Machine Translation

Tom Ko

Ancient translation

- The Rosetta stone

Ancient Egyptian

Demotic

Ancient Greek

Ancient translation

- The Rosetta stone

The trend in machine translation (MT)

- Rule-based MT
- Statistical MT
- Neural MT

Rule-based MT

Figure 25.3 The Vauquois (1968) triangle.

Direct Translation

- Word-by-word processing
- Bilingual dictionary

Direct Translation

- Word-by-word processing
- Bilingual dictionary

Input:	Mary didn't slap the green witch
After 1: Morphology	Mary DO-PAST not slap the green witch
After 2: Lexical Transfer	Maria PAST no dar una bofetada a la verde bruja
After 3: Local reordering	Maria no dar PAST una bofetada a la bruja verde
After 4: Morphology	Maria no dió una bofetada a la bruja verde

Figure 25.6 An example of processing in a direct system

(25.10) Mary didn't slap the green witch

Maria no dió una bofetada a la bruja verde

Mary not gave a slap to the witch green

Languages differ in structural ways

- 成龍 到 香港 買東西
- Jackie Chan went shopping in Hong Kong

- Direct approach is too focused on individual words.

Transfer approach

- Alter the structure of the source language to make it conform to the rules of the target language.

Figure 25.10 A simple transformation that reorders adjectives and nouns

Transfer approach

Figure 25.12 The result of Yamada and Knight's (2001) syntactic transformations from English order (SVO) to Japanese order (SOV) for the sentence *He adores listening to music* (*kare ha ongaku wo kiku no ga daisuki desu*). This transform would require rules for moving verbs after their NP and VP complements, and changing prepositions to postpositions.

The Interlingua approach

- A distinct set of rules is needed for each pair of languages in the previous approaches.
- Can we extract the meaning of the source language and expressing that meaning in the target language?
- We need a representation scheme.

The Interlingua approach

$$\begin{bmatrix} \text{EVENT} & \text{SLAPPING} \\ \text{AGENT} & \text{MARY} \\ \text{TENSE} & \text{PAST} \\ \text{POLARITY} & \text{NEGATIVE} \\ \text{THEME} & \begin{bmatrix} \text{WITCH} & \\ \text{DEFINITENESS} & \text{DEF} \\ \text{ATTRIBUTES} & \begin{bmatrix} \text{HAS-COLOR} & \text{GREEN} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 25.14 Interlingual representation of *Mary did not slap the green witch*.

Rule-based MT approaches

- Direct approach
  - Chinese: 守 株 待 兔
  - English: defend the tree and wait for a rabbit
- Transfer approach
  - To overcome the structural differences.
  - English: waiting for a rabbit under a tree
- Interlingua approach
  - English: a lazy living style

How details we should translate?

- “a wild goose chase”
- 追逐一只野鹅
- 徒劳无功的事
- 追逐一只野生有长脖子短腿在水中生活的鸟类

Statistical MT

- A good translation should be both faithful to the source language and natural as an utterance in the target language.

$$\text{best-translation } \hat{T} = \text{argmax}_T \text{ faithfulness(T,S) fluency(T)}$$

- Factorize the problem

$$\begin{aligned}
\hat{E} &= argmax_E P(E|F) \\
&= argmax_E \frac{P(F|E)P(E)}{P(F)} \\
&= argmax_E \underbrace{P(F|E)}_{\text{Translation model}} \underbrace{P(E)}_{\text{Language model}}
\end{aligned}$$

The translation model

- The job of the translation model, given an English sentence *E* and a foreign sentence *F*, is to assign a probability that *E* generates *F*.

- How to learn the translation model P(F|E)?
  - We need a large amount of parallel data.

Som a l'any 50 abans de Crist. Tota la Gàl·lia és ocupada pels romans... Tota? No! Un llogaret del Nord habitat per gals indomables rebutja una i altra vegada ferotgement l'invassor. La vida doncs no és gens planera per als legionaris romans dels petits campaments de Babaòrum, Aquàrium, Laundànum i Petibònum...

The year is 50 B.C. Gaul is entirely occupied by the Romans. Well, not entirely... One small village of indomitable Gauls still holds out against the invaders. And life is not easy for the Roman legionaries who garrison the fortified camps of Totorum, Aquarium, Laudanum and Compendium...

The translation model

- Given the parallel data:
  - a wild goose chase
  - 追逐一只野鹅

- How to compute P(F|E) ?

The translation model

- Can we break down P(F|E) into $\prod_{t=1}^{T} p(f_t|e_t)$?

- We need to know the aligned word pairs.

- Introduce the alignment “a” into the model:
  - P(F,a|E)

What is alignment?

