

# Multi-Hop Paragraph Retrieval for Open-Domain Question Answering

ACL2019論文紹介

**2018年9月11日**

武田浩一  
名古屋大学大学院情報学研究科  
価値創造研究センター

# Multi-Hop Paragraph Retrieval for Open-Domain Question Answering

Yair Feldman and Ran El-Yaniv (Technion – Israel Institute of Technology), pp.2296-2309

**Question:** The football manager who recruited David Beckham managed Manchester United during what time-frame?

**Context 1:** The 1995–96 season was Manchester United's fourth season in the Premier League ... Their triumph was made all the more remarkable by the fact that *Alex Ferguson* ... had drafted in young players like Nicky Butt, David Beckham, Paul Scholes and the Neville brothers, Gary and Phil.

**Context 2:** Sir *Alexander Chapman Ferguson*, CBE (born 31 December 1941) is a Scottish former football manager and player who managed Manchester United from 1986 to 2013. He is regarded by many players, managers and analysts to be one of the greatest and most successful managers of all time.

多ホップ質問の例：Context 1が最初の手掛かりとマッチ(first hop, 緑色)、Context 2で2番目の手掛かりとマッチ(second hop, 紫色)、2つの文脈に共通する固有表現(Alex Ferguson、青色)を回答

多ホップ・オープンドメイン質問応答では2つ以上の関連パラグラフを検索する(NTTの西田らの手法がHotpotQAでSoTAを達成した報告例がある)。

[Chen et al. 2017] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. "Reading wikipedia to answer open-domain questions," ACL 2017

[Clark et al. 2018] Christopher Clark and Matt Gardner. "Simple and effective multi-paragraph reading comprehension," ACL 2018

- 現在のQAデータセットは提示された文脈に質問の解答が存在する単一ホップ推論(読解と呼ばれる)課題が中心
- 実問題の解決にはマルチホップで、明示的な文脈提示のないオープンドメイン型の質問応答(ODQA)手法が必要
- 先行研究としては、Chenら(Chen et al. 2017)がWikipediaを利用し、tf-idfによる関連パラグラフ検索で単一ホップのODQAを提案
- 本論文では[Clark et al. 2018]の読解モジュールを参考に多ホップのODQA手法を提案し、HotpotQA(多ホップ)で2位、SQuAD-Open(単一ホップ)でSoTAを達成。

Setting	Method	Answer		Sup Fact		Joint	
		EM	F <sub>1</sub>	EM	F <sub>1</sub>	EM	F <sub>1</sub>
distractor	Baseline (Yang et al., 2018)	44.44	58.28	21.95	66.66	11.56	40.86
	Our Reader	51.56	65.32	44.54	75.27	28.68	54.08
full wiki	Baseline (Yang et al., 2018)	24.68	34.36	5.28	40.98	2.54	17.73
	TF-IDF + Reader	27.55	36.58	10.75	42.45	7.00	21.47
	MUPPET (sentence-level)	30.20	39.43	16.57	46.13	11.38	26.55
	MUPPET (paragraph-level)	<b>31.07</b>	<b>40.42</b>	<b>17.00</b>	<b>47.71</b>	<b>11.76</b>	<b>27.62</b>

Method	EM	F <sub>1</sub>
DrQA (Chen et al., 2017)	28.4	-
DrQA (Chen et al., 2017) (multitask)	29.8	-
R <sup>3</sup> (Wang et al., 2018a)	29.1	37.5
DS-QA (Lin et al., 2018)	28.7	36.6
Par. Ranker + Full Agg. (Lee et al., 2018)	30.2	-
Minimal (Min et al., 2018)	34.7	42.6
Multi-step (Das et al., 2019)	31.9	39.2
BERTserini (Yang et al., 2019)	38.6	46.1
TF-IDF + Reader	34.6	41.6
MUPPET (sentence-level)	<b>39.3</b>	<b>46.2</b>
MUPPET (paragraph-level)	35.6	42.5

## HotpotQA

(Wikipediaに対する多ホップの質問データセット、質問数112K)

## SQuAD-Open

(SQuADの質問のみとWikipedia全体を対にしたデータセット、質問数100K)

