Build Robust QA System with Small Datasets using Data Augmentation and Answer Length Penalty

Stanford CS224N {Default} Project

Jingchao Zhang
Stanford Center for Professional Development
Stanford University
jingchao@stanford.edu

Hongbo Miao
Stanford Center for Professional Development
Stanford University
hongbo.miao@stanford.edu

Abstract

Data scarcity has been a common issue in domain adaption for language models.[1] For real-world QA applications, the number of labeled QA datasets are limited by various difficulties. For instance, it is very expensive to generate QA dataset in the medical[2] and legal[3] fields due to the required domain expertise. Therefore, effective approaches for data augmentation are desired. There are two main approaches for language data augmentation, *i.e.* back translation and token perturbation. In this work, a pre-trained DistilBERT[4] model is used to perform a QA task. It is firstly fine-tuned on three large datasets, *i.e.*, SQuAD[5], NewsQA[6], and Natural Questions[7]. The fine-tuned model is then adapted to three new domains with very small training datasets of 127 samples each. By implementing several data augmentation approaches and a length penalty technique, we managed to achieve an EM score of 42.064 and an F1 score of 59.982 on the oo-domain test datasets, which are improved by 32.8% and 26.6% respectively compared to the baseline. We found that data augmentation is particularly helpful to improve F1 score, while answer length penalty contributes to the improvement of EM score.

1 Key Information to include

- Mentor: Christopher Wolff

2 Introduction

The recent development of computer hardware and machine learning algorithms have enabled the training of very large models with billions of parameters. For instance, the state-of-the-art natural language processing (NLP) model GPT-3 has 175 billion parameters.[8] On the other hand, large models demand more training data. In recent years, a model-centric to data-centric trend has attracted growing attention in the research field of AI.[9] Domain adaptation in question answering (QA) is one of the research topics that could benefit from enriched training datasets.[10] A robust QA model allows people to efficiently embed and extract knowledge, which is a critical task given the explosion of data at the internet age.

There are two main issues associated with domain adaptation in QA. First of all, a pretrained model is generally required to perform the domain adaptation task. However, general AI practitioners do not have enough computing power to fine-tune very large language models such as GPT-3. On the other hand, the training datasets in some domains are difficult to acquire. For instance, high-quality QA datasets in the fields of medical[2] and legal[3] are expensive to collect due to the nature of domain expertise. Therefore, light-weight models which can be domain adapted on limited datasets are urgently needed.

In this work, we fine-tune a pre-trained DistilBERT[4] model on SQuAD[5], NewsQA[6], and Natural Questions datasets[7]. Each dataset has 50,000 training samples and 4000-12000 validation samples.

DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than bert-base-uncased, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark.[11] The fine-tuned model is then adapted to three new domains each with 127 QA training samples. The three domains are DuoRC[12], RACE[13], and RelationExtraction[14]. Several data augmentation approaches are explored and an answer length penalty is implemented to increase the performance of the adapted QA model.

3 Related Work

Transfer learning is a machine learning technique where a pretrained model is employed to new datasets.[15] It has become increasingly important in modern AI. Given the variety and similarity among different research fields, it is impractical and inefficient to train new models on every new dataset. The transfer learning technique has been successfully used in many fields such as multi-language text classification[16], text sentiment assessment[17], image classification[18], and human activity classification[19].

Domain adaptation is a particular type of transfer learning. In the context of QA, domain adaptation is often required when a pre-trained language model needs to be applied to a new field due to data shift.[20] Domain adaption is related to an important task in machine learning, which is model generalization beyond training data distribution. Aside from QA, domain adaptation has also been explored in the field of computer vision[21], transfer component analysis[22], and adversarial discrimination[23].

