Universitat Politècnica de València

Master in Artificial Intelligence, Pattern Recognition and Digital Imaging

2023-2024

MACHINE TRANSLATION

5. Advanced Topics in Neural Machine Translation

Francisco Casacuberta

fcn@prhlt.upv.es

October 24, 2023

Index

Index

Why embeddings?

- Traditional approach to MT: Discrete representation of words and sentences.
- Most techniques of machine learning are developped in continuous spaces (i.e. vector spaces)
- In machine learning, (deep) neural networks are good models for many aplications.

Units

- Byte level text representation [Wang AAAI 2020]
- Character.
- Sub-word (BPE, SentencePiece)
- Word.
- Phrase.
- Sentence.
- Paragraph.
- Document.

Embeddings

- Unit representation as vectors.
- Many NLP applications.
- Word embeddings from monolingual corpus: i.e. word2vec (Sec. 2, Chap. 3).
- Word embeddings from a training process: Sub-product of sequence-to-sequence training (Sec. 4-6, 3).
- Word embeddings from a character embedding using CNN (Sec. 7, Chap. 3).
- Word embeddings from pre-trained LMs: i.e. BERT (topic 5 in Chap. 5), ELMo [Peters NAACL 2018], GPT-2, GPT-3, ...
- Multilingual word embeddings: Represent words from multiple languages in a single distributional vector space [Chen+ EMNLP 2018].

A common vector space for word embeddings

[Conneau+ ICLR 2018][Artetxe+ ACL 2018]

- Learn a linear mapping $\widehat{W}$ between the source W_X^E and the target W_Y^E embeddings:

$$\widehat{W} = \underset{W}{\text{argmin}} \, \|W \; W_X^E - W_Y^E\|$$

- Align monolingual word embeddings.
  - Supervised, from a train bilingual dictionary.
  - Unsupervised using adversarial training,

- Toolkit MUSE [Conneau+ ICLR 2018]
  `https://github.com/facebookresearch/MUSE`

- Toolkit VecMap [Artetxe+ ACL 2018]
  `https://github.com/artetxem/vecmap`

Sentence embeddings

- Sum, product, arithmetic or grametric mean of word embeddings.
- LASER: Multilingual Sentence Embeddings [Artetxe arXiv 2019].
- Doc2Vec [Le & Mikolov arXiv 2014].
- SentenceBERT from BERT [Reimers & Gurevych arXiv 2019].
- InferSent [Conneau arXiv 2018].
- Universal Sentence Encoder [Cer arXiv 2018].

Index

Why pre-trained models?

- Training neural machine translation models from scratch is expensive.
- Large bilingual corpora are necessary.
- For many task-specific translation models there are low resources.
- Pre-trained models have proved to be useful in many scenarios. And in translation?
  - Combination of pretrained encoders and pretrained decoders?
  - Multilingual pretrained language models: Prompt engineering.
- Large Language Models in 2023 [Dilmegani 2023]:
  `https://research.aimultiple.com/large-language-models/`
  `https://research.aimultiple.com/large-language-model-training/`

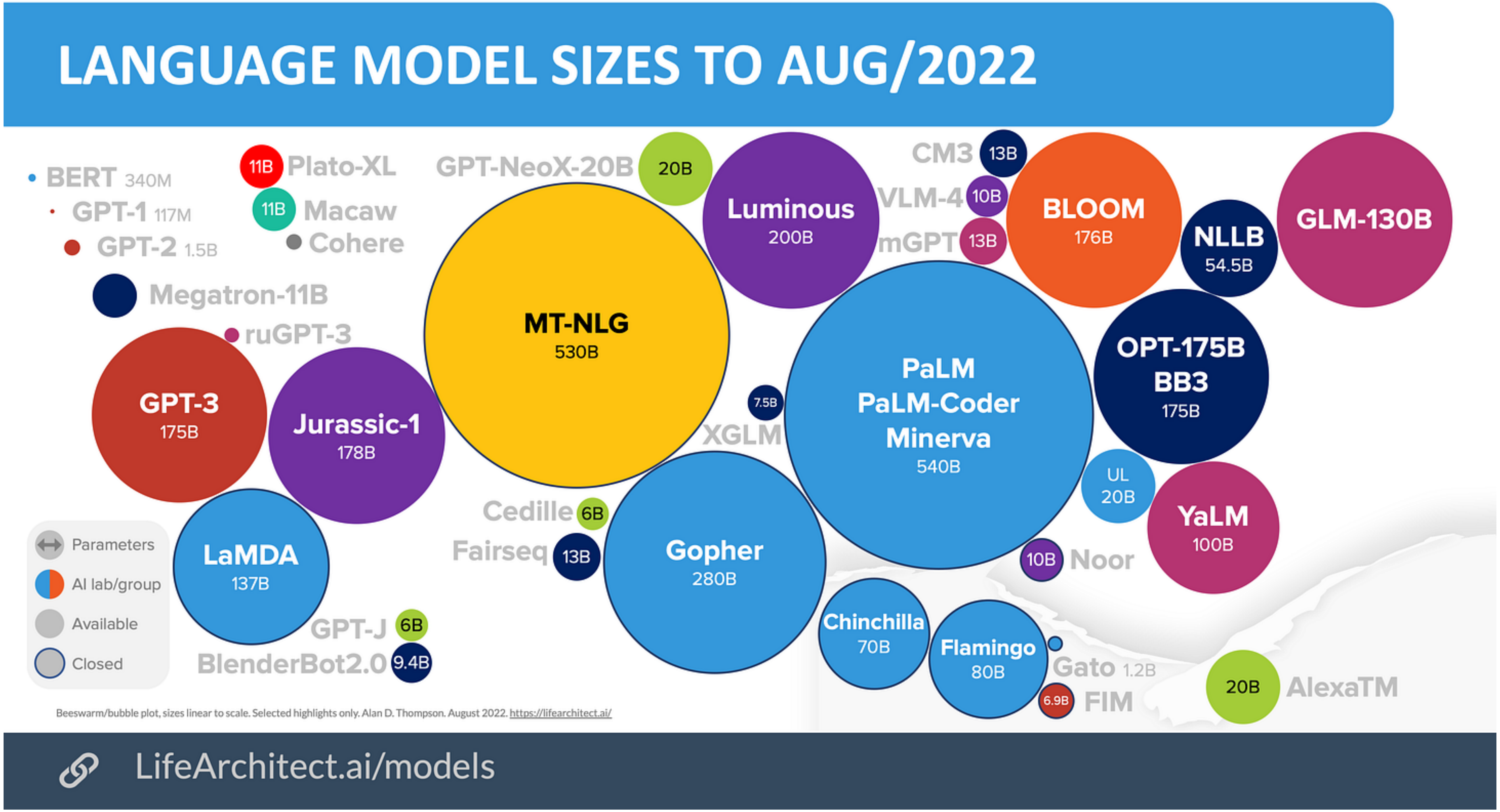
Pre-trained models

- Bidirectional Encoder Representations from Transformers (BERT). Based on the Transformer encoder. [Devlin NAACL 2019] -Google-
- Generative Pre-trained Transformer (GPT, GPT-2, GPT-3 and GPT-4). Based on the Transformer decoder. [Radforf openAI 2018] -OpenAI-
  Demo: https://gpt3demo.com/apps/openai-gpt-3-playground
- ChatGPT (GPT-3.5 or Davinci -OpenAI-), GPT-J (Eleuther AI), nano GPT https://github.com/karpathy/nanoGPT.
- Visual ChatGPT: ChatGPT + Stable Diffusion 2.
- Generalized Autoregressive Pretraining (XLNet). Decoder of Transformer without masks + permutation language modelling. [Yang NIPS 2020] -Google-
- BART: Full encoder-decoder (monolingual and multilingual) Transformer [Liu 2020] -Facebook AI-
- T5: Text-to-Text Transfer Transformer (Trained with Colossal Clean Crawled Corpus (C4)). Complete Transformer. [Raffel 2020] -Google-