- In MT, alignment is the word-level correspondence between source sentence and target sentence.
- Typological differences between languages lead to complicated alignments!

Spurious word

- Some words has no counter part.

Figure 25.20 The alignment of the **spurious** Spanish word a to the English NULL word e_0.

Many-to-one

Figure 25.21 An alignment between an English and a French sentence, in which each French word does not align to a single English word, but each English word aligns to one French word. Adapted from Brown et al. (1993).

One-to-many

- We call “implemented” a *fertile* word.

Figure 25.19 An alignment between an English and a French sentence, after Brown et al. (1993). Each French word aligns to a single English word.

One-to-many

- Some words are very fertile.

il a m' entarté

He hits me with a pie

Many-to-many

Figure 25.22 An alignment between an English and a French sentence, in which there is a many-to-many alignment between English and French words. Adapted from Brown et al. (1993).

Learning the alignment

- We learn P(F,a|E) as a combination of many factors, including:
  - Probability of particular words aligning (also depends on position in sent)
  - Probability of particular words having particular fertility (number of corresponding words)
- Alignments are missing in the training data, like in the case of ASR, we need to use EM algorithm.

Decoding for SMT

$$
\begin{aligned}
\hat{E} &= argmax_E P(E|F) \\
&= argmax_E \frac{P(F|E)P(E)}{P(F)} \\
&= argmax_E \underbrace{P(F|E)}_{\text{Translation model}} \underbrace{P(E)}_{\text{Language model}}
\end{aligned}
$$

- Viterbi with beam pruning

Decoding for SMT

Pictures from Stanford CS224n

The phrase-based translation model

- Use phrases (sequences of words) as well as single words as the fundamental units of translation.

Neural MT

- Neural Machine Translation (NMT) is a way to do Machine Translation with a *single neural network*
- The neural network architecture is called sequence-to-sequence (aka seq2seq) and it involves two RNNs: the encoder RNN and the decoder RNN.

Recall the RNN structure

- Image from Hung-yi Lee's PPT

The encoder-decoder framework

- The encoder RNN produces an encoding of the source sentence.
- The decoder RNN acts like an RNN LM that generates the target sentence.

Pictures from Stanford CS224n

Other tasks

- The encoder-decoder framework is useful for many tasks.
- Many NLP tasks can be phrased as sequence-to-sequence:
  - Summarization (long text → short text)
  - Dialogue (previous utterances → next utterance)
  - Parsing (input text → output parse as sequence)
  - Code generation (natural language → Python code)

Conditional Language Model

- The sequence-to-sequence model can be regarded as a **Conditional Language Model**.
  - **Language Model** because the decoder is predicting the next word given the previous words of the target sentence *y*
  - **Conditional** because its predictions are *also* conditioned on the source sentence *x*

Neural MT

- NMT directly calculates P(E|F):
- $P(E|F) = P(e_1|f)\ P(e_2|e_1, f)\ \ldots\ P(e_T|e_1 \ldots e_{T-1}, f)$

Training a NMT system

Advantages of NMT

- Compared to SMT, NMT has many advantages:
  - Better performance
    - More fluent
    - Better use of context
    - Better use of phrase similarities
  - A single neural network to be optimized end-to-end
    - No subcomponents to be individually optimized
  - Requires much less human engineering effort
    - No feature engineering
    - Same method for all language pairs

Disadvantages of NMT

- Compared to SMT:
  - NMT is less interpretable
    - Hard to debug
  - NMT is difficult to control
    - For example, can’t easily specify rules or guidelines for translation

The impact on the world

- **2014**: First seq2seq paper published
- **2016**: Google Translate switches from SMT to NMT

- SMT systems, built by hundreds of engineers over many years, outperformed by NMT systems trained by a handful of engineers in a few months

Evaluation of MT system

- **BLEU** (**B**ilingual **E**valuation **U**nderstudy)
- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:
  - *n*-gram precision (usually for 1, 2, 3 and 4-grams)
  - Plus a penalty for too-short system translations

Cand 1:	It is a guide to action which ensures that the military always obeys the commands of the party
Cand 2:	It is to insure the troops forever hearing the activity guidebook that party direct
Ref 1:	It is a guide to action that ensures that the military will forever heed Party commands
Ref 2:	It is the guiding principle which guarantees the military forces always being under the command of the Party
Ref 3:	It is the practical guide for the army always to heed the directions of the party

Figure 25.31 Intuition for Bleu: one of two candidate translations of a Chinese source sentence shares more words with the reference human translations.

Simple precision

- A simple unigram precision metric would be to count the number of words in the candidate translation that occur in some reference translation, and divide by the total number of words in the candidate translation.
- If a candidate translation had 10 words, and 6 of them occurred in at least one of the reference translations, we would have a precision of 6/10 = 0.6
- However, a flaw is that it rewards candidates that have extra repeated words.

