Generalizing Question Answering System with Pre-trained Language Model Fine-tuning

Dan Su*, **Yan Xu***, **Genta Indra Winata**, **Peng Xu**,
Hyeondey Kim, **Zihan Liu**, **Pascale Fung**
Center for Artificial Intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
{dsu, yxucb, giwinata, pxuab}@connect.ust.hk,
{hdkimaa, zliucr}@connect.ust.hk, pascale@ece.ust.hk

Abstract

With a large number of datasets being released and new techniques being proposed, Question answering (QA) systems have witnessed great breakthroughs in reading comprehension (RC) tasks. However, most existing methods focus on improving in-domain performance, leaving open the research question of how these models and techniques can generalize to out-of-domain and unseen RC tasks. To enhance the generalization ability, we propose a multi-task learning framework that learns the shared representation across different tasks. Our model is built on top of a large pre-trained language model, such as XLNet, and then fine-tuned on multiple RC datasets. Experimental results show the effectiveness of our methods, with an average Exact Match score of 56.59 and an average F1 score of 68.98, which significantly improves the BERT-Large baseline by 8.39 and 7.22, respectively.

1 Introduction

Reading comprehension (RC) is a fundamental human skills needed to answer questions that require knowledge of the world and understanding of natural language. This task is essential for intelligent dialogue systems to quickly respond in a search engine or a product recommendation system. Recently, we have witnessed several breakthroughs in question answering (QA) systems, such as bidirectional attention flow (BiDAF) (Seo et al., 2017), the attention over attention mechanism (AoA) (Cui et al., 2017), and a multi-hop architecture using gated-attention readers (Dhingra et al., 2017).

A large number of QA datasets have been proposed in recent years for single-hop and multi-hop reasoning applications (Rajpurkar et al., 2016; Lai et al., 2017; Saha et al., 2018; Trischler et al., 2017; Joshi et al., 2017). However, each QA dataset is built for a particular domain and focus (Talmor and Berant, 2019). Dataset passages cover different topics, such as movies (Saha et al., 2018), news (Trischler et al., 2017), and biomedicine (Tsatsaronis et al., 2012). Also, the styles of questions (e.g., entity-centric, relational, other tasks reformulated as QA, etc.), the sources (e.g., crowd-workers, domain experts, exam writers, etc.), and the relationship of the question to the passage are different among datasets (e.g., collected as independent vs. dependent on evidence, multi-hop, etc). The availability of such datasets promotes the development of models that work well for only a specific domain. However, little attention (Chung et al., 2017; Sun et al., 2018) has been paid towards generalization, i.e., building QA systems that can generalize well on different datasets and transfer to new domains quickly.

One major factor that could contribute to generalization, is effective contextual representation (Talmor and Berant, 2019). Recently, models pre-trained on a large unlabeled corpus, by adding an extra final layer and fine-tuning on task-specific supervised data, obtained breakthrough performances on many language understanding tasks such as the GLUE benchmark and the SQuAD QA task (Radford et al., 2018; Devlin et al., 2019; Yang et al., 2019). This indicates the power of pre-trained language models in representing contextual information. Thus, we adopt XLNet (Yang et al., 2019), the state-of-the-art pre-trained language model as our language representation.

Another critical issue related to generalization is how to adapt to new QA tasks using few or even no prior training examples. McCann et al. (2018); Liu et al. (2019); Talmor and Berant (2019) show that promising results can be obtained in transferring to new domains by training models on multiple tasks simultaneously using multi-task learn-

* These two authors contributed equally.

Proceedings of the Second Workshop on Machine Reading for Question Answering, pages 203–211
Hong Kong, China, November 4, 2019.

ing. Multi-task learning explores the relationships between different tasks by capitalizing on relatedness while mitigating interference from dissimilarities, thus forcing models to learn useful representations more generally by unifying tasks under a single perspective. Thus, a model, which is trained on multiple source QA datasets, can achieve robust generalization and transferring ability.

To summarize, we present our work for the MRQA 2019 shared task on generalization. We propose to use multi-task learning on different source QA datasets and fine-tune XLNet (Yang et al., 2019), to build a QA system which has general linguistic intelligence.

