Computer Science Tripos

Part II Project Proposal Coversheet

Please fill in Part 1 of this form and attach it to the front of your Project Proposal.

Part 1

Name: Yuan Xiaotian Thomas

CRSID: xty20

College: Downing

Overseers: (Initials) AM, CO

Title of Project: Data Augmentation Techniques in Natural Language Processing

Date of submission: 14/10/2022

Will Human Participants be used? No

Project Originator: Yiren Zhao

Project Supervisor: Yiren Zhao, Robert Mullins

Directors of Studies: Robert Harle

Special Resource Sponsor:

Special Resource Sponsor:

Part 2

Overseer Signature 1: --

Overseer Signature 2: --

Overseers signatures to be obtained by Student Administration.

Overseers Notes:

Part 3

SA Date Received:

SA Signature Approved:

Computer Science Tripos Part II Project Phase 3 Proposal

Thomas Yuan

October 14, 2022

Data Augmentation Techniques in Natural Language Processing

1 Introduction and Description

Data augmentation is a technique used in deep learning to increase the amount of data available for training, usually by modifying existing copies of data. By artificially generating more data points, data augmentation helps deal with the issue of limited or scarce data, improving the quality and quantity of the dataset such that the trained model is better and is less prone to overfitting. Some commonly used techniques include synonym replacement and back translation which uses a translation model to translate text to and from other languages in order to obtain syntactically different sentences. Other techniques such as insertion and deletion adds noise to the data by inserting and deleting words randomly.

Data augmentation in natural language processing is more interesting than general deep learning tasks because of the nature of natural languages and how there are interesting rules to manipulate text such that the syntax is different, but the semantics are still the same, for example swapping words with their synonyms or changing the structure of the sentence. This results in new augmentation techniques, namely paraphrasing-based techniques apart from the usual noising and sampling-based techniques that utilize statistical properties to perform data augmentation.

Moreover, data augmentation has also been widely used in computer vision to help train models to perform tasks such as identifying objects in pictures, and some notable techniques in recent years include CutMix [4] and MixUp [5]. Therefore this project aims to explore data augmentation techniques in natural language processing tasks and see how they affect the performance of the trained model, especially under varying scarcities of data, as well as look into how specific data augmentation techniques in computer vision can be applied in natural language processing.

To achieve this, I will be training neural network models on the tasks text classification and seq2seq machine translation, and using data augmentation techniques including paraphrasing and noising-based ones. For each of these techniques, I will be applying them to the dataset, and training the model using this augmented dataset. Afterwards, I will be measuring the performance of the model, using accuracy in classifying texts for text classification and BLEU score [3] for machine translation to evaluate the quality of the translation. I will compare the performance between these augmented datasets and also unaugmented datasets to see how these techniques affect the performance of the model.

2 Starting Point

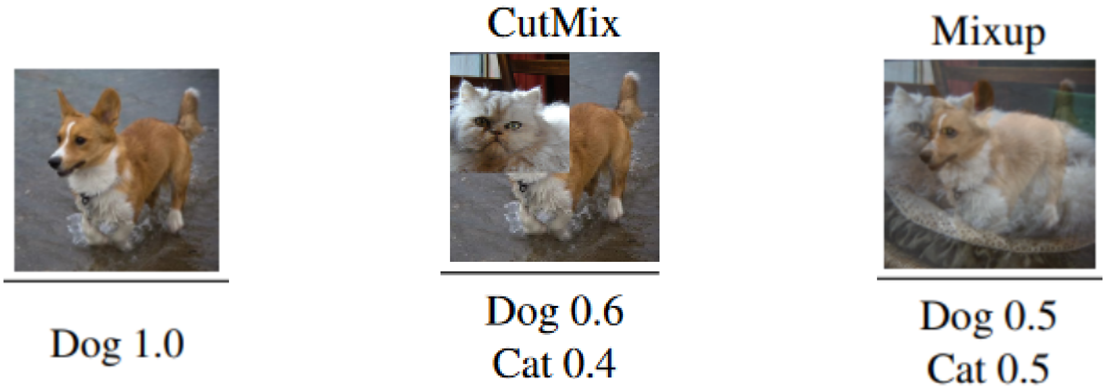
The external libraries that will be used for this project include open source library PyTorch, as well as sub-libraries torchtext and torchnlp. These libraries will be used to implement the neural network model and to help prepare datasets. During September 2022, I went through the PyTorch tutorial and learnt how to implement a basic neural network model to classify images, which was my first time using PyTorch. I have also read through some documentation of torchtext and torchnlp, but no code on NLP-specific neural network models has been written yet. This was also my first time reading into data augmentation, where I read through 3-4 papers on data augmentation linked to me by supervisors, including some surveys of techniques used in NLP. Before that, I had no prior experience with deep learning implementations or data augmentation.

