深圳河套學院
Shenzhen Loop Area Institute

Natural Language Processing and Large Language Models

Autumn, 2025

深圳河套學院
Shenzhen Loop Area Institute

Midterm and Final Project:

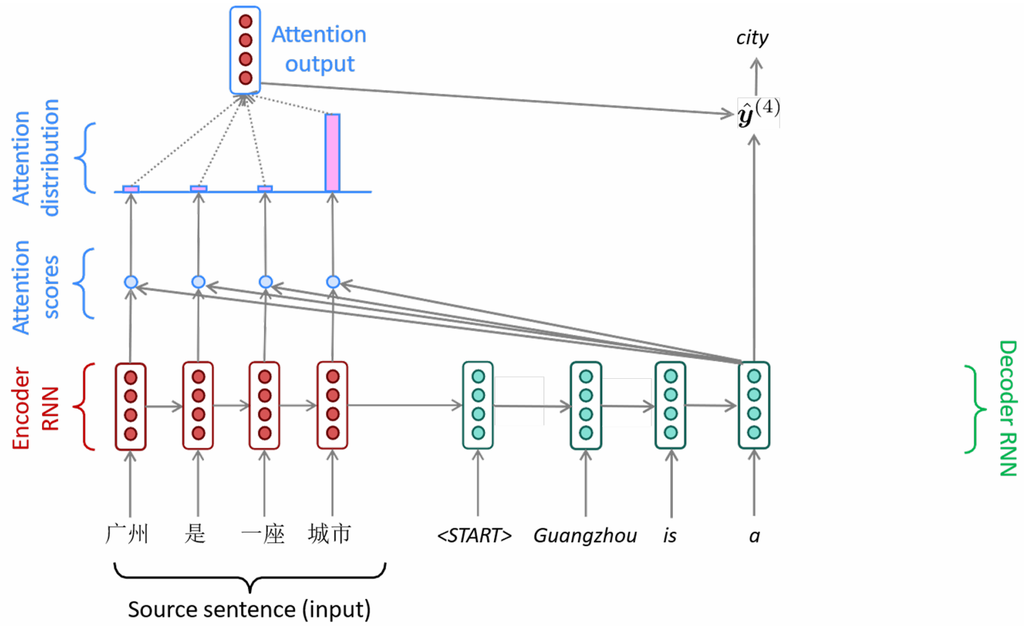
Machine Translation between Chinese and English

The Goal: To implement Chinese–English machine translation using RNN and Transformer model, and compare their performance and architectural differences.

Outline

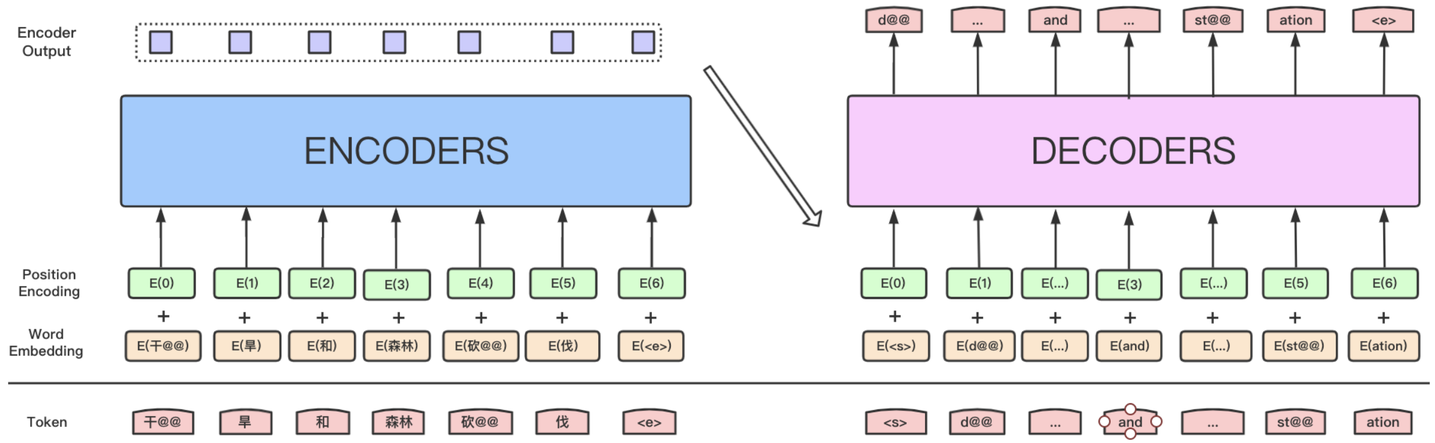
- Assignment Requirements
- Datasets Description
- Metrics Description
- Submission Requirements
- Scoring Criteria
- Reference

Assignment Requirements



1. **RNN-based NMT:** Build and train a RNN-based NMT, including:
   - Model: Implement a model using either GRU or LSTM, with both the encoder and decoder consisting of two unidirectional layers.
   - Attention mechanism: Implement the attention mechanism and investigate the impact of different alignment functions—such as dot-product, multiplicative, and additive—on model performance.
   - Training policy: Compare the effectiveness of Teacher Forcing and Free Running strategies.
   - Decoding policy: Compare the effectiveness of greedy and beam-search decoding strategies.