# Multi-Hop Paragraph Retrieval for Open-Domain Question Answering

本論文の多ホップ文脈検索+読解システム(MUPPET)の構成  $A=\phi(Q,KS)$ となる $\phi$ を学習

知識源  $KS$

$= \{P_1, \dots, P_{|KS|}\}$

パラグラフ  $P_i$

$= (p_1, \dots, p_{l_i})$

は  $l_i$  個の token 列

質問  $Q$

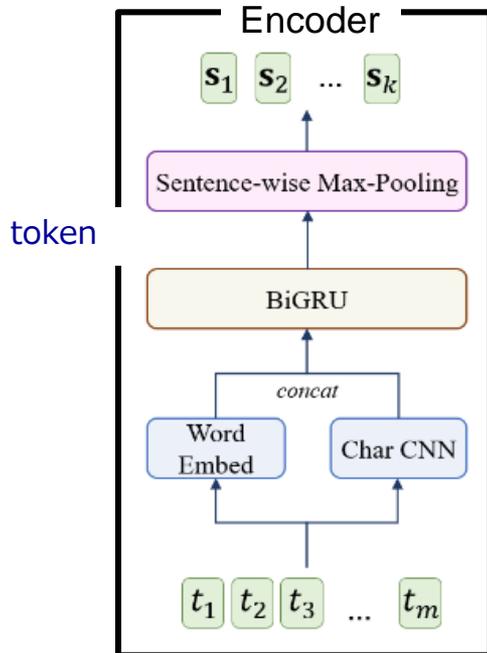
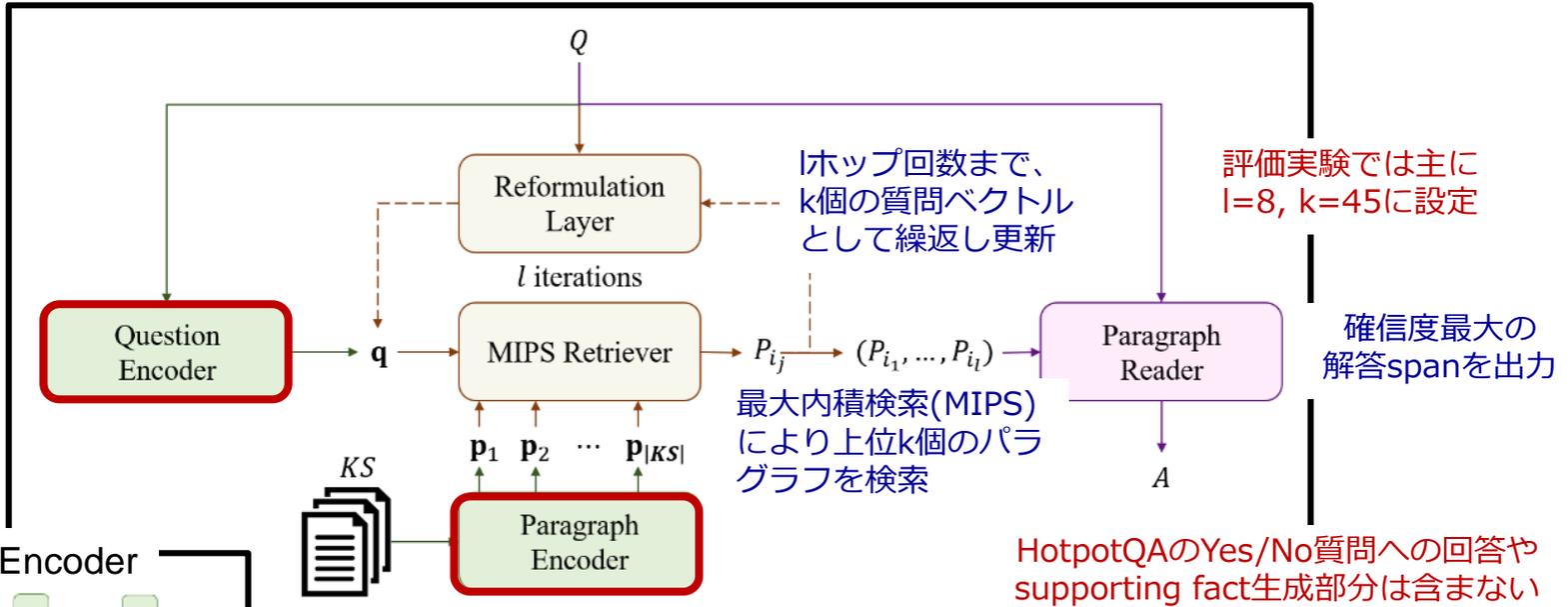
$= (q_1, \dots, q_m)$

