Neural Machine Translation

Sami Virpioja

University of Helsinki, firstname.lastname@helsinki.fi
&
Utopia Analytics, firstname.lastname@utopiaanalytics.com

19th March 2024

Slides: Sami Virpioja, Stig-Arne Grönroos

About the lecturer

D.Sc. (Tech.), 2012
Post-doc 2013–2018

Research Scientist
2013–2016

University Researcher
2019–

Lead Data Scientist
2017–

Goals of the lecture

Neural machine translation

- Why NMT is the mainstream approach?
- What path led to the current state-of-the-art NMT?
- How are the current state-of-the-art NMT systems built?
- What are the challenges and limitations for the systems?

Outline

Motivation

Model architectures

Data, preprocessing, and learning techniques

State-of-the-art and future

Outline

Motivation

Model architectures

Data, preprocessing, and learning techniques

State-of-the-art and future

Why neural machine translation?

Ability to generalize

Model similarity of related words and phrases

- Semantically related: synomyms, paraphrases, ...
- Morphologically related: inflections, derivations, compounds

Avoid sparsity problems encountered in phrase-based MT.

Flexibility

Different context vectors are easy to include as input.

Enables paragraph and document-level modeling.

Integration

Easier to combine with other sources of information:
Text in other languages, speech, images, videos, ...

Multitask learning

Paradigm shift to NMT

Dominant paradigm since the latter half of the 2010's.

Reasons for the paradigm shift

Increased computation power (GPUs).

Matured deep learning software frameworks and libraries: TensorFlow, (Py)Torch, Chainer, etc.

Improvements in training algoritms for neural networks:

- Adam (Kingma and Ba 2014),
- Layer normalization (Ba, Kiros, and Hinton 2016),
- Dropout (Srivastava et al. 2014).

Cross-pollination between fields of research

Success of deep learning in computer vision and speech recognition inspired NMT.

Later, NMT architectures such as Attention and Transformers spread to other fields.

Some NMT toolkits

- Fairseq
- Joey NMT
- Marian
- OpenNMT
- Sockeye
- Trax
- ...
- (Hugging Face)

Outline

Motivation

Model architectures

Data, preprocessing, and learning techniques

State-of-the-art and future

Reminder: The autoregressive language model

$$P(X) = \sum_i P(x_i \mid x_0, \ldots, x_{(i-1)})$$

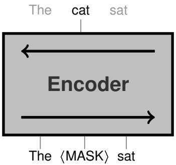
Here is a fragment of text. Tell me how this fragment might go on.

According to this model of the statistics of human language, what words are likely to come next?

Types of language models

Masked LM

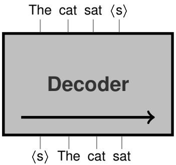
a.k.a. encoder-only



E.g. BERT

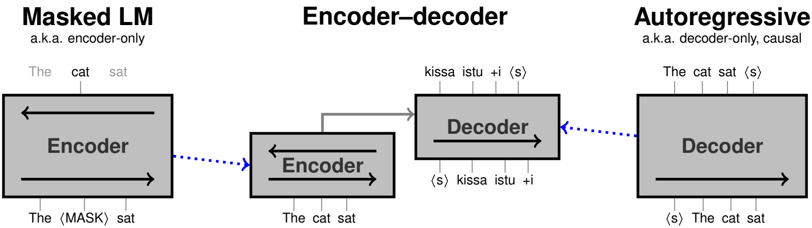
Autoregressive

a.k.a. decoder-only, causal



E.g. GPT

Types of language models



E.g. BERT

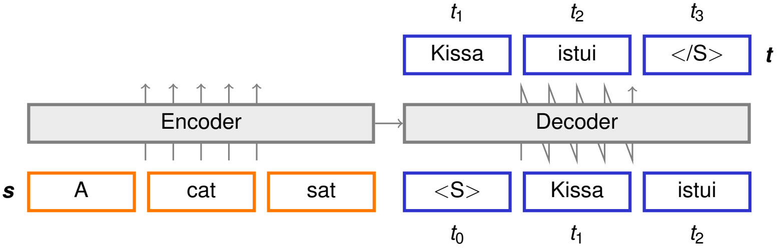
NMT models, but also e.g. T5

E.g. GPT

MT systems are conditional language models

$$P(\; t_j \mid t_0, \ldots, t_{(j-1)}, \; \boldsymbol{s})$$

$$P(\; \text{“istui”} \mid \text{“<S>”, “Kissa”}, \; \boldsymbol{s})$$



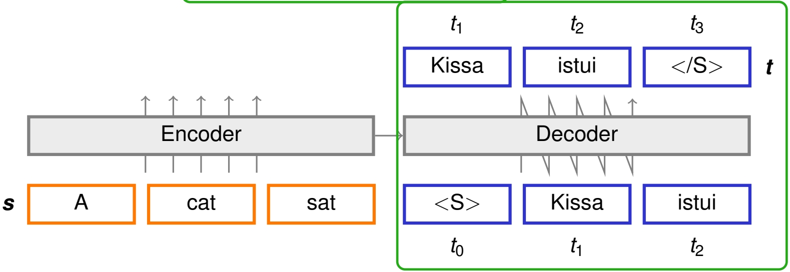
A (data-driven) MT system is a conditional language model.

Predicts the target conditioned on the source.

