Neural Machine Translation & Machine Translation Evaluation

Stig-Arne Grönroos

Silo.AI, stig.gronroos@silo.ai
University of Helsinki, stig-arne.gronroos@helsinki.fi

22th March 2022

About lecturer

D.Sc. (Tech.), 2021

PostDoc, 2022–

Senior AI Scientist 2020–

Goals of the lecture

Neural machine translation

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

Evaluation of machine translation

- How are machine translation systems evaluated manually?
- How do the standard automatic metrics work,
  and how can they be improved?
- What are the limitations of the metrics?

*https://slator.com/whitepapers/slator-neural-machine-translation-report-2018/

Part I

Neural Machine Translation

Why neural machine translation?

Ability to generalize

Model similarity of related words

- 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, (Theano), 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

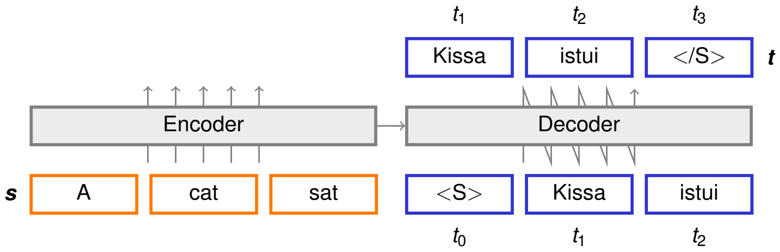
Trax

...

MT systems are conditional language models

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

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



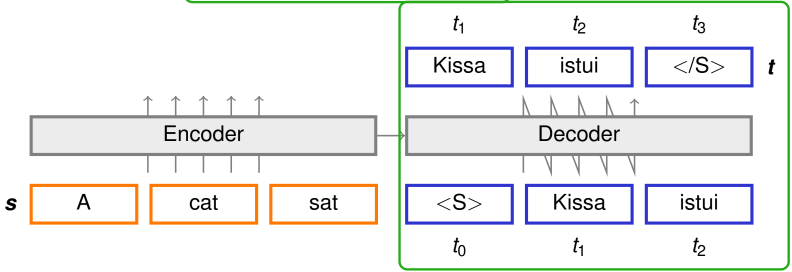
A (data-driven) MT system is a conditional language model.

Predicts the target conditioned on the source.

MT systems are conditional language models

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

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



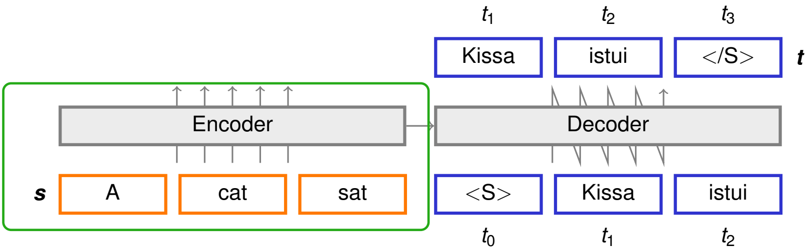
A (data-driven) MT system is a conditional language model.

Predicts the target conditioned on the source.

MT systems are conditional language models

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

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



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

Predicts the target conditioned on the source.

Background: 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?

Background: 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 Lecture 3: Embeddings such as word2vec give fixed-length vectors for each of the units in the sequence.

Background: 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 Lecture 3: Embeddings such as word2vec give fixed-length vectors for each of the units in the sequence.

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

Background: 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?

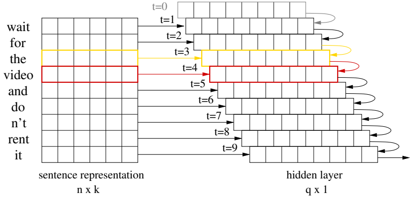
Remember from Lecture 3: Embeddings such as word2vec give fixed-length vectors for each of the units in the sequence.

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

Remember from Lecture 8: Neural network language models are able to store information over long contexts.

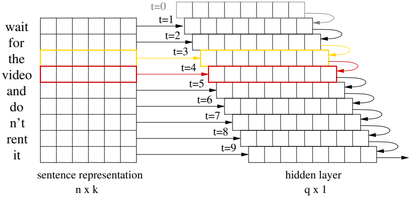
Sequence encoding

Recurrent neural networks:
Take the last hidden state as sentence embedding.

