

# Multi-step Retriever-Reader Interaction for Scalable Open-domain Question Answering

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Andrew McCallum  
ICLR 2019 (poster)

# Objectif : open-domain QA

Trouver la réponse à une *question* dans partir d'un *corpus volumineux* de textes

# Objectif : open-domain QA

Trouver la réponse à une *question* dans partir d'un *corpus volumineux* de textes

## Classiquement deux étapes

- retrieval
- reading

# Objectif : open-domain QA

Trouver la réponse à une *question* dans partir d'un *corpus volumineux* de textes

## Classiquement deux étapes

- retrieval
- reading

**Problème** : souvent le modèle de retrieval est basique (tf-idf) et n'apprend pas.

→ ex : DrQA (Chen et al. 2017) / DocQA (Clark & Gardner 2018)

# Solution : apprendre en même temps retriever et reader

R<sup>3</sup>: Reinforced Reader-Ranker for Open-Domain Question Answering (Wang et al. 2017)

## Optimisation par RL

→ le retriever apprend à sélectionner les paragraphes sur lesquels le reader performe bien

# Solution : apprendre en même temps retriever et reader

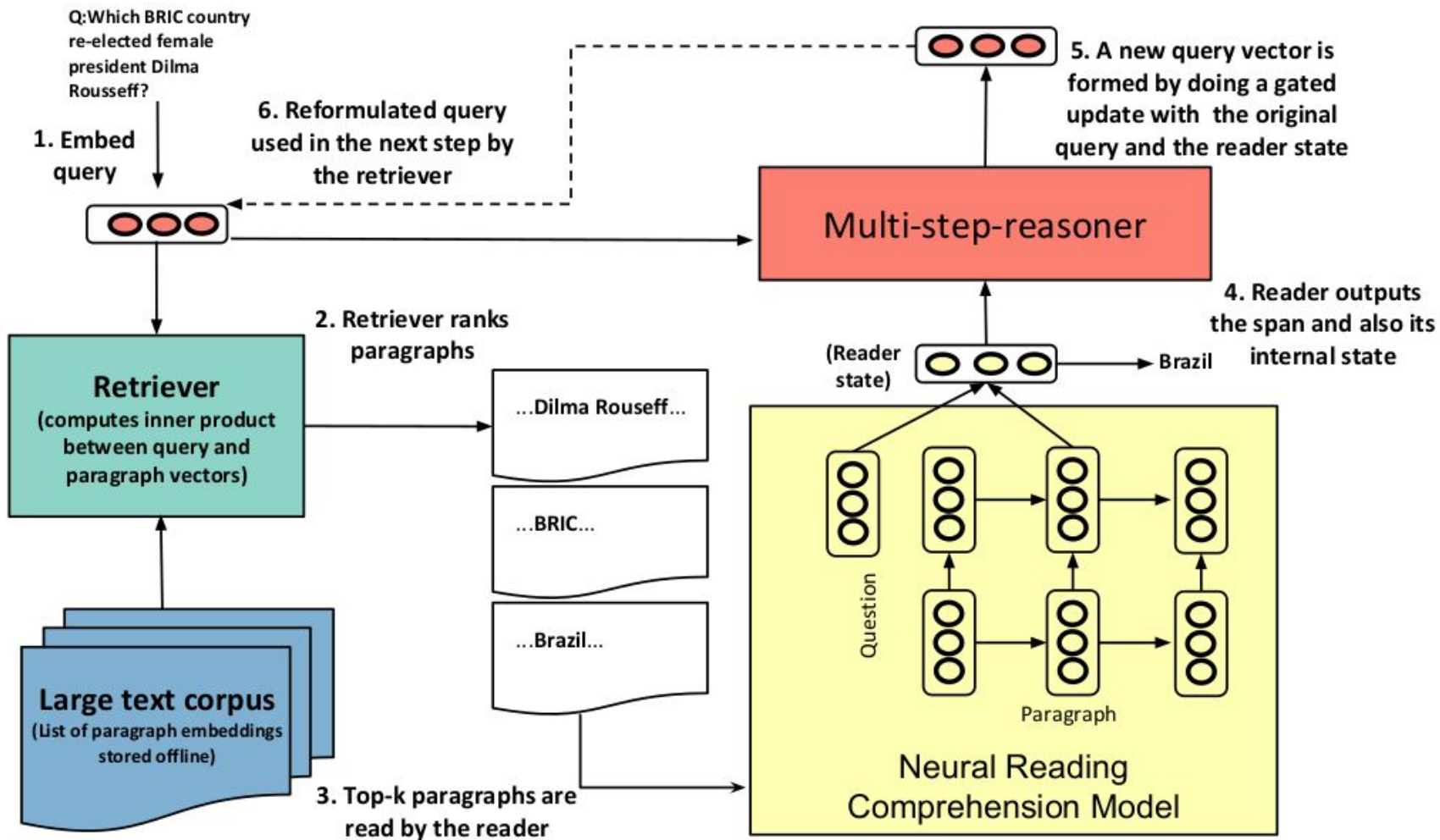
R<sup>3</sup>: Reinforced Reader-Ranker for Open-Domain Question Answering (Wang et al. 2017)

