DyREx: Dynamic Query Representation for Extractive Question Answering

Urchade Zaratiana[1,2]*, **Niama El Khbir**[2], **Dennis Núñez**[2], **Pierre Holat**[1,2], **Nadi Tomeh**[2], **Thierry Charnois**[2]
[1]FI Group, [2]LIPN, Université Sorbonne Paris Nord - CNRS UMR 7030

Abstract

Extractive question answering (ExQA) is an essential task for Natural Language Processing. The dominant approach to ExQA is one that represents the input sequence tokens (question and passage) with a pre-trained transformer, then uses two learned query vectors to compute distributions over the start and end answer span positions. These query vectors lack the context of the inputs, which can be a bottleneck for the model performance. To address this problem, we propose *DyREx*, a generalization of the *vanilla* approach where we dynamically compute query vectors given the input, using an attention mechanism through transformer layers. Empirical observations demonstrate that our approach consistently improves the performance over the standard one. The code and accompanying files for running the experiments are available at `https://github.com/urchade/DyReX`.

1 Introduction

Extractive question answering is a challenging task where the goal is to extract the answer span given a question and a passage as inputs [Rajpurkar et al., 2016, Kwiatkowski et al., 2019]. The prevailing approach achieves Extractive question answerin (ExQA) by firstly producing a contextualized representation of the input, which is a concatenation of the question and the passage, using a pre-trained transformer model. Two learned query vectors are then used to compute a probability distribution over this input sequence representation to produce the start and end positions of the answer span. This approach has demonstrated very strong and hard-to-beat results, which makes it the de facto approach to extractive QA [Devlin et al., 2019, Liu et al., 2019, Joshi et al., 2020].

However, despite their high performance, we argue that these methods remain suboptimal since the query vectors used to compute the start and end distributions are static, i.e., they are independent of the input sequence, which can be a bottleneck for improving the performance of the model. Hence, we propose to extend this by allowing the queries to dynamically aggregate information from the input sequence to better answer the question. Our method, *DyREx*, iteratively refines the initial query representations, allowing them to aggregate information from the source sequence through attention mechanism [Bahdanau et al., 2015, Vaswani et al., 2017]. More specifically, we make use of an L-layers transformer decoder architecture, which allows (1) interaction between the queries through self-attention to model the interdependence between the start and end of the answer span, and allows (2) interaction between queries and the input sequence through cross-attention, which specializes the queries to a specific input question and passage, giving more flexibility than a static representation.

We conduct extensive experiments on several extractive Question Answering benchmarks, including SQuad [Rajpurkar et al., 2016] and MRQA datasets [Fisch et al., 2019]. Experimental results demonstrate that our approach consistently improves the performance over the standard approach.

[*] Correspondence to: `zaratiana@lipn.fr`

NeurIPS 2022 2nd Workshop on Efficient Natural Language and Speech Processing, New Orleans.

2 Model

2.1 Background: *Vanilla* QA model

We describe here the mainstream approach to extractive Question Answering tasks. In all the following, we call it the ExQA *vanilla* approach. It is typically performed by feeding the input text sequence $\{x_i\}_{i=1}^N$ (the concatenation of the question Q and the passage D containing the answer) into a pre-trained language model such as BERT [Devlin et al., 2019], producing contextualized token representations $\{\mathbf{h}_i\}_{i=1}^N \in \mathbb{R}^d$, d being the embedding dimension of the model. Then, to compute the probability of the *start* and *end* positions of the answer span, the following estimators are used:

$$p(\textit{start} = i|Q, D) = \frac{\exp(\mathbf{q}_s^T \mathbf{h}_i)}{\sum_{i'=1}^N \exp(\mathbf{q}_s^T \mathbf{h}_{i'})} \qquad p(\textit{end} = j|Q, D) = \frac{\exp(\mathbf{q}_e^T \mathbf{h}_j)}{\sum_{j'=1}^N \exp(\mathbf{q}_e^T \mathbf{h}_{j'})} \tag{1}$$

Where $\mathbf{q}_s$ and $\mathbf{q}_e \in \mathbb{R}^d$ are respectively the *start* and *end* queries, randomly initialized and updated during model learning. The training objective is to minimize the sum of the negative log-likelihood of the correct start and end positions $(\hat{i}, \hat{j})$:

$$\mathcal{L} = -\log p(\textit{start} = \hat{i}|Q, D) - \log p(\textit{end} = \hat{j}|Q, D) \tag{2}$$

This approach was first proposed by Devlin et al. [2019], and is now used by most of the work on transformer-based extractive question answering [Liu et al., 2019, Joshi et al., 2020, Shi et al., 2022].