Data augmentation is widely used across all domains of machine learning and deep learning. It is a technique to enrich the training datasets by making modifications to the existing datasets.[24] In computer vision, data augmentation can be easily applied by rotating, shifting, cropping, shrinking or stretching the original picture.[25] However, data augmentation in the field of NLP needs to be handled more carefully since a slight change in the context could radically change the meaning of a sentence.[26]

Answer length penalty and brevity penalty (BP) are techniques to improve QA model performance by penalizing long or short question answers.[27] The BP technique penalizes generated translations that are too short compared to the nearest reference length, and does so in an exponentially decaying manner. The brevity penalty makes up for the fact that the BLEU score has no recall term. The length penalty used a similar approach to force the QA model to generate shorter answers.[28]

4 Approach

4.1 Back translation

Back translation is a language augmentation technique where the source language is first translated to a target language, then translated back to the original language. An important parameter when implementing back translation in QA tasks is the "success rate", which is defined by checking whether the generated context contains the original answer in the QA pair. For example, a successful back translation looks like below, where the answer **heart attack** is preserved in the translated sentence:

Original Text	Ray Eberle died of a **heart attack** in Douglasville, Georgia on August 25, 1979, aged 60.
Vietnamese	Ray Eberle qua đi t mt cn đau tim Douglasville, Georgia vào ngày 25 tháng 8 năm 1979, tui 60.
Back translation	Ray Eberle died from a **heart attack** in Douglasville, Georgia on August 25, 1979, at the age of 60.

On the other hand, a "failed" translation losses the answer words during the translation.

Original Text	Ray Eberle died of a **heart attack** in Douglasville, Georgia on August 25, 1979, aged 60.
Spanish	Ray Eberle murió de un ataque al corazón en Douglasville, Georgia, el 25 de agosto de 1979, a los 60 años.
Back translation	Ray Eberle passed away from a **heart failure** at the age of 60 in Douglasville, Georgia, on August 25, 1979.

After each of the successfully back translation, the new context and answer start index are updated and appended to the original training dataset. In this work, we implemented two types of back translations, vanilla and looped.

4.1.1 Vanilla back translation

In the vanilla back translation method, the back translation is applied only once to each context in the out-of-domain QA training datasets. We used the M2M100 [29] seq-to-seq model for back translation. Specifically, the pre-trained model 'facebook/m2m100_1.2B' from Hugging Face [30] is used for out-of-domain text translation. The 'facebook/m2m100_1.2B' is selected over the 'facebook/m2m100_418M' because the 1.2B model gives a more accurate target language translation. A total of 99 languages are used for the back translation. The full list of languages are shown in the appendix. It is worth noting that the M2M100[29] model claims to translate 100 languages but we found out that only 99 languages are supported. Specifically, there is a duplicate for the Occitan language where two entries "post 1500" and "oc" are provided.

4.1.2 Looped back translation

To generate more training data using back translation, we further implemented a looped version where the "successful" cases are translated back and forth between the source and target languages until certain criterion is reached. The looped back translation will stop if any of the following conditions is met: 1) The translated context is identical with previous ones; 2) The answer words are lost; 3) The loop reached 10 times. Here is a back translation example with two loops:

Original Text	Ray Eberle died of a **heart attack** in Douglasville, Georgia on August 25, 1979, aged 60.
Vietnamese	Ray Eberle qua đi t mt cn đau tim Douglasville, Georgia vào ngày 25 tháng 8 năm 1979, tui 60.
1st translation	Ray Eberle died from a **heart attack** in Douglasville, Georgia on August 25, 1979, at the age of 60.
Vietnamese	Ray Eberle qua đi vì mt cn đau tim Douglasville, Georgia vào ngày 25 tháng 8 năm 1979, hng th 60 tui.
2nd translation	Ray Eberle passed away of a **heart attack** in Douglasville, Georgia on August 25, 1979, aged 60.

We found that the looped back translation could further enrich the augmented language contexts compared to vanilla back translation.

4.2 Token perturbation

Token perturbation is another effective approach for language augmentation. Given a training context, it will randomly choose and perform simple transformations of texts such as swap, insertion, delete, and replacement. In this work, the NlpAug[31] model is implemented to perform token perturbations on the oodomain training datasets. Unlike back translation, token perturbation could eliminate the missing QA answer problem by implementing a "stop word", which will not be transformed during the token perturbation process. This guarantees a 100% "success rate".

4.2.1 Context substitution

The context substitution leverages contextual word embeddings to find top *n* similar word for augmentation. The substitution rate is set at 0.3 which means 30% of the whole sentence will be substituted according to the contextual embeddings calculation.

Original text	Ray Eberle died of a heart attack in Douglasville, Georgia on August 25, 1979, aged 60.
Context substitution	Ray Eberle died of a heart attack in Montgomery, Georgia date august 10th, 1954, December 93

4.2.2 Context insertion

Instead of substitution, the insertion method will inject new words to random positions according to contextual word embeddings calculation. The same augmentation rate of 0.3 is used which means 30% more words will be added to the original sentence.

