Machine Translation and Sequence-to-sequence Models

http://phontron.com/class/mtandseq2seq2017/

Graham Neubig

Carnegie Mellon University
CS 11-731
What is Machine Translation?

*kare wa ringo wo tabeta.*

He ate an apple.
What are Sequence-to-sequence Models?

Sequence-to-sequence Models

Machine translation:
kare wa ringo wo tabeta → he ate an apple

Tagging:
he ate an apple → PRN VBD DET PP

Dialog:
he ate an apple → good, he needs to slim down

Speech Recognition:

→ he ate an apple

And just about anything...:
1010000111101 → 00011010001101
Why MT as a Representative?

Useful!

KUSHINIKIZA! Google Translate SAVES BABY in Irish roadside birth

Do no evil? We literally save lives now

13 Feb 2015 at 12:01, John Leyden

Quick-thinking Irish paramedics turned to Google Translate to communicate with a pregnant woman who spoke Swahili, allowing her to safely give birth.

Source: The Register

Imperfect...

Global MT Market Expected To Reach $983.3 Million by 2022

Source: Grand View Research
MT and Machine Learning

**Big Data!** Billions of words for major languages
... but little for others

**Well-defined, Difficult Problem!**
Use for algorithms, math, etc.

**Algorithms Widely Applicable!**
MT and Linguistics

Trina Baker is a good person

Morphology!  이니까요 is a variant of  이다 (to be)

Syntax!  should keep subject together

Semantics!  “Trina” is probably not a man...

... and so much more!
Class Organization
Class Format

• Before class:
  • Read the assigned material
  • Ask questions via web (piazza/email)

• In class:
  • Take a small quiz about material
  • Discussion/questions
  • Pseudo-code walk
  • Programming (TAs/Instructor will supervise)
Assignments

• **Assignment 1:** Create a neural sequence-to-sequence modeling system. Turn in code to run it, and a short 1-2 page report.

• **Assignment 2:** Create a symbolic sequence-to-sequence modeling system. Similarly turn in code/report.

• **Final project:** Come up with an interesting new idea and test it.
Assignment Instructions

- Bring your computer to every class and make a Github account.
- We recommend you implement in the following libraries:
  - DyNet: for neural networks (C++ or Python)
  - OpenFST: for transducers, if you use them (C++)
  - pyfst: for transducers in Python
- It is OK to work in small groups up to 3, particularly for the final project. If you do so, please use a shared git repository and commit the code that you write, and in reports note who did what part of the project.
Class Grading

- Short quizzes: 20%
- Assignment 1: 20%
- Assignment 2: 20%
- Final Project: 40%
Class Plan

1. Introduction (Today): 1 class
2. Language Models: 4 classes
3. Neural MT: 2 classes
3. Evaluation/Data: 2 classes
4. Symbolic MT: 4 classes
5. Algorithms: 2 classes
7. Advanced Topics: 11 classes
8. Final Project Discussion: 2 classes

#1 Due
#2 Due
Project Due
Guest Lectures

• Bob Frederking: Knowledge-based Translation
• LP Morency: Something Multi-modal

(Date TBD)
Statistical Machine Translation
Statistical Machine Translation

F = kare wa ringo wo tabeta .
E = He ate an apple .

Probability model: \( P(E|F;\Theta) \)

Parameters
Problems in SMT

• **Modeling:** How do we define \( P(E|F; \Theta) \)?

• **Learning:** How do we learn \( \Theta \)?

• **Search:** Given \( F \), how do we find the highest scoring translation?

\[
E' = \arg \max_E P(E|F; \Theta)
\]

• **Evaluation:** Given \( E' \) and a human reference \( E \), how do we determine how good \( E' \) is?
Part 1: Neural Models
Given multiple candidates, which is most likely as an English sentence?

- $E_1 = \text{he ate an apple}$
- $E_2 = \text{he ate an apples}$
- $E_3 = \text{he insulted an apple}$
- $E_4 = \text{preliminary orange orange}$

• Definition of language modeling
• Count-based n-gram language models
• Evaluating language models
• **Implement**: n-gram language model
Language Models 2: Log-linear Language Models

- Log-linear language models
- Stochastic gradient descent
- Features for language modeling
- **Implement**: Log-linear language model
Language Models 3: Neural Networks and Feed-forward LMs

- Neural networks and back-propagation
- Feed-forward neural language models
- Mini-batch training
- **Implement:** Feed-forward LM

```plaintext
<s>    <s>     this     is        a      pen    </s>
```
Language Models 4: Recurrent LMs

- Recurrent neural networks
- Vanishing Gradient and LSTMs/GRUs
- Regularization and dropout