2.2 Our model: *DyREx*

The learned query vectors $\mathbf{q}_s$ and $\mathbf{q}_e$ in the vanilla approach are shared among all sentences and are context insensitive. We presume that using such static queries is a constraining factor for performance improvement, so we propose to extend this approach by allowing the queries to dynamically aggregate information from the input sequence to allow the model to better adapt to the context.

In our model, the initial start and end query representations $\mathbf{q}_s^0$ and $\mathbf{q}_e^0$ are concatenated and fed to an L-layers transformer decoder [Vaswani et al., 2017] to obtain dynamic representations $\mathbf{q}_s^L$ and $\mathbf{q}_e^L$:

$$\mathbf{Q}^L = \texttt{Trans\_Dec}_L(\mathbf{Q}^0, \mathbf{H}) \tag{3}$$

with $\mathbf{Q}^i = [\mathbf{q}_e^i, \mathbf{q}_s^i]$ the concactenated queries at layer i and $\mathbf{H} = [\mathbf{h}_0, \mathbf{h}_1, ..., \mathbf{h}_N]$ the concatenated token representations, and $\texttt{Trans\_Dec}_L$ being an L-layers transformer decoder.

More specifically, the i-th transformer layer consists of a bi-directional self-attention module $\texttt{self-att}_i$ applied between the queries to model the interdependence between the start and the end positions of the answer, a cross-attention $\texttt{cross-att}_i$ which updates the query representations by aggregating information from the input sequence embeddings, and finally a two-layer point-wise feedforward network $\texttt{FFN}_i$ with GeLU activation [Hendrycks and Gimpel, 2016]:

$$\begin{aligned} \tilde{\mathbf{Q}}^i &= \texttt{self-att}_i(\texttt{Q} = \mathbf{Q}^i, \texttt{K} = \mathbf{Q}^i, \texttt{V} = \mathbf{Q}^i) \\ \hat{\mathbf{Q}}^i &= \texttt{cross-att}_i(\texttt{Q} = \tilde{\mathbf{Q}}^i, \texttt{K} = \mathbf{H}, \texttt{V} = \mathbf{H}) \\ \mathbf{Q}^{i+1} &= \texttt{FNN}_i(\hat{\mathbf{Q}}^i) \end{aligned} \tag{4}$$

Furthermore, an `Add-Norm` (skip connection [He et al., 2016] + layer normalization [Ba et al., 2016]) is inserted after each of the components as in Vaswani et al. [2017], but we do not show it here for better readability. Moreover, both `self-att` and `cross-att` are multi-head scaled dot-product attention from Vaswani et al. [2017], and the embedding dimension and the number of attention heads of the decoder layers are the same as for the token representation layer.

Finally, to compute the start and the end answer position probabilities, we use the same estimator as the *vanilla* model in equation 1, substituting $\mathbf{q}_s$ and $\mathbf{q}_e$ by $\mathbf{q}_s^L$ and $\mathbf{q}_e^L$ respectively. Note that the *vanilla* model is a particular case of our model with a number of decoder layers $L = 0$.

Train size	Models	Datasets				
		SQuAD	HotpotQA	TriviaQA	NewsQA	NaturalQs
256	*Vanilla*	65.74	53.23	28.49	35.80	41.87
	DyREx	**70.75**	**57.08**	**41.66**	**43.77**	**45.57**
512	*Vanilla*	69.72	58.70	45.39	43.24	48.36
	DyREx	**77.19**	**61.15**	**52.57**	**49.20**	**55.37**
1024	*Vanilla*	74.01	62.54	51.87	50.61	53.42
	DyREx	**79.42**	**67.95**	**57.59**	**54.26**	**61.67**
Full	*Vanilla*	90.64	79.95	76.31	68.02	77.79
	DyREx	**91.01**	**80.55**	**77.37**	**68.53**	**78.58**

Table 1: **Main results**. We reported experimental results for different sizes of training datasets using SpanBERT [Joshi et al., 2020] for token representation.