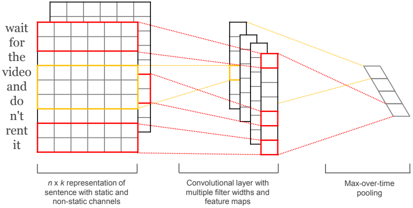


Sequence encoding

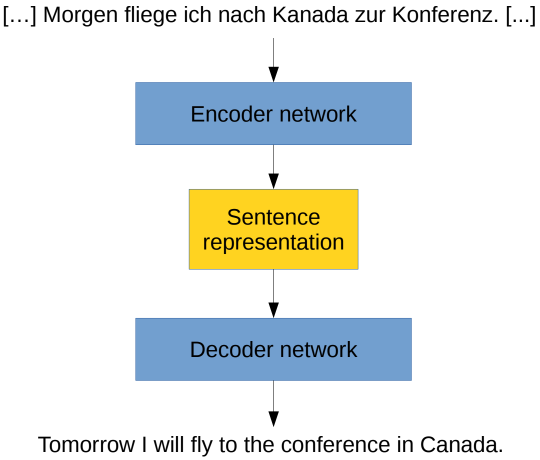
Recurrent neural networks: Take the last hidden state as sentence embedding.



Alternative: Convolutional neural networks (Kim 2014)



Encoder-decoder model



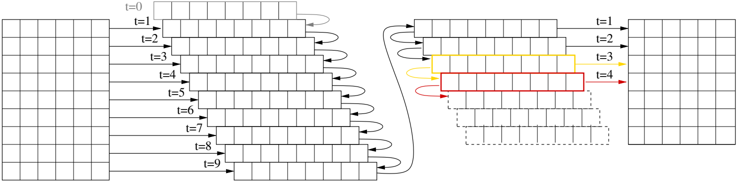
Sequence decoding

How to implement the decoder?

Sequence decoding

How to implement the decoder?

Again, we can use a neural network language model — just initialize the hidden state with the sentence representation from encoder!

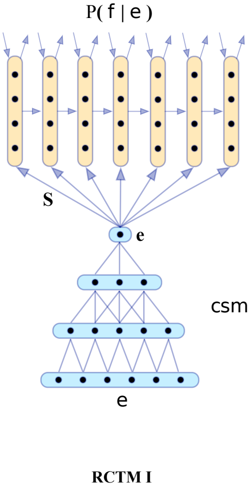


First complete NMT systems

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



Recurrent neural networks: Gates

Vanishing gradient problem: Error signal decreases exponentially with the number of layers in backpropagation and gradient-based learning.

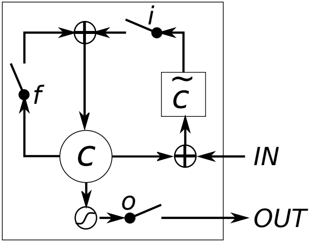
The RNN encoder must process entire sentence before sentence encoding is ready: The long path makes it hard to learn relevant information from first time steps (beginning of sentence).

Solution:

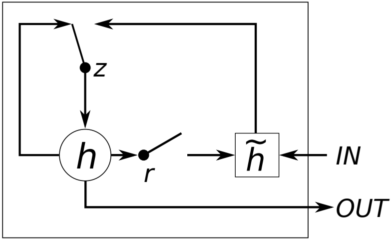
- Predict what information to keep and what to forget from the state representation.
- Gates: sigmoid activation (0–1) followed by pointwise multiplication with the target signal.

Recurrent neural networks: Gated units

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)

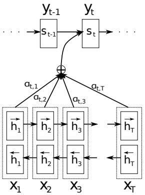
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Attention model

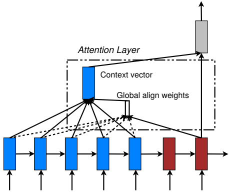
Even with gated units, it is hard to decode a sensible target sentence from a single embedded source vector.

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.



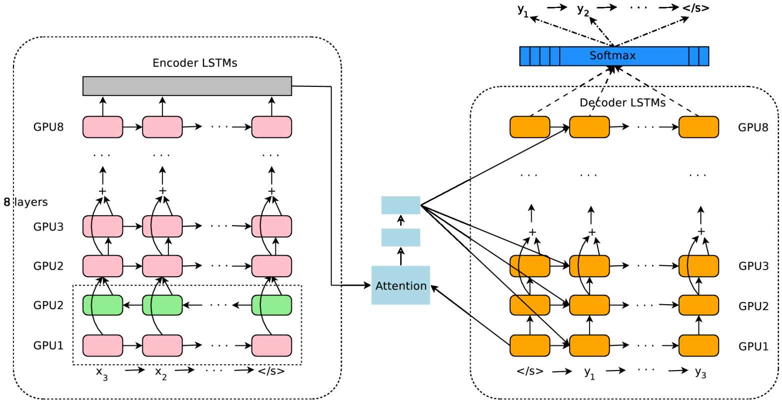
(Bahdanau, Cho, and Bengio 2015)