は  $m$  個の token 列

解答  $A$

$= (a_1, \dots, a_n)$

は  $n$  個の token 列



tokenを順次BiGRUに読み込んで文脈表現 $c_1, \dots, c_m$ を生成、**文単位の**k個の埋込み表現に区分し、max-pooling-over-timeを計算 (質問ベクトルの更新にはtokenベクトルを使用)

$$(c_1, c_2, \dots, c_m) = \text{BiGRU}(t_1, t_2, \dots, t_m).$$

$$s_i = \max(c_{i_1}, c_{i_2}, \dots, c_{i_l}).$$

知識源や質問のtoken  $t_1, \dots, t_m$ は、事前学習されたGloVe単語埋め込み、および文字レベルの埋込みのCNNによるmax-poolingを連結(d次元)

$$t^c = \max(\text{CNN}(t_1^c, t_2^c, \dots, t_{l_t}^c))$$

$$t = [t^w; t^c]. \text{ (実験ではELMoも結合)}$$

# Multi-Hop Paragraph Retrieval for Open-Domain Question Answering

本論文の多ホップ文脈検索+読解システム(MUPPET)の構成  $A=\phi(Q,KS)$ となる $\phi$ を学習

知識源  $KS$

$= \{P_1, \dots, P_{|KS|}\}$

パラグラフ  $P_i$

$= (p_1, \dots, p_{l_i})$

は  $l_i$  個の token 列

質問  $Q$

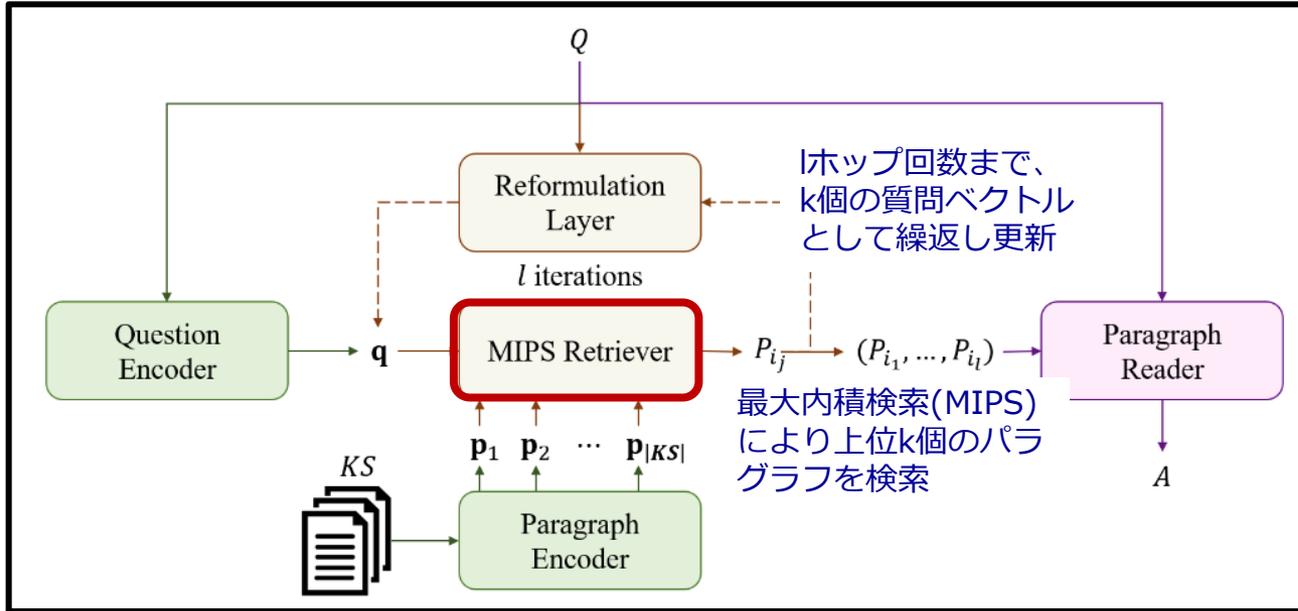
$= (q_1, \dots, q_m)$

は  $m$  個の token 列

解答  $A$

$= (a_1, \dots, a_n)$

は  $n$  個の token 列



確信度最大の  
解答 span を出力

質問  $Q$  に関連する知識源のパラグラフ  $P = (s_1, \dots, s_k)$  は以下の  
ように計算し、 $rel(Q, P)$  値の上位  $k$  個のパラグラフを保持

