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: Jiaxin Wang

CRSID: jw2117

College: Emmanuel

Overseers: (Initials) AFB/SK

Title of Project: A Recursive Recurrent Neural Network Decoder for Grammatical Error Correction

Date of submission: 8th Oct 2021

Will Human Participants be used? No

Project Originator: Jiaxin Wang

Signature: Jiaxin Wang

Project Supervisor: Dr Zheng Yuan
Dr Christopher Bryant

Signature: ZY CJBryant

Directors of Studies: Dr Thomas Sauerwald

Signature: Thomas Sauerwald

Special Resource Sponsor:

Signature: ----------

Special Resource Sponsor:

Signature: ----------

Above signatures to be obtained by the Student

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:

PROJECT PROPOSAL

Introduction

Statistical machine translation (SMT) is an approach to machine translation based on statistical models. Grammatical error correction (GEC), which is the task of producing grammatically correct text given potentially erroneous text while preserving its meaning, can be seen as a translation process from an erroneous source to a correct target. As such, it is possible to build an SMT system and apply it to the task of GEC.

A typical SMT system consists of four main components: The language model (LM), the translation model (TM), the reordering model and the decoder (Yuan, 2017). The LM computes the probability of a given sequence being valid. The TM builds a translation table which contains mappings of words/phrases between source and target corpora. The reordering model learns about phrase reordering of translation. The decoder finds a translation candidate who is most likely to be the translation of the source sentence. In terms of GEC, this would be the most probable correction to the original erroneous sentence.

A *recursive recurrent neural network* (R^2NN) was proposed by Liu et al. (2014) for SMT. It has features of both recursive neural networks and recurrent neural networks. This R^2NN network can be used to model the decoding process in SMT. A three-step training method was also given in the paper to train the R^2NN network.

This project aims to implement the proposed R^2NN decoder and build an SMT-based GEC system with it. The GEC system should aim to correct all types of errors, including grammatical, lexical, and orthographical errors. Considering that the original paper used data from IWSLT 2009 dialog task which could be out-of-date, this project will instead make use of the BEA-2019 shared task dataset, which is publicly available at (https://www.cl.cam.ac.uk/research/nl/bea2019st/), for training and testing purposes. Its performance will be evaluated against a baseline SMT system, namely Moses (Koehn et al., 2007). Moses is an open-source toolkit for SMT, and it has been used for the task of GEC (Yuan and Felice, 2013). When building the Moses baseline system, a language model is to be built and several figures such as the distortion score can be calculated. Since the proposed R^2NN decoder is to be used with a 5-gram language model and the training process requires distortion scores, it can make use of the calculated scores and the built language model when building Moses. After both SMT systems (i.e. Moses and R^2NN) are built, it is natural to evaluate the performance of the R^2NN decoder against Moses decoder.

Starting Point

The Part IB Computer Science Tripos course Artificial Intelligence [1] gives an introduction to neural networks and explains how forwarding and backpropagation works. The paper on R^2NN (Liu, et al., 2014) demonstrates the idea of combining recursive neural networks and recurrent neural networks, but implementation details or example codes are not provided. At the time of writing this proposal, I have no experience with using recursive neural networks or recurrent neural networks, which means I will need to study relevant materials for this project.

An example SMT system is available on Moses website[2]. By following the tutorial and build a Moses SMT system I should gain familiarity with the various models used in SMT.

The datasets I plan to use in this project are available on the BEA 2019 Shared Task website[3] for non-commercial purposes. The corpora provided are standardised, making it easy to perform pre-processing of data.

Project Structure

The project mainly consists of the following components:

1. Data preparation
2. Implementation of an SMT-based GEC system using Moses
    a. Language model training
    b. Translation and reordering model training
    c. Testing
3. Implementation of an SMT-based GEC system using R^2NN model
    a. Phrase pair embedding implementation
        i. Implementation of a one-hidden-layer neural network
        ii. Implementation of a recurrent neural network
        iii. Translation confidence score training
    b. R^2NN decoder implementation
        i. R^2NN model implementation
        ii. R^2NN model training
    c. Testing
4. Evaluation
    a. Comparison of performances of Moses and the R^2NN model

Success Criteria

Core Tasks

[1] https://www.cl.cam.ac.uk/teaching/2021/ArtInt/
[2] http://www.statmt.org/moses/index.php?n=Main.HomePage
[3] https://www.cl.cam.ac.uk/research/nl/bea2019st/

This project will be successful if the following have been achieved.