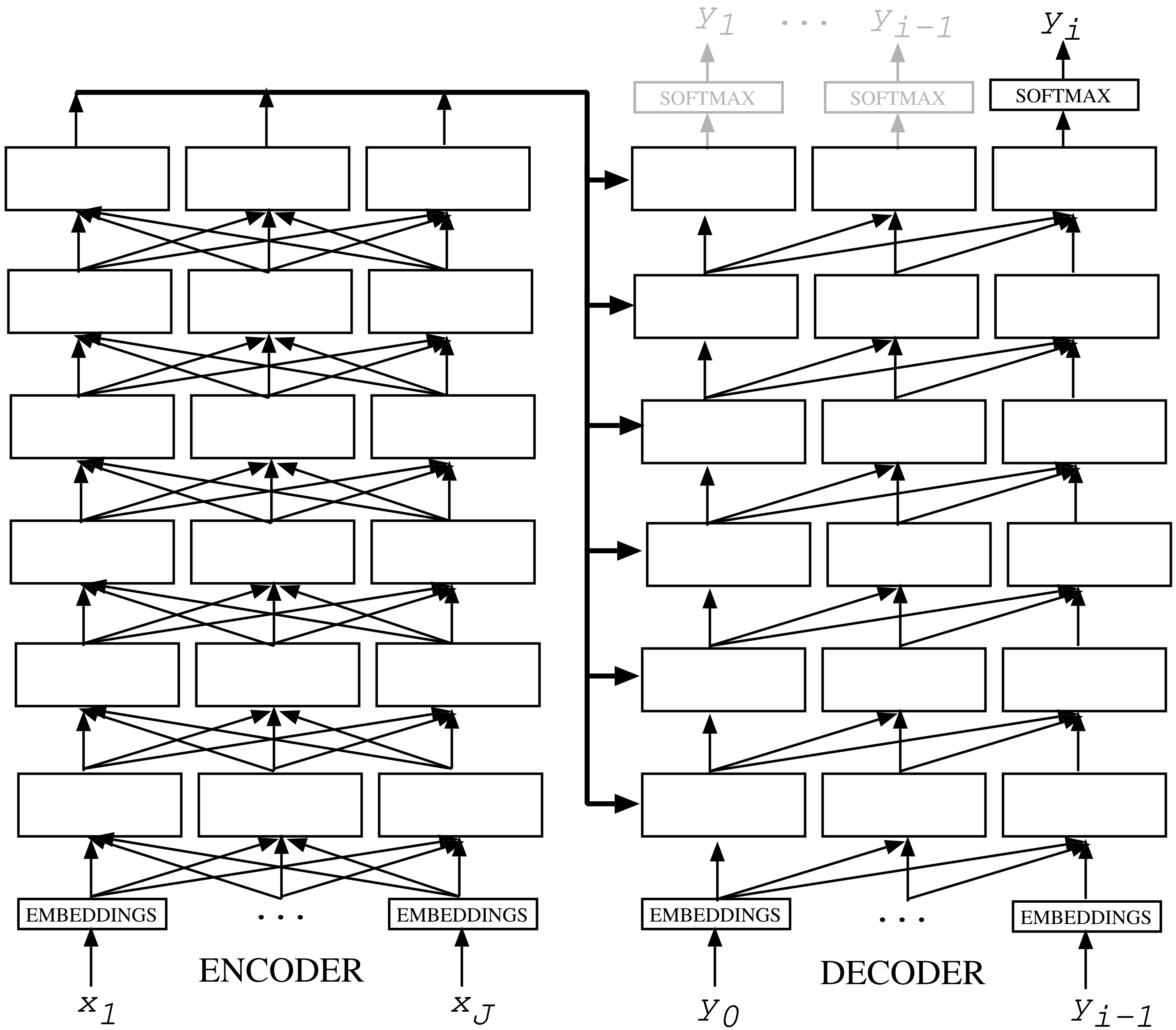
Pre-trained models

- Megatron-Turing Natural Language Generation (MT-NLG). English 530B. [Shoeybi arXiv 2019] -NVIDIA-
- Cross-lingual Language Model Pretraining (XLM). Transformer-based [Conneau NIPS 2019] -Facebook AI-
- Gopher. [Borgeaud DeepMind 2021] -DeepMind-
- Minerva-PaLM. 540G. Maths. [Lewkowycz arXiv 2022] -Google-
- Large Language Model Meta AI: LLaMA. 65B (1.4Trillon tokens) . -Meta AI-
- Survey: [Sun SCTS 2020], [Wang Engineering 2022].
- A platform for using pre-trained models: Hugging Face.
  https://huggingface.co/
- Talk on “Pre-training Methods for Neural Machine Translation” [Wang ACL 2021]
  https://sites.cs.ucsb.edu/~lilei/TALKS/2021-ACL/pre-training_nmt_ACL_tutorial_2021.pdf