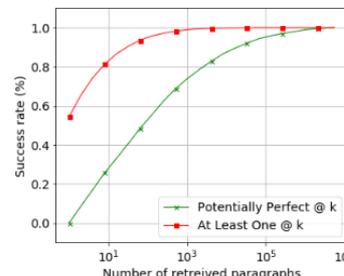
$$rel(Q, P) = \max_{i=1, \dots, k} \sigma \left( \begin{bmatrix} s_i \\ s_i \odot \mathbf{q} \\ s_i \cdot \mathbf{q} \\ \mathbf{q} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ w_3 \\ \mathbf{w}_4 \end{bmatrix} + b \right),$$

あるパラグラフ中に質問  $Q$  と高い関連度  
を持つ文  $s_i$  が **1つ含まれていればよい**

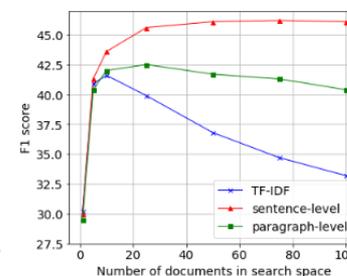
$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^N y_i \log (rel(Q_i, P_i)) + (1 - y_i) \log (1 - rel(Q_i, P_i)),$$

MIPSの計算対象となるパラグラフ数をtf-idfベースの  
パラグラフ検索器により事前に制限する(横軸)

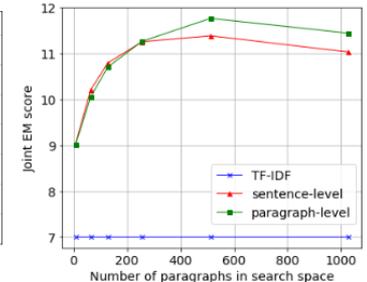
最低 1 つの正解を含むパラグラフを検索できるが、複数は難しい  
さらに検索パラグラフ数を増やしても正答率は改善しない



