university of groningen / faculty of science and engineering / bernoulli institute

Retriever-Extractor-based System for Question Answering

Matthijs van der Lende, s4325621, m.r.van.der.lende@student.rug.nl
Niclas Müller-Hof, s4351495, n.j.muller-hof@student.rug.nl
Dries Wedda, s4745329, a.f.wedda@student.rug.nl

Abstract: In the realm of Natural Language Processing, accurately extracting answers from text to solve questions has become a pivotal role. An extractive language model like question answering guarantees the question's results to be found within a span of text according to context documents. Our work targets the enhancement of question-answering models through transfer learning, specifically fine-tuning a RoBERTa-based extractive reader from a retrieval/reader model. The research aims to ascertain the extent to which such a model, pre-trained on diverse datasets, can adapt to specific domains when further trained on targeted datasets like NewsQA. Our results show that the RoBERTa-based model outperformed a BERT-based model which was not pretrained on a QA data set. Furthermore, both models struggled to generalize to OOD data.

1 Introduction

In the evolving field of Natural Language Processing (NLP), the ability to accurately extract answers from textual data has emerged as a cornerstone of technological advancement, powering applications from virtual assistants to information retrieval systems. Central to this endeavor is the development of reading comprehension-based question-answering (QA) models ((Chen, Fisch, Weston, & Bordes, 2017), (Kwon, Trivedi, Jansen, Surdeanu, & Balasubramanian, 2018), (Zhu, Zeng, & Huang, 2019)), which aim to understand and extract precise information from a given context in response to specific inquiries.

Extracting specific details from textual documents presents a notable challenge due to the variability and free-form nature of language, which can render rule-based extraction methods insufficient, especially across diverse context domains and varying question formulations. This complexity highlights the need for advanced machine learning solutions, particularly language models, which excel in understanding and interpreting textual contexts within written documents ((Pearce, Zhan, Komanduri, & Zhan, 2021), (Xu, Liang, Huang, & Xiang, 2021)). Moreover, the advantages of the extractive question-answering model become evident when compared to the pitfalls of hallucination seen in generative large language models (LLMs) (Bang et al., 2023). Such extractive models underscore the necessity for more sophisticated information retrieval techniques, showcasing their effectiveness in accurately identifying and returning specific text boundaries within documents.

For this reason, we investigate the possibilities of using transfer learning on a retriever/reader model. More specifically, we are fine-tuning the extractive reader from a pre-trained model. The pre-trained question-answering model that we are using is built upon the RoBERTa-based architecture. RoBERTa (Robustly Optimized BERT Approach) is a robustly optimized version of BERT, using dynamic masking, more extensive training data, and has been optimized by removing the Next Sentence Prediction (NSP) task. (Y. Liu et al., 2019). By leveraging a pre-trained model, we want to explore how the model can be fine-tuned with specific datasets to excel in particular domains or tasks. This approach not only aims to enhance the model's performance on specialized tasks, but also to investigate its adaptability and efficiency in processing and understanding language and context-specific to those tasks.

The report is structured as follows: in Section 2 we discuss background work to QA and language models. We discuss our model architecture and methods in Section 3. Section 4 will be a description of our experiment and an analysis of our experimental results. Finally, we conclude the report and propose future work in Section 5.

2 Related Work

In this section, we will highlight relevant prior research. First, we will discuss various architectures suitable for reading comprehension. Then, we will highlight the usage of transfer learning in NLP. Finally, we will mention how the generalization ability of a QA system can be evaluated.

Reading Comprehension In a survey on reading comprehension methods, S. Liu, Zhang, Zhang, Wang, and Zhang (2019) mention that BERT is a suitable method for reading comprehension tasks, acknowledging its high computational cost. They also describe that while ELMo is a competitive method, its model capacity might be limited due to the use of the bidirectional LSTM. Zeng, Li, Li, Hu, and Hu (2020) also mention that modern deep learning architectures such as RoBERTa and XLNet reach state-of-the-art performance on reading comprehension. Overall, we conclude that modern deep learning methods can yield high performance, but the computational cost and scarcity of data to train a deep learning model is a practical hurdle.