(Luong, Pham, and Manning 2015)

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

Adding layers



Google NMT (Wu et al. 2016)

Some results: https://research.googleblog.com/2016/09/a-neural-network-for-machine.html

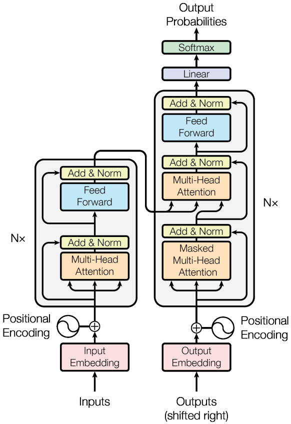
Transformer architecture

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

Can we cope without them?

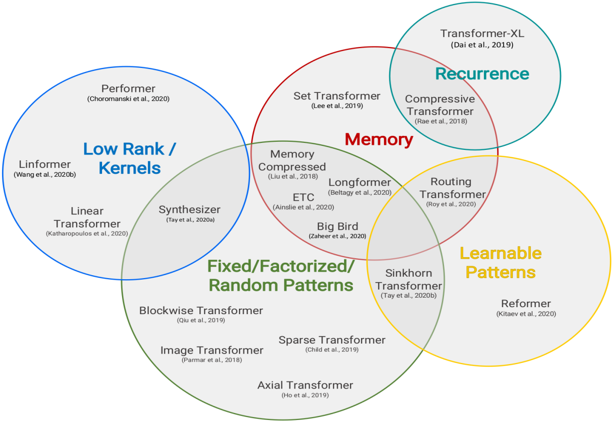
"Attention is all you need" — Google's Transformer architecture (Vaswani et al. 2017)

Multiple layers of attention networks in both encoder and decoder



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

Latest Transformer variants



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

(Kitaev, Kaiser, and Levskaya 2020) (Wang et al. 2020) (Choromanski et al. 2020) (Beltagy, Peters, and Cohan 2020) (Tay et al. 2020b) (Roy et al. 2021) (Jaegle et al. 2021)

Break-out groups

Let’s keep in mind the distinction between

- problems to be solved (e.g. “I want to understand this foreign language”), and
- solution methods ("I’ll apply this machine learning method")

A single method (family) can be useful for many problems/tasks.

Break-out groups

Translation is

mapping from one arbitrary length sequence to another arbitrary length sequence,

- (form of the task)

where the sequences are in different natural languages.

- (the semantics of the task)

The encoder-decoder NMT model is one kind of sequence-to-sequence model.

Discuss in break-out groups (5 min):

Other tasks that you can use an NMT architecture for?

Same form, different semantics.

Findings from break-out sessions

Text in, text out

Text summarization

- Long text $\mapsto$ Short text

Writing style transfer

- Text $\mapsto$ Text with same meaning, but as if written by another author

Paraphrasing

- Sentence $\mapsto$ Sentence with same meaning, but expressed differently

Punctuation prediction

- Text without punctuation $\mapsto$ Text with punctuation

NLG from structured data

- Database lookup $\mapsto$ Sentence containing the information

Modernizing language

- Archaic language $\mapsto$ Modern language

Dialogue systems

- Preceding dialogue $\mapsto$ Next utterance

Findings from break-out sessions (cont.)

Text in, text out (cont.)

Question answering

- Question $\mapsto$ Answer

Sentence fusion

- Two sentences about the same subject $\mapsto$ A single sentence combining them

Grammar correction

- Incorrent text $\mapsto$ Corrected text

Humor generation

- Non-joke $\mapsto$ Joke

Story generation

- Synopsis $\mapsto$ Story

Code generation

- Problem description $\mapsto$ Source code

Findings from break-out sessions (cont.)

Text in, tags out

Parsing

- Text $\mapsto$ Dependency/constituency parse

Lemmatization

- A word as characters $\mapsto$ Lemma as characters

Morphological analysis

- A word as characters $\mapsto$ Lemma and morphological tags

Morphological reinflection

- Wordform and desired tags $\mapsto$ Inflected form

Canonical segmentation

- A word as characters $\mapsto$ Canonical morphemes

Named entity regognition

- Sentence $\mapsto$ NER tags

Other modalities

Speech recognition

- Speech audio $\mapsto$ Transcription

Image/video captioning

- Image/video $\mapsto$ Description

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?