MT systems are conditional language models

$$P(t_j \mid t_0, \ldots, t_{(j-1)}, \boldsymbol{s})$$

$$P(\text{“istui”} \mid \text{“<S>”, “Kissa”}, \boldsymbol{s})$$



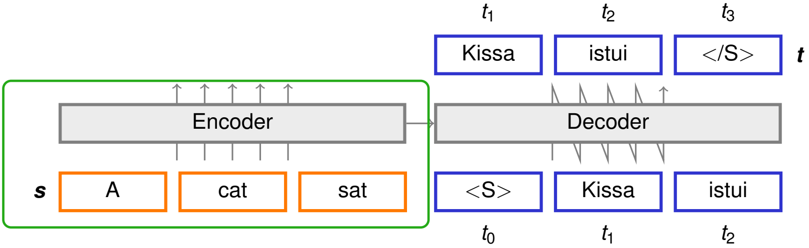
A (data-driven) MT system is a conditional language model.

Predicts the target conditioned on the source.

MT systems are conditional language models

$$P(\;t_j \;\mid\; t_0, \ldots, t_{(j-1)}, \;\boldsymbol{s})$$

$$P(\;\text{“istui”} \;\mid\; \text{“<S>”, “Kissa”}, \;\boldsymbol{s})$$



A (data-driven) MT system is a conditional language model.

Predicts the target conditioned on the source.

History lesson

Let's look at some of the breakthroughs leading towards current SOTA architectures, and the challenges that inspired these breakthroughs.

History: Embedding variable-length sequences

How to encode sequences (words, phrases, sentences) $x_1, x_2, \ldots, x_n$ of variable length $n \geq 1$ to fixed length representations?

History: Embedding variable-length sequences

How to encode sequences (words, phrases, sentences) $x_1, x_2, \ldots, x_n$ of variable length $n \geq 1$ to fixed length representations?

Remember from Word2vec lecture: Embeddings such as word2vec give fixed-length vectors for each of the units in the sequence, but the sequence itself is variable-length.

History: Embedding variable-length sequences

How to encode sequences (words, phrases, sentences) $x_1, x_2, \ldots, x_n$ of variable length $n \geq 1$ to fixed length representations?

Remember from Word2vec lecture: Embeddings such as word2vec give fixed-length vectors for each of the units in the sequence, but the sequence itself is variable-length.

But how to combine them? Summing or averaging discards the sequence order.

History: Embedding variable-length sequences

How to encode sequences (words, phrases, sentences) $x_1, x_2, \ldots, x_n$ of variable length $n \geq 1$ to fixed length representations?

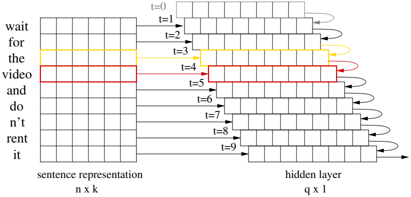
Remember from Word2vec lecture: Embeddings such as word2vec give fixed-length vectors for each of the units in the sequence, but the sequence itself is variable-length.

But how to combine them? Summing or averaging discards the sequence order.

Remember from LM Lecture: Neural network language models are able to store information over long contexts.

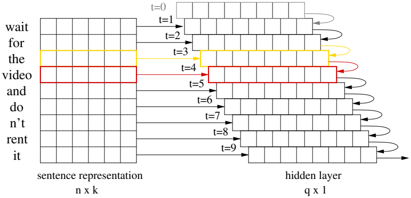
History: Sequence encoding with RNN and CNN

Recurrent neural networks: Take the last hidden state as sentence embedding. (Elman 1990; Mikolov et al. 2010)

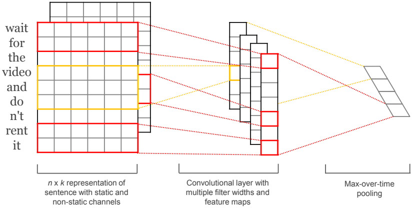


History: Sequence encoding with RNN and CNN

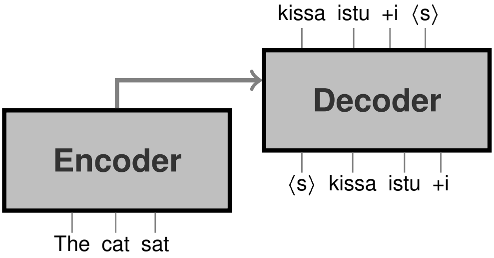
Recurrent neural networks: Take the last hidden state as sentence embedding. (Elman 1990; Mikolov et al. 2010)



Alternative: Convolutional neural networks (Fukushima 1980; Kim 2014)



History: Encoder-decoder model



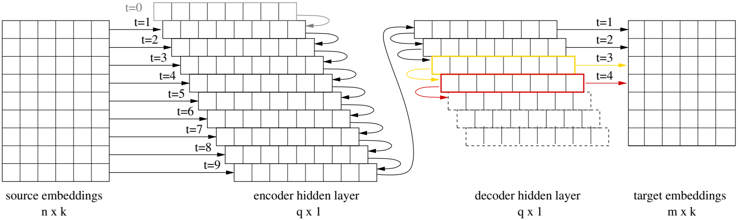
History: Sequence decoding

How to implement the decoder?

History: Sequence decoding

How to implement the decoder?

Again, we can use an RNN language model
— just initialize the hidden state with the sentence representation from encoder!

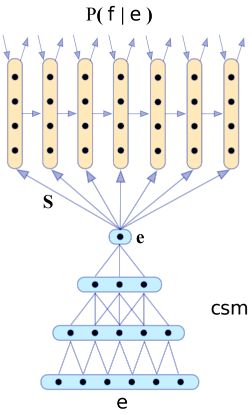


History: First complete NMT systems

Kalchbrenner and Blunsom 2013: Encode with convolutional neural networks (CNN), decode with recurrent neural network (RNN) language model

Sutskever, Vinyals, and Le 2014: Encode and decoder with RNN with long short-term memory (LSTM) units

Cho et al. 2014b: Encode with RNN with gated recursive units (GRU) or gated recursive CNN, decoder with RNN with GRUs

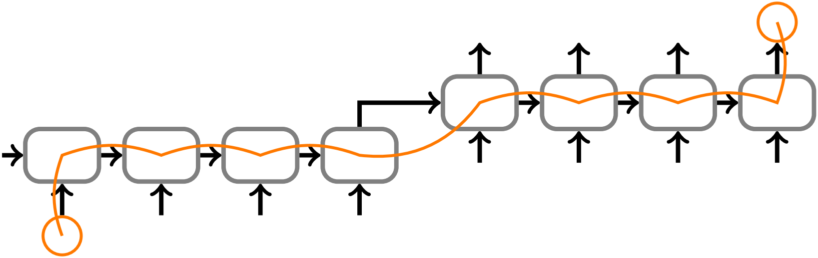


RCTM I

History: Vanishing gradient problem

The error signal decreases exponentially with the number of layers in backpropagation and gradient-based learning.