Transfer Learning This is a common challenge in NLP, which gives rise to the popularity of transfer learning (TL) (Alyafeai, AlShaibani, & Ahmad, 2020). The idea is to use a model that has already been trained on a similar task and use that model on the task at hand. In our project, a model pretrained on a QA data set could be beneficial. The abundance of pre-trained models makes it feasible to use and train deep neural networks in the absence of a strong GPU and a large data set. While it might appear that TL is a general-purpose solution to solve common issues faced in deep learning, the no free lunch theorem suggests that TL might also have its pitfalls. Indeed, TL might suffer from catastrophic interference (Iman, Arabnia, & Rasheed, 2023). Catastrophic interference refers to the scenario where a pre-trained model forgets its prior knowledge after being trained on a new data set. In our project, we use a baseline model which is not pretrained on a QA data set, and we use another model which is trained on a QA data set.

Generalization The aim of our project is to design a QA system that can generalize well. To this extent, we got inspired by the MRQA 2019 Shared Task (Fisch et al., 2019). In this task, participants were provided a training set and a validation set, and the QA systems were subsequently evaluated on hidden test sets. The provided data set and the test sets originate from different sources, such that the test set evaluates the model's Out-of-Distribution (OOD) error. This entails that the shared task focuses on OOD generalization, where the training data and test data follow a different distribution (J. Liu et al., 2023). In our project, we therefore used test sets which are OOD, as described in Section 4.

3 Methods

The approach that has been taken to designing the QA system is a variant of the retriever/reader model called reader-extractor (Kešelj, 2024) (see Figure 3.1). As the name suggests, the model is divided into two stages. The first stage is *retrieval.* Here, the model is given a query to find appropriate documents from a document collection, database, or the internet. For this QA system/model, we assume an information retrieval system that searches based on the similarity of the query and document embeddings, where the set of retrievable documents is set at inference time. The second stage is *extraction.* Here, a BERT-based transformer model is used to extract a candidate substring of the retrieved document, called a *span*, that answers the associated query. This type of NLP task is called span extraction and is a form of reading comprehension. The first step of the QA system can also be skipped by providing an associated context or passage directly, alongside a query vector. This is what is done at training time to fine-tune the reader.

3.1 Information Retrieval

In the full retriever-reader formulation, we assume that the context string associated with a question can be retrieved instead of given directly. The information retrieval module is called the *retriever.*

The information retrieval technique used here is known as semantic search, which involves using vector embeddings of the question (also known as the query here) $\mathbf{q}$ to retrieve suitable candidate documents $(\mathbf{d}_i)_{i=1,..,k}$ from a document collection $(\mathbf{d}_j)_{j=1,...,l}$ (where $k \leq l$) using a similarity metric. Similarity is traditionally measured by the cosine similarity:

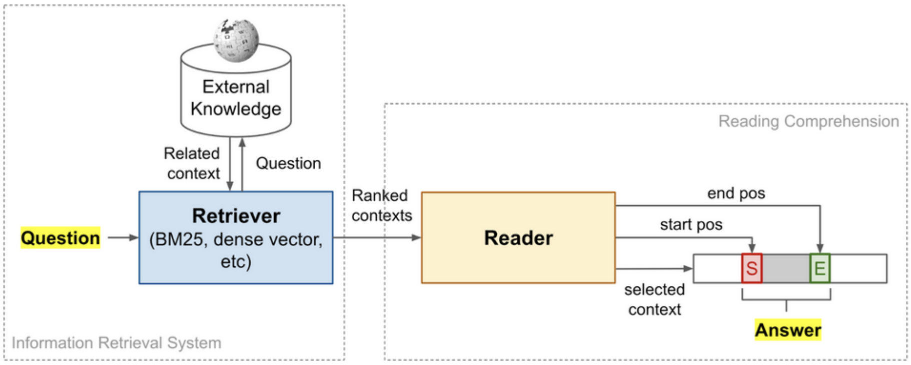

Figure 3.1: The retriever-reader QA framework combined with an extractor-based reader (Weng, 2020).

$$\text{cosine}(\mathbf{q}, \mathbf{d}_i) = \frac{\mathbf{q} \cdot \mathbf{d}_i}{\|\mathbf{q}\| \|\mathbf{d}_i\|},$$

where $\|\cdot\|$ is the L_2/Euclidean norm. The intuition is that two vectors in an embedding space are similar semantically if they point in a similar direction. This is the case if the dot product is positive. The cosine similarity additionally normalizes by the lengths of the two embedding vectors.