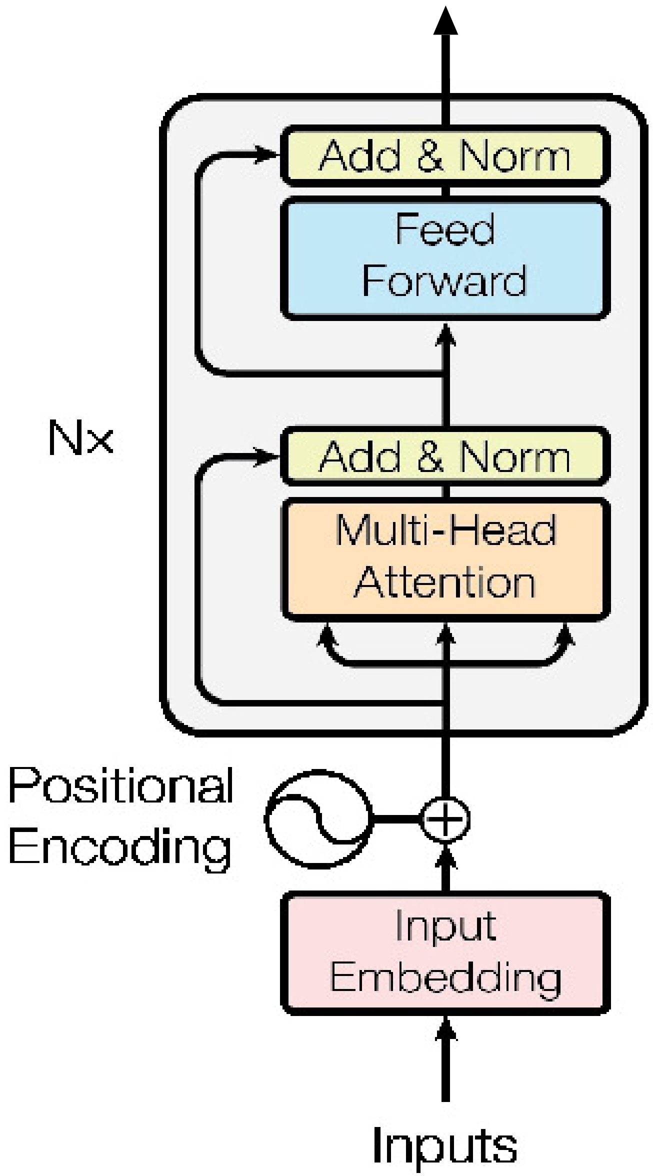
Pre-trained models



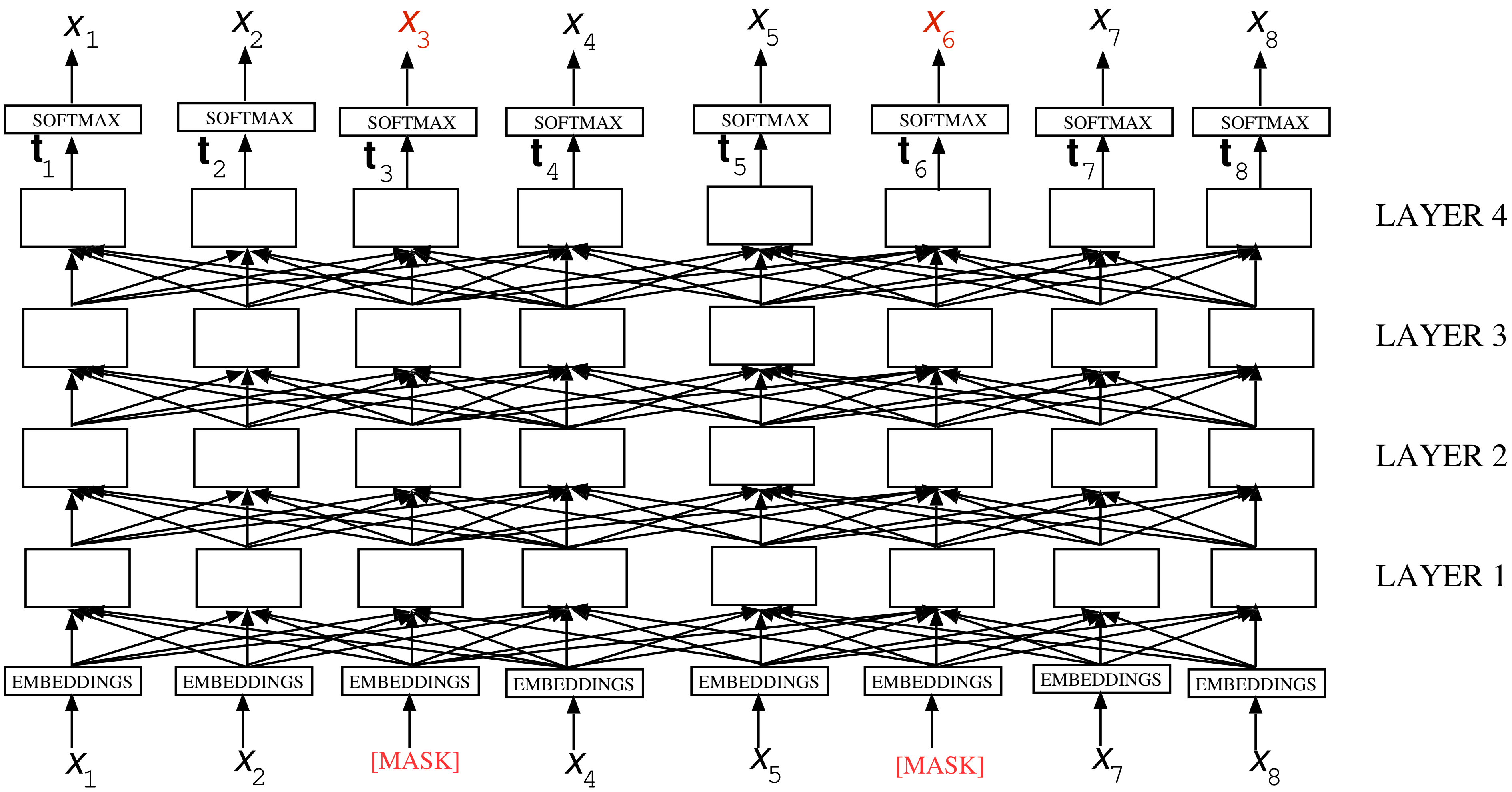
Transformer model



Encoder in Transformer (Vaswani 2017)



BERT model


x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
SOFTMAX
$\mathbf{t}_1$
$\mathbf{t}_2$
$\mathbf{t}_3$
$\mathbf{t}_4$
$\mathbf{t}_5$
$\mathbf{t}_6$
$\mathbf{t}_7$
$\mathbf{t}_8$
LAYER 4
LAYER 3
LAYER 2
LAYER 1
EMBEDDINGS
x_1
x_2
[MASK]
x_4
x_5
[MASK]
x_7
x_8

The layers in BERT (encoder of Transformer)

Given a source sentence x_1^J,

- Initialization: $\mathbf{h}_j^0 = \mathbf{x}_j = \mathcal{E}(x_j) \quad 1 \leq j \leq J$
- In layer l of the encoder($1 \leq l < L$)
  - Self-attention model:

$$\mathbf{u}_j^{l+1} = \mathbf{a}(\mathbf{h}_1^l, \ldots, \mathbf{h}_J^l, \mathbf{h}_j^l) \quad 1 \leq i \leq J$$

  - Feed-forward network:

$$\mathbf{h}_j^{l+1} \;=\; \mathbf{F}_f(\mathbf{u}_j^{l+1}) \quad 1 \leq j \leq J$$

- In layer L

$$p(\cdot) \;=\; \mathbf{f}_{sm}(\mathbf{t}_j) \;=\; \mathbf{f}_{sm}(\mathbf{W}\ \mathbf{h}_j^L) \quad 1 \leq j \leq J$$

where $p(\cdot)$ is a probabilistic distribution on V_X.

The input of BERT [Devlin NAACL 2019]

Word embeddings: For each input token sum the token embedding plus segment embedding (for two input sentences) plus position embedding.

Embeddings	$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$
Positional embedding	E^p_0	E^p_1	E^p_2	E^p_3	E^p_4	E^p_5	E^p_6	E^p_7	E^p_8	E^p_9
	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
Sentence embeddings	E^s_0	E^s_1	E^s_2	E^s_3	E^s_4	E^s_5	E^s_6	E^s_7	E^s_8	E^s_9
	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
Tokens embeddings	$E^t_{\text{[CLS]}}$	$E^t_{x_1}$	$E^t_{x_2}$	$E^t_{\text{[MASK]}}$	$E^t_{x_4}$	$E^t_{\text{[SEP]}}$	$E^t_{x'_1}$	$E^t_{\text{[MASK]}}$	$E^t_{x'_3}$	$E^t_{\text{[MASK]}}$
Tokens	[CLS]	x_1	x_2	[MASK]	x_4	[SEP]	x'_1	[MASK]	x'_3	[MASK]
Input		x_1	x_2	x_3	x_4		x'_1	x'_2	x'_3	x'_4
	sentence A						sentence B			

$$\mathbf{x}_j = \mathcal{E}(x_j) = E^t_{\bar{x}_j} + E^p_j + E^s_j \quad 0 \leq j \leq 9$$

$$E^s_j = \begin{cases} E_A & 0 \leq j \leq 5 \\ E_B & 6 \leq j \leq 9 \end{cases}$$

Masked language model [Devlin NAACL 2019]

- Training masked language with BERT:
  - Masked language model: during training mask some percentage of the input tokens at random, and then predict those masked tokens.
  - In 15% of positions at random:
    - 80% the token is substituted by [MASK].
    - 10% the token is substituted by another random token.
    - 10% the token remains unchanged.
  - The output corresponding to [CLS] is used for classification.
  - Slow convergence.

- Fine tuning with BERT:
  - For classification: adding a classifier layer in the correspondient output of the [CLS] token.

Masked language model

- Given a sentence x_1^J, with $x_j \in \mathcal{V}_X$ for $1 \leq j \leq J$, let $\mathcal{J} \subset \{1, \ldots, J\}$.

