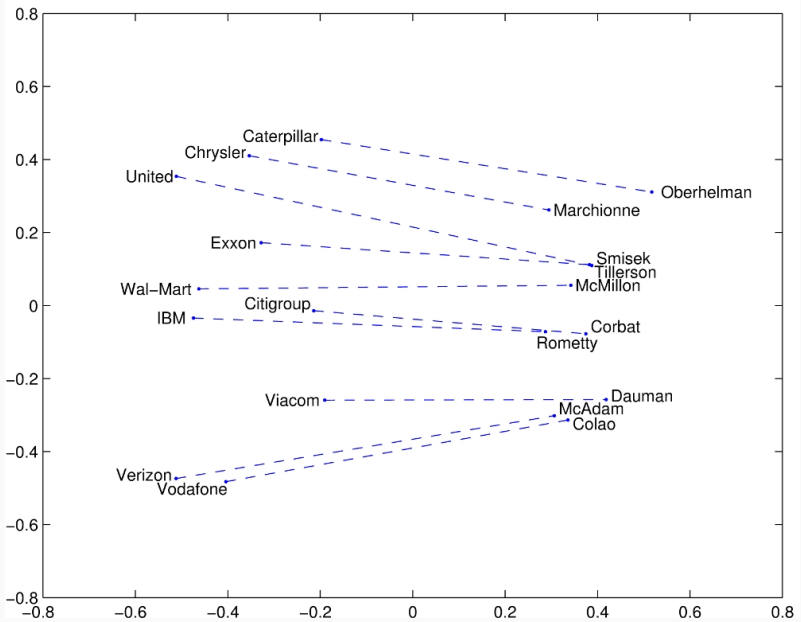
Word Embedding - GloVe

Characteristics

6 Billions tokens, vocabulary size 400k, word vector dimension 100, based on Wikipedia 2014 and Gigaword 5.

Linear Properties





Images Source: https://nlp.stanford.edu/projects/glove/

F1-Score

		Property Value	
		N	P
Classification Value	N	True Negative (TN)	False Negative (FN)
	P	False Positive (FP)	True Positive (TP)

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2}{\frac{1}{Pr} + \frac{1}{Re}} = 2\frac{Pr\ Re}{Pr + Re} \tag{3}$$

F1-Score - Example

		Property Value	
		N	P
Classification Value	N	True Negative (TN)	False Negative (FN)
	P	False Positive (FP)	True Positive (TP)

Question Who was Albert Einstein?

Paragraph Albert Einstein was a theoretical physicist.

Answer a theoretical physicist

Inferred Answer Einstein was a theoretical

Score Analysis Albert Einstein was a theoretical physicist.

TP = 2; FN = 1; FP = 2; TN = 2

Precision = 2/(2+2) = 1/2; Recall = 2/(2+1) = 2/3; F1 = 0.5714