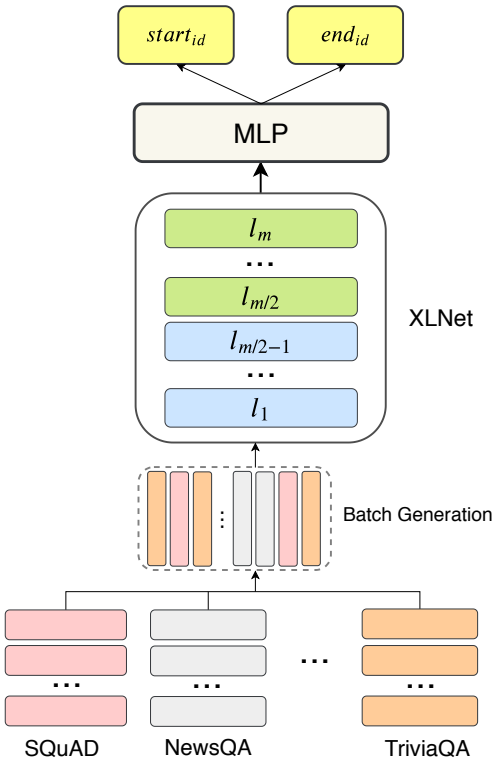

Figure 1: The model architecture. **GPU-version:** The blue boxes (first half) of XLNet layers remain unchanged during fine-tuning and only green boxes are updated due to the GPU's memory limitation. **TPU-version:** All layers of XLNet are fine-tuned.

2 Related Work

2.1 Pre-trained Language Models

Fine-tuning pre-trained language models via supervised learning has become the key to achieving state-of-the-art performance in various natural language processing (NLP) tasks. Among them, BERT (Devlin et al., 2019) extracts contextual meaning through bidirectional encoding with a masked language model and a next-sentence prediction objective. Recently, XLNet (Yang et al., 2019), a permutation language model, was introduced to leverage the bidirectional context and overcome the drawbacks of BERT due to its autoregressive nature. XLNet-based models have already achieved better performance than BERT-based models on many NLP tasks.

2.2 Question Answering

Unlike traditional knowledge-based QA (Kalyanpur et al., 2012), nowadays, many QA systems involve natural language understanding and knowledge of the world. Many datasets, such as SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (Yang et al., 2018), NaturalQuestions (Kwiatkowski et al., 2019), DROP (Dua et al., 2019), RACE (Lai et al., 2017), DueRC (Saha et al., 2018), BioASQ (Tsatsaronis et al., 2012), TextbookQA (Kembhavi et al., 2017), and RelationExtraction (Levy et al., 2017), have been published for specific QA tasks. Among all these tasks, one of the most widely studied one is extractive QA, which is to find a directly mentioned span in the article which answers the particular question. Although many studies on extractive QA have achieved significant improvements by leveraging attention-based models and pre-trained language representations, QA models might still perform poorly in unseen domains due to the data scarcity.

2.3 Multi-task Learning

Liu et al. (2019) proposed a multi-task learning framework-based pre-trained language model (MT-DNN) that leverages nine *natural language understanding* (NLU) datasets and outperforms BERT models. MT-DNN classifies NLU tasks into four classes and uses different loss functions for different task classes, which avoids the model overfitting on a single task by regularizing the language representation.

Meanwhile, Talmor and Berant (2019) proposed MultiQA, which leverages five large QA datasets and five small QA datasets. Merging various extractive QA datasets in training brings general improvement, and achieves the state-of-the-art performance on five QA datasets, which illustrates that training with multiple datasets improves both generalization and transferability.

Dataset	Source	Question	Multi-hop
In-Domain Datasets			
SQuAD	Wikipedia	Crowd	No
NewsQA	News	Crowd	No
TriviaQA	Snippets	Trivia	No
SearchQA	Snippets	Trivia	No
HotpotQA	Wikipedia	Crowd	Yes
NQ	Wikipedia	Query	No
Out-of-Domain Datasets			
DROP	Wikipedia	Crowd	Yes
RACE	Exam	Expert	Yes
DuoRC	Movie Plot	Crowd	No
BioASQ	Biomedical	Crowd	No
TQA	Textbook	Crowd	No
RE	Wikipedia	Crowd	No

Table 1: Characterization of the training and development datasets. *TQA*, *NQ* and *RE* are the abbreviations for *TextbookQA*, *NaturalQuestions* and *RelationExtraction*, respectively.

3 Methodology

3.1 Baseline

MRQA organizers have released the BERT-base and BERT-large models as baselines implemented using the AllenNLP (Gardner et al., 2018) platform. [1] The BERT transformer receives a passage and a question that is separated by an [SEP] token. On top of this, the baseline models deploys a linear layer to find the corresponding span which answers the question from the passage.