The information retrieval module is initialized as follows: we take a document collection $(d_i)_{i=1,\ldots,l}$ and pass it to an encoder BERT_D to get a dataset of pre-computed document vectors $(\mathbf{d}_i)_{i=1,\ldots,l}$. After this is done, another (or the same) encoder BERT_Q is used to get an embedding for the query, and then the cosine similarity is computed between the query embedding $\mathbf{q}$ and all the document embeddings $(\mathbf{d}_i)_{i=1,\ldots,l}$ to compute a score for each of the documents (the higher, the more of a match). This score can then be used to get a document ranking.

3.2 Extractive Reader

The training algorithm for fine-tuning the extractive reader proceeds as follows: We consider a train dataset $\mathcal{D} = (q_i, p_i, a_i)_{i=1,\ldots,N}$ of question-context-answer strings. Where q_i, p_i, a_i are all assumed to be strings over the English alphabet. The provided context string p_i, also known as a passage, allows one to skip the information retrieval step and fine-tune the reader independently of the full QA system. In the preprocessing step, the strings are assumed to be tokenized. The goal of the reader is to compute the probability that a span a in a document/passage p is the answer to the query q, $P(a|q,p)$. A simplifying assumption is made that this probability can be estimated by:

$$P(a|q,p) = P_{\text{start}}(a_{\text{start}}|q,p)P_{\text{end}}(a_{\text{end}}|q,p), \tag{3.1}$$

where $a_{\text{start}}, a_{\text{end}}$ is the start and end token of the span. $P_{\text{start}}(\cdot), P_{\text{end}}(\cdot)$ are probability measures describing the probability a token is the start and end token of a span, respectively.

This means that for a tokenized passage $(p_1, \ldots, p_n)$ we require that for each token we output the probability of it being the start and end token of the answer span. To allow for this the reader, a BERT language model is modified to include two linear layers with weight vectors $\mathbf{s}$, $\mathbf{e}$ respectively (called the span-start and span-end embeddings) that will output the probability that a token p_i is the start and end position of the span.

More specifically, assume the query and passage are tokenized as $(q_1, \ldots, q_n)$ and $(p_1, \ldots, p_m)$ respectively. Then the full token sequence given to the BERT encoder is $(\texttt{[CLS]}, q_1, \ldots, q_n, \texttt{[SEP]}, p_1, \ldots, p_m)$, where `[SEP]` is a separation token. The `[CLS]` token is used to estimate the probability that there is no answer. Questions with no answers have the `[CLS]` token as an answer (see Figure 3.2).

Let $(\mathbf{p}_1, \ldots, \mathbf{p}_m)$ be the associated output embeddings corresponding to passage tokens $(p_1, \ldots, p_m)$ after being passed through the encoder of the reader model. Then to get the start probability $P_{\text{start}}(p_i|q,p) = P_{\text{start}_i}$, we compute the dot product between the output embedding $\mathbf{p}_i$ for the token and the start-embedding vector $\mathbf{s}$ and pass it through a softmax, normalizing over all other token embeddings $\mathbf{p}_j$ in passage:

$$P_{\text{start}_i} = \frac{\exp(\mathbf{s} \cdot \mathbf{p}_i)}{\sum_j \exp(\mathbf{s} \cdot \mathbf{p}_j)}, \tag{3.2}$$