(a) TF-IDF retrieval results



(b) SQuAD-Open



(c) HotpotQA

# Multi-Hop Paragraph Retrieval for Open-Domain Question Answering

本論文の多ホップ文脈検索+読解システム(MUPPET)の構成  $A=\varphi(Q,KS)$ となる $\varphi$ を学習

知識源 **KS**

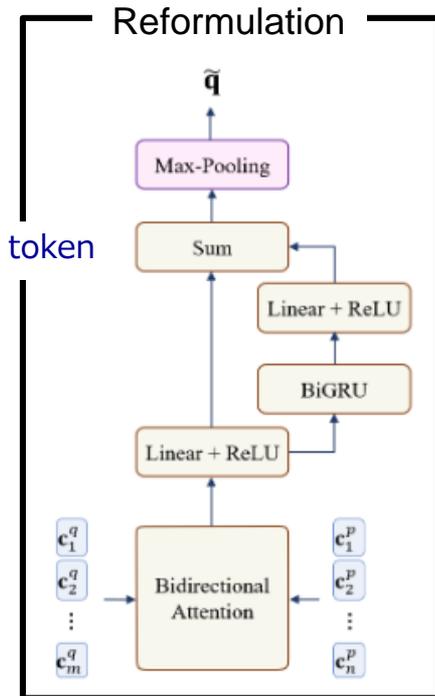
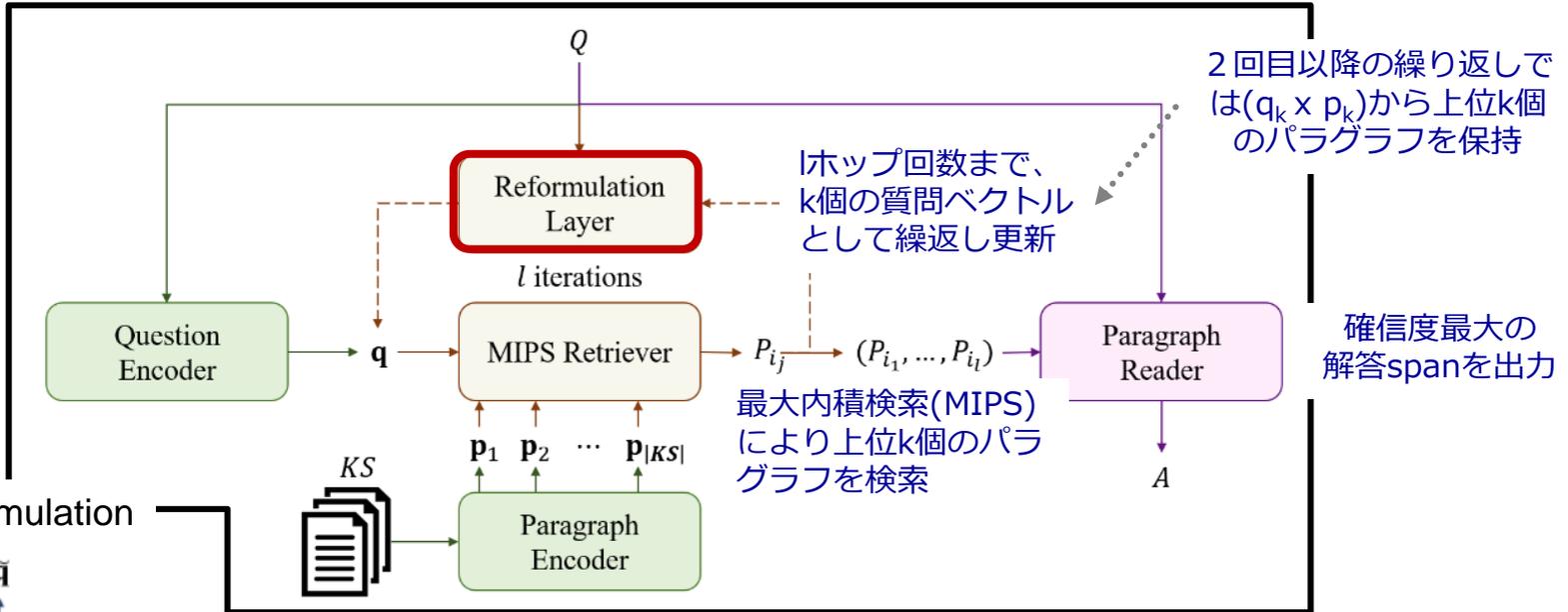
$= \{P_1, \dots, P_{|KS|}\}$   
 パラグラフ  $P_i$   
 $= (p_1, \dots, p_{l_i})$   
 は  $l_i$  個の token 列

質問 **Q**

$= (q_1, \dots, q_m)$   
 は  $m$  個の token 列

解答 **A**

$= (a_1, \dots, a_n)$   
 は  $n$  個の token 列



質問QとパラグラフPのtoken単位のアテンションを計算([Clark et al. 2018]と同じ)

$$a_{ij} = \mathbf{w}_1^a \cdot \mathbf{c}_i^q + \mathbf{w}_2^a \cdot \mathbf{c}_j^p + \mathbf{w}_3^a \cdot (\mathbf{c}_i^q \odot \mathbf{c}_j^p),$$

パラグラフPのtokenのattended vector  $\mathbf{a}_i$

query-to-context vector  $\mathbf{a}^p$

$$\alpha_{ij} = \frac{e^{a_{ij}}}{\sum_{j=1}^{n_p} e^{a_{ij}}}, \quad \mathbf{a}_i = \sum_{j=1}^{n_p} \alpha_{ij} \mathbf{c}_j^p.$$

$$m_i = \max_{1 \leq j \leq n_p} a_{ij}, \quad \beta_i = \frac{e^{m_i}}{\sum_{i=1}^{n_q} e^{m_i}}$$

$$\mathbf{a}^p = \sum_{i=1}^{n_q} \beta_i \mathbf{c}_i^q.$$

$$\mathbf{c}_i^q, \mathbf{a}_i, \mathbf{c}_i^q \odot \mathbf{a}_i \text{ and } \mathbf{a}^p \odot \mathbf{a}_i$$