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

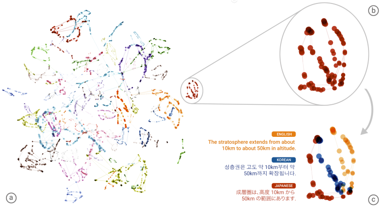
- Each language pair is one task.

Multilingual settings:

- one-to-many
- many-to-one
- many-to-many

Cross-lingual transfer: Zero-shot and universal translation

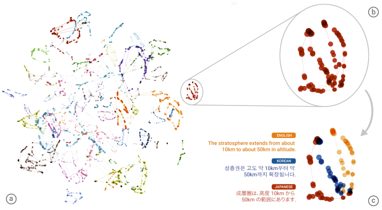
Many-to-many translation enables new language pairs without training data (“zero-shot”) or explicit pivot language.



Google’s multilingual NMT (Johnson et al. 2016)

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.



Google's multilingual NMT (Johnson et al. 2016)

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

How effective is cross-lingual 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.

Very 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↔Cs, En↔De, En↔Ru.

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.

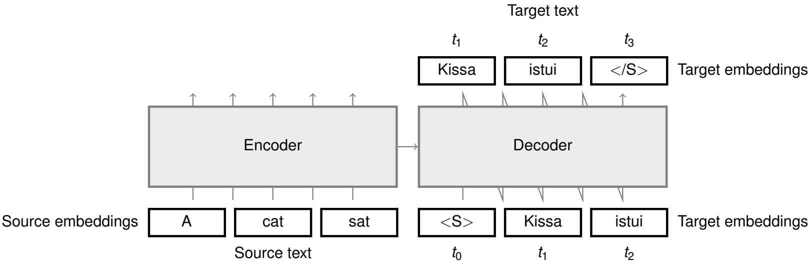
1. Pretrained source or target embeddings
2. Pretrained subnetwork (encoder or decoder)
3. Pretraining entire network (encoder and decoder)
   - E.g. finetuning a multilingual LM for MT (Liu et al. 2020)

Monolingual corpora: Pretraining

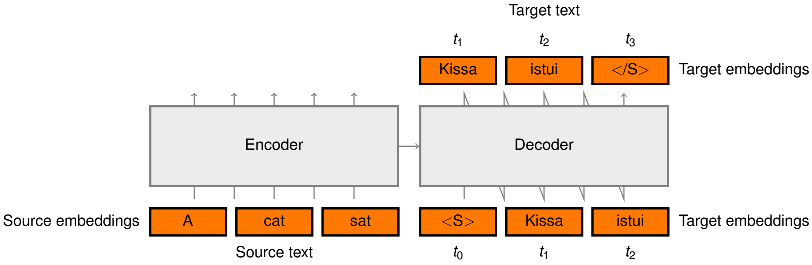
Sequential transfer: Train a component of the model on monolingual data.