- A masked sentence of x_1^J by $\mathcal{J}$ is a sentence $\bar{x}_1^J$, such that:

$$\bar{x}_j = \begin{cases} \text{[MASKED]} & j \in \mathcal{J} \\ x_j & \text{otherwise} \end{cases} \quad \text{for} \quad 1 \leq j \leq J$$

- The probability of the masked words is:

$$p(\{x_j : j \in \mathcal{J}\} \mid \bar{x}_1^J) = \prod_{j \in \mathcal{J}} p(x_j \mid \bar{x}_1^J) = \prod_{j \in \mathcal{J}} \mathbf{f}_{sm}(\mathbf{W}_o\, \mathbf{h}_j^L)_{i(x_j)}$$

- Given a set of traning sentences $\{x^{(k)}\}_{k=1}^N$, the training loss is

$$\mathcal{L}(\mathbf{W}) \;=\; -\sum_{k=1}^{N} \log p(\{x_j^{(k)} : j \in \mathcal{J}_k\} \mid \bar{x}^{(k)})$$

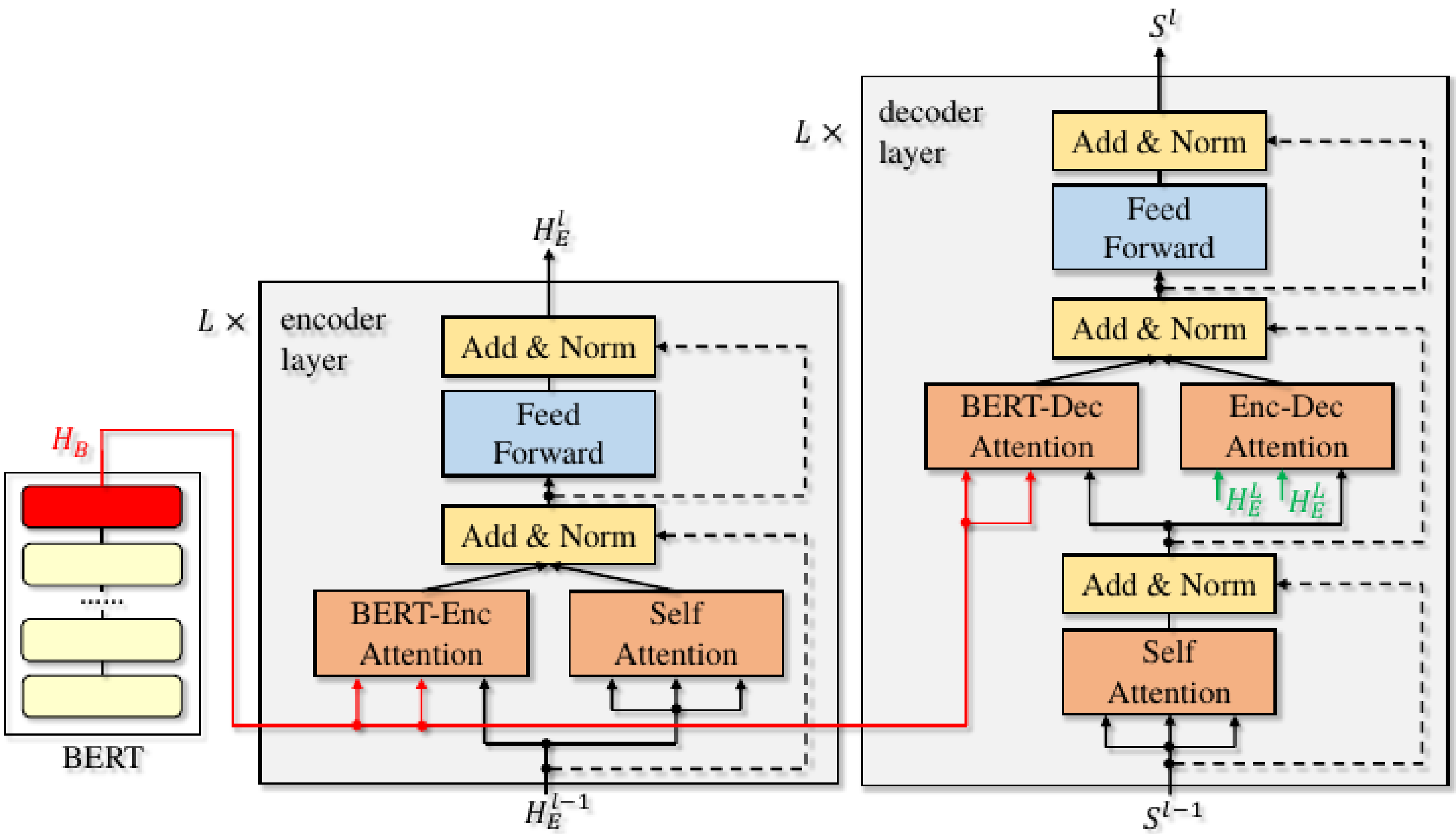
BERT model [Devlin NAACL 2019]

- # Layers=12 or 24; # layer size = 768 or 1024; # self-attention heads = 12 or 16.
- Sentence embedding:
  - The first output token corresponding to [CLS] previous to the softmax operation.
  - The sum of the word embeddings $\sum_{j=1}^{J} \mathbf{t}_j$
  - The mean the word embeddings $\frac{1}{J}\sum_{j=1}^{J} \mathbf{t}_j$
- BERT as a Markov Random Field Language Model [Wang NeuralGen 2019]
- Extension: Multilingual BERT (mBERT) [Pires ACL 2019]
  - Monolingual text of Wikipedia from 104 languages.
  - Shared Word-piece vocabulary.
  - Good for zero-shot cross-lingual model transfer.
  - No marker of language is used.
  - mBERT presents syntactic properies across languages [Chi ACL 2020].

Other pre-trained BERT-like models

- ALBERT: A Lite BERT [Lan arXiv 2019]
- SBERT: Siamese and triplet network structures [Reimers EMNLP 2019] 5 architecture [Lewis ACL 2020]
- RoBERTa: A Robustly Optimized BERT Preraining Approach. [Liu arXiv 2019]
- Char-Bert; BERT at character level [Ma COLING 2020]
- DistilBERT: a smaller general-purpose language representation model [Sahn NeuroIPS 2019]
- ExpBERT, GAN-BERT, MobileBERT, DeeBERT, schuBERT, SentiBERT, BERTRAM, CluBERT, MTSI-BERT, SiBERT, FlauBERT, NegBERT, AraBERT, BioBERT, SciBERT, ClinicalBERT, TransBERT, DocBERT, PatentBERT, XLNet, SpanBERT, ... [ACL 2020][LREC 2020]

BERT-fused for machine translation [Radford openAI 2018]



BLEU in IWSLT'14 En→De	
Standard Transformer	28.6
Bert-fused model	30.5

Translation Language Model (XML) [Conneau NIPS 2019]

				x_3				y_2		y_4
	Encoder									
Embeddings	$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$
Positional embedding	E^p_0	E^p_1	E^p_2	E^p_3	E^p_4	E^p_0	E^p_1	E^p_2	E^p_3	E^p_4
	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
Language embeddings	E^l_0	E^l_1	E^s_2	E^l_3	E^l_4	E^l_5	E^l_6	E^l_7	E^l_8	E^l_9
	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
Tokens embeddings	$E^t_{\text{[CLS]}}$	$E^t_{x_1}$	$E^t_{x_2}$	$E^t_{\text{[MASK]}}$	$E^t_{x_4}$	$E^t_{\text{[SEP]}}$	$E^t_{y_1}$	$E^t_{\text{[MASK]}}$	$E^t_{y_3}$	$E^t_{\text{[MASK]}}$
Tokens	[CLS]	x_1	x_2	[MASK]	x_4	[SEP]	y_1	[MASK]	y_3	[MASK]
Input		x_1	x_2	x_3	x_4		y_1	y_2	y_3	y_4
		English					French			