3 Experimental setup

Datasets We evaluate our model on English Machine Reading Comprehension datasets including SQuAD [Rajpurkar et al., 2016], HotpotQA [Yang et al., 2018], TriviaQA [Joshi et al., 2017], NewsQA [Trischler et al., 2016], and Natural Questions [Kwiatkowski et al., 2019]. We preprocess the datasets using standard approaches, then we evaluate our model using the F1 and Exact Match (EM) metrics implemented on the MRQA Github repository.

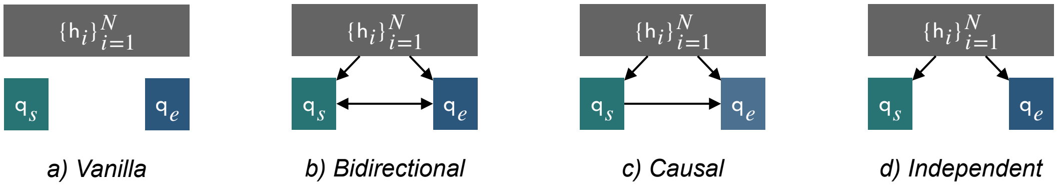
Hyperparameters We adopt a similar experimental setup of Ram et al. [2021] and Joshi et al. [2020]. For all experiments, we use SpanBERT for token representations. We fine-tune our model using hyperparameters from the default configuration of the HuggingFace Transformers library [Wolf et al., 2020]. We use Adam optimizer with a learning rate of 3.10^{-5}, where a warm-up stage is set for the first 10% of the steps, then decay the learning rate linearly for the rest of the training steps. We employed a batch size of 12, and we trained for a maximum of 5 epochs for full-sized datasets, and for few-shot models, we train the models up to either 2500 steps or 10 epochs. For *DyREx*, unless specified, we employed a 3-layers transformer decoder for the query representations, where each attention layer has 8 heads and the same embedding dimension as the contextualized token embeddings. To run the *vanilla* ExQA model, we borrowed the code from [Ram et al., 2021] GitHub repository. We trained all the models in a server with V100 GPUs.

4 Results

Table 1 presents the obtained results for the different datasets, employing SpanBERT for token representations. Our approach is very effective in few-shot settings compared to the *vanilla* approach. For instance, our model exceeds the *vanilla* approach by 7.47% F1 score on SQuAD (512 samples), 5.41% on HotpotQA (1024 samples), 12.76% on TriviaQA (256 samples), and 7.97% on NewsQA (256 samples). These empirical results show the effectiveness of our approach and demonstrate that contextual queries are important when only a few data are available.

5 Component analysis for *DyREx*

We perform a more in-depth analysis to inspect the contribution of the separate components of our *DyREx* architecture. We study the influence of the number of layers of the decoder. We also examine various masking strategies of self-attention of *DyREx* decoder: *Bidirectional* which allows full attention between queries, *Causal* with causal masking (the start query cannot attend the end query), and *Independent* which fully masks the attention between the queries. For all the studies, we employed the SQuAD dataset, a SpanBERT representation, and averaged results across three different seeds.



a) Vanilla b) Bidirectional c) Causal d) Independent

Figure 1: **Queries and tokens interactions**. This figure shows the different interactions (or attention) between the queries (q_s and q_e) and the input tokens ($\{\mathbf{h}_i\}_{i=1}^N$). a) *Vanilla*: No interaction, either token-query or query-query (i.e., static queries). b) *Bidirectional*: dynamic queries with bidirectional query interaction. c) *Causal*: dynamic queries with causal relation (q_s influence q_e). d) *Independent*: queries are independent of each other but remain dynamic.

# Layers	F1	EM
0	90.64 ± 0.10	83.08 ± 0.09
1	91.08 ± 0.05	83.56 ± 0.31
2	90.92 ± 0.12	83.50 ± 0.17
3	91.01 ± 0.03	83.35 ± 0.07
4	$\mathbf{91.17 \pm 0.05}$	83.57 ± 0.14
5	91.08 ± 0.07	$\mathbf{83.64 \pm 0.17}$

Table 2: Results for different number of transformer decoder layers.