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

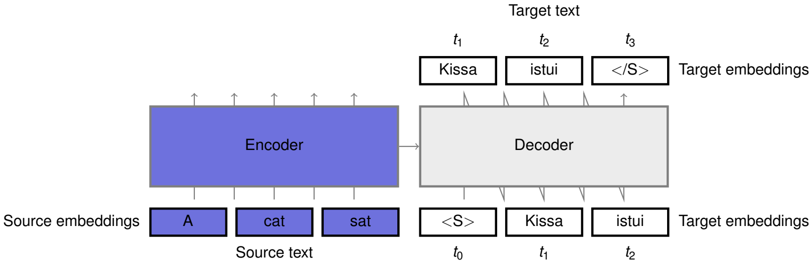


Parameter sharing in NMT transfer learning



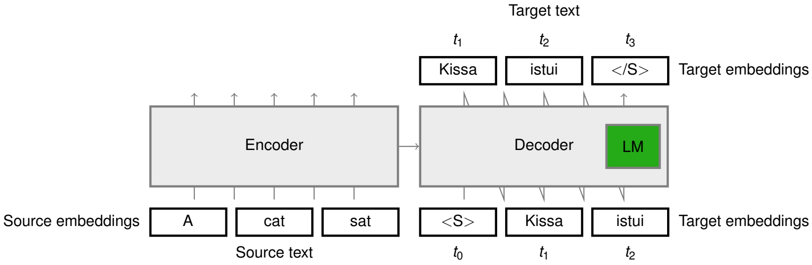
Pretrained embeddings

Parameter sharing in NMT transfer learning



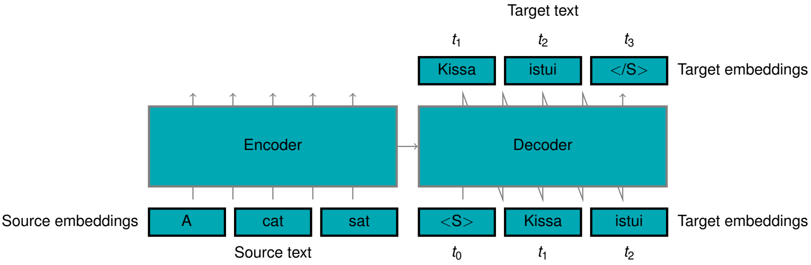
Pretrained encoder

Parameter sharing in NMT transfer learning



Language model fusion

Parameter sharing in NMT transfer learning



Full parameter sharing

Parameter sharing in NMT transfer learning

For (cross-lingual) transfer, parameters have to shared between languages.

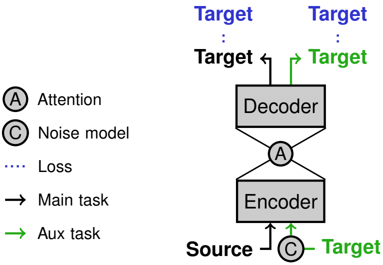
- Full sharing: All model parameters shared (mark languages with special tokens "⟨TO_FI⟩" or embeddings)
- Partial sharing: Share only a subnetwork (e.g. encoder)
- Soft sharing: Learn a dependency between the parameters instead of sharing them directly (Platanios et al. 2018)

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

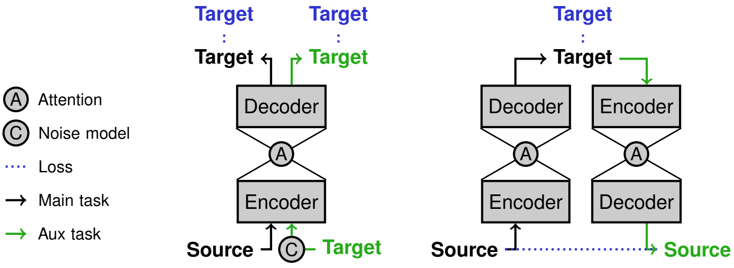
Parallel transfer: Use multi-task learning with source-to-source or target-to-target autoencoding as an additional task.



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.

Attention model may limit the use of characters.

- Softmax operation on input tokens.

Decoder output symbols

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

Conclusion

Subword units (morphological segmentation if available, or statistical subwords) may be a good compromise.

Multilingual units

Current standard practice in segmentation:

- SentencePiece (Kudo 2018)

Still popular:

- Byte-pair encoding (BPE) (Sennrich, Haddow, and Birch 2016b)
- See Lecture 6 for details

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.

- Identical words will have the same segmentation in all languages.
- The NMT system can learn to make character-by-character copy of rare names.

Challenges

Training is computationally very expensive.

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

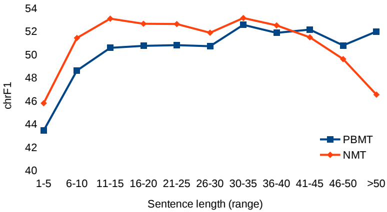
NMT is a "black box" system.

- 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)

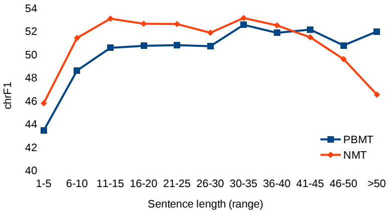


  - Now the challenge is in document-level translation.
- Good fluency, but sometimes very misleading translations — can be less predictable than PBMT

Challenges (cont.)

Translation quality issues

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

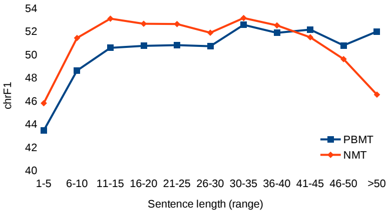


  - Now the challenge is in document-level translation.
- Good fluency, but sometimes very misleading translations — can be less predictable than PBMT
  - EN: Stealing food is a common crime in student halls.