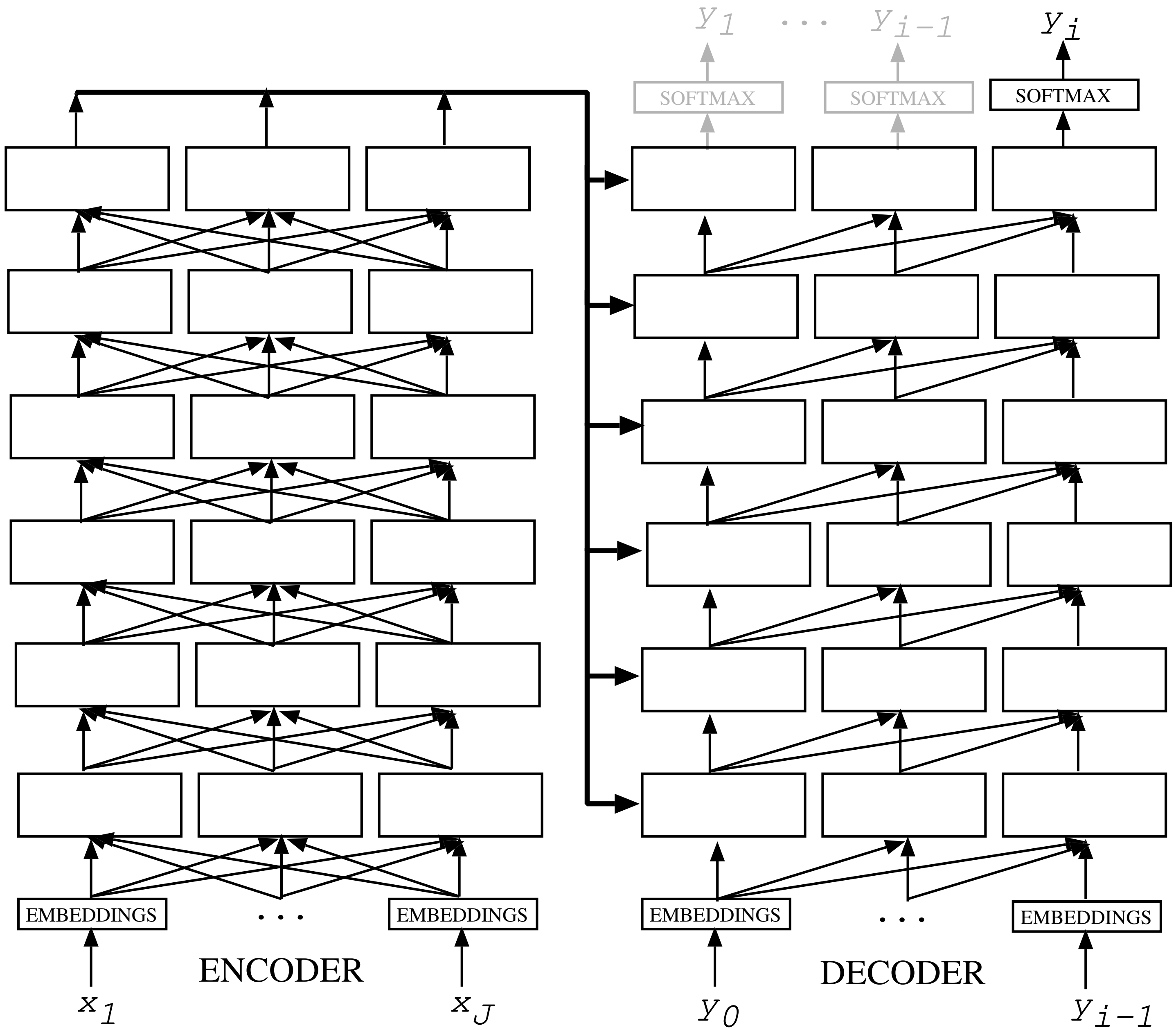
$$\mathbf{x}_j = \begin{cases} E^t_{x_j} + E^p_j + E^l_j & 0 \leq j \leq 4 \\ E^t_{y_j} + E^p_j + E^l_j & 5 \leq j \leq 9 \end{cases}$$

$$E^l_j = \begin{cases} E_{\text{English}} & 0 \leq j \leq 4 \\ E_{\text{French}} & 6 \leq j \leq 9 \end{cases}$$

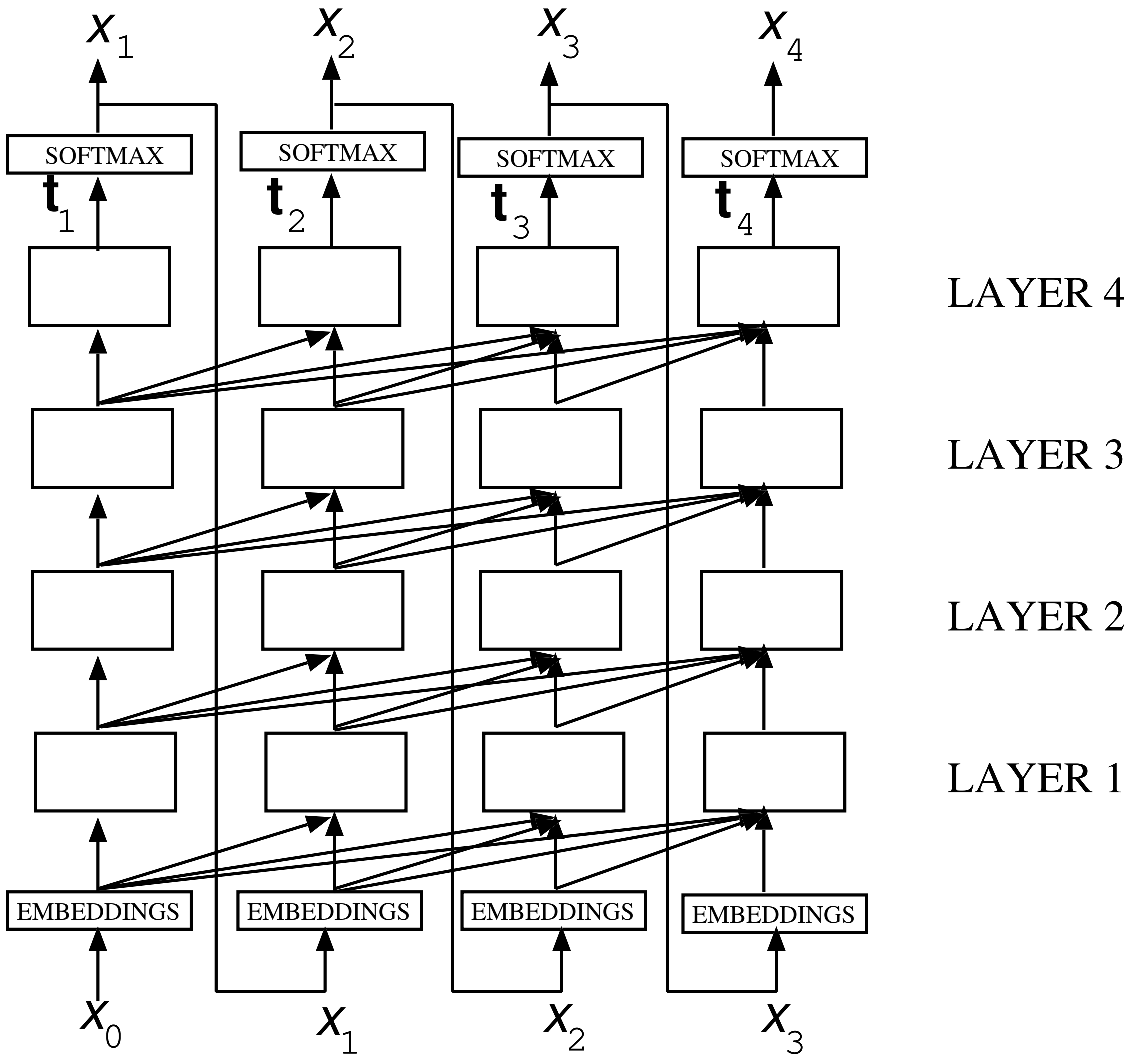
Generative Pre-trained Transformer [Zhu ICLR 2020]

- Encoder-based LM can not be generative models in a “natural” way.
- Decoder-based LM as a autorregresive model can generate sentences
- GPT family:
  - GPT (OpenAI): 802 million tokens. 117 million parameters [Radford openAI 2018].
  - GPT-2 (OpenAI): 984 million tokens (40 GB of text). 1.5 billion parameters [Radford openAI 2019].
  - GPT-3 (OpenAI): 300 billion tokens. 175 billion parameters [Brown openAI 2020].
  - GPT-J (EleutherAI): 400 billion tokens (600 GB of text). 6 billion parameters [BGao arXiv 2020].
  - Gopher (DeepMind): 300 billion tokens. 280 billion parameters [Rae arXiv 2021].
  - Chinchilla (DeepMind): 1.4 trillion tokens. 530 billion parameters. [Hoffman arXiv 2022].
  - GPT-4 (OpenAI): 100 trillion parameters. 2023.
- Prompting (Self supervised learning?)