Candidate:	the	the	the	the	the	the	the
Reference 1:	the	cat	is	on	the	mat	
Reference 2:	there	is	a	cat	on	the	mat

Figure 25.32 A pathological example showing why Bleu uses a modified precision metric. Unigram precision would be unreasonably high (7/7). Modified unigram precision is appropriately low (2/7).

Modified precision

- First count the maximum number of times a word is used in any single reference translation.
- The count of each *candidate* word is then clipped by this maximum *reference* count.
- Thus the modified unigram precision in the example in Fig. 25.32 would be 2/7, since Reference 1 has a maximum of 2 *the* s.

Candidate:	the	the	the	the	the	the	the
Reference 1:	the	cat	is	on	the	mat	
Reference 2:	there	is	a	cat	on	the	mat

Figure 25.32 A pathological example showing why Bleu uses a modified precision metric. Unigram precision would be unreasonably high (7/7). Modified unigram precision is appropriately low (2/7).

BLEU

- BLEU uses a **modified N-gram precision** metric.
- (The modified bigram precision for Candidate 1 is 10/17, and for Candidate 2 is 1/13.)
- Bleu uses unigram, bigrams, trigrams, and often quadri-grams; it combines these modified N-gram precisions together by taking their geometric mean.

Cand 1:	It is a guide to action which ensures that the military always obeys the commands of the party
Cand 2:	It is to insure the troops forever hearing the activity guidebook that party direct
Ref 1:	It is a guide to action that ensures that the military will forever heed Party commands
Ref 2:	It is the guiding principle which guarantees the military forces always being under the command of the Party
Ref 3:	It is the practical guide for the army always to heed the directions of the party

Figure 25.31 Intuition for Bleu: one of two candidate translations of a Chinese source sentence shares more words with the reference human translations.

BLEU

- In addition, BLEU adds a brevity penalty (BP) over the whole corpus to penalize candidate translations that are too short.
- The BP is computed as follow:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

- where c is the total length of the candidate translation corpus and r is the **effective reference length.**
- Overall, the BLEU score is computed as follow:

$$\text{Bleu} = \text{BP} \times \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$$

Is BLEU useful?

- BLEU is useful but imperfect
  - There are many valid ways to translate a sentence.
  - A good translation can get a poor BLEU score because it has low *n*–gram overlap with the human translation.
  - BLEU performs poorly at comparing systems that have radically different architectures.
  - BLEU is most appropriate when evaluating incremental changes to a single system.

Another big impact

- Researchers have done a lot of work on the "vanilla" seq2seq NMT system.
- There is a piece of work creating another impact on the world.
- That is the **attention** mechanism.

The encoder-decoder framework

- Can you observe any problem with this architecture?

Pictures from Stanford CS224n
http://web.stanford.edu/class/cs224n/slides/cs224n-2021-lecture07-nmt.pdf

The bottleneck problem

- The fixed size embedding needs to capture *all information* about the source sentence. This creates a bottleneck.

Pictures from Stanford CS224n

Core idea

- On each step of the decoder, focus on a particular part of the source sentence.

- How to implement this idea in the neural network?

The attention mechanism

- This example is copied from Stanford course CS224n: http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

The attention mechanism

The attention mechanism

The attention mechanism

The attention mechanism

The attention mechanism

The attention mechanism

The attention mechanism

The attention mechanism

The attention mechanism

The attention mechanism

The attention mechanism

Advantages of Attention

- Significantly improves NMT performance
  - It's very useful to allow decoder to focus on certain parts of the source
- Solves the bottleneck problem
  - Attention allows decoder to look directly at source; bypass bottleneck
- Helps with vanishing gradient problem
  - Provides shortcut to faraway states
- Provides some interpretability
  - By inspecting attention distribution, we can see what the decoder was focusing on. (the alignment)
  - This is cool because we never explicitly trained an alignment system
  - The network just learned alignment by itself

Impact of attention

- Encoder-decoder + attention architecture has become the new standard architecture. There is a lot of work done on the basic attention architecture.
- Attention is a general deep learn technique as it can be used in many architectures and many tasks.
- Later, it inspires the self-attention mechanism and then the Transformer.