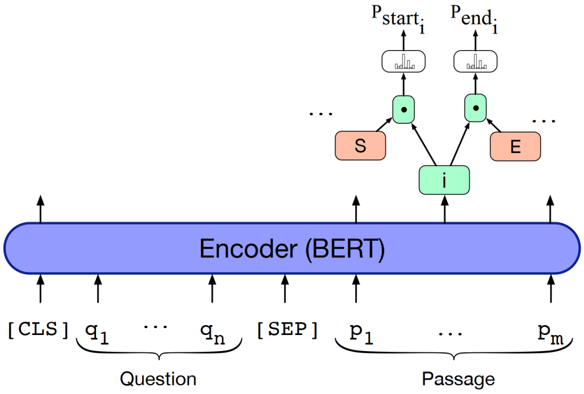

Figure 3.2: An encoder model (using BERT) for span-based question answering from reading-comprehension-based question answering tasks (Kešelj, 2024). S, E are additional weight matrices that are trained for span labelling.

and analogously for the end token probability:

$$P_{\mathrm{end}_i} = \frac{\exp(\mathbf{e} \cdot \mathbf{p}_i)}{\sum_j \exp(\mathbf{e} \cdot \mathbf{p}_j)}.$$

$\mathbf{s}$ and $\mathbf{w}$ are additional parameters that are trained during fine-tuning.

Given the aforementioned dataset $\mathcal{D}$, the training loss for fine-tuning is the negative sum of the log-likelihoods of the correct start and end position for each token:

$$\mathcal{L}(\boldsymbol{\theta}) = -\log P_{\mathrm{start}_i} - \log P_{\mathrm{end}_i},$$

where i is the token index and $\boldsymbol{\theta}$ is a vector of all the model parameters (or a selection of the parameters in case some are frozen).

At inference time, the model can output candidate answers spans sorted by a confidence score, which is the probability $P(a|q,p)$ mentioned before. The highest scoring span can be chosen as the model prediction.

4 Experiments and Results

Our repository is available on GitHub [1].

4.1 Data

We are utilizing the NewsQA dataset from HuggingFace (`lucadiliello/newsqa`) which has been formatted and filtered for question answering. The original data has been split and taken from the MRQA 2019 Shared Task dataset also provided in HuggingFace (MRQA) (Fisch et al., 2019). The NewsQA dataset comprises CNN news articles, providing a real-world context for natural language processing (NLP) tasks. Our experiment involves using a predefined split of the dataset, with 74,160 samples in the training set and 4,212 samples in the validation set. Each training sample consists of:

- Input u: This comprises a pair of context and question. The context is a passage extracted from a CNN news article having a length between roughly 300 to 700 tokens, with the highest frequency at the upper end of this range. This provides the information necessary to answer the question. The questions are shorter, with most of the questions being between 5 and 10 tokens, and the mode being around 7 or 8 tokens.

- Teacher output: This is the answer to the question posed, extracted from the context. These are quite brief, predominantly 2 to 4 tokens in length, with 2-token answers being the most common.

An example of a training sample is:

- Context: "The Venezuelan government is placing a higher tax on alcohol and cigarettes"

- Question: Venezuela hiked taxes on what items?
- Answer: alcohol and cigarettes

To further evaluate how well the reader evaluates we take a selection of out of distribution test sets from the `mrqa` dataset. This dataset includes NewsQA as part of the train and validation set, hence we know the test sets are out of distribution. The datasets are listed Table 4.1.

Dataset	**Number of Samples**
TextbookQA	1503
DROP	1503
BioASQ	1504
RACE	674
DuoRC Paraphraser	1501

Table 4.1: Datasets and Number of Samples with HuggingFace Links.

4.2 Model Selection

For semantic search in the information retrieval step use the `multi-qa-mpnet-base-dot-v1` model from HuggingFace which is designed for sentence similarity. As a Baseline for our extractive reader transformer model, we take the `distilbert-base-uncased` model, which is a distilled BERT model that has not been trained on any question answering datasets. For our primary reader model, we take a RoBERTa model, `roberta-base-squad2-distilled`, which has been pretrained on the SQuAD 2.0 (Rajpurkar, Zhang, Lopyrev, & Liang, 2016) dataset.