Assignment Requirements



2. Transformer-based NMT: Build and train a Transformer-based NMT, including:

- From scratch: Build a Chinese-to-English translation model using the Transformer architecture with an encoder-decoder structure and train it from scratch.
- Architectural Ablation: Train from scratch and compare the effects of different position embedding schemes (e.g., absolute vs. relative) and normalization methods (e.g., LayerNorm vs. RMSNorm).
- Hyperparameter Sensitivity: Train from scratch with varying batch sizes, learning rates, and model scales to assess their impact on translation performance.
- From pretrained language model: Fine-tune a pretrained language model (e.g., T5) to adapt it for neural machine translation and evaluate its performance in comparison with models trained from scratch.

Assignment Requirements

3. **Analysis and Comparison:** Conduct a comprehensive comparison between the RNN-based and Transformer-based NMT models in terms of:
   - Model architecture (e.g., sequential vs. parallel computation, recurrence vs. self-attention),
   - Training efficiency (e.g., training time, convergence speed, hardware requirements),
   - Translation performance (e.g., BLEU score, fluency, adequacy),
   - Scalability and generalization (e.g., handling long sentences, low-resource scenarios),
   - Practical trade-offs (e.g., model size, inference latency, ease of implementation).

Outline

- Assignment Requirements
- Datasets Description
- Metrics Description
- Submission Requirements
- Scoring Criteria
- Reference

Datasets Introduction

- **Description:** The compressed package contains four JSONL files, corresponding respectively to the small training set, large training set, validation set, and test set, with sizes of 100k, 10k, 500, and 200 samples. Each line in a JSONL file contains one parallel sentence pair. The final model performance will be evaluated based on results on the test set.

- **Data access:** https://piazza.com/class_profile/get_resource/mfzcdlplb7n1n0/mifeas10tj31pl.

Note: If computational resources are limited, you may train using only 10k parallel sentence pairs from the small training set. However, you are encouraged to explore training with the large training set.

Datasets Preprocessing

- **Data Cleaning**: Remove illegal characters and filter out rare words; filter or truncate excessively long sentences.
- **Tokenization**: Split input sentences into tokens, where each substring carries relatively complete semantic meaning to facilitate learning meaningful embedding representations. For English, words are naturally separated by spaces and punctuation. You can directly use tokenization tools such as NLTK or statistical subword segmentation methods like BPE (Byte-Pair Encoding) or WordPiece; For Chinese, use dedicated word segmentation tools—for example, Jieba (lightweight) or HanLP (larger but higher accuracy).
- **Vocabulary Construction**: Build a statistical vocabulary from the tokenized data. Consider filtering out low-frequency words to prevent the vocabulary from becoming excessively large.
- **Word Embedding Initialization**: It is recommended to initialize embeddings with pretrained word vectors and allow them to be fine-tuned during training.
- **Implementation Flexibility**: In addition to the provided baseline implementation (Jieba for Chinese + space-based tokenization for English), students are encouraged to explore more advanced tokenization strategies.

Outline

- Assignment Requirements
- Datasets Description
- Metrics Description
- Submission Requirements
- Scoring Criteria
- Reference

Metrics

- BLEU

$$\text{precision}_n = \frac{\sum_{C \in corpus} \sum_{\text{n-gram} \in C} \text{count-in-reference}_{\text{clip}}(\text{n-gram})}{\sum_{C \in corpus} \sum_{\text{n-gram} \in C} \text{count}\ (\text{n-gram})}$$

$$\text{BLEU-4} = \min\left(1, \frac{\text{output-length}}{\text{reference-length}}\right) \prod_{i=1}^{4} \text{precision}_i$$

Outline

- Assignment Requirements
- Datasets Description
- Metrics Description
- **Submission Requirements**
- Scoring Criteria
- Reference

Submission Requirements

- **Source code:** You may submit your source code and checkpoints to a GitHub repository and provide a one-click inference script named “inference.py” to facilitate testing with your model.
- **Project report:** The project report (in PDF format, named “ID_name.pdf”, for example, "250010001_Zhang San.pdf") must include a description of the model architecture, an explanation of the code implementation and completion process, an analysis of experimental results (with the clarification that grading will not be based on the final BLEU score), as well as visualization-based analysis and personal reflections. Please clearly specify the code repository URL in a designated section on the first page of your report.
- **Final submission:** You only need to submit the project report to https://piazza.com platform.
- **DDL**: December 28th

Presentation

- **Presentation duration:** 10 mins presentation & 5 mins QA (15 mins for each group)
- **Group composition:** The class will be divided into 18 groups (~7 students each). You may form your own team; otherwise, we will randomly assign teammates for you. Each group will deliver one joint presentation, but every student must submit their own individual project report by the deadline.
- **DDL**: December 28th

Outline

- Assignment Requirements
- Datasets Description
- Metrics Description
- Submission Requirements
- Scoring Criteria
- Reference

Scoring Criteria

Content	Score
RNN-based NMT Implementation & Experiments	15%
Transformer-based NMT Implementation & Experiments	25%
Comparative Analysis & Discussion	5%
Project Report	5%
Presentation	50%

Outline

- Assignment Requirements
- Datasets Description
- Metrics Description
- Submission Requirements
- Scoring Criteria
- Reference

Reference

- Seq2Seq Machine Translation Tutorial: https://docs.pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html
- Tokenization Tools:
  1. Jieba (Chinese word segmentation tool): https://github.com/fxsjy/jieba;
  2. SentencePiece (English and multilingual subword tokenization tool): https://github.com/google/sentencepiece
- Papers:
  1. Attention is All You Need (Vaswani et al., 2017)
  2. Neural Machine Translation by Jointly Learning to Align and Translate (Dzmitry et al., 2015)
  3. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (Colin et al., 2020)
- Pretrained Language Model Weight: https://huggingface.co/google-t5/t5-base/tree/main