Original text	Ray Eberle died of a heart attack in Douglasville, Georgia on August 25, 1979, aged 60.
Context insertion	Ray Eberle died of a heart attack in northern Douglasville, Georgia east on August 17 25, was 1979, by aged hardly 60.

4.2.3 Synonym replacement

Unlike the context substitution which used the contextual word embeddings to find new words, the synonym replacement technique leverages semantic meaning to substitute words. The augmentation rate is also set as 30%.

Original text	Ray Eberle died of a heart attack in Douglasville, Georgia on August 25, 1979, aged 60.
Synonym replacement	Ray Eberle died of a heart attack in Douglasville, GA on Aug xxv, 1979, aged threescore.

4.3 Length Penalty

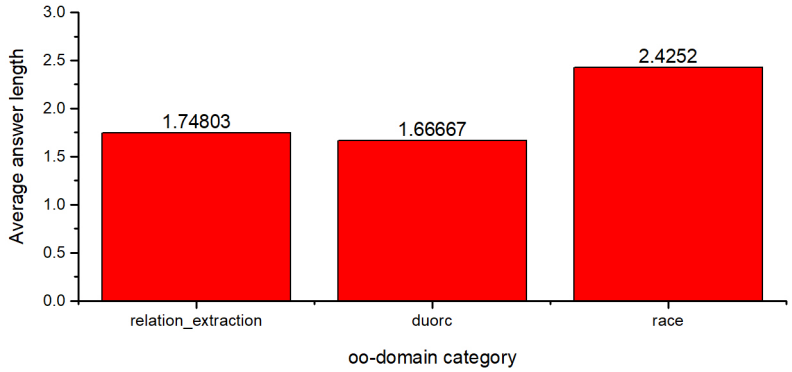

Figure 1: Average answer length for oo-domain training datasets.

While exploring the oo-domain training and validation datasets, we found out that the answers in all categories are very brief, ranging from 1-2 words per answer. The calculated mean answer lengths are shown in Fig. 1. Therefore, instead of implementing a brevity penalty, a length penalty is used to force the model to generate shorter answers during fine-tuning.[28] An additional loss term is added to penalize long answers, which is expressed as $L = sum(diagonal(p_{start} * p_{end} *$

$mask\_matrix(k_{length})))$. The p_{start} and p_{end} are the predicted logits for start position and end position. For a given position *i*, all answer tokens after position [*i* + *k*] are considered invalid, where *k* is a hyperparameter for the length penalty threshold. $mask\_matrix(k_{length})$ is a pre-computed matrix with shape sequence_length * sequence_length where, for each column *i*, $mask\_matrix(i : i + k_{length} + 1, i) = 0$ and other values are 1. The length penalty is then added to the original loss L_{QA}. Based on the results in Fig. 1, *k* is set as 3 in this work.

5 Experiments

5.1 Data

The datasets used in this work are summarized in Table 1. The in-domain datasets are first used to train on the DistilBERT model. The oo-domain training datasets are used for data augmentation. The augmented datasets are then used for domain adaptation.

Table 1. In-domain and oo-domain datasets

Dataset	Question Source	Passage Source	Train	dev	Test
		in-domain datasets			
SQuAD[5]	Crowdsourced	Wikipedia	50000	10,507	-
NewsQA[6]	Crowdsourced	News articles	50000	4,212	-
Natural Questions[7]	Search logs	Wikipedia	50000	12,836	-
		oo-domain datasets			
DuoRC[12]	Crowdsourced	Movie reviews	127	126	1248
RACE[13]	Teachers	Examinations	127	128	419
RelationExtraction[14]	Synthetic	Wikipedia	127	128	2693

5.2 Evaluation method

Performance is measured via two metrics: Exact Match (EM) score and F1 score. The EM score is a binary measure (i.e. true/false) of whether the system output matches the ground truth answer exactly. F1 is a less strict metric – it is the harmonic mean of precision and recall. When evaluating on the validation or test sets, the maximum F1 and EM scores across the three human-provided answers are taken for that question. Finally, the EM and F1 scores are averaged across the entire evaluation datasets to get the final reported scores.