4.3 Evaluation

For evaluation, we use the same metrics that are used in the Stanford Question Answering Dataset (SQuAD). Namely, the Exact Match (EM) metric and the F_1 score. The EM metric gives the percentage of predicted answers that match the gold answer. Here, gold answer is the exact answer span that is given in a train sample. The F_1 score is computed based on the overlap between tokens in the predicted answer and gold answer:

$$\begin{aligned} \text{Precision} &= \tfrac{\text{\# overlapping tokens}}{\text{\# tokens in predicted answers}} \\ \text{Recall} &= \tfrac{\text{\# overlapping tokens}}{\text{\# tokens in gold answer}} \\ F_1 &= 2 \times \tfrac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}. \end{aligned} \tag{4.1}$$

4.4 Experimental Setup

We first discuss hyperparameter tuning. Hyperparameter tuning on these transformer models is very computationally expensive. For this reason, Sheng (2021) only performed hyperparameters on the number of epochs. This same approach has been taken here, except due to computational and time constraints, the train set is limited to 4000 samples. The NewsQA validation set F_1 score is taken on $1, 2, 3, 5$ and 7 epochs to roughly estimate how many epochs are needed (see Figure 4.1). Similar to research by Sheng (2021), we concluded that 3 is a suitable number of epochs. The other hyperparameters have been chosen based on typical values used in similar HuggingFace models on question answering tasks that are known to 'work well' (see Table 4.2). The max sequence length/maximum number of tokens is set to 512, the maximum allowed on BERT models, based on maximum context size from the training set.

Training is performed by fine-tuning the baseline and primary model on the NewsQA train set with the mentioned hyperparameters (we assume the encoder is automatically modified to be comptable with span extraction if needed). The resulting models are denoted as (`Matthijs0/DistilBERT`) DistilBERT and Distilled-RoBERTa (`Distilled-RoBERTa`) respectively. For evaluation, we evaluate DistilBERT (the baseline) and Distilled-RoBERTa by computing EM and F_1 scores on the NewsQA validation set. Finally, at test time, we perform the same evaluation on a collection of out of distribution datasets.

Parameter	**Value**
Batch Size	16
Number of Epochs	3
Max Sequence Length	512
Learning Rate	2×10^{-5}
Optimizer	AdamW
Learning Rate Schedule	LinearWarmup
Weight Decay	0.01
Embeddings Dropout Probability	0.1

Table 4.2: Hyperparameters for extractive reader models.

4.5 Results

The evaluation metrics on the evaluation and test sets can be found in Table 4.3 and 4.4. Finally, we also perform inference on question-context pairs to see if reader output is sensible. We now show reader model inference on a selection of examples ("no answer" is used to indicate there is no span that answers the question):

- **Question:** What is virtual memory?
- **Context:** Virtual memory is a crucial abstraction mechanism used in Operating Systems. It involves the separation of logical memory as perceived by developers from physical memory. Each process then has its own virtual address space, which is broken up into pages (which can be partially in main or non-volatile storage).
- **Predictions:** a crucial abstraction mechanism used in Operating Systems (confidence: 0.85), no answer (confidence: 0.15).
- **Question:** What is NLP?
- **Context:** It is the design and analysis of computational algorithms and representations for processing human natural language. It is about automating the analysis, generation, and acquisition of human ("natural") language.
- **Predictions:** Natural Language Processing (confidence: 0.81), the design and analysis of computational algorithms and representations for processing human natural language (confidence: 0.77), no answer (confidence: 0.04).
- **Question:** What is a cat?
- **Context:** I'm ChatGPT, an artificial intelligence language model created by OpenAI. My purpose is to assist users like you in generating text, answering questions, providing information, and engaging in conversations on a wide range of topics. Whether you need help with writing, learning, brainstorming ideas, or just want to chat, I'm here to lend a hand! Feel free to ask me anything, and I'll do my best to assist you.
- **Predictions:** No answer (confidence: 0.71), ChatGPT (confidence: 0.16), artificial intelligence language model created by OpenAI (confidence: 0.15).