Transformer model



GPT model


x_1
x_2
x_3
x_4
SOFTMAX
SOFTMAX
SOFTMAX
SOFTMAX
t_1
t_2
t_3
t_4
LAYER 4
LAYER 3
LAYER 2
LAYER 1
EMBEDDINGS
EMBEDDINGS
EMBEDDINGS
EMBEDDINGS
x_0
x_1
x_2
x_3

The layers in GPT (decoder of Transformer)

For $1 \leq j \leq J$, given a prefix of a sentence x_1^{j-1}, for generating word x_j:

- Initialization: $\mathbf{h}_j^0 = \mathbf{x}_{j-1} = \mathcal{E}(x_{j-1})$
- In layer l of the encoder($1 \leq l < L$)
  - Self-attention model:

$$\mathbf{u}_j^{l+1} = \mathbf{a}(\mathbf{h}_1^l, \ldots, \mathbf{h}_j^l, \mathbf{h}_j^l)$$

  - Feed-forward network:

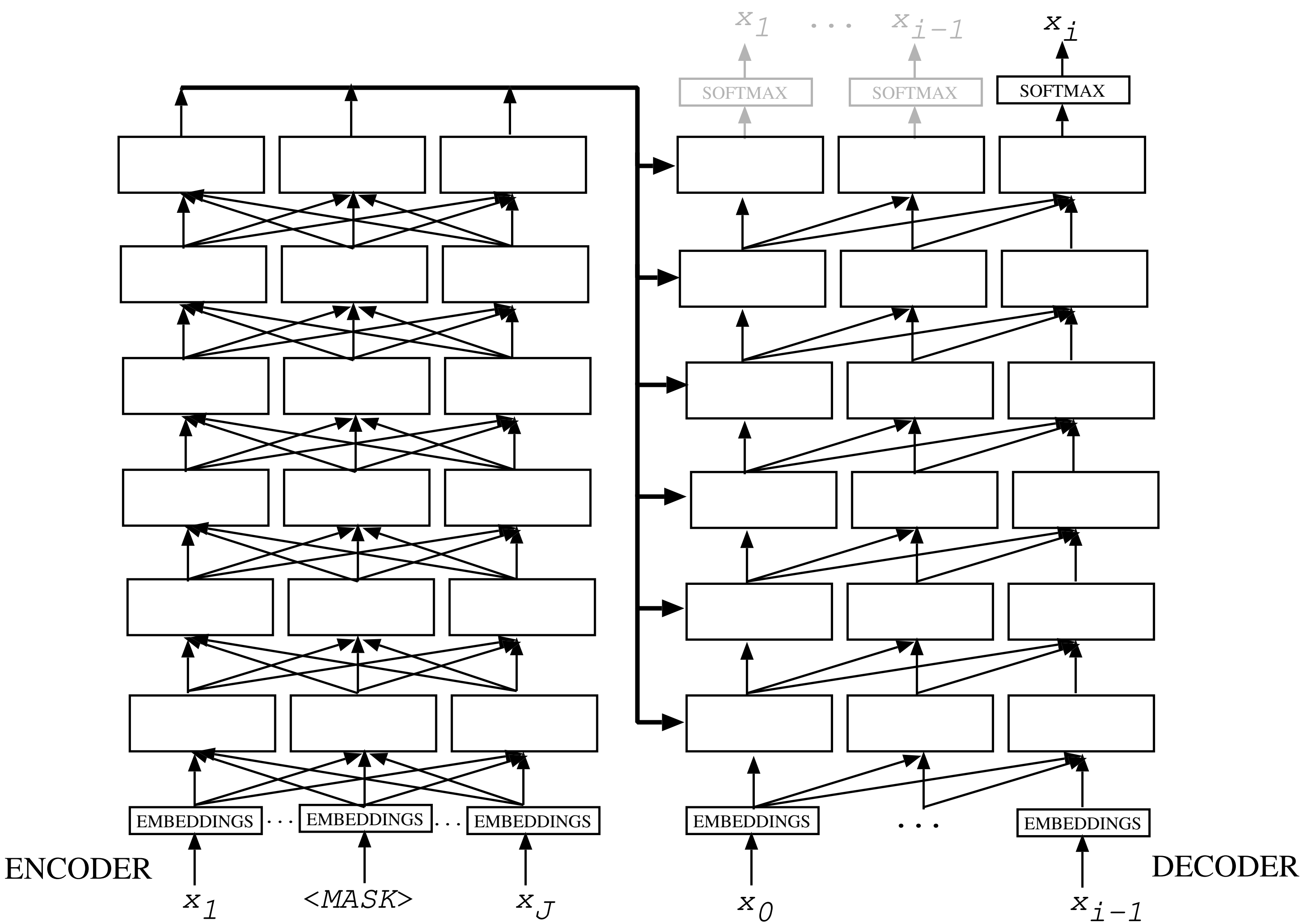
$$\mathbf{h}_j^{l+1} \; = \; \mathbf{F}_f(\mathbf{u}_j^{l+1})$$

- In layer L

$$p(\cdot) \; = \; \mathbf{f}_{sm}(\mathbf{t}_j) \; = \; \mathbf{f}_{sm}(\mathbf{W}\ \mathbf{h}_j^L)$$

where $p(\cdot)$ is a probabilistic distribution on V_X.

BART: Denoising Sequence-to-Sequence Pre-Training [Lewis ACL 2020]



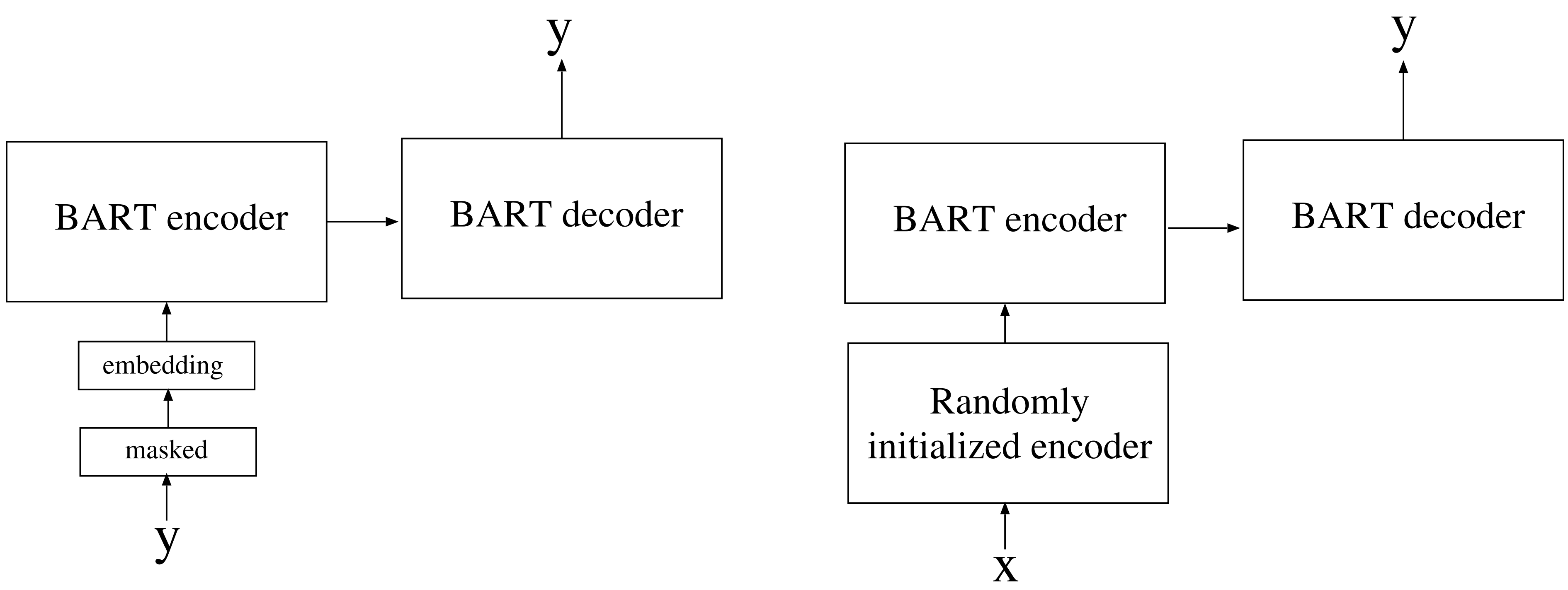
BART: Denoising Sequence-to-Sequence Pre-Training

[Wang & Li ACL 2020]

- Trained by corrupting documents and then optimizing a reconstruction loss.
  - Token masking.
  - Token deletion.
  - Sentence permutation.
  - Document Rotation.