## Optimisation par RL

→ le retriever apprend à sélectionner les paragraphes sur lesquels le reader performe bien

**Problème** pour chaque question, une représentation différente du texte est calculée → **c'est long !**

**Contribution proposée** : calculer **une** représentation du texte, mais **reformuler** la question



# Retriever

Pour chaque paragraphe  $p$ , une *single-vector representation* est calculée par BiLSTM puis self-attention :

$$b_j = \frac{\exp(\mathbf{w} \cdot \mathbf{p}_j)}{\sum_{j'} \exp(\mathbf{w} \cdot \mathbf{p}_{j'})}$$

$$\mathbf{p} = W_s \sum_{j'} b_{j'} \cdot \mathbf{p}_{j'}$$

Ce calcul est fait **une seule fois**



# Retriever

Question  $q$

Ranking des paragraphes par similarité  $\text{score}(\mathbf{p}, \mathbf{q}) = \langle \mathbf{p}, \mathbf{q} \rangle$

Proposent d'accélérer ce calcul par *maximum inner product search (MIPS)* et nearest neighbor SGTtree

**Training** supervision distante

# Reader

Appliqué sur les top-k retrieved paragraphs

Modèle de reading comprehension classique (DrQA, BiDAF, etc.)

**Training** supervision classique des spans (NLL)

# Multi-step reasoner

Reformulation de la query grâce à l'état caché du Reader :

- $L$  représentation de la question par le Reader
- $m_j$  représentation de chaque token du passage

$$\alpha_j = \frac{\exp(\mathbf{m}_j \cdot \mathbf{L})}{\sum_{j'} \exp(\mathbf{m}_{j'} \cdot \mathbf{L})} \quad \mathbf{S} = \sum_j (\alpha_j \cdot \mathbf{m}_j)$$

L'état caché est passé dans un GRU pour obtenir la nouvelle question  $q_{\{t+1\}}$

$$\mathbf{q}_{t+1} = \text{GRU}(\mathbf{q}_t, \mathbf{S})$$

**Training** RL avec reward F1 score du span choisi

# Multi-step reasoner

La reformulation se fait dans l'**espace latent** !

→ différent de AQA : Ask the Right Questions: Active Question Reformulation with Reinforcement Learning, Buck et al. 2017

# Résultats

SOTA sur Quasar-T et TriviaQA-unfiltered + de bons résultats

Model	Quasar-T		SearchQA		TRIVIAQA-unfiltered		SQUAD-open	
	EM	F1	EM	F1	EM	F1	EM	F1
GA <a href="#">Dhingra et al. 2016</a>	26.4	26.4	-	-	-	-	-	-
BIDAF <a href="#">Seo et al. 2017</a>	25.9	28.5	28.6	34.6	-	-	-	-
AQA <a href="#">Buck et al. 2018</a>	-	-	40.5	47.4	-	-	-	-
R <sup>3</sup> <a href="#">Wang et al. 2018a</a>	35.3	41.7	49.0	55.3	47.3	53.7	29.1	37.5
DS-QA <a href="#">Lin et al. 2018</a>	37.27 <sup>2</sup> (42.2)	43.63 <sup>2</sup> (49.3)	<b>58.5</b>	<b>64.5</b>	48.7	56.3	28.7	36.6
MINIMAL <a href="#">Min et al. 2018</a>	-	-	-	-	-	-	<b>34.7</b>	<b>42.5</b>
Dr.QA baseline	36.87	45.49	51.36	58.24	48.0	52.13	27.1	-
multi-step-reasoner (Dr.QA)	<b>39.53</b>	<b>46.67</b>	55.01	61.61	<b>51.94</b>	<b>61.66</b>	31.93	39.22
multi-step-reasoner (BiDAF)	<b>40.63</b>	<b>46.97</b>	56.26	61.36	-	-	-	-

# Résultats

SOTA sur Quasar-T et TriviaQA-unfiltered + de bons résultats

## Retrieval

Model	P@1	P@3	P@5
R <sup>3</sup> (Wang et al., 2018a)	40.3	51.3	54.5
Our Retriever	35.7	49.6	56.3
+ multi-step (7 steps)	<b>42.9</b>	<b>55.5</b>	<b>59.3</b>

+ gros gains en complexity et rapidité