The RNN encoder must process entire sentence before sentence encoding is ready: The long path makes it hard to learn relevant information from first time steps (beginning of sentence).



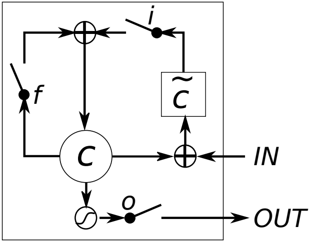
History: Gated units in recurrent neural networks

Solution:

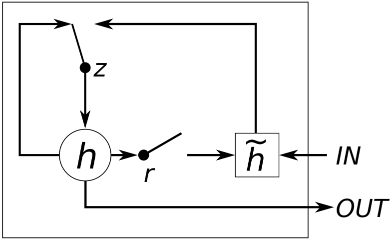
- Predict what information to keep and what to forget from the state representation.
- Gates: sigmoid activation (0–1) followed by pointwise multiplication with the target signal.

History: Gated units in recurrent neural networks

LSTM and GRU are two gate architectures with similar performance (Chung et al. 2014)



Long short-term memory
(Hochreiter and Schmidhuber 1997)



Gated recurrent unit
(Cho et al. 2014a)

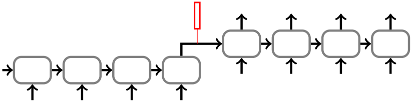
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

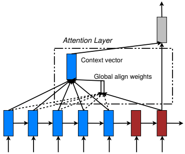
History: Attention model

Even with gated units, it is hard to decode a sensible target sentence from a single embedded source vector.

Encoder provides embeddings for each input unit — allow decoder to look at them.

Attention model: At each decoder time step, predict which parts of the source encoding are relevant for next output.

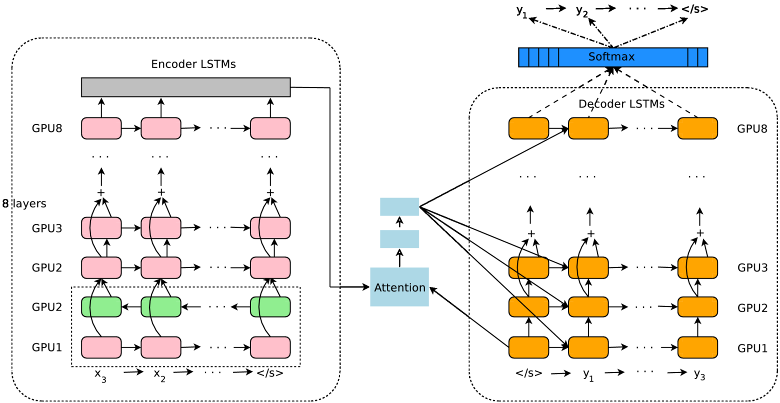




(Luong, Pham, and Manning 2015)
(Bahdanau, Cho, and Bengio 2015)

http://distill.pub/2016/augmented-rnns/#attentional-interfaces

History: Adding layers



Google NMT (Wu et al. 2016)

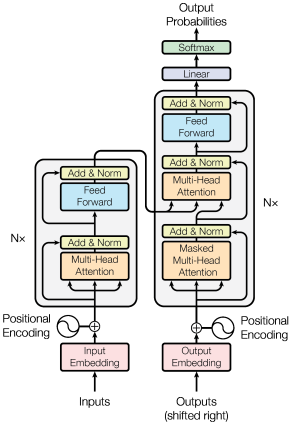
History: Transformer architecture

Recurrent networks require sequential computation ($O(n)$ for n units in sentence)

Can we cope without them?

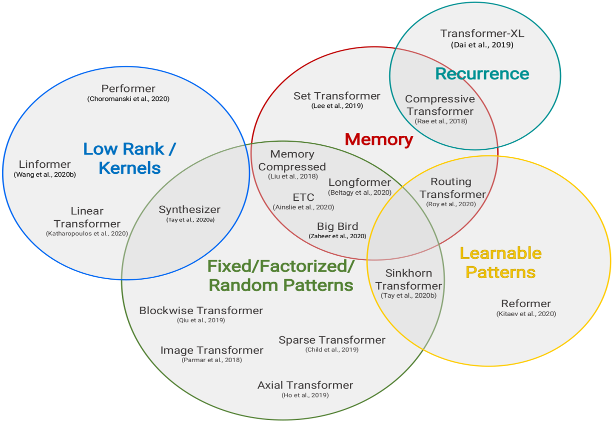
"Attention is all you need" — Google's Transformer architecture (Vaswani et al. 2017)

Multiple layers of attention networks in both encoder and decoder



https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Transformers: the sequels



Taxonomy of efficient Transformer architectures (Tay et al. 2020).

Mixture of Experts

In Mixture of Experts (MoE), a gating network selects which subnetworks to use for the example.

- A sparse network: not all parameters are used each time.
- Can be combined with the Transformer architecture.