A general definition of attention

- Given a set of vector *values*, and a vector *query*, compute a weighted sum of the values, dependent on the query.
- Attention operates on **queries**, **keys** and **values.**
  - **queries:** $\boldsymbol{q_1}, \boldsymbol{q_2}, \ldots, \boldsymbol{q_T}$ for $\boldsymbol{q_i} \in \mathbb{R}^{\boldsymbol{d}}$
  - **keys:** $\boldsymbol{k_1}, \boldsymbol{k_2}, \ldots, \boldsymbol{k_T}$ for $\boldsymbol{k_i} \in \mathbb{R}^{\boldsymbol{d}}$
  - **values:** $\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_T}$ for $\boldsymbol{v_i} \in \mathbb{R}^{\boldsymbol{e}}$
- The operations are as follows:
  - Compute key-query affinities
  - Compute attention weights
  - Compute the weighted sum of values

$$e_{ij} = q_i^\top k_j$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

$$\text{output}_i = \sum_j \alpha_{ij} v_j$$

Self-attention

Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou,and Y. Bengio, "A structured self-attentive sentence embedding," Proceedings of the International Conference on Learning Representations, 2017.

Where it is often used?

- In sentence-level classification, e.g.
  - Speaker classification
  - Emotion classification
  - Sentence type classification
  - Intention classification

- Convert a “sum” operation into a “weighted-sum” operation.

A sum operation

A weighted-sum operation

Issues with RNN

- **Problem**: RNNs take **O(sequence length)** steps for distant word pairs to interact.

- "The computer(s) which I had just put into the machine room on the fifth floor is (are) crashing."

Issues with RNN

- **Problem**: RNNs take **O(sequence length)** steps for distant word pairs to interact.

- "The computer(s) which I had just put into the machine room on the fifth floor is (are) crashing."

Issues with RNN

- Hard to learn long-distance dependencies (gradient problems)

- Forward and backward passes have **O(sequence length)** unparallelizable operations
  - Can't exploit the GPU
  - Training on large dataset is very slow

Self-attention

RNN

Attention

Self-attention as building blocks

Problems of self-attention

- 1. Self-attention mechanism doesn't consider the order information
- 2. No nonlinearities for deep learning magic
- 3. The mechanism always look at the future, but the decoder should have no future words to look at

Problem 1: Ordering

- output_2 is the same even when "who" and "food" exchange their positions.

$$e_{ij} = q_i^\top k_j$$

$$\text{output}_i = \sum_j \alpha_{ij} v_j \qquad \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

Implant order information

- Positional embeddings
- Modify the **queries**, **keys**
  - **queries:** $\tilde{q}_i = q_i + p_i$
  - **keys:** $\tilde{k}_i = k_i + p_i$
  - where $p_i \in \{1, 2, \ldots, T\}$ are the positional embeddings

Non-learnable positional embeddings

- **Sinusoidal position representations**

$$p_i = \begin{pmatrix} \sin(i/10000^{2*1/d}) \\ \cos(i/10000^{2*1/d}) \\ \vdots \\ \sin(i/10000^{2*\frac{d}{2}/d}) \\ \cos(i/10000^{2*\frac{d}{2}/d}) \end{pmatrix}$$

Learnable positional embeddings

- Absolute positional embeddings
- Relative positional embeddings
- Dependency syntax-based position

Problem 2: No nonlinearities

Add nonlinearities between layers

Problem 3: no future words

- The decoder has no future words to look at

Masking the future

- As the decoder has no future words to look at in the prediction stage, we have to maintain the same condition during in training stage.
- To enable parallelization, we **mask out attention** to future words by setting attention scores to $-\infty$.

$$e_{ij} = \begin{cases} q_i^\top k_j, j \leq i \\ -\infty, j > i \end{cases}$$

Solutions to the problems

- 1. Self-attention mechanism doesn't consider the order information
  - Solution: Implant order information in the input
- 2. No nonlinearities for deep learning magic
  - Solution: Add nonlinearities between self-attention layers
- 3. The mechanism always look at the future, but the decoder should have no future words to look at
  - Solution: Use masking

The Transformer

- Attention is all you need

Pictures from Stanford CS224n
http://web.stanford.edu/class/cs224n/slides/cs224n-2021-lecture09-transformers.pdf

The Transformer encoder

Pictures from Stanford CS224n
http://web.stanford.edu/class/cs224n/slides/cs224n-2021-lecture09-transformers.pdf

The Transformer decoder

Pictures from Stanford CS224n
http://web.stanford.edu/class/cs224n/slides/cs224n-2021-lecture09-transformers.pdf

Is the Transformer perfect?

- Quadratic computation
  - Some proposal solutions:
  - Linformer [Wang et al., 2020]
  - BigBird [Zaheer et al., 2021]

Readings

- Speech and Language Processing, version 2
  - Chapter 25
- "Neural Machine Translation by Jointly Learning to Align and Translate"
  - Bahdanau et al., 2014
  - https://arxiv.org/abs/1409.0473
- "Attention Is All You Need"
  - Vaswani et al., 2017
  - https://arxiv.org/abs/1706.03762