Neural Machine Translation Decoding

Philipp Koehn

8 October 2020
Inference

- Given a trained model

  ... we now want to translate test sentences

- We only need execute the "forward" step in the computation graph
Word Prediction

Eyi → Embed → Embed → Output Word Embeddings

yii → Output Word

tti → Softmax → Output Word Prediction

Si → RNN → RNN → Decoder State

Ci → Input Context
Selected Word

Output Word Embeddings

Output Word

Output Word Prediction

Decoder State

Input Context

the
cat
cat
this
this
of
of
fish
fish
there
there
dog
dog
these
these

Embed

Embed

Softmax

RNN

RNN

Eyi

yi

yi

ti

Si

Ci

Ey
Embedding

E \( y_i \)  
\( y_i \)  
\( t_i \)  
\( S_i \)  
\( C_i \)  
Output Word Embeddings  
Output Word  
Output Word Prediction  
Decoder State  
Input Context  

\( y_i \) the
ca
this
of
fish
there
dog
these
Distribution of Word Predictions

\[ y_i \]

- the
- cat
- this
- of
- fish
- there
- dog
- these
Select Best Word

The best word is "the".
Select Second Best Word

\[ y_i \quad \text{the} \quad \text{cat} \quad \text{this} \quad \text{of} \quad \text{fish} \quad \text{there} \quad \text{dog} \quad \text{these} \]

- The
- This
Select Third Best Word

\[ y_i \]

- the
- cat
- this
- of
- fish
- there
- dog
- these

---

Philipp Koehn
Machine Translation: Neural Machine Translation Decoding
8 October 2020
Use Selected Word for Next Predictions

$y_i$ the cat this this of fish there there dog these these

the this these
Select Best Continuation

$y_i$ the
cat
this
of
fish
there
dog
these
cat
the
dog
these
this
of
fish
there
dog
these
Select Next Best Continuations

The diagram illustrates a process for selecting the next best continuations in a translation task. The input sequence $y_i$ consists of words such as 'the', 'cat', 'this', 'of', 'fish', 'there', 'dog', and 'these'. The system then predicts possible continuations, such as 'cat', 'this', 'these', 'cats', 'dog', and 'cats'. The diagram shows the probabilistic nature of these predictions, with arrows indicating possible continuations from the input sequence.
Continue...
Beam Search
Best Paths
Beam Search Details

- Normalize score by length
- No recombination (paths cannot be merged)
ich glaube aber auch, er ist clever genug um seine Aussagen vage genug zu halten, so dass sie auf verschiedene Art und Weise interpretiert werden können.
ensembling
Ensembling

- Train multiple models

- Say, by different random initializations

- Or, by using model dumps from earlier iterations

  (most recent, or interim models with highest validation score)
Decoding with Single Model

E \( y_i \)  
\[ \text{Embed} \]

\( y_i \)  
\[ \text{Embed} \]

Output Word Embeddings

Output Word

Output Word Prediction

Decoder State

RNN\( t_i \)  
\[ \text{Softmax} \]

\( s_i \)  
\[ \text{RNN} \]

Input Context

RNN\( c_i \)

y_i: the
cat
this
of
fish
there
dog
these

Philipp Koehn
Machine Translation: Neural Machine Translation Decoding
8 October 2020
### Combine Predictions

<table>
<thead>
<tr>
<th></th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
<th>Model Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>0.54</td>
<td>0.52</td>
<td>0.12</td>
<td>0.29</td>
<td>0.37</td>
</tr>
<tr>
<td>cat</td>
<td>0.01</td>
<td>0.02</td>
<td>0.33</td>
<td>0.03</td>
<td>0.10</td>
</tr>
<tr>
<td>this</td>
<td>0.11</td>
<td>0.01</td>
<td>0.06</td>
<td>0.14</td>
<td>0.08</td>
</tr>
<tr>
<td>of</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>0.08</td>
<td>0.02</td>
</tr>
<tr>
<td>fish</td>
<td>0.00</td>
<td>0.12</td>
<td>0.15</td>
<td>0.00</td>
<td>0.07</td>
</tr>
<tr>
<td>there</td>
<td>0.03</td>
<td>0.03</td>
<td>0.00</td>
<td>0.07</td>
<td>0.03</td>
</tr>
<tr>
<td>dog</td>
<td>0.00</td>
<td>0.00</td>
<td>0.05</td>
<td>0.20</td>
<td>0.06</td>
</tr>
<tr>
<td>these</td>
<td>0.05</td>
<td>0.09</td>
<td>0.09</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Ensembling

• Surprisingly reliable method in machine learning

• Long history, many variants:
  - bagging, ensemble, model averaging, system combination, ...