3.2 XLNet

Model XLNet (Yang et al., 2019) is a recently proposed generalized autoregressive pre-training model for language understanding which naively follows the Transformer(-XL) (Dai et al., 2019) architecture. Instead of the bidirectional encoding structure used in BERT (Devlin et al., 2019), XLNet leverages a permutation language modeling objective and target-aware representations with a two-stream attention mechanism to enable the model to capture the context on both sides. Besides the datasets which are also used in the pre-training procedure of BERT (Devlin et al., 2019), XLNet involves Giga5 (Parker et al., 2011), ClueWeb 2012-B (an extension version of Callan et al. (2009)) and Common Crawl (Buck et al., 2014) for pre-training. XLNet captures general semantic meanings and produces effective representations to generalize language understanding. BERT is inferior to XLNet because it suffers significantly from the independence assumption and input noise, which prevent BERT from modeling the dependency between targets and result in a pre-training-finetune discrepancy.

Fine-tuning The common strategy in leveraging a pre-trained model is to fine-tune it with an additional linear layer or multilayer perceptron (MLP) on top and adapt it to specific tasks. Empirically, XLNet (Yang et al., 2019) achieves striking results when applied to other tasks through fine-tuning methods, and outperforms the previous state-of-the-art results on 18 tasks, including QA. The results shown in Yang et al. (2019) on the RACE and SQuAD datasets, showing that only an XLNet single model outperforms humans and the best ensemble by 7.6 and 2.5 points in EM, undoubtedly reveal the effectiveness of XLNet on QA tasks.

3.3 Attention-over-Attention

Attention-based neural networks have become a stereotype in most extractive QA systems and is well-known for its capability of learning the importance of distribution over the inputs. attention-over-attention (AoA) mechanism (Cui et al., 2017) is successful because it can generate an "attended attention" which considers the interactive information from both the query-to-document and document-to-query perspectives. Its effectiveness has been proved on public datasets such as the CNN, Children's Book Test, and SQuAD datasets.

4 Experiments

4.1 Preprocessing

The original setting of the sequence length is 512 in the XLNet-large model, but because of the constraint on the computational ability of a single GPU, a trade-off is made between the size of the context and the performance of the model. The sequence length is set as 340 when fine-tuning on the GPU but kept at 512 on the tensor processing unit (TPU). All the datasets are tokenized with SentencePiece (Kudo and Richardson, 2018) and uniformed in lower cases.

4.2 Data Analysis

Datasets Under the scenario of this task, the model should be trained on six training datasets.

[1] https://github.com/mrqa/MRQA-Shared-Task-2019

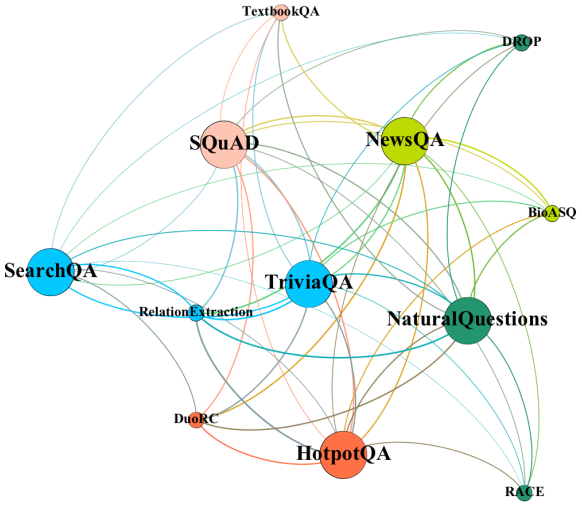

Figure 2: The visualization of the similarity between different datasets using the force-directed placement algorithm via the Gephi platform (Grandjean, 2015). We leverage the Louvain method (Blondel et al., 2008) to automatically cluster the node (datasets) into several communities and mark each community with different colors.

	GPU	TPU
Fine-tuned Layers	12 - 24 (13)	1 - 24 (24)
Floating Point	16	32
Training Batch Size	4	48
Sequence Length	340	512
MLP Layer Size	512, 384, 1	1024, 1

Table 2: Difference of hyper-parameters and the MLP structure when fine-tuning XLNet model on GPU and TPU.

Six in-domain datasets and six out-of-domain datasets are offered as development sets for evaluation. The characterization of the corresponding datasets is shown in Table 1. The twelve known datasets differ from each other in terms of the source of the data, the type of questions, and whether inference (multi-hop) is required during QA. Moreover, the sources of the data on the development datasets are more diverse and not fully covered by the training datasets, which indicates that the generalization ability of the representations produced by the model can significantly improve the performance on the development datasets.