	F1	EM
Static		
a) *Vanilla*	90.64 ± 0.10	83.08 ± 0.09
Dynamic		
b) *Bidirectional*	$\mathbf{91.01 \pm 0.03}$	83.35 ± 0.07
c) *Causal*	91.00 ± 0.05	$\mathbf{83.47 \pm 0.17}$
d) *Independent*	90.87 ± 0.09	83.14 ± 0.44

Table 3: Results for different interactions/attentions between the queries.

Influence of # of layers Table 2 shows that performance is generally better with more layers. The best performance is attained when using 4 or 5 layers, but a lower number of layers can also obtain competitive results. However, the results are much weaker without a decoder layer, i.e., without a static query representation.

Type of self-attention Table 3 shows the obtained results using the various attention or masking strategies for the decoder. We see that fully masking (d) attention between queries provides the weakest result, while *Causal* and *Bidirectional* attention perform similarly. This shows that the interaction between queries is important.

6 Related Work

Early ExQA models based on deep learning, such as BidAF [Seo et al., 2017], Match-LSTM [Hu et al., 2017], QaNet [Yu et al., 2018] and others, were highly specialized and heavily engineered. The arrival of BERT and pre-trained language models completely transformed the domain by introducing a simple yet effective approach: the *Vanilla* ExQA (section 2.1) [Devlin et al., 2019]. Since its introduction, this has been the dominant approach to extractive QA [Liu et al., 2019, Joshi et al., 2020]. For instance, [Fajcik et al., 2021] proposes to model the joint probability of the spans, instead of modeling the probability of a span's start and end positions independently, but it does not outperform the *Vanilla* approach. In this work, we extend the vanilla model not by modifying the objective function but by learning a richer representation of the query parameters.

7 Conclusion

In this paper, we propose *DyREx*, a method to dynamically compute query representations to calculate the start and end positions of answer spans in extractive question answering. Our approach consistently outperforms the dominant approach on a wide range of QA datasets, and the gain is even more significant in a few-shot scenario. In future work, it would be interesting to adapt *DyREx* for multi-span extraction tasks such as Named Entity Recognition and Keyphrase Extraction.

Acknowledgments

This work is partially supported by a public grant overseen by the French National Research Agency (ANR) as part of the program Investissements d'Avenir (ANR-10-LABX-0083). This work was granted access to the HPC/AI resources of [CINES/IDRIS/TGCC] under the allocation 20XX-AD011013682 made by GENCI.

References

Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

Martin Fajcik, Josef Jon, Santosh Kesiraju, and Pavel Smrz. Rethinking the objectives of extractive question answering. *ArXiv*, abs/2008.12804, 2021.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5801. URL `https://aclanthology.org/D19-5801`.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *ArXiv*, abs/1606.08415, 2016.

Annie Hu, Cindy Wang, and Brandon Yang. Question answering using match-lstm and answer pointer. 2017.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017. URL `https://arxiv.org/abs/1705.03551`.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 01 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00300. URL `https://doi.org/10.1162/tacl_a_00300`.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 08 2019. ISSN 2307-387X. doi: 10.1162/tacl_a_00276. URL `https://doi.org/10.1162/tacl_a_00276`.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL `https://arxiv.org/abs/1907.11692`.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016. URL `https://arxiv.org/abs/1606.05250`.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. Few-shot question answering by pretraining span selection. 2021. doi: 10.48550/ARXIV.2101.00438. URL https://arxiv.org/abs/2101.00438.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603, 2017.

Han Shi, JIAHUI GAO, Hang Xu, Xiaodan Liang, Zhenguo Li, Lingpeng Kong, Stephen M. S. Lee, and James Kwok. Revisiting over-smoothing in BERT from the perspective of graph. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=dUV91uaXm3.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset, 2016. URL https://arxiv.org/abs/1611.09830.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Transformers: State-of-the-art natural language processing. In *EMNLP*, 2020.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *CoRR*, abs/1809.09600, 2018. URL http://arxiv.org/abs/1809.09600.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension, 2018. URL https://arxiv.org/abs/1804.09541.