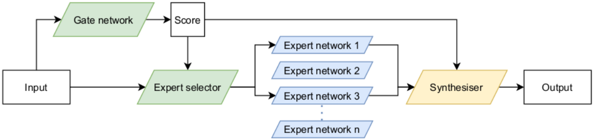


Figure 1: An illustrative example of an MoE layer. In this example, expert 1 and expert 3 are selected by the gate for computation.

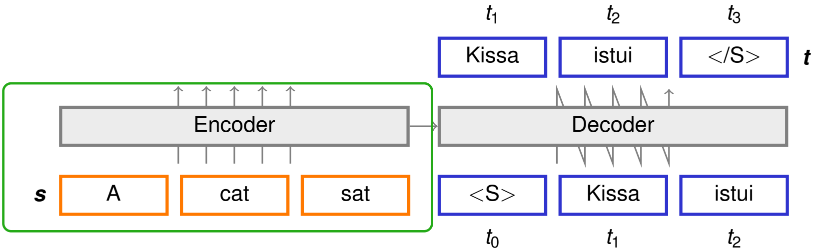
- Figure from (He et al. 2021)

Discuss in groups (a few minutes)

What happens if the conditioning on the source fails to be learned?

$$P(\quad t_j \quad | \quad t_0, \ldots, t_{(j-1)}, \quad \boxed{\boldsymbol{s}})$$

$$P(\quad \text{“istui”} \quad | \quad \text{“<S>”, “Kissa”}, \quad \boxed{\boldsymbol{s}})$$



Examples

EN source: Stealing food is a common crime in student halls.

ET source: Laktoosi puhul see nii ju ongi!
(That's the case with lactose!)

Examples

EN source: Stealing food is a common crime in student halls.
FI: Lapsenteko on yhteistä rikollisuutta.
(Making children is shared crime.)

ET source: Laktoosi puhul see nii ju ongi!
(That's the case with lactose!)

EN: I've been thinking about it.

Outline

Motivation

Model architectures

Data, preprocessing, and learning techniques

State-of-the-art and future

Is Transformer all you need?

At the moment, Transformer is the state-of-the-art and *de facto* standard in NMT.

Is Transformer all you need?

At the moment, Transformer is the state-of-the-art and *de facto* standard in NMT.

But the model architecture is not everything!

Is Transformer all you need?

At the moment, Transformer is the state-of-the-art and *de facto* standard in NMT.

But the model architecture is not everything!

Especially for low-resource language pairs and morphologically rich languages, we need methods for:

Is Transformer all you need?

At the moment, Transformer is the state-of-the-art and *de facto* standard in NMT.

But the model architecture is not everything!

Especially for low-resource language pairs and morphologically rich languages, we need methods for:

1. Learning from bilingual data in other languages
2. Using monolingual corpora in source or target language
3. Selecting input and output units

Transfer learning

Current machine learning methods are data-hungry.

The easiest way to improve performance is to train on larger data.

Transfer learning

Current machine learning methods are data-hungry.

The easiest way to improve performance is to train on larger data.

Either collect more data for the task (expensive!),

Transfer learning

Current machine learning methods are data-hungry.

The easiest way to improve performance is to train on larger data.

Either collect more data for the task (expensive!),
or figure out a way to use existing data sets.

Transfer learning

Current machine learning methods are data-hungry.

The easiest way to improve performance is to train on larger data.

Either collect more data for the task (expensive!),

or figure out a way to use existing data sets.

- Labeled data for other tasks.
- Unlabeled data.

Labeled and unlabeled in the context of MT

Let's say the goal is a English-to-Finnish system.

Labeled data for this task: English-Finnish sentence pairs

- Input English sentence
- is labeled by output Finnish sentence.

Labeled and unlabeled in the context of MT

Let's say the goal is a English-to-Finnish system.

Labeled data for this task: English-Finnish sentence pairs

- Input English sentence
- is labeled by output Finnish sentence.

Labeled data for another task:

- e.g. English-Estonian sentence pairs.

Labeled and unlabeled in the context of MT

Let's say the goal is a English-to-Finnish system.

Labeled data for this task: English-Finnish sentence pairs

- Input English sentence
- is labeled by output Finnish sentence.

Labeled data for another task:

- e.g. English-Estonian sentence pairs.

Unlabeled data:

- Monolingual English,
- or monolingual Finnish.

Transfer learning techniques

Transfer learning: Use knowledge gained from solving one task in a related task.

How are the different learning tasks timed?

- Sequential transfer
- Parallel transfer
- Mix: Scheduled multi-task learning

Transfer learning techniques

Sequential transfer

Parallel transfer

Mix: Scheduled multi-task learning

Transfer learning techniques

Sequential transfer

- Often called just "transfer learning"

1. Train a system on one task (“pretraining”),
2. then transfer the knowledge,
3. and finally continue training on another task (“fine-tuning”).

Parallel transfer

Mix: Scheduled multi-task learning

Transfer learning techniques

Sequential transfer

- Often called just "transfer learning"

1. Train a system on one task (“pretraining”),
2. then transfer the knowledge,
3. and finally continue training on another task (“fine-tuning”).

- Risk: catastrophic forgetting
- Benefit: the second task doesn’t need to be known when training the first!

Parallel transfer

Mix: Scheduled multi-task learning

Transfer learning techniques

Sequential transfer

- Often called just "transfer learning"

1. Train a system on one task (“pretraining”),
2. then transfer the knowledge,
3. and finally continue training on another task (“fine-tuning”).

- Risk: catastrophic forgetting
- Benefit: the second task doesn’t need to be known when training the first!

Parallel transfer

- Often called "multi-task learning"
- Learn multiple related tasks at the same time.

Mix: Scheduled multi-task learning

Transfer learning techniques

Sequential transfer

- Often called just "transfer learning"

1. Train a system on one task (“pretraining”),
2. then transfer the knowledge,
3. and finally continue training on another task (“fine-tuning”).