Similarity Evaluation Following the similarity evaluation method utilized in Talmor and Berant (2019), we fine-tune XLNet with an additional MLP on a single GPU using the six training datasets separately, and then evaluate the model on all the in-domain and out-of-domain development sets. More details about fine-tuning the XLNet model on the GPU are mentioned in §4.4. The evaluation results can be found in Table 3. When evaluating the in-domain datasets, the similarity can be computed as

$$Similarity = \frac{P_{ij}}{P_j} + \frac{P_{ji}}{P_i}, \quad (1)$$

where P_{ij} refers to the F1 score when fine-tuning XLNet on dataset D_i and evaluating it on D_j, while P_i refers to the F1 score when fine-tuning and evaluating on D_i. When evaluating the similarity between the in-domain datasets and out-of-domain datasets,

$$Similarity = \frac{2 \cdot P_{ij}}{P_j}, \quad (2)$$

where dataset D_j is one of the in-domain datasets, while D_i is among the out-of-domain datasets.

We visualize the datasets using the force-directed placement algorithm (Fruchterman and Reingold, 1991) for a more intuitive view, which is shown in Figure 2. Each node represents a dataset, and the in-domain datasets and out-of-domain datasets are distinguished by the size of the node. The nodes are linked by a set of edges acting as the springs, pulling nodes towards one another, while non-linked nodes are pushed apart. The weights of the edges act as the pulling force, influencing the distance and the relative position among nodes. In our case, we consider the similarities between nodes (datasets) as the pulling force. The nodes with higher similarity tend to be pulled closer and vise versa.

From Figure 2, the out-of-domain datasets tend to be pushed to the boundary of the figure, which indicates that they have lower similarity with the in-domain datasets. Except for the RelationExtraction dataset, all the out-of-domain datasets only have a strong relationship with one or two in-domain datasets but are positioned far from the others. This implies that to achieve consistently good performance on out-of-domain datasets, data samples from all the in-domain datasets are needed.

4.3 Data Feeding Methods

Empirically, the data feeding order when training and fine-tuning has a great impact on the performance of the model. In terms of the fine-tuning

Datasets	SQuAD	NewsQA	TriviaQA	SearchQA	HotpotQA	NQ
(I) SQuAD	**93.25**	84.99	67.67	43.42	83.48	83.52
(I) NewsQA	60.84	**72.43**	44.13	23.76	56.75	59.13
(I) TriviaQA	66.70	67.50	**76.24**	67.99	64.32	69.21
(I) SearchQA	35.43	43.70	60.16	**79.27**	40.21	54.11
(I) HotpotQA	69.28	64.65	54.12	34.07	**80.09**	64.78
(I) NQ	57.28	66.78	52.36	38.24	63.17	**80.60**
(O) DROP	51.07	33.62	30.04	16.20	48.07	49.54
(O) RACE	48.25	46.67	34.96	19.22	39.57	47.72
(O) DuoRC	61.73	61.45	48.66	29.49	54.24	59.18
(O) BioASQ	70.64	64.48	59.61	49.78	65.46	69.44
(O) TextbookQA	52.93	55.08	46.30	34.90	37.39	58.77
(O) RE	84.62	69.14	73.08	64.47	81.80	81.31

Table 3: Results for XLNet models that are only fine-tuned on a single training set but tested on all the in-domain and out-of-domain development sets. The models are fine-tuned on a single GPU following the GPU-version architecture that is further explained in §4.4. *NQ* and *RE* are the abbreviations for *NaturalQuestions* and *RelationExtraction*, respectively. All the results shown in the table are the corresponding F1 scores.

procedure with the six training sets, we propose two methods for data feeding.

The first method follows the idea of **multi-task learning**. In this task, because the six training sets differ in several aspects as explained in §4.2, we consider them different tasks and leverage the model to fully explore the general semantic representations of the samples in the training datasets. During multi-task learning, we combine all the training datasets and shuffle them to reduce the reliance on the model on the order of the data.

The second method is similar to curriculum learning (Bengio et al., 2009), but because of the sparse relation among the datasets, it's not practical to evaluate the difficulty and the degree of learning. So we simply propose to fine-tune the model using the training sets that are shuffled separately **one after another** with the same training steps.