Table 4.3: Exact Match (EM) and F_1 score on the NewsQA validation set of different models trained on the NewsQA training set.

Model	EM/F_1
DistilBERT	41.93/54.91
Distilled-RoBERTa	50.02/63.61

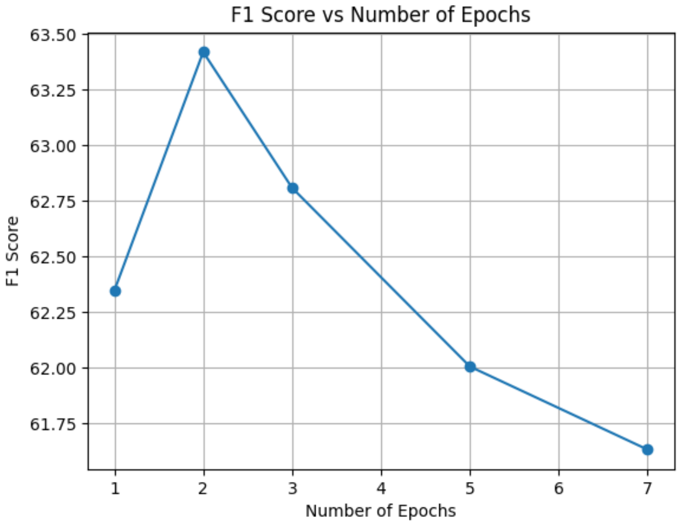

Figure 4.1: F_1 score on the NewsQA validation set over number of epochs.

Table 4.4: Exact Match (EM) and F_1 score on OOD test sets of different models trained on the NewsQA training set.

Model	RACE		BioASQ		DROP		DuoRC		TextbookQA	
	EM	F_1	EM	F_1	EM	F_1	EM	F_1	EM	F_1
DistilBERT	20.33	30.39	30.59	46.37	13.11	20.32	36.38	46.52	24.02	33.23
Distilled-RoBERTa	30.27	43.16	37.17	55.58	24.15	31.54	46.97	58.08	35.26	47.94

4.6 Analysis

In examining the model predictions, we observe distinct behaviors across different question-context pairs. For instance, when asked “What is virtual memory?”, the model correctly identifies “a crucial abstraction mechanism used in Operating Systems” as the answer with high confidence. However, it also provides a second prediction of “no answer” with relatively lower confidence. This dual prediction suggests the model’s recognition of uncertainty in its response, while still demonstrating recognition of what is meant by virtual memory based on the context.

Similarly, when queried about “What is NLP?”, the model exhibits proficiency by offering multiple plausible answers, including “Natural Language Processing” and a detailed explanation of NLP, all with high confidence scores. This versatility in providing relevant and informative responses showcases the model’s ability to extract multiple plausible answers to a question.

In the case of the question “What is a cat?”, the model correctly identifies that there is no answer and provides the prediction of “no answer” with relatively high confidence. This recognition of the absence of a suitable response demonstrates the model’s ability to acknowledge situations where the question cannot be answered based on the provided context.

Overall, Distilled-RoBERTa demonstrates superior performance in terms of EM/F1 scores compared to DistilBERT on both the NewsQA validation set and out-of-distribution test sets. This indicates that Distilled-RoBERTa does benefit from being pre-trained on QA dataset. The F1/EM scores suggest that the Distilled-RoBERTa has overall worse out of distribution performance than validation performance. It is worth noting in general regarding these metrics that the reader model can output a prediction that sufficiently answers the question, but does not align closely with the gold answer.

5 Conclusion

Our results show that the Distilled-RoBERTa model, which was pretrained on a QA data set, consistently outperforms the DistilBERT model, which was not pretrained on a QA data set. This showcases how transfer learning can be beneficial in NLP, even though it should not be forgotten that there do exist cases in which transfer learning can be harmful.