- Risk: catastrophic forgetting
- Benefit: the second task doesn’t need to be known when training the first!

Parallel transfer

- Often called "multi-task learning"
- Learn multiple related tasks at the same time.

Mix: Scheduled multi-task learning

- e.g. multi-task pretraining + multi-task fine-tuning

Cross-lingual transfer: Settings

Given training data between languages A and B, can it help translating from language C to D?

Cross-lingual transfer: Settings

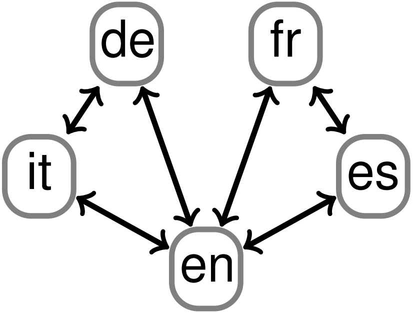
Given training data between languages A and B, can it help translating from language C to D?

Training a multilingual MT system is a multi-task training scenario:

- Each language pair is one task.
- Multilingual settings:
  - one-to-many
  - many-to-one
  - many-to-many
- Can also combine with monolingual tasks.

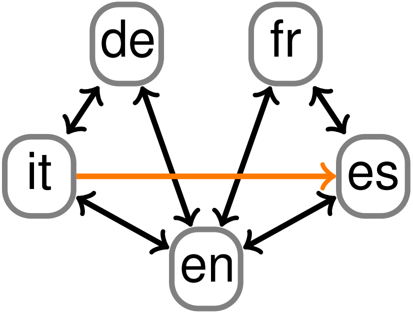
Cross-lingual transfer: Zero-shot and universal translation

Many-to-many translation enables new language pairs without training data ("zero-shot") or explicit pivot language.



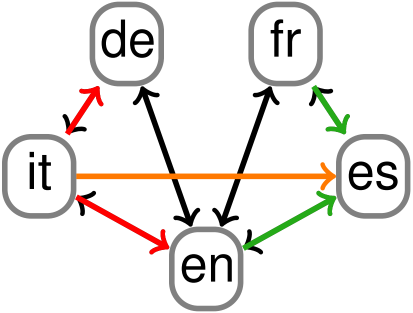
Cross-lingual transfer: Zero-shot and universal translation

Many-to-many translation enables new language pairs without training data ("zero-shot") or explicit pivot language.



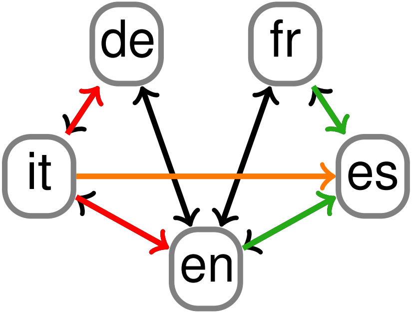
Cross-lingual transfer: Zero-shot and universal translation

Many-to-many translation enables new language pairs without training data ("zero-shot") or explicit pivot language.



Cross-lingual transfer: Zero-shot and universal translation

Many-to-many translation enables new language pairs without training data ("zero-shot") or explicit pivot language.



Universal translation: Extension of many-to-many translation to cover all languages.

How effective is cross-lingual transfer?

When model capacity is insufficient and tasks are too different, you get interference (negative transfer).

The accepted wisdom used to be that cross-lingual transfer

- is good for medium and low-resource languages,
- but for high-resource pairs bilingual was better.

Recently this was put in question

- Facebook AI's WMT 2021 News task submission (Tran et al. 2021)
- Large enough multilingual models outperform single-pair models even for high-resource language pairs like En$\leftrightarrow$Cs, En$\leftrightarrow$De, En$\leftrightarrow$Ru.

Massively multilingual machine translation

No Language Left Behind: Scaling Human-Centered Machine Translation (Costa-jussà et al. 2022)

- Massively multilingual model from Meta AI.
- Supports 200 languages,
  including both high- and low-resource languages.
- Transformer Mixture-of-Experts (MoE),
  54.5B parameters in total.

Using monolingual corpora

There is no separate language model component in NMT.

Using monolingual corpora

There is no separate language model component in NMT.

How to exploit abundant monolingual data?

Using monolingual corpora

There is no separate language model component in NMT.

How to exploit abundant monolingual data?

Approaches:

- Pretraining
- Autoencoding
- Back-translation

Monolingual corpora: Pretraining

Sequential transfer: Train a component of the model on monolingual data.

Monolingual corpora: Pretraining

Sequential transfer: Train a component of the model on monolingual data.

1. Pretrained source or target embeddings

Monolingual corpora: Pretraining

Sequential transfer: Train a component of the model on monolingual data.

1. Pretrained source or target embeddings
2. Pretrained subnetwork (encoder or decoder)

Monolingual corpora: Pretraining

Sequential transfer: Train a component of the model on monolingual data.