4.4 Fine-tuning Methods

Various fine-tuning methods based on XLNet are tested to identify the most effective method to achieve better generalization performance. During the fine-tuning procedure, all the methods share a learning rate of 1×10^{-5}.

Fine-tuning on TPU The trend of the pre-trained models for language understanding (Yang et al., 2019; Devlin et al., 2019) is to achieve better performance with larger models, but this leads to their reliance on better computational resources. Even the fine-tuning procedure of XLNet (Yang et al., 2019) is hard to handle in a normal GPU such as GTX 1080Ti, because of the memory size and the processing speed. To make it possible to fine-tune the XLNet model and adapt it to QA tasks on a single GPU, we make modifications to the MLP structure and the hyper-parameters, which are listed in Table 2. For the model on the GPU, only the last 13 layers are further tuned. Except for the reduction of the three hyper-parameters mentioned above, the MLP structure is also changed from a single large linear layer to a deeper but smaller structure.

To fulfill the fine-tuning procedure on the original structure of XLNet with a larger additional linear layer and achieve better performance on development sets and test sets, we take advantage of the TPU (Jouppi et al., 2017) from the Google cloud service. The TPU is a machine learning-oriented application-specific integrated circuit. It has a larger memory and faster computational speed than a GPU, since it consists of a large high bandwidth memory (HBM) and 32-bit floating-point multiply-accumulate systolic array matrix unit. In contrast to the computational power of a GTX 1080Ti (11.34 Tflops of 32-bit floating-point computation and 11 GB of memory), the TPU has 420 Tflops of a 32-bit floating-point computational speed and a 128 GB HBM, which allow us to train a deeper and larger model at a faster speed.

Fine-tuning with MLP Leveraging an MLP as the additional structure for fine-tuning a pre-

Dev Datasets	Multi-task XLNet-large		XLNet-large	
	EM	**F1**	**EM**	**F1**
DROP	40.45	48.93	38.79	48.78
RACE	34.12	49.23	**39.02**	**51.08**
DuoRC	54.63	64.64	50.50	60.62
BioASQ	54.79	70.12	52.06	70.67
TextbookQA	53.76	62.88	48.77	58.86
RelationExtraction	71.27	83.67	66.79	81.75
Average	51.50	63.25	49.32	61.96

Table 4: Results of models fine-tuned with different data feeding methods on development datasets. Both of the models are fine-tuned based on the off-the-shelf XLNet-large pre-trained model on a single GPU. We combine all the training datasets and shuffle the data to fine-tune the multi-task XLNet-large model, while for the other, we feed the data in the following order: SQuAD, NewsQA, TriviaQA, SearchQA, HotpotQA and NaturalQuestions.

Dev Datasets	MLP + GPU		AoA + GPU		MLP + TPU		BERT Large Baseline	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
DROP	40.45	48.93	34.20	43.59	41.04	51.11	33.91	43.50
RACE	34.12	49.23	33.83	48.47	37.22	50.46	28.96	41.42
DuoRC	54.63	64.64	53.03	62.47	51.70	63.14	43.38	55.14
BioASQ	54.79	70.12	56.32	71.58	59.62	74.02	49.74	66.57
TextbookQA	53.76	62.88	52.03	61.49	55.50	65.18	45.62	53.22
RelationExtraction	71.27	83.67	69.10	82.63	76.47	86.23	72.53	84.68
Average	51.50	63.25	49.75	61.71	53.59	65.02	45.69	57.42

Table 5: Results of multi-task models that are fine-tuned with the methods described in §4.4. Compared with the BERT-large baseline, XLNet shows its effectiveness and generalization ability on QA tasks and outperforms the BERT-large model, but the enormous amount of parameters in the XLNet model causes the performance of the model to be constrained by the access to better computational resources.

Test Datasets	Multi-task XLNet-large		BERT-large Baseline	
	EM	**F1**	**EM**	**F1**
BioProcess	56.16	72.91	46.12	63.63
ComplexWebQuestions	54.73	61.39	51.80	59.05
MCTest	64.56	78.72	59.49	72.20
QAMR	56.36	72.47	48.23	67.39
QAST	75.91	88.80	62.27	80.79
TREC	49.85	63.36	36.34	53.55
Dev Average	53.59	65.02	45.69	57.42
Test Average	59.59	72.94	50.71	66.10
Average	56.59	68.98	48.20	61.76

Table 6: Results on test datasets. The multi-task XLNet-large model is the final submission model that is fine-tuned on the TPU with 15k training steps.

trained model is a common strategy of task adaptation. In this task, we test the performance of XLNet with an MLP when fine-tuning on both the GPU and TPU. Because of the limitation of the memory size on the GPU, the MLP structure differs from that on the TPU. More details are shown in Table 2.