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

Completely optional additional reading

Most of these topics are discussed in further detail in my PhD thesis (Grönroos 2020).

- Section 3.2.2 Neural models (Sequence2sequence models)
- Section 5.2.3 Neural machine translation (History of NMT)
- Section 3.4.2 Transfer and Multi-task learning
- Section 5.3.3 Multilingual translation
- Section 5.3.4 Exploiting monolingual data
- Section 5.3.1 Vocabulary construction (Subword units)

Not in the exam.

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.

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.

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.

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.

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.

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

Bibliography (cont.)

Kitaev, Nikita, Lukasz Kaiser, and Anselm Levskaya (2020). "Reformer: The Efficient Transformer". In: *International Conference on Learning Representations*.

Wang, Sinong et al. (2020). "Linformer: Self-attention with linear complexity". In: *arXiv preprint arXiv:2006.04768*.

Choromanski, Krzysztof et al. (2020). "Masked language modeling for proteins via linearly scalable long-context transformers". In: *arXiv preprint arXiv:2006.03555*.

Beltagy, Iz, Matthew E Peters, and Arman Cohan (2020). "Longformer: The long-document transformer". In: *arXiv preprint arXiv:2004.05150*.

Tay, Yi et al. (2020b). "Synthesizer: Rethinking self-attention in transformer models". In: *arXiv preprint arXiv:2005.00743*.

Roy, Aurko et al. (2021). "Efficient content-based sparse attention with routing transformers". In: *Transactions of the Association for Computational Linguistics* 9, pp. 53–68.

Bibliography (cont.)

Jaegle, Andrew et al. (2021). "Perceiver: General perception with iterative attention". In: *International Conference on Machine Learning*. PMLR, pp. 4651–4664.

Johnson, Melvin et al. (2016). "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation". In: *CoRR* abs/1611.04558. URL: http://arxiv.org/abs/1611.04558.

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.

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.)

Platanios, Emmanouil Antonios et al. (Aug. 2018). "Contextual Parameter Generation for Universal Neural Machine Translation". en. In: *arXiv:1808.08493 [cs, stat]*. arXiv: 1808.08493. URL: http://arxiv.org/abs/1808.08493.

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.

Bibliography (cont.)

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.

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.

Bibliography (cont.)

Grönroos, Stig-Arne (2020). "Machine translation into morphologically rich low-resource languages". Aalto University publication series DOCTORAL DISSERTATIONS; 202/2020. PhD thesis. Aalto University, 200 + app. 188. ISBN: 978-952-64-0169-0 (electronic); 978-952-64-0168-3 (printed). URL: http://www.waino.org/Gronroos_-_Machine_translation_into_morphologically_rich_lowresource_languages_thesis_2020.pdf.

Part II

Machine Translation Evaluation

Outline

Human evaluation

Automatic evaluation

Meta-evaluation

How to evaluate MT systems?

Final evaluation should depend on the intended application

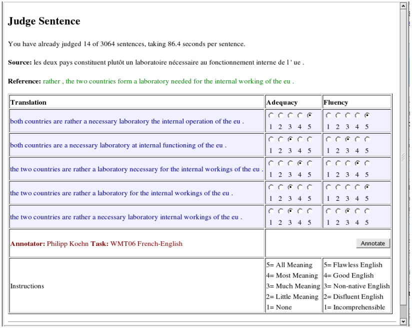
- Understanding text as it is; skimming/gisting $\rightarrow$ Human evaluation
- Aid for human translations $\rightarrow$ Decrease in translation time
- Multilingual information retrieval $\rightarrow$ IR evaluation

Human evaluation: Direct assessment

Given translation output and source and/or reference translation, how good the translation is?

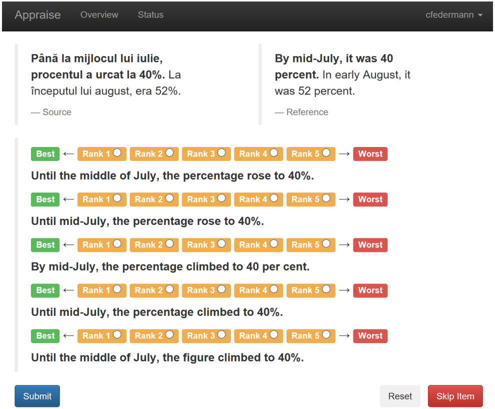
Adequacy: Does the output convey the same meaning?

Fluency: Is the output good and fluent language?