1. Data is pre-processed for the use of training the models.
2. A baseline SMT-based GEC system is built.
3. A one-hidden-layer neural network is built to learn the translation confidence score.
4. A recurrent neural network is built to learn the translation confidence score.
5. An R^2NN model is implemented.
6. An SMT-based GEC system making use of the R^2NN decoder is built.
7. A comparison between the performances of the two systems is performed.

POSSIBLE EXTENSIONS

1. Train both GEC systems on an alternative GEC dataset and evaluate their performances
2. Experiment with different language models for both GEC systems
3. Experiment with alternative phrase pair embeddings for the GEC system using R^2NN

Timetable

9th Oct – 22nd Oct 2021

Background research on: statistical machine translation (SMT), Moses SMT, recursive neural networks, recurrent neural networks. Get familiar with relevant libraries.

Friday 15th Oct 2021 (5pm)

Final Proposal Deadline

23rd Oct – 5th Nov 2021

Build a Moses SMT system. Prepare the data for training the SMT.

6th Nov – 19th Nov 2021

Begin to implement phrase pair embedding. Build a one-hidden-layer neural network to learn translation confidence.

20th Nov – 3rd Dec 2021

Continue implementing phrase pair embedding. Build a recurrent neural network to learn translation confidence.

4th Dec – 21st Jan 2022 [Christmas Break]

Build an R^2NN decoder. Start with building a recursive neural network, then add recurrent input vectors to it. Train the parameters of the R^2NN decoder.

22nd Jan – 4th Feb 2022

Evaluate the performance of Moses SMT and the R^2NN-based SMT. Write the progress report.

Friday 4th Feb 2022 (12 noon)

Progress Report Deadline

5th Feb – 18th Feb 2022

Start with possible extensions if the core project is finished. Begin writing the dissertation.

19th Feb – 4th Mar 2022

Continue working on extensions if the core project is finished and writing the dissertation.

5th Mar – 18th Mar 2022

Continue writing the dissertation.

19th Mar – 22nd Apr 2022 [Easter Break]

Continue writing the dissertation. Send in the draft for review.

23rd Apr – 6th May 2022

Submit the dissertation.

Friday 13th May 2022 (12 noon)

Dissertation Deadline

Resources Declaration

For this project I will be using my personal laptop (1.9GHz quad-core Intel Core i7 processor, 8GB RAM, Windows 10). I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure. I will back up the entire project on Google Drive and use git for revision control. I might use a cloud GPU (available in Google Colab) to accelerate the speed of training the neural networks if needed.

The dataset from BEA-2019 shared task, which will be used in this project, can be found on (https://www.cl.cam.ac.uk/research/nl/bea2019st/). The training time for small models using only the public data from BEA-2019 shared task is expected to be a few hours, and for bigger models (if I plan to use additional data) it could be a few

days. I should keep this in mind and leave enough time for training when implementing my project.

References

Liu, S., Yang, N., Li, M. & Zhou, M., 2014. *A Recursive Recurrent Neural Network for Statistical Machine Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, Evan Herbst, 2007. *Moses: Open Source Toolkit for Statistical Machine Translation.* Association for Computational Linguistics.

Yuan, Z., 2017. *Grammatical error correction in non-native English*.

Yuan, Z., Felice, M., 2013. *Constrained grammatical error correction using Statistical Machine Translation,* Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen and Ted Briscoe. 2019. *The BEA-2019 Shared Task on Grammatical Error Correction.* In Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications (BEA-2019), pp. 52–75, Florence, Italy, August. Association for Computational Linguistics.