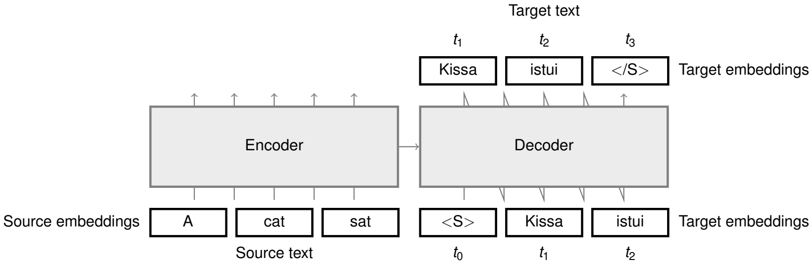
1. Pretrained source or target embeddings
2. Pretrained subnetwork (encoder or decoder)
3. Pretraining entire network (encoder and decoder)
   - E.g. finetuning a multilingual LM for MT (Liu et al. 2020)

Monolingual corpora: Pretraining

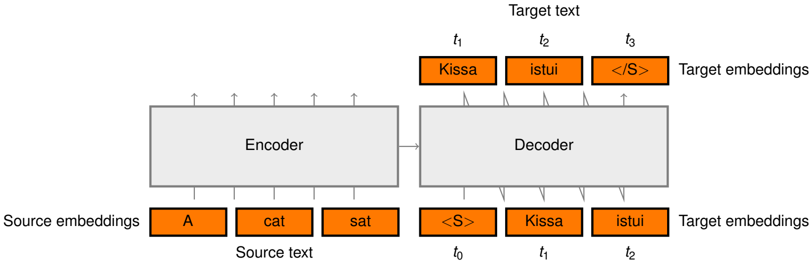
Sequential transfer: Train a component of the model on monolingual data.

1. Pretrained source or target embeddings
2. Pretrained subnetwork (encoder or decoder)
3. Pretraining entire network (encoder and decoder)
   - E.g. finetuning a multilingual LM for MT (Liu et al. 2020)
4. Language model fusion

Parameter sharing in NMT transfer learning

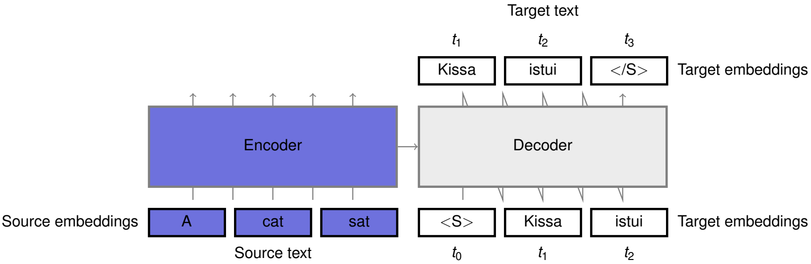


Parameter sharing in NMT transfer learning



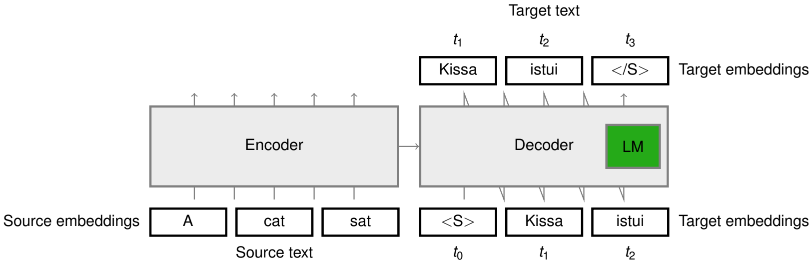
Pretrained embeddings

Parameter sharing in NMT transfer learning



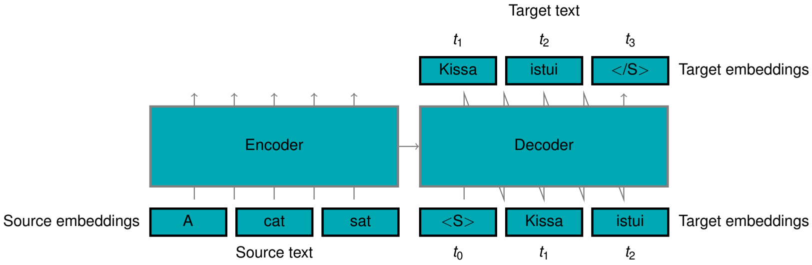
Pretrained encoder

Parameter sharing in NMT transfer learning



Language model fusion

Parameter sharing in NMT transfer learning



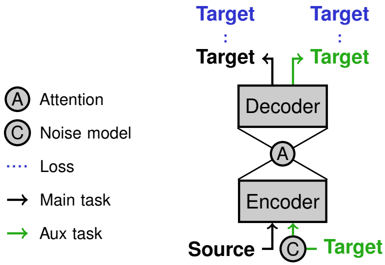
Full parameter sharing

Monolingual corpora: Autoencoding

Parallel transfer: Use multi-task learning with source-to-source or target-to-target autoencoding as an additional task.

Monolingual corpora: Autoencoding

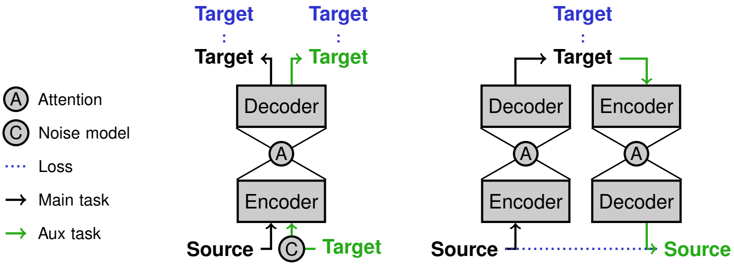
Parallel transfer: Use multi-task learning with source-to-source or target-to-target autoencoding as an additional task.



Denoising autoencoder
for target language

Monolingual corpora: Autoencoding

Parallel transfer: Use multi-task learning with source-to-source or target-to-target autoencoding as an additional task.



Denoising autoencoder for target language

Dual learning (Xia et al. 2016)

Monolingual corpora: Back-translation

Let's say the goal is a English-to-Finnish system.

Monolingual corpora: Back-translation

Let's say the goal is a English-to-Finnish system.

First train a Finnish-to-English system and translate any monolingual Finnish corpora with it.

Monolingual corpora: Back-translation

Let's say the goal is a English-to-Finnish system.

First train a Finnish-to-English system and translate any monolingual Finnish corpora with it.

Use results as additional training material.

- Synthetic training data.

Monolingual corpora: Back-translation

Let's say the goal is a English-to-Finnish system.