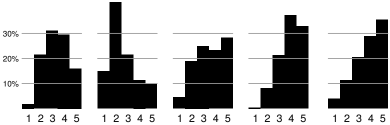
Human evaluation: Ranking

Given N translation output and source, order them from best to worst.



Human evaluation: Agreement

Evaluators disagree (WMT 2006):



Inter-evaluator agreement can be measured with Kappa coefficient:

$$K = \frac{p(A) - p(E)}{1 - p(E)}$$

$p(A)$ = proportion of agreement

$p(E)$ = agreement by chance

Ranking provides more consistent results than direct assessment.

Evaluating translator efficiency gain

How does the average translation time per sentence change?

- From scratch
- Using only translation memory
- Between different MT systems

Challenges:

- Translators have different experience and ways of working
- High variability between translation segments
- Easiest cases often solved by translation memories
- How to present the translation in the UI

Needs lots of data or complicated setup and advanced analysis (e.g. mixed-effect regression models).

Why automatic evaluation?

Manual evaluation is expensive

MT researchers rarely have the resources.

Annual competitions (WMT shared tasks) help somewhat.

Manual evaluation is slow

Cannot be used during development.

Especially not for optimization of model parameters and hyperparameters.

Challenges in automatic evaluation

Why MT evaluation is more difficult than in ASR evaluation? Why not use word error rate (WER)?

Challenges in automatic evaluation

Multiple correct answers: Ideally there should be several reference translations made by different persons.

Graded correctness: Word choices, grammatical correctness, emphasis (“koira jahtasi kissaa” vs. “kissaa koira jahtasi”), style (“kick the bucket” vs. “die”), ...

Usefulness depends on intended use.

- Translator’s tool: Long segments that require no changes
- Skimming: Meaning should be correct; fluent enough for easy understanding
- Information retrieval: Terminology important; fluency and grammatical correctness do not matter

Global edit distance metrics

Word and letter error rates do not account possible variations in word order.

Edit distance with moves is an NP-hard problem.

Solutions:

- TER: Shift operation + greedy search (Snover et al. 2006)
- SPEDE: Limited-distance word swapping (Wang and Manning 2012)

Local metrics

Concentrate on small parts of the full text at a time.

Similarity to IR metrics:

- **Precision:** Every item should be found in the reference.
- **Recall:** Anything in the reference should not be left out.

Observing individual words in not adequate (word order!)

Local metrics: BLEU

BLEU ("Bilingual Evaluation Understudy") (Papineni et al. 2002) was one of the first metrics to report high correlation with human judgments of quality.

Log-linear model parameters can be tuned directly for the score.

$$\text{BLEU} = \min\left(1, \frac{\texttt{output-length}}{\texttt{reference-length}}\right)\left(\prod_{i=1}^{4} \texttt{precision}_i\right)^{\frac{1}{4}}$$

Typically calculated over entire corpus (system-level evaluation).

Example:

(by Philipp Koehn, http://www.statmt.org/book/)

Example

SYSTEM A: $\boxed{\text{Israeli officials}}$ responsibility of $\boxed{\text{airport}}$ safety
2-GRAM MATCH 1-GRAM MATCH

REFERENCE: Israeli officials are responsible for airport security

SYSTEM B: $\boxed{\text{airport security}}$ $\boxed{\text{Israeli officials are responsible}}$
2-GRAM MATCH 4-GRAM MATCH

Metric	System A	System B
precision (1gram)	3/6	6/6
precision (2gram)	1/5	4/5
precision (3gram)	0/4	2/4
precision (4gram)	0/3	1/3
brevity penalty	6/7	6/7
BLEU	0%	52%

Local metrics: Problems in BLEU

Does not work for languages with no word boundaries.

Single word or n-gram is scored 0 or 1.

- Inflections: “translation” vs. “translations”
- Derivations: “[he] made translations” vs. “[he] translated”
- Compounds: “Arbeits Geberverband” vs. “Arbeitgeberverband” *(employers’ organization)*

Poor measure of adequacy for morphologically rich languages.

Beyond word-based metrics

Preprocessing (stemming, morphological segmentation)

- METEOR (Banerjee and Lavie 2005; Denkowski and Lavie 2011)
- AMBER (Chen and Kuhn 2011)

Characted-based measures

- char-BLEU (Denoual and Lepage 2005)
- Weighted character F-score (chrF3) (Popović 2015)

Combine with word similarity calculation

- Alignment based on character similarity (Homola, Kuboň, and Pecina 2009)
- Tolerant BLEU (Libovický and Pecina 2014)
- LeBLEU (Virpioja and Grönroos 2015)

Semantic similarity using contextual embeddings

- BERTscore (Zhang et al. 2019)

How to evaluate evaluation metrics?