• Works because errors are random, but correct decisions unique
reranking
Right-to-Left Inference

- Neural machine translation generates words right to left (L2R)

  the → cat → is → in → the → bag → .

- But it could also generate them right to left (R2L)

  the ← cat ← is ← in ← the ← bag ← .

**Obligatory notice:** Some languages (Arabic, Hebrew, ...) have writing systems that are right-to-left, so the use of “right-to-left” is not precise here.
Right-to-Left Reranking

- Train both L2R and R2L model

- Score sentences with both
  ⇒ use both left and right context during translation

- Only possible once full sentence produced → re-ranking
  1. generate n-best list with L2R model
  2. score candidates in n-best list with R2L model
  3. chose translation with best average score
Inverse Decoding

- Recall: Bayes rule
  \[ p(y|x) = \frac{1}{p(x)} p(x|y) p(y) \]

- Language model \( p(y) \)
  - trained on monolingual target side data
  - can already be added to ensemble decoding

- Inverse translation model \( p(x|y) \)
  - train a system in the reverse language direction
  - used in reranking
Reranking

- Several models provide a score each
  - regular model
  - inverse model
  - right-to-left model
  - language model

- These scores could be just added up

- Typically better: weighting the score to optimize translation quality
Training Reranker

**Training**
- Training input sentences
  - Base model
    - Decode
    - N-best list of translations
      - Reference translations
      - Additional features
        - Combine
          - Labeled training data
            - Learn
              - Reranker

**Testing**
- Test input sentence
  - Base model
    - Decode
    - N-best list of translations
      - Additional features
        - Combine
          - Reranker
            - Translation
Learning Reranking Weights

• Minimum error rate training (MERT)
  – optimize one weight at a time, leave others constant
  – check how different values change n-best lists
  – only a some threshold values change ranking
    → can be done exhaustively

• Pairwise Ranked Optimization (PRO)
  – for each sentence in tuning set
  – for each pair of translations in n-best list
  – check which one is a better translation, leaving everything else fixed
  – create a training example
    ( difference in feature values → { better, worse } )
  – train linear classifier that learns weights for each feature

• This has not been explored much in neural machine translation
Lack of Diversity

Translations of the German sentence
Er wollte nie an irgendeiner Art von Auseinandersetzung teilnehmen.

He never wanted to participate in any kind of confrontation.
He never wanted to take part in any kind of confrontation.
He never wanted to participate in any kind of argument.
He never wanted to take part in any kind of argument.
He never wanted to participate in any sort of confrontation.
He never wanted to take part in any sort of confrontation.
He never wanted to participate in any kind of controversy.
He never wanted to take part in any kind of controversy.
He never intended to participate in any kind of confrontation.
He never intended to take part in any kind of confrontation.
He never wanted to take part in some sort of confrontation.
He never wanted to take part in any sort of controversy.
Increasing Diversity

• Monte Carlo decoding
  – no beam search, i.e., beam size 1
  – when selecting words to extend the beam ...
  – ... do not select the top choice
  – ... do select word randomly based on their probability
  – 10% chance to choose a word with 10% probability

• Diversity bias term
  – extension of regular beam search
  – add a cost for extending a hypothesis based on rank of word choice
    * most probable word: no cost
    * second most probable word: cost $c$
    * third most probable word: cost $2c$
  ⇒ prefer to extend many different hypotheses
constraint decoding
Specifying Decoding Constraints

• Overriding the decisions of the decoder

• Why?
  ⇒ translations have followed strict terminology
  ⇒ rule-based translation of dates, quantities, etc.
  ⇒ interactive translation prediction
XML Schema

The `<x translation="Router"> router </x>` is `<wall/>

a model `<zone>` Psy X500 Pro `<zone>`.

- The XML tags specify to the decoder that
  - the word `router` to be translated as `Router`
  - `The router is`, to be translated before the rest (`<wall/>`)
  - brand name `Psy X500 Pro` to be translated as a unit (`<zone>`, `</zone>`)
The <x translation="Router"> router </x> is a model Psy X500 Pro.

- Satisfying constraints typically costly (overriding model-best choices)
- Solution: separate beam, based on how many constraints satisfied
Grid Search

Beam 0

Beam 1
Grid Search

Beam 0

Beam 1

Beam 2
• Two hypothesis that fulfill the constraint
  – first one has relevant input words in attention focus
  – second one does not have relevant input words in attention focus
Considering Alignment

• When satisfying a constraint...
  – minimum amount of attention needs to be paid to source
  – use alignment scores as additional cost

• When not satisfying a constraint...
  – block out attention to words not covered by constraint