5.3 Experimental details

For the in-domain DistilBert model fine-tuning, we tested trainings from 1 epoch to 5 epochs and it was discovered that the model performance does not improve beyond 3 epochs. A batch size of 16 is used in all models. The AdamW optimizer[32] is used to minimize the loss. The learning rate is set as 3e-5. The random seed for each model is set the same at 42. Since the maximum context size that can be encoded by BERT is 512, each (question, paragraph) is converted into multiple chunks of size 384 with a stride of 128. The in-domain training with 3 epochs took about 1 hour on one NVIDIA A100 GPU. The domain adaptation only took a few minutes on the same hardware.

For vanilla back translation, 99 different languages are used and each language is used only once per QA context. For looped back translation, the same 99 languages are used and multiple passes are used for each language until a stop criterion is met. For context substitution, context insertion and synonym replacement, each QA context is augmented 10 times with a 30% augmentation rate.

5.4 Results

The five highest and five lowest back translation "success" rates are shown in Fig. 2. The rate is calculate by dividing the number of translations containing the QA answer words by the total number

of back translations. Note that all back translations can be considered successful from a syntactic perspective. The "success" rate used in this work is based on the QA answer words preservation. The five highest "success" rate languages are English, Tagalog, Cebuano, Spanish, Sundanese. The five lowest "success" rate languages are Burmese, Wolof, Ganda, Fulah and Nepali. Technically the English language should not be considered as part of the back translation since the target and source languages should be different. Therefore, the high "success" rate of English is not surprising. It is worth noting that 3/5 of the highest "success" rate languages are from Asian. While 3/5 of the lowest "success" rate languages are from African.

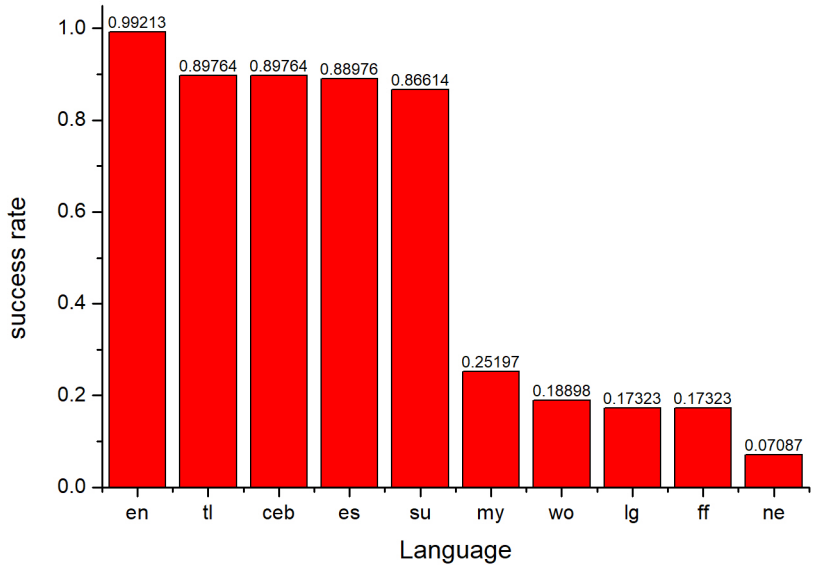

Figure 2: Five highest and five lowest success rates for back translation.

Table 2. Model performance results (EM/F1)

Model	in-domain val	oo-domain val	oo-domain test
Baseline	52.97/69.11	31.68/47.38	
Baseline + oo-domain	53.01/69.04	32.23/48.64	
Vanilla back translation	54.35/70.71	36.65/51.27	
Looped back translation	54.17/70.08	37.17/**52.22**	**42.064/59.982**
Context substitution	54.25/70.88	31.94/48.52	
Context insertion	53.65/69.45	34.82/50.04	
Synonym replacement	54.66/70.88	33.51/49.73	
Length penalty + Looped BT	54.58/69.43	**37.43**/51.71	

Final model performances are summarized in Table 2. The highest oo-domain F1 validation score is provided by the looped back translation method at 52.22. The highest oo-domain EM validation score is achieved by combining length penalty and looped back translation (BT). The final test score are 42.064 and 59.982 for EM and F1 respectively from the "Looped back translation" augmentation approach. The "Length penalty + Looped BT" model does not provide better test score.