First train a Finnish-to-English system and translate any monolingual Finnish corpora with it.

Use results as additional training material.

- Synthetic training data.

This technique is called back-translation (Sennrich, Haddow, and Birch 2016a).

Monolingual corpora: Back-translation

Let's say the goal is a English-to-Finnish system.

First train a Finnish-to-English system and translate any monolingual Finnish corpora with it.

Use results as additional training material.

- Synthetic training data.

This technique is called back-translation (Sennrich, Haddow, and Birch 2016a).

Bad translations on the source side do not matter too much.

Monolingual corpora: Back-translation

Let's say the goal is a English-to-Finnish system.

First train a Finnish-to-English system and translate any monolingual Finnish corpora with it.

Use results as additional training material.

- Synthetic training data.

This technique is called back-translation (Sennrich, Haddow, and Birch 2016a).

Bad translations on the source side do not matter too much.

Large gains, but double work in training.

Lexical units in NMT

Limiting issues in phrase-based MT:

- Many tokens per sentence makes decoding more difficult.
- Different number of tokens in source and target sentence makes word alignment more difficult.

No such restrictions in NMT!

Units for encoder and decoder

Encoder input symbols

Words: large vocabulary, rare words, OOVs.

- But factors (e.g. morphological analysis) easy to integrate.

Using characters may slow down attention too much.

- Softmax operation on input tokens.

Decoder output symbols

Computational complexity increases with vocabulary size due to softmax in output layer.

Conclusion

Subword units are a good compromise.

- Morphological segmentation (if available)
- Statistical subwords (strong encoders compensate for non-sensible segmentations)

Multilingual units

Current standard practice in segmentation:

- SentencePiece (Kudo 2018)

Still popular:

- Byte-pair encoding (BPE) (Sennrich, Haddow, and Birch 2016b)

Joint segmentation: The source and target language corpora — or more languages in a multilingual system — can be combined as a single training corpus for SentencePiece / BPE.

- Very practical for massively multilingual models: no need for language-specific preprocessing.

Outline

Motivation

Model architectures

Data, preprocessing, and learning techniques

State-of-the-art and future

Challenges

Training SOTA-size models is computationally expensive.

- Increasing the number of layers improves results but requires more GPU/TPU resources.
- Distributed training over enormous number of GPUs.

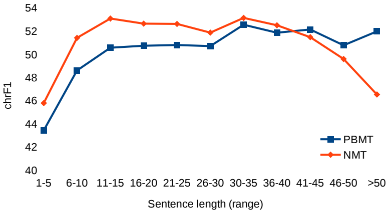
NMT is a "black box" system.

- No "phrase table" to observe or modify.
- Inconvinient especially for translation industry, where correct terminology is very important.

Challenges (cont.)

Translation quality issues

- Problems with long texts.
  - Long sentences used to be problematic (Toral and Sánchez-Cartagena 2017)



  - Now the challenge is in document-level translation.
- Good fluency, but sometimes very misleading translations — can be less predictable than PBMT

NMT quality on par with human translators?

Sometimes human evaluation has indicated that NMT would be on the level of human translation.

E.g. paper by Microsoft Research:
"Achieving Human Parity on Automatic Chinese to English News Translation" (Hassan Awadalla et al. 2018)

- Direct assessment (score 0-100) by bilingual humans.
- No statistically significant difference between NMT output and reference translations by humans!

NMT quality on par with human translators?

Caveats:

- Are the human translators professionals? Are they translating to their native language?
- How about the human evaluators?
  - Do they understand what to judge (e.g. fluency vs. adequacy)? Even bad NMT is fluent.
  - Skill and time spent: ability to notice subtle differences.
  - Bilingual vs evaluators only speaking target language (use source, or only reference?)
  - Is the document context available?

Toral et al. 2018, Läubli et al. 2020

https://www.linkedin.com/pulse/
microsoft-mt-reaches-parity-bad-human-translation-tommi-nieminen

Large language models as translators

LLMs are often trained on multilingual data.

- Zero-shot translation

LLMs fine-tuned for conversational usage (e.g. ChatGPT) can be asked to do various tasks, including translation.

- Prompt engineering

How good they are in translation compared to NMT models?

ChatGPT vs commercial translation services

Table 8: Performance of GPT-4 (Date: 2023.03.15) for multilingual translation.

System	De⇒En	En⇒De	Zh⇒En	En⇒Zh	De⇒Zh	Ro⇒Zh
Google	45.04	41.16	31.66	43.58	38.71	39.05
DeepL	49.23	41.46	31.22	44.31	40.46	38.95
Tencent	n/a	n/a	29.69	46.06	40.66	n/a
ChatGPT (Direct)	43.71	38.87	24.73	38.27	34.46	30.84
ChatGPT (Direct$_{new}$)	n/a	n/a	n/a	n/a	30.76	27.51
ChatGPT (Pivot$_{new}$)	n/a	n/a	n/a	n/a	34.68	34.19
GPT-4	46.00	45.73	28.50	42.50	38.16	37.84

Jiao et al. 2023

- Direct prompt: "Please provide the [TGT] translation for these sentences:"
- Pivot prompt: "Please provide the [PIV] translation first and then the [TGT] translation for these sentences one by one:"

Future of NMT

If LLMs are already so good translators, do we need separate NMT models?

- Multiple questions: Architecture, size, training.

Future of NMT

If LLMs are already so good translators, do we need separate NMT models?

- Multiple questions: Architecture, size, training.

Architecture: Is encoder-decoder redundant?

- Translation Language Models of similar size perform equally well (Gao et al. 2022).

Future of NMT (cont.)

Size: "GPT-scale" LLMs have NMT challenges multiplied: computation, memory, ...

- Only large corporations have resources for training.
- Not possible to run the models locally.
- Environmental impact.

