Today’s Schedule

• History of Machine Translation
• A2: the big picture
  • Task 1: BLEU Score
  • Task 2: Encoder-Decoder
  • Task 3: Training and Testing
• Marking: How each tasks are evaluated
Today’s Schedule

• History of Machine Translation
• A2: the big picture
  • Task 1: BLEU Score
  • Task 2: Encoder-Decoder
  • Task 3: Training and Testing
• Marking: How each tasks are evaluated
The early years of MT

When I look at an article in Russian, I say: "This is really written in English, but has been coded in some strange symbols. I will now proceed to decode."

[Warren Weaver, 1947]

Rough History of Machine Translation Research

It turns out MT is difficult

Beginning of MT

It turns out MT is difficult

Beginning of MT

Disinterest in MT research (esp. in USA after ALPAC report)

Rough History of Machine Translation Research

It turns out MT is difficult

Diversity and the number of installed systems for MT had increased (esp. after ALPAC report)

Beginning of MT

Rough History of Machine Translation Research

Example-based Machine Translation (EBMT) [Nagao, 1981]

Rough History of Machine Translation Research

Example-based Machine Translation (EBMT) [Nagao, 1981]

Statistical Machine Translation (SMT) [Brown et al., 1993]

Rough History of Machine Translation Research

Example-based Machine Translation (EBMT) [Nagao, 1981]
Statistical Machine Translation (SMT) [Brown et al., 1993]
Phrase-based SMT [Koehn et al., 2003]

Rough History of Machine Translation Research

Example-based Machine Translation (EBMT) [Nagao, 1981]

Phrase-based SMT [Koehn et al., 2003]

Statistical Machine Translation (SMT) [Brown et al., 1993]

Neural Machine Translation (NMT) [Sutskever et al., 2014]

Example-based Machine Translation (EBMT) [Nagao, 1981]

Statistical Machine Translation (SMT) [Brown et al., 1993]

Phrase-based SMT [Koehn et al., 2003]

Neural Machine Translation (NMT) [Sutskever et al., 2014]

Today’s Schedule

• History of Machine Translation
• A2: the big picture
• Task 1: Preprocess Inputs
• Task 2: Compute n-gram counts
• Marking: How each tasks are evaluated
A2 – Historically: Noisy Channel Model

We need a language model, a translation model, and a decoder.

\[ E^* = \arg \max_E P(F|E)P(E) \]
Currently:

Noisy Channel Model

We need an **encoder, a decoder, and a search algorithm.**
Task 1. BLEU Score

File: “a2_bleu_score.py”

4 functions:

1. grouper(seq, n):
2. n_gram_precision(reference, candidate, n):
3. brevity_penalty(reference, candidate):
4. BLEU_score(reference, hypothesis, n):
Task 1. BLEU Score

BLEU Score is calculated like in the lecture slides.

Notes:
- No capping.
- Only 1 reference and 1 candidate at a time.
- Don’t include SOS and EOS tokens in the calculations.
Task 2. Encoder and Decoder

File: “a2_encoder_decoder.py”

4 Classes:

1. Encoder(EncoderBase):
2. DecoderWithoutAttention(DecoderBase):
3. DecoderWithAttention(DecoderWithoutAttention):
4. EncoderDecoder(EncoderDecoderBase):
Notes:

• Does not follow 100% from the slides. Look at the handouts!

• To correctly implement the code you’ll need to look and understand other files (a2_abcs.py).

  DO NOT CHANGE ANYTHING IN THE FILE.
Task 3. Train and Test

File: “a2_training_and_testing.py”

3 Functions:

1. def train_for_epoch
2. def compute_batch_total_bleu
3. def compute_average_bleu_over_dataset
General Development

- You will not need any imports that are not already specified.
- Do not change any files that come implemented.
- Check Piazza for updates to the code.
- When dealing with matrices (i.e., tensors) in the neural network, use vectorized/matrix operations.
(A large) Portion of it will be auto-marked.

- Your code will be tested individually (by file).
- Your code must adhere to the specifications for function calls to work.
- Do **NOT** hardcode any paths.
- It must work on CDF (test your code).
- There is a GPU component which will be tough to finish on time if you leave it to the last second.
Marking (Cont)

Initially:

Correct Code: A B C D

Your Code: A B C D
If we are testing “A”:

Correct Code: A ▶️ B ▶️ C ▶️ D

Your Code: A ▶️ B ▶️ C ▶️ D
If we are testing “B”:

Correct Code: A B C D

Your Code: A B C D