The discrepancy between the test error in Table 4.4 and the validation error in Table 4.3 suggests that the models struggle to generalize to the OOD test data. From a machine learning perspective, this makes sense, as the probability distribution that is modelled in the training set differs from the probability distribution that is modelled in the test set. In future research, one could use the setup of a task that aimed to train a semantic segmentation model that generalizes well to OOD data (Wang, Zheng, Ma, Lu, & Zhong, 2022). In this task, the training set, the validation set, and the test set follow different probability distributions, which perhaps can make the model validation more robust.

Furthermore, it should be acknowledged that question answering is a broad topic, and that our model is limited to reading comprehension. For example, in a closed-book question answering setting, the model constructs a knowledge base at training time, which it has to rely on for its answering at inference time. This is in contrast with the reading comprehension setting, where the context is provided to the model, and information retrieval is limited.

Notes

[1] For the source code of our project, refer to `https://github.com/matthjs/nlp-final-project`

References

Alyafeai, Z., AlShaibani, M. S., & Ahmad, I. (2020). A survey on transfer learning in natural language processing. *arXiv preprint arXiv:2007.04239*.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., ... Fung, P. (2023). *A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity.*

Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. In R. Barzilay & M.-Y. Kan (Eds.), *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1870–1879). Vancouver, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P17-1171 doi: doi:10.18653/v1/P17-1171

Fisch, A., Talmor, A., Jia, R., Seo, M., Choi, E., & Chen, D. (2019). MRQA 2019 shared task: Evaluating generalization in reading comprehension. In A. Fisch, A. Talmor, R. Jia, M. Seo, E. Choi, & D. Chen (Eds.), *Proceedings of the 2nd workshop on machine reading for question answering* (pp. 1–13). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-5801 doi: doi:10.18653/v1/D19-5801

Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A review of deep transfer learning and recent advancements. *Technologies*, *11*(2), 40.

Kešelj, V. (2024). *Speech and Language Processing (third edition draft).* Retrieved from https://web.stanford.edu/~jurafsky/slp3/

Kwon, H., Trivedi, H., Jansen, P., Surdeanu, M., & Balasubramanian, N. (2018). Controlling information aggregation for complex question answering. In L. Azzopardi, G. Pasi, A. Hanbury, & B. Piwowarski (Eds.), *Advances in information retrieval - 40th european conference on ir research, ecir 2018, proceedings* (pp. 750–757). Springer-Verlag. (Publisher Copyright: © Springer International Publishing AG, part of Springer Nature 2018.; 40th European Conference on Information Retrieval, ECIR 2018 ; Conference date: 26-03-2018 Through 29-03-2018) doi: doi:10.1007/978-3-319-76941-7_72

Liu, J., Shen, Z., He, Y., Zhang, X., Xu, R., Yu, H., & Cui, P. (2023). *Towards out-of-distribution generalization: A survey.*

Liu, S., Zhang, X., Zhang, S., Wang, H., & Zhang, W. (2019). Neural machine reading comprehension: Methods and trends. *Applied Sciences*, *9*(18), 3698.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). *Roberta: A robustly optimized bert pretraining approach.*

Pearce, K., Zhan, T., Komanduri, A., & Zhan, J. (2021). *A comparative study of transformer-based language models on extractive question answering.*

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *Squad: 100,000+ questions for machine comprehension of text.*

Sheng, K. (2021). Improve distilibert-based question answering model performance on out-of-domain datasets by mixing right experts.. Retrieved from `https://api.semanticscholar.org/CorpusID:236924127`

Wang, J., Zheng, Z., Ma, A., Lu, X., & Zhong, Y. (2022). *Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation.*

Weng, L. (2020, 10). *How to Build an Open-Domain Question Answering System?* Retrieved from `https://lilianweng.github.io/posts/2020-10-29-odqa/`

Xu, P., Liang, D., Huang, Z., & Xiang, B. (2021). *Attention-guided generative models for extractive question answering.*

Zeng, C., Li, S., Li, Q., Hu, J., & Hu, J. (2020). A survey on machine reading comprehension—tasks, evaluation metrics and benchmark datasets. *Applied Sciences*, *10*(21), 7640.

Zhu, C., Zeng, M., & Huang, X. (2019). *Sdnet: Contextualized attention-based deep network for conversational question answering.*