を連結、ReLU活性化と残差接続で最終的な質問ベクトル $\mathbf{q}$ を生成

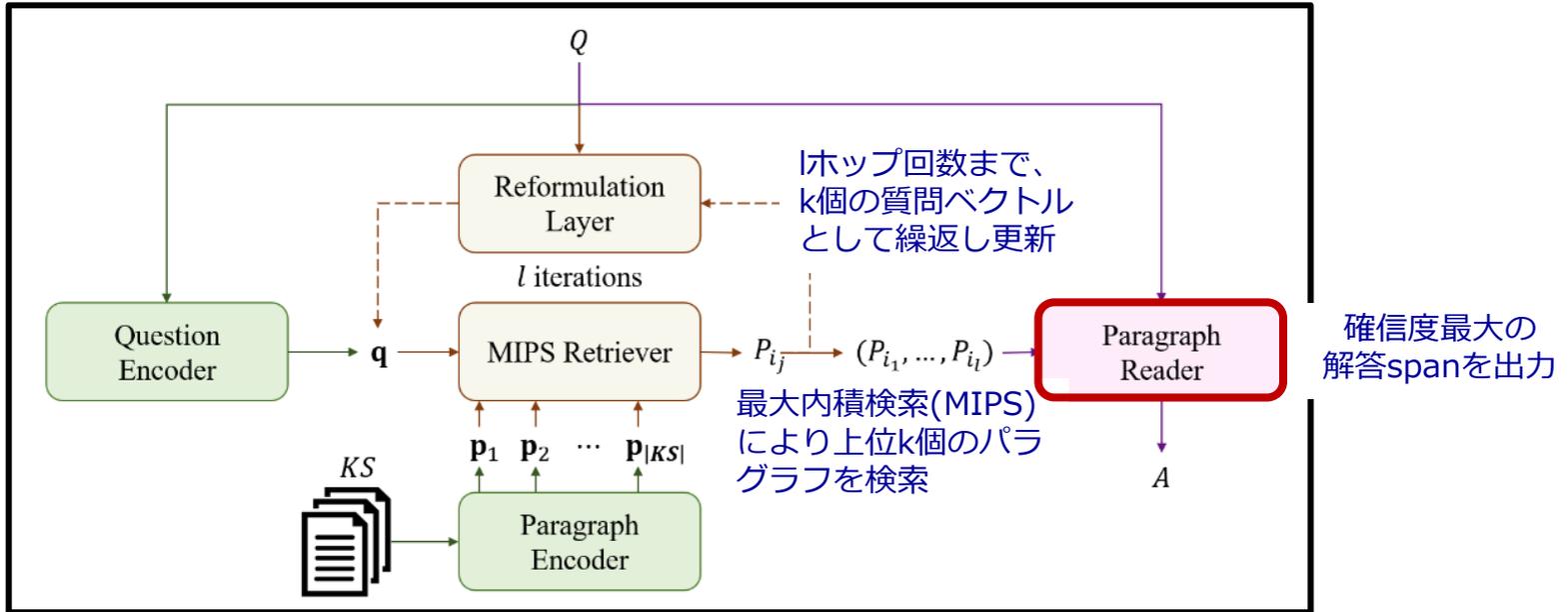
# Multi-Hop Paragraph Retrieval for Open-Domain Question Answering

本論文の多ホップ文脈検索+読解システム(MUPPET)の構成  $A=\phi(Q,KS)$ となる $\phi$ を学習

知識源  $KS$   
 $= \{P_1, \dots, P_{|KS|}\}$   
 パラグラフ  $P_i$   
 $= (p_1, \dots, p_{l_i})$   
 は  $l_i$  個の token 列

質問  $Q$   
 $= (q_1, \dots, q_m)$   
 は  $m$  個の token 列

解答  $A$   
 $= (a_1, \dots, a_n)$   
 は  $n$  個の token 列



Paragraph Readerは、質問 $Q$ と対応付けられたパラグラフ集合 $P^Q$ 中の各 $P_j$ について下記のnegative log-likelihood式を最適化する。ここで $s_{ij}$ は、 $P_j$ に含まれるtoken  $t_i$ が解答のspan開始位置(start)となる予測スコアであり、 $A_j$ は $P_j$ に含まれる解答のtoken集合である