Future of NMT (cont.)

Size: “GPT-scale” LLMs have NMT challenges multiplied: computation, memory, ...

- Only large corporations have resources for training.
- Not possible to run the models locally.
- Environmental impact.

Training: Adapting conversational LLMs to specific tasks and domains is not simple (prompt engineering).

Future of NMT (cont.)

How to create accessible and efficient translation models?

- Re-usable models and modular architectures
- Efficient fine-tuning (e.g. low-rank adaptation)
- Knowledge distillation (small model trained on large model's output)
- Related on-going projects at University of Helsinki: FoTran, HPLT, GreenNLP

Bibliography

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980. URL: http://arxiv.org/abs/1412.6980.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E Hinton (2016). "Layer normalization". In: *arXiv preprint arXiv:1607.06450*. URL: https://arxiv.org/abs/1607.06450.

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Elman, Jeffrey L. (Mar. 1990). "Finding Structure in Time". en. In: *Cognitive Science* 14.2, pp. 179–211. ISSN: 03640213.

Bibliography (cont.)

Mikolov, Tomáš et al. (2010). "Recurrent Neural Network Based Language Model". en. In: *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pp. 1045–1048.

Fukushima, Kunihiko (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological cybernetics* 36.4, pp. 193–202.

Kim, Yoon (2014). "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1746–1751. URL: http://www.aclweb.org/anthology/D14-1181.

Bibliography (cont.)

Kalchbrenner, Nal and Phil Blunsom (2013). "Recurrent Continuous Translation Models". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1700–1709. URL: http://www.aclweb.org/anthology/D13-1176.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. V Le (2014). "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., pp. 3104–3112. URL: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Cho, Kyunghyun et al. (2014b). "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, pp. 103–111. URL: http://www.aclweb.org/anthology/W14-4012.

Bibliography (cont.)

Chung, Junyoung et al. (2014). "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *CoRR* abs/1412.3555. URL: http://arxiv.org/abs/1412.3555.

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). "Long Short-term Memory". In: *Neural Comput.* 9.9, pp. 1735–1780. ISSN: 0899-7667. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Cho, Kyunghyun et al. (2014a). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734. URL: http://www.aclweb.org/anthology/D14-1179.

Bibliography (cont.)

Luong, Thang, Hieu Pham, and Christopher D. Manning (2015). “Effective Approaches to Attention-based Neural Machine Translation”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1412–1421. URL: http://aclweb.org/anthology/D15-1166.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *ICLR*. URL: http://arxiv.org/pdf/1409.0473v6.pdf.

Wu, Yonghui et al. (2016). “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: *CoRR* abs/1609.08144. URL: http://arxiv.org/abs/1609.08144.pdf.

Vaswani, Ashish et al. (2017). “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Bibliography (cont.)

Tay, Yi et al. (2020). *Efficient Transformers: A Survey*. arXiv: 2009.06732 [cs.LG].

He, Jiaao et al. (2021). “FastMoE: A fast mixture-of-expert training system”. In: *arXiv preprint arXiv:2103.13262*. URL: https://arxiv.org/abs/2103.13262.

Tran, Chau et al. (2021). “Facebook AI’s WMT21 News Translation Task Submission”. In: *Proceedings of the Sixth Conference on Machine Translation*. Online: Association for Computational Linguistics, pp. 205–215. URL: https://aclanthology.org/2021.wmt-1.19.

Costa-jussà, Marta R et al. (2022). “No language left behind: Scaling human-centered machine translation”. In: *arXiv preprint arXiv:2207.04672*.

Liu, Yinhan et al. (2020). “Multilingual denoising pre-training for neural machine translation”. In: *Transactions of the Association for Computational Linguistics* 8, pp. 726–742.

Bibliography (cont.)

Xia, Yingce et al. (Nov. 2016). "Dual Learning for Machine Translation". en. In: *arXiv:1611.00179 [cs]*. arXiv: 1611.00179. URL: http://arxiv.org/abs/1611.00179.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016a). "Improving Neural Machine Translation Models with Monolingual Data". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 86–96. URL: http://www.aclweb.org/anthology/P16-1009.

Kudo, Taku (2018). "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates". en. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL) (Volume 1: Long Papers)*, pp. 66–75. URL: http://arxiv.org/abs/1804.10959.

Bibliography (cont.)

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016b). "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725. URL: http://www.aclweb.org/anthology/P16-1162.

Toral, Antonio and Víctor M. Sánchez-Cartagena (2017). "A Multifaceted Evaluation of Neural versus Phrase-Based Machine Translation for 9 Language Directions". In: *CoRR* abs/1701.02901. URL: http://arxiv.org/abs/1701.02901.

Hassan Awadalla, Hany et al. (2018). "Achieving Human Parity on Automatic Chinese to English News Translation". In: *ArXiv e-prints*. URL: https://arxiv.org/abs/1803.05567.

Bibliography (cont.)

Toral, Antonio et al. (2018). "Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation". In: *Proceedings of the Third Conference on Machine Translation (WMT): Research Papers*, pp. 113–123. URL: https://www.aclweb.org/anthology/W18-6312.

Läubli, Samuel et al. (2020). "A Set of Recommendations for Assessing Human–Machine Parity in Language Translation". In: *Journal of Artificial Intelligence Research* 67, pp. 653–672.

Jiao, Wenxiang et al. (2023). "Is ChatGPT a good translator? Yes With GPT-4 As The Engine". In: *arXiv preprint arXiv:2301.08745v4*. URL: https://arxiv.org/abs/2301.08745v4.

Gao, Yingbo et al. (Nov. 2022). "Is Encoder-Decoder Redundant for Neural Machine Translation?" In: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 562–574. URL: https://aclanthology.org/2022.aacl-main.43.