Fine-tuning with AoA Layer We also test the performance of the model when fine-tuning XLNet with an AoA layer on a single GPU. In this case, we add an additional AoA layer between the output layer of XLNet and MLP mentioned above. In the practical implementation of this method, the representations of the context and the query need

to be split from the output of XLNet, while we can get the corresponding representation directly and separately when using BERT.

4.5 Results

Comparison between Data Feeding Methods

Table 4 shows the performance of the XLNet models fine-tuned with the two data feeding methods mentioned in §4.3 on the development sets. Both models are fine-tuned with an additional MLP on a single GPU based on XLNet-large. For the single-task XLNet model, we feed the data in the following order: SQuAD, NewsQA, TriviaQA, SearchQA, HotpotQA, and NaturalQuestions. In general, the multi-task data feeding method outperforms the method in which the datasets are fed one after another. On further observation, multi-task learning tends to enable the model to achieve uniform generalization performance on unseen datasets, while the single-task feeding method better benefits the tasks that are similar to the last task that is involved during fine-tuning. The fact that the single-task model achieves better performance on RACE than that using the multi-task learning method is related to the higher similarity between RACE and NaturalQuestions, which we can figure out from Figure 2.

Comparison between Fine-tuning Methods

The results of the experiments on different fine-tuning methods are shown in Table 5. All the experiments are evaluated on the development sets. Although the AoA layer improves the performance of BERT on the SQuAD dataset, which can be seen on the SQuAD leaderboard, it fails to improve generalization performance on XLNet. Moreover, while it takes 300k training steps to finish fine-tuning, we only need 100k training steps to fine-tune the XLNet model with an MLP (refer to §4.4) on this QA task. The XLNet model fine-tuned with an MLP on the TPU achieves the best performance, both on average and on each development dataset. It outperforms the baseline by a large margin, but only requires 15k training steps for fine-tuning. The TPU shows its effectiveness on training with its ability to afford a larger model, batch size, and sequence length.

Comparison with Baseline

The results on the test sets shown in Table 6 indicate that the multi-task XLNet-large model fine-tuned with a larger linear layer on the TPU consistently outperforms the BERT-large baseline by a huge margin. On the test set, our XLNet based model fine-tuned under the multi-task learning setting shows its robust generalization and transferring ability over the baseline.

5 Conclusion

In this paper, we propose a multi-task framework to improve the generalization ability of question answering systems by leveraging large pre-trained language models. Experimental results indicate the effectiveness of our methods on broader QA tasks, with an average Exact Match score of 56.59 and an average F1 score of 68.98, which are significantly higher than the BERT-large baseline results by 8.39 and 7.22, respectively.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Christian Buck, Kenneth Heafield, and Bas Van Ooyen. 2014. N-gram counts and language models from the common crawl. In *LREC*, volume 2, page 4. Citeseer.

Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. 2009. Clueweb09 data set.

Yu-An Chung, Hung-Yi Lee, and James Glass. 2017. Supervised and unsupervised transfer learning for question answering. *arXiv preprint arXiv:1711.05345*.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 593–602.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proc. of NAACL*.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Thomas MJ Fruchterman and Edward M Reingold. 1991. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Martin Grandjean. 2015. GEPHI – Introduction to Network Analysis and Visualization. Http://www.martingrandjean.ch/gephi-introduction/.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. 2017. In-datacenter performance analysis of a tensor processing unit. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pages 1–12. IEEE.

Aditya Kalyanpur, Siddharth Patwardhan, BK Boguraev, Adam Lally, and Jennifer Chu-Carroll. 2012. Fact-based question decomposition in deepqa. *IBM Journal of Research and Development*, 56(3.4):13–1.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4999–5007.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, june. *Linguistic Data Consortium, LDC2011T07*, 12.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Amrita Saha, Rahul Aralikatte, Mitesh M Khapra, and Karthik Sankaranarayanan. 2018. Duorc: Towards complex language understanding with paraphrased

reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2018. Improving machine reading comprehension with general reading strategies. *arXiv preprint arXiv:1810.13441*.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

George Tsatsaronis, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, Thierry Artieres, Michael R Alvers, Matthias Zschunke, et al. 2012. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *2012 AAAI Fall Symposium Series*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.