3 Substance and Structure of the Project

The project uses the following key concepts:

Seq2seq machine translation Seq2seq is a machine learning approach commonly used in machine translation, which uses a recurrent neural network to transform one sequence into another sequence. In this project it will be implemented as an encoder and a decoder (along with some other components for optimization) transforming sequences of English words into German words.

CutMix CutMix is a data augmentation technique used in computer vision, where it cuts out portions of images with different labels and pieces them together, with the label being a mix of the labels the original images have. 1b This method aims to help the network learn about a wider range of features such that the trained model can generalize better.



(a) An image of a dog and its label.

(b) An image of the dog and a cat pieced together using the CutMix algorithm, with corresponding labels

(c) An image of the dog and a cat mixed together using the MixUp algorithm, with corresponding labels.

Figure 1: Unaugmented and augmented dog images

MixUp MixUp is a data augmentation technique used in computer vision, where features and their labels are mixed together 1c. It is implemented using the following formula

$$\hat{x} = \lambda x_i + (1 - \lambda)x_j, \quad \text{where } x_i, x_j \text{ are raw input vectors}$$

$$\hat{y} = \lambda y_i + (1 - \lambda)y_j, \quad \text{where } y_i, y_j \text{ are one-hot label encodings}$$

As a result, neural networks become less prone to memorizing corrupt labels and networks are not overconfident about the correlation between features and labels.

The project has the following main work items:

Preliminary Reading I would have to perform some further reading on PyTorch, torchtext and torchnlp to become more familiar with them. I would also need to read more into the different augmentation methods and decide on how exactly I should implement them.

Neural network models I would have to implement a text classification model and a seq2seq English-to-German machine translation model.

Data augmentation techniques I would have to implement synonym replacement, back translation, insertion and deletion. For extensions, I would have to implement these for German as well, and also I would have to implement CutMix and MixUp.

Training and Testing I would have to train my networks on datasets, and implement performance measurements to measure the performance of my trained networks. For extensions, I would have to train my networks on larger datasets.

Dissertation I will work on the dissertation throughout the project along with writing the code. I will finish the dissertation after writing the source code and performing all the training and testing required.

4 Success Criterion

In order to be considered a success, the project should have achieved the following:

1. Implemented a neural network model for text classification, trained and tested on the AGNews [6] corpus, measure accuracy and make comparison with other models that usually achieve 85-95% accuracy
2. Implemented a neural network model for seq2seq machine translation, trained and tested on the IWSLT [2] dataset, measure BLEU score and make comparison with other models that usually achieve a BLEU score of 20-30
3. Implemented data augmentation functions including synonym replacement, back translation, insertion and deletion
4. Train and evaluate implemented network models using augmented datasets and evaluating performance using accuracy for text classification and BLEU score for machine translation
5. Evaluate the effectiveness of data augmentation techniques with accuracy and BLEU score using p-tests

5 Extensions

Some potential extensions for the project include:

1. Use a larger dataset (WMT'16 [1]) for training and testing the models, evaluate performance of models using above-mentioned performance metrics
2. Augment both target and source text in translation text using the same data augmentation techniques but for German
3. Implement CutMix and MixUp and evaluate performance of models trained using these augmentation techniques

6 Timetable and Planning of Work

I have broken down my workload into 15 work items, including 2 breaks, each of which should last around 2 weeks. I have included a description of each of these work items, the weeks they span, and some milestones that should be completed by the end of that particular work item.