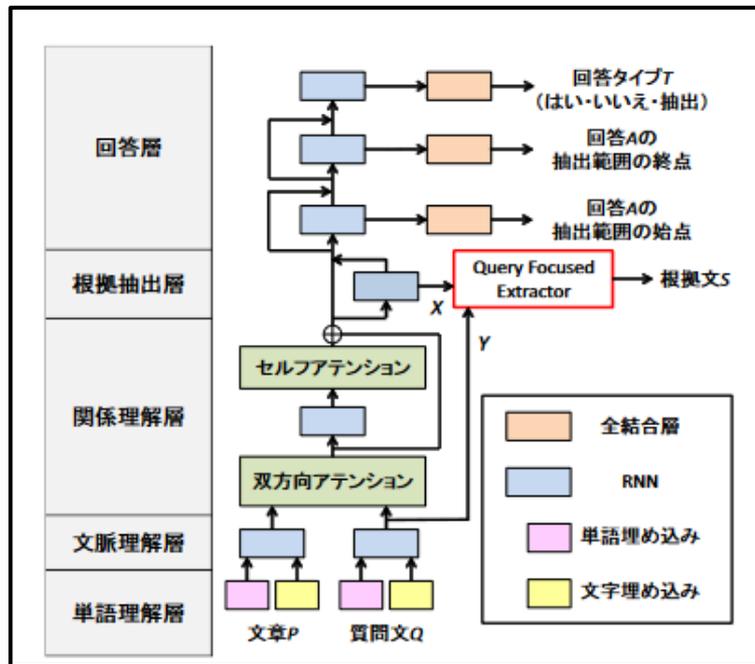
$$\mathcal{L}_{start} = -\log \left( \frac{\sum_{j \in P^Q} \sum_{k \in A_j} e^{s_{kj}}}{\sum_{j \in P^Q} \sum_{i=1}^{n_j} e^{s_{ij}}} \right),$$

すべての候補パラグラフについて解答spanの確信度を正規化(shared-normalization) [Clark et al. 2019]

学習時には解答spanの終了位置(end)に関する式との和による下記の目的関数を最小化する

$$\mathcal{L}_{span} = \mathcal{L}_{start} + \mathcal{L}_{end}.$$

# 考察



- 読解の基本的アーキテクチャは従来手法を継承
- 新たなチャレンジとして、多ホップのQAタスクに注目
- 本論文のFeldmanらの手法では
  - 文単位に質問文との関連度を計算し、知識源から候補パラグラフを検索する
  - 候補パラグラフに基づいて元の検索ベクトルを更新し、L回の繰り返しによって多ホップQAを実現することで検索精度を向上できた(ただしHotpotQAではパラグラフベクトルのほうが高精度だった)
- 既に本論文の精度を大きく上回る結果が出ている

HotpotQAの質問例(左)とLeaderboard (fullwiki) 2019/09/04  
 すでに本論文の回答精度を大きく上回るシステムが報告されている

西田らの読解システム, “抽出型要約との同時学習による回答根拠を提示可能な機械読解”, 言語処理学会 第25回年次大会, 2019年3月

質問応答と抽出型複数文要約(根拠文生成)のマルチタスク学習によりHotpotQAでSoTAを達成

**Paragraph A, Return to Olympus:**  
 [1] *Return to Olympus* is the only album by the alternative rock band Malfunkshun. [2] It was released after the band had broken up and after lead singer Andrew Wood (later of Mother Love Bone) had died of a drug overdose in 1990. [3] Stone Gossard, of Pearl Jam, had compiled the songs and released the album on his label, Loosegroove Records.

**Paragraph B, Mother Love Bone:**  
 [4] *Mother Love Bone* was an American rock band that formed in Seattle, Washington in 1987. [5] The band was active from 1987 to 1990. [6] Frontman Andrew Wood's personality and compositions helped to catapult the group to the top of the burgeoning late 1980s/early 1990s Seattle music scene. [7] Wood died only days before the scheduled release of the band's debut album, "Apple", thus ending the group's hopes of success. [8] The album was finally released a few months later.

**Q:** What was the former band of the member of Mother Love Bone who died just before the release of "Apple"?  
**A:** Malfunkshun  
**Supporting facts:** 1, 2, 4, 6, 7

Leaderboard (Fullwiki Setting)

In the fullwiki setting, a question-answering system must find the answer to a question in the scope of the entire Wikipedia. Similar to in the distractor setting, systems are evaluated on the accuracy of their answers (Ans) and the quality of the supporting facts they use to justify them (Sup).

Model	Code	Ans		Sup		Joint	
		EM	F <sub>1</sub>	EM	F <sub>1</sub>	EM	F <sub>1</sub>
1 BERT pip. (single model) <small>Anonymous</small>		45.32	57.34	38.67	70.83	25.14	47.60
2 Entity-centric BERT Pipeline (single model) <small>Anonymous</small>		41.82	53.09	26.26	57.29	17.01	39.18
3 GoldEn Retriever (single model) <small>Anonymous</small>		37.92	48.58	30.69	64.24	18.04	39.13