Fine-Tune on Neural Machine Translation [Wang & Li ACL 2020]

Replace BART encoder embedding layer with a new randomly initialized encoder



Fine-Tuning [Ruder 2021]

- Adaptive fine-tuning (unsupervised): a way to bridge os distributions that shifts in distribution by fine-tuning the model on data that is closer to the distribution of the target data.

- Behavioural fine-tuning (supervised): Given a target labels.

- Parameter-efficient fine-tuning: Part of the parameter set is frozen.

- Text-to-text fine-tuning: Prompt learning.

- Mitigating fine-tuning instabilities: Differente techinique to deal with instabilities (low learning rates, early stopping, ...)

Index

Why multilingual translation?

- For low-resource languages.
- For no resources: zero-shot translation models.
- To take profit of common features in similar languages.
- A single engine instead many engines.

Scenarios for multilingual neural machine translation

[Dabre+ arXiv 2020]

- Multi-way translation. The goal is constructing a single NMT system for one-to-many, many-to-one or many-to-many translation using parallel corpora for more than one language pair. Parallel corpora are available for each language pairs.

- Low resource translation. (a) a high-resource language pair is available to assist a low-resource language pair. (b) no direct parallel corpus for the low-resource pair and a pivot language is used.

- Multi-source translation. Documents that have to be translated into more than one language [Zoph+ arXiv 2016].

- Multilingualism with RBMT and SMT: Interlingua approach.

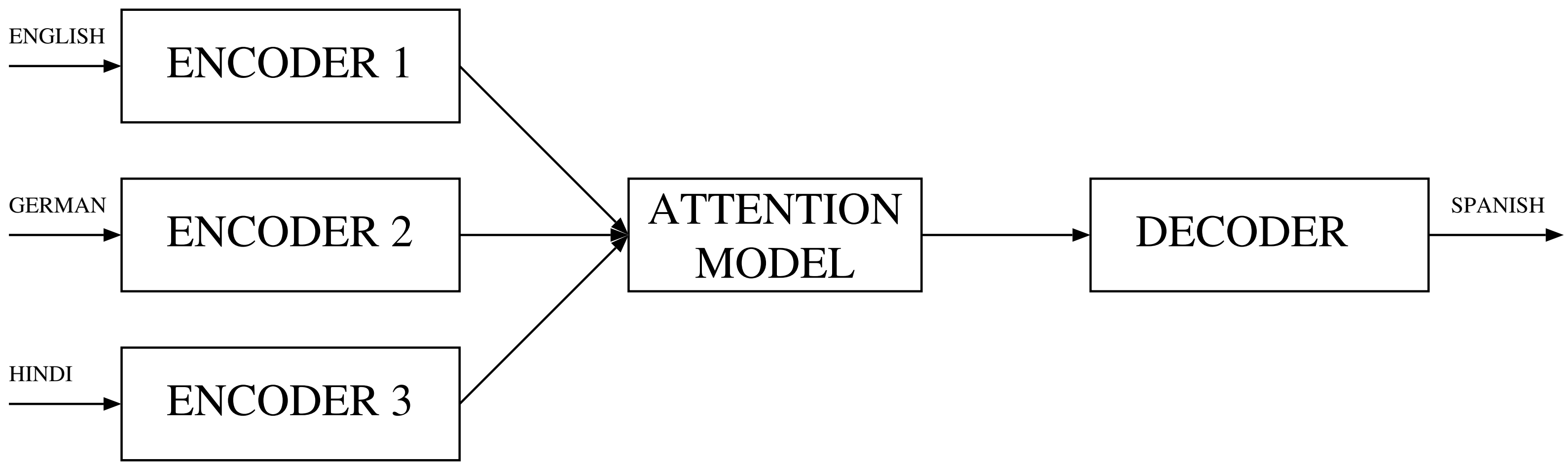
Models for multi-way neural machine translation

- One encoder for all source languages or one encoder for each source language.

- One decoder for all target languages or one decoder for each target language.

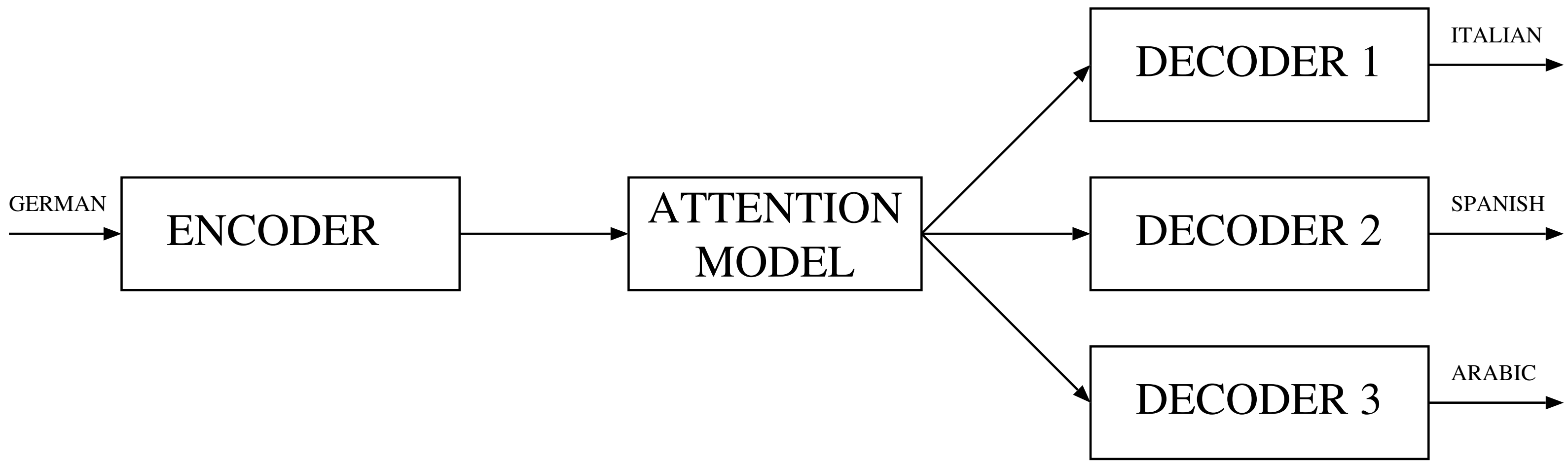
Multi-way neural machine translation (I)