Date	Description	Milestones
1. 17/10 - 30/10	- Familiarize myself with PyTorch, torchtext and torchnlp - Do more reading and research into how NLP tasks are implemented - Implement a neural network model for text classification and Seq2Seq machine translation	Finished implementing the two models and tested that they work according to expected using small training datasets
2. 31/10 - 13/11	- Implement data preparation and preprocessing functions such that the datasets can be used for augmentation and models - Read more into the various augmentation methods, specifically back-translation as well as MixUp, figure out exact implementation details including what libraries and algorithms to use - Implement synonym replacement and back-translation (augmentation techniques)	- Finished implementing the two mentioned augmentation techniques, tested that they work according to expected on data - Finished preparing datasets
3. 14/11 - 27/11	- Implement insertion and deletion (augmentation techniques) - Decide on hyperparameters to use for the models - Run all implemented augmentation techniques on dataset to create new datasets that I will be training the networks on	- Finished implementing the three mentioned augmentation techniques, tested that they work according to expected on data - Created new datasets using the implemented augmentation techniques
4. 28/11 - 11/12	Break	

5. 12/12 - 25/12	- Implement performance measurement functions, train and test networks with and without augmentation - Collect data on performance differences, plot graphs, analyze performance differences	Collected data on performance of all five different augmentation techniques, done comparison, visualized differences
6. 26/12 - 8/1	- Extension 1: Implement training network using larger datasets (WMT'18) and measure performance change - Complete progress report and presentation	- All base success criteria should be met at this point - Extension 1: Finish implementation and testing - Finish progress report and presentation
7. 9/1 - 22/1	Extension 2: Implement augmentation techniques for augmenting target language (German) in machine translation, including synonym replacement and MixUp (can reuse same insertion and deletion function implemented before)	Submit progress report
8. 23/1 - 5/2	- Extension 2: Train network and run performance tests - Extension 3: Implement CutMix - Start on write-up	Extension 2: Finish implementation and testing

9. 6/2 - 19/2	- Extension 3: Implement MixUp - Write chapter 1-2 of dissertation	Complete chapter 1-2 of dissertation
10. 20/2 - 5/3	- Extension 3: Train network and run performance tests - Write chapter 3-4 of dissertation	- Extension 3: Finish implementation and testing - Complete chapter 3-4 of dissertation - Source code should be completed by now, and no major changes should be made
11. 6/3 - 19/3	- Continue on chapter 5 and bibliography of dissertation - Polishing touches on dissertation	Complete first draft
12. 20/3 - 2/4	Break	Send off first draft to supervisors and Dos by 2/4
13. 3/4 - 16/4	Markups and improvements on write-up	Send off second draft to supervisors and DoS
14. 17/4 - 1/5	Markups and improvements on write-up	- Ideally have gone through at least 3-4 iterations of write-up - Submit final draft write-up and source code
15. 2/5 -12/5	Buffer Time	

7 Resource Declaration

I will primarily be using my own laptop (Thinkpad T14, 2.1GHz CPU, 16GB RAM, 475GB Storage) and my desktop PC (3.6GHz CPU, GTX1660Ti GPU, 16GB RAM, 512GB SSD, 2TB HDD, Windows 10 OS and Linux Ubuntu 22.04) for programming and local testing. In the event that both machines fail, I would use MCS computers before I can repair my machines. I accept full responsibility for these two machines and I have made contingency plans to protect myself against hardware and/or software failure.

I will also have access to a machine with 4 GPUs hosted by the CST department, which I will also use to perform local training and testing if my machine is insufficient. I will be training and testing my neural networks using the University HPC Wilkes3 service, which professor Robert Mullins (robert.mullins@cl.cam.ac.uk) has granted me access to. I assume the availability of these two machines over the duration of my project.

I will be using Github for version control and backing up of source code, and I will be using Overleaf and local storage to write and store my local dissertation, as well as backing it up regularly on Google Drive.

References

[1] Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics.

[2] Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan, December 14-15 2017. International Workshop on Spoken Language Translation.

[3] 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, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[4] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[5] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[6] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.