Goals

- Correct: better systems have higher scores
- Interpretable: intuitive interpretation of translation quality
- Consistent: repeated use gives the same results
- Low cost: efficient computation, no extra work or linguistic resources needed
- Tuning compatible: can be used to tune translation systems

WMT Metrics shared task

- Long-running comparison of evaluation metrics
- Correlation with human evaluation scores

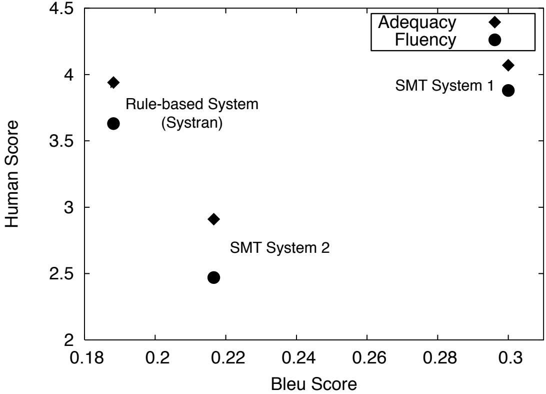
How to evaluate evaluation metrics?

Even if a metric works for comparing similar MT systems, it should not to be trusted for comparing very different ones.

Example from http://www.statmt.org/book/:

Evidence of Shortcomings of Automatic Metrics

Rule-based vs. statistical systems



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?

See e.g. https://www.linkedin.com/pulse/microsoft-mt-reaches-parity-bad-human-translation-tommi-nieminen
or (Toral et al. 2018; Läubli et al. 2020)

Bibliography

Snover, Matthew et al. (2006). "A Study of Translation Edit Rate with Targeted Human Annotation". In: *Proceedings of Association for Machine Translation in the Americas*, pp. 223–231. URL: http://www.cs.umd.edu/~snover/pub/amta06/ter_amta.pdf.

Wang, Mengqiu and Christopher Manning (2012). "SPEDE: Probabilistic Edit Distance Metrics for MT Evaluation". In: *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, pp. 76–83. URL: http://www.aclweb.org/anthology/W12-3107.

Papineni, Kishore et al. (2002). "BLEU: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA, USA: Association for Computational Linguistics, pp. 311–318. URL: http://www.aclweb.org/anthology/P02-1040.

Bibliography (cont.)

Banerjee, Satanjeev and Alon Lavie (2005). “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”. In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 65–72. URL: http://www.aclweb.org/anthology/W/W05/W05-0909.

Denkowski, Michael and Alon Lavie (2011). “Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems”. In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, pp. 85–91. URL: http://www.aclweb.org/anthology/W11-2107.

Chen, Boxing and Roland Kuhn (2011). “AMBER: A Modified BLEU, Enhanced Ranking Metric”. In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, pp. 71–77. URL: http://www.aclweb.org/anthology/W11-2105.

Bibliography (cont.)

Denoual, Etienne and Yves Lepage (2005). "BLEU in characters: towards automatic MT evaluation in languages without word delimiters". In: *Companion Volume to the Proceedings of the Second International Joint Conference on Natural Language Processing*, pp. 81–86. URL: https://www.aclweb.org/anthology/I/I05/I05-2014.pdf.

Popović, Maja (2015). "chrF: character n-gram F-score for automatic MT evaluation". In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, pp. 392–395. URL: http://aclweb.org/anthology/W15-3049.

Homola, Petr, Vladislav Kuboň, and Pavel Pecina (2009). "A Simple Automatic MT Evaluation Metric". In: *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Athens, Greece: Association for Computational Linguistics, pp. 33–36. URL: http://www.aclweb.org/anthology/W09-0403.

Bibliography (cont.)

Libovický, Jindřich and Pavel Pecina (2014). "Tolerant BLEU: a Submission to the WMT14 Metrics Task". In: *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, pp. 409–413. URL: http://www.aclweb.org/anthology/W14-3353.

Virpioja, Sami and Stig-Arne Grönroos (2015). "LeBLEU: N-gram-based Translation Evaluation Score for Morphologically Complex Languages". In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, pp. 411–416. URL: http://aclweb.org/anthology/W15-3052.

Zhang, Tianyi et al. (2019). "BERTScore: Evaluating Text Generation with BERT". In: *International Conference on Learning Representations*.

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.