To summarize, It can be observed that both token perturbation and back translation can improve the model performance on oo-domain adaptation. However, the back translation approaches provide better domain adaptation results compared to the three token perturbation approaches. Overall, we think data augmentation is an effective approach to enhance domain adaptation performance with scarce training datasets. Moreover, answer length penalty is very effective in improving the EM score of the final model. Both the EM and F1 scores are improved compared to the baseline model.

However, the addition of length penalty does not yield higher oo-domain test results compared to looped back translation itself. The effect of answer length penalty is shown below:

Answers without length penalty	Answers with length penalty
briefcase of shredded blank paper	shredded blank paper
Bean and her best friend, Ivy	Ivy and Bean
mitochondrial is a protein that in humans is encoded by the C14orf159 gene (chromosome 14 open reading frame 159	chromosome 14 open reading frame 159

It can be observed that the answer length penalty approach can effectively shorten the predicted answers.

6 Analysis

6.1 Back Translation

Overall, both vanilla and looped back translations provide better model performances compared to the token perturbation approaches. The back translation approaches provide more comprehensive text transformations. It has combined effects of context substitution, insertion, deletion and etc. Therefore, the overall performance is better. Below are some examples of wrong predictions with back translation augmentation. In the first example, "television" are predicted as "27-inch table computer", which could have caused by a side-effect of back translation of using different expressions of certain words. The second example shows a much longer answer than the correct answer, which suggests the importance of length penalty. The last example is missing punctuation, which we believe is not specifically associated with the back translation augmentation approach.

Predicted answers	Correct answers
27-inch table computer	Televisions
Bean and her best friend, Ivy	Bean
his mother	his mother.

6.2 Token Perturbation

Token perturbations are effective approaches for increasing the number of contexts in the training datasets. However, the augmented context are less accurate compared to those generated by back translations. An example is shown below. Even though the token perturbation method successfully generated a new context based on the original context, the name, location and time have all been transformed and only the QA answer is preserved. We believe these factors contribute to its inferior performance compared to back translation.

Original	Ray Eberle died of a heart attack in Douglasville, Georgia on August 25, 1979, aged 60.
Back translation	Ray Eberl died of a heart attack in Douglasville, Georgia on 25 August 1979 at the age of 60.
Token perturbation	Fred Eberle died of a heart attack in Albany, Georgia since March 25, 1987, aged 60.

Some of the typical errors are shown in the table below. First of all, token perturbation also gives long and incorrect answers which could be due to the context substitution and insertion during data augmentation. Some of the answers are entirely different which could be caused by the synonym replacement.

Predicted answers	Correct answers
a 27-inch table computer	Televisions
Ivy and Bean Make the RulesBy Annie Barrows Bean's older sister Jessie	Bean
Charles Savage Homer.	his mother.

6.3 Length Penalty

The length penalty approach is very effective in improving the EM scores by forcing the model to generate shorter answers. Thus improving the probability of an exact match. On the other hand, a shorter answer increases the probability of missing the correct answer words which gives a lower F1 score compared to the data augmentation approach. Some of the incorrect predictions are shown below. Compared to above approaches, the generated answers are indeed much shorter.

Predicted answers	Correct answers
table computer	Televisions
Ivy and Bean	Bean
his mother	his mother.

7 Conclusion

In this work, we implemented several data augmentation approaches and an answer length penalty technique to improve the domain adaptation performance with scarce training datasets. The explored data augmentation approaches include vanilla back translation, looped back translation, context substitution, context insertion, and synonym replacement. We found that back translation is a more effective approach for language augmentation compared to token perturbation. The looped back translation approach gives the best oo-domain test performance with EM and F1 scores of 42.064 and 59.982, respectively, which are 26.6% and 32.8% improvement compared to the baseline model. Both data augmentation and answer length penalty can help improve the domain adaptation performance. The augmentation approaches are particularly helpful in improving F1 scores while length penalty is more helpful with improving EM scores. In the future, it would be interesting to explore the combined effects of token perturbation with back translation for data augmentation. Also, it is worth investigating the reasons why Asian languages have better "success" rates than African languages when performing back translation on English.

References

[1] Amar Prakash Azad, Dinesh Garg, Priyanka Agrawal, and Arun Kumar. Deep domain adaptation under label scarcity. In *8th ACM IKDD CODS and 26th COMAD*, CODS COMAD 2021, page 101–109, New York, NY, USA, 2021. Association for Computing Machinery.