**Implement:** Recurrent neural network LM
Neural MT 1: Encoder-decoder Models

- Encoder-decoder Models
- Searching for hypotheses
- Mini-batched training
- **Implement**: Encoder-decoder model
Neural MT 2: Attentional Models

<table>
<thead>
<tr>
<th>could</th>
<th>you</th>
<th>recommend</th>
<th>an</th>
<th>inexpensive</th>
<th>restaurant</th>
<th>?</th>
<th>&lt;s&gt;</th>
</tr>
</thead>
</table>

- Attention in its various varieties
- Unknown word replacement
- Attention improvements, coverage models
- **Implement**: Attentional model
Data and Evaluation
Creating Data

- Preprocessing
- Document harvesting and crowdsourcing
- Other tasks: dialog, captioning
- Implement: Find/preprocess data
Evaluation

- Human evaluation
- Automatic evaluation
- Significance tests and meta-evaluation
- **Implement**: BLEU and measure correlation
Symbolic Translation Models
Symbolic Methods 1: IBM Models

- The IBM/HMM models
- The EM algorithm
- Finding word alignments
- **Implement**: Word alignment

太郎 が 花子 を 訪問 し た。
taro visited hanako.

太郎 が 花子 を 訪問 し た。
taro visited hanako.
Symbolic Methods 2: Monotonic Symbolic Models

- Models for sequence transduction
- The Viterbi algorithm
- Weighted finite-state transducers
- **Implement**: A part-of-speech tagger
Symbolic Methods 3: Phrase-based MT

F = \textit{watashi wa CMU de kouen wo okonaimasu}.

E = \textit{I will give a talk at CMU}.

- Phrase extraction and scoring
- Reordering models
- Phrase-based decoding
- **Implement**: Phrase extraction or decoding
Symbolic Methods 4: Tree-based MT

- Graphs and hyper-graphs
- Synchronous context free grammars
- Tree substitution grammars
- Implement: Search over hyper-graphs

give a talk at CMU

CMU de kouen wo okonaimasu

N_0 PP_0-1 VP_0-5 PP_2-3 VP_2-5 VP_4

x2 at x1

x2 x1

give

a talk at CMU

give a talk at CMU
Algorithms
Algorithms 1: Search

- Beam search and cube pruning
- Hypothesis recombination
- Future costs, A* search
- Implement: Beam search
# Algorithms 2: Parameter Optimization

<table>
<thead>
<tr>
<th></th>
<th>LM</th>
<th>TM</th>
<th>RM</th>
<th>Highest</th>
</tr>
</thead>
<tbody>
<tr>
<td>○ Taro visited Hanako</td>
<td>0.2*-4</td>
<td>0.3*-3</td>
<td>0.5*-1</td>
<td>-2.2</td>
</tr>
<tr>
<td>✗ the Taro visited the Hanako</td>
<td>0.2*-5</td>
<td>0.3*-4</td>
<td>0.5*-1</td>
<td>-2.7</td>
</tr>
<tr>
<td>✗ Hanako visited Taro</td>
<td>0.2*-2</td>
<td>0.3*-3</td>
<td>0.5*-2</td>
<td>-2.3</td>
</tr>
</tbody>
</table>

- Loss functions
- Deciding the hypothesis space
- Optimization criteria
- **Implement**: Optimization of NMT or PBMT
Advanced Topics
Other Sequence-to-sequence Tasks

he ate an apple → PRN VBD DET PP
he ate an apple → good, he needs to slim down
he ate an apple → he ate an apple

- Case studies about task-specific models
  - Consistency constraints in tagging
  - Diversity objectives in dialog
  - Dynamic programming in speech
- Implement: Try models on other tasks
Ensembling/System Combination

- Ensembles and distillation
- Post-hoc hypothesis combination
- Reranking
- **Implement**: Ensembled decoding
Hybrid Neural-symbolic Models

- Symbolic models with neural components
- Neural models with symbolic components
- **Implement**: Implement lexicons in NMT or neural feature functions
Multi-lingual and Multi-task Learning

- Learning for multiple tasks
- Learning for multiple languages
- **Implement**: Implement a multi-lingual neural system
Subword Models

reconstructed

\[ \text{re} + \text{construct} + \text{ed} \]

- Character models
- Subword models
- Morphology models

**Implement**: Implement subword splitting
Document Level Models

- Document level modeling
- Document level evaluation
- Stream decoding
- **Implement**: Document level measures
For Next Class
Homework

• Read n-gram language modeling materials

• Get software working on your machine (doing all at once may be more efficient?)
  • By Thursday 1/19: Python
  • By Tuesday 1/24: Numpy
  • By Thursday 1/26: DyNet