[2] Sheng Shen, Yaliang Li, Nan Du, Xian Wu, Yusheng Xie, Shen Ge, Tao Yang, Kai Wang, Xingzheng Liang, and Wei Fan. On the generation of medical question-answer pairs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8822–8829, 2020.

[3] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. Jec-qa: A legal-domain question answering dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9701–9708, 2020.

[4] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[5] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

[6] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset, 2017.

[7] 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. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[9] Andrew ng launches a campaign for data-centric ai. `https://www.forbes.com/sites/gilpress/2021/06/16/andrew-ng-launches-a-campaign-for-data-centric-ai/?sh=598df1fd74f5`.

[10] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[11] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

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

[13] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations, 2017.

[14] Wenxuan Zhou and Muhao Chen. An improved baseline for sentence-level relation extraction, 2021.

[15] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[16] Joey Tianyi Zhou, Ivor W Tsang, Sinno Jialin Pan, and Mingkui Tan. Heterogeneous domain adaptation for multiple classes. In *Artificial intelligence and statistics*, pages 1095–1103. PMLR, 2014.

[17] Chang Wang and Sridhar Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *Twenty-second international joint conference on artificial intelligence*, 2011.

[18] Yin Zhu, Yuqiang Chen, Zhongqi Lu, Sinno Jialin Pan, Gui-Rong Xue, Yong Yu, and Qiang Yang. Heterogeneous transfer learning for image classification. In *Twenty-fifth aaai conference on artificial intelligence*, 2011.

[19] Maayan Harel and Shie Mannor. Learning from multiple outlooks. *arXiv preprint arXiv:1005.0027*, 2010.

[20] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

[21] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[22] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2):199–210, 2010.

[23] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[24] David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

[25] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[26] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*, 2021.

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[28] Ranjie Duan. Alp-net : Robust few-shot question-answering with adversarial training , meta learning , data augmentation and answer length penalty. 2021.

[29] Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. Beyond english-centric multilingual machine translation, 2020.

[30] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[31] Edward Ma. Nlp augmentation. https://github.com/makcedward/nlpaug, 2019.

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

A Appendix (optional)

List of languages used in the back translation method:

Afrikaans (af), Amharic (am), Arabic (ar), Asturian (ast), Azerbaijani (az), Bashkir (ba), Belarusian (be), Bulgarian (bg), Bengali (bn), Breton (br), Bosnian (bs), Catalan; Valencian (ca), Cebuano (ceb), Czech (cs), Welsh (cy), Danish (da), German (de), Greeek (el), English (en), Spanish (es), Estonian (et), Persian (fa), Fulah (ff), Finnish (fi), French (fr), Western Frisian (fy), Irish (ga), Gaelic; Scottish Gaelic (gd), Galician (gl), Gujarati (gu), Hausa (ha), Hebrew (he), Hindi (hi), Croatian (hr), Haitian; Haitian Creole (ht), Hungarian (hu), Armenian (hy), Indonesian (id), Igbo (ig), Iloko (ilo), Icelandic (is), Italian (it), Japanese (ja), Javanese (jv), Georgian (ka), Kazakh (kk), Central Khmer (km), Kannada (kn), Korean (ko), Luxembourgish; Letzeburgesch (lb), Ganda (lg), Lingala (ln), Lao (lo), Lithuanian (lt), Latvian (lv), Malagasy (mg), Macedonian (mk), Malayalam (ml), Mongolian (mn), Marathi (mr), Malay (ms), Burmese (my), Nepali (ne), Dutch; Flemish (nl), Norwegian (no), Northern Sotho (ns), Occitan (post 1500) (oc), Oriya (or), Panjabi; Punjabi (pa), Polish (pl), Pushto; Pashto (ps), Portuguese (pt), Romanian; Moldavian; Moldovan (ro), Russian (ru), Sindhi (sd), Sinhala; Sinhalese (si), Slovak (sk), Slovenian (sl), Somali (so), Albanian (sq), Serbian (sr), Swati (ss), Sundanese (su), Swedish (sv), Swahili (sw), Tamil (ta), Thai (th), Tagalog (tl), Tswana (tn), Turkish (tr), Ukrainian (uk), Urdu (ur), Uzbek (uz), Vietnamese (vi), Wolof (wo), Xhosa (xh), Yiddish (yi), Yoruba (yo), Chinese (zh), Zulu (zu)