Several languages to one language



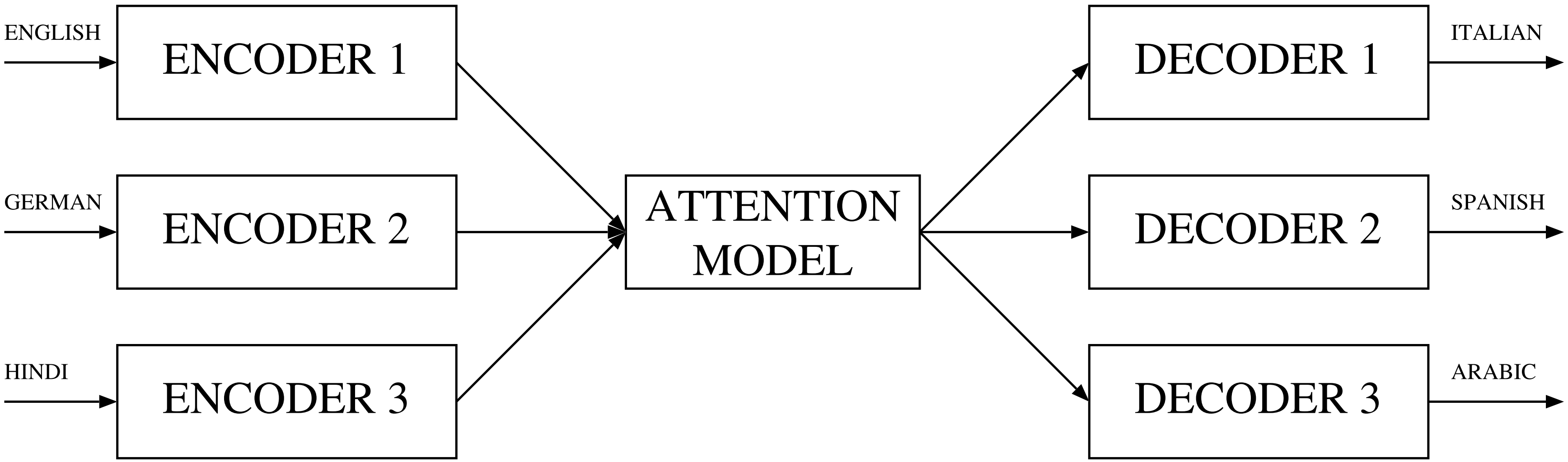
Multi-way neural machine translation (II)

One language to several languages



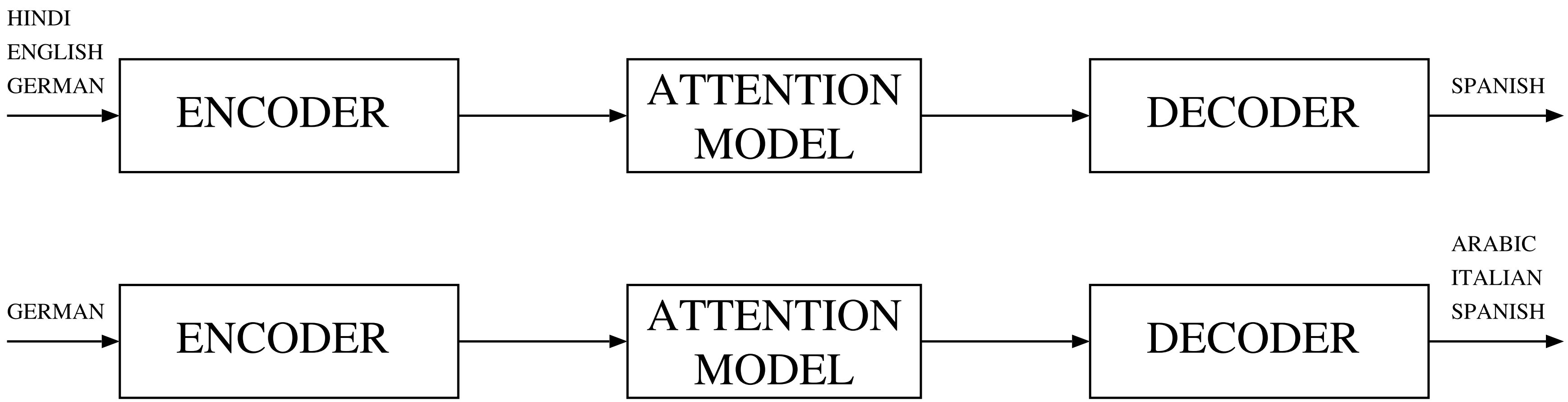
Multi-way neural machine translation (III)

Several languages to several languages



Multi-way neural machine translation

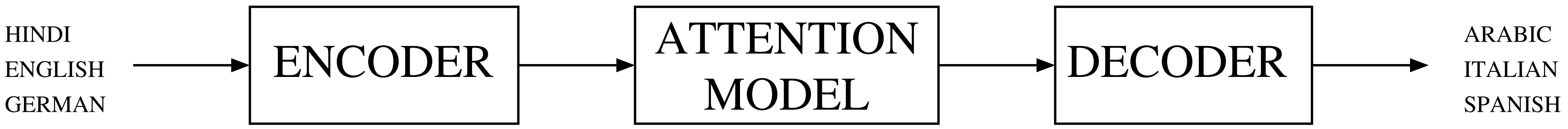
One encoder/decoder for all languages.



The language tag trick: A tag is added to each sentence to identify the language.

Multi-way neural machine translation

One encoder/decoder for all languages.



- Training:
  - Parallel corpus English-Arabic.
  - Parallel corpus Hindi-Italian.
  - Parallel corpus German-Spanish.

- Inference:
  - English to Arabic.
  - Hindi to Italian.
  - German to Spanish.
  - English to Italian.
  - English to Spanish.
  - Hindi to Spanish.
  - Hindi to Arabic.
  - German to Italian.
  - German to Arabic.

Training with MNMT [Dabre+ arXiv 2020]

- Regular training criterion for one language pair given a training bilingual corpus T:

$$\widehat{\mathbf{W}} = \operatorname*{argmax}_{\mathbf{W}} \mathcal{F}_T(\mathbf{W}) \equiv \operatorname*{argmax}_{\mathbf{W}} \sum_{(x_1^J, y_1^I) \in T} \log p(y_1^I \mid x_1^J; \mathbf{W})$$

- Training criterion for a set of language pairs ($L \subset S \times D$) pairs given a the corresponding training bilingual corpus T_l for $l \in L$:

$$\widehat{\mathbf{W}} = \operatorname*{argmax}_{\mathbf{W}} \sum_{l \in L} \mathcal{F}_{T_l}(\mathbf{W}) \equiv \operatorname*{argmax}_{\mathbf{W}} \sum_{l \in L} \sum_{(x_1^J, y_1^I) \in T_l} \log p(y_1^I \mid x_1^J; \mathbf{W})$$

- Single stage parallel/joint training
  - For models with separate encoders and decoders, each batch consists of sentence pairs for a specific language pair whereas for fully shared models, a single batch can contain sentence pairs from multiple language pairs.

MNMT for low-resources languages pairs [Dabre+ arXiv 2020]

- The high-resource and low-resource language pairs share the same target language. Jointly training both language pairs.
- Fine tune: First, they trained a parent model on a high-resource language pair. The child model is initialized with the parent's parameters wherever possible and trained on the small parallel corpus for the low-resource pair.
- Lexical Transfer.
- Syntactic Transfer

MNMT for unseen languages pairs [Dabre+ arXiv 2020]

- Zero-shot translation: The MNMT system has not been trained for the unseen language pair, but the system is able to generate reasonable target language translations for the source sentence.

- Zero-resource translation: Using a pivot language i.e. by synthetic corpus generation using a pivot language.

Some experimental results [Cuevas TFM 2020]

- From one language to several languages.
- One decoder for all target languages & one decoder for each target language.
- Toolkit: NMT-keras.
- Corpus: Europarl.
  - Source language: English.
  - Target languages: Spanish, German and French.

Some experimental results [Cuevas TFM 2020]

Experiment	Source language	Target language	BLEU	TER
Baseline	English	Spanish	23.6	58.0
		French	24.7	60.0
		German	15.0	68.3
One decoder	English	Spanish	21.3	61.0
		French	22.4	63.0
		Spanish	22.5	60.0
		German	14.4	73.6
		French	22.4	63.8
		German	13.5	74.2
Two decoders	English	Spanish	25.4	57.0
		French	25.5	59.0
		Spanish	25.4	56.9
		German	17.0	68.0
		French	26.6	59.0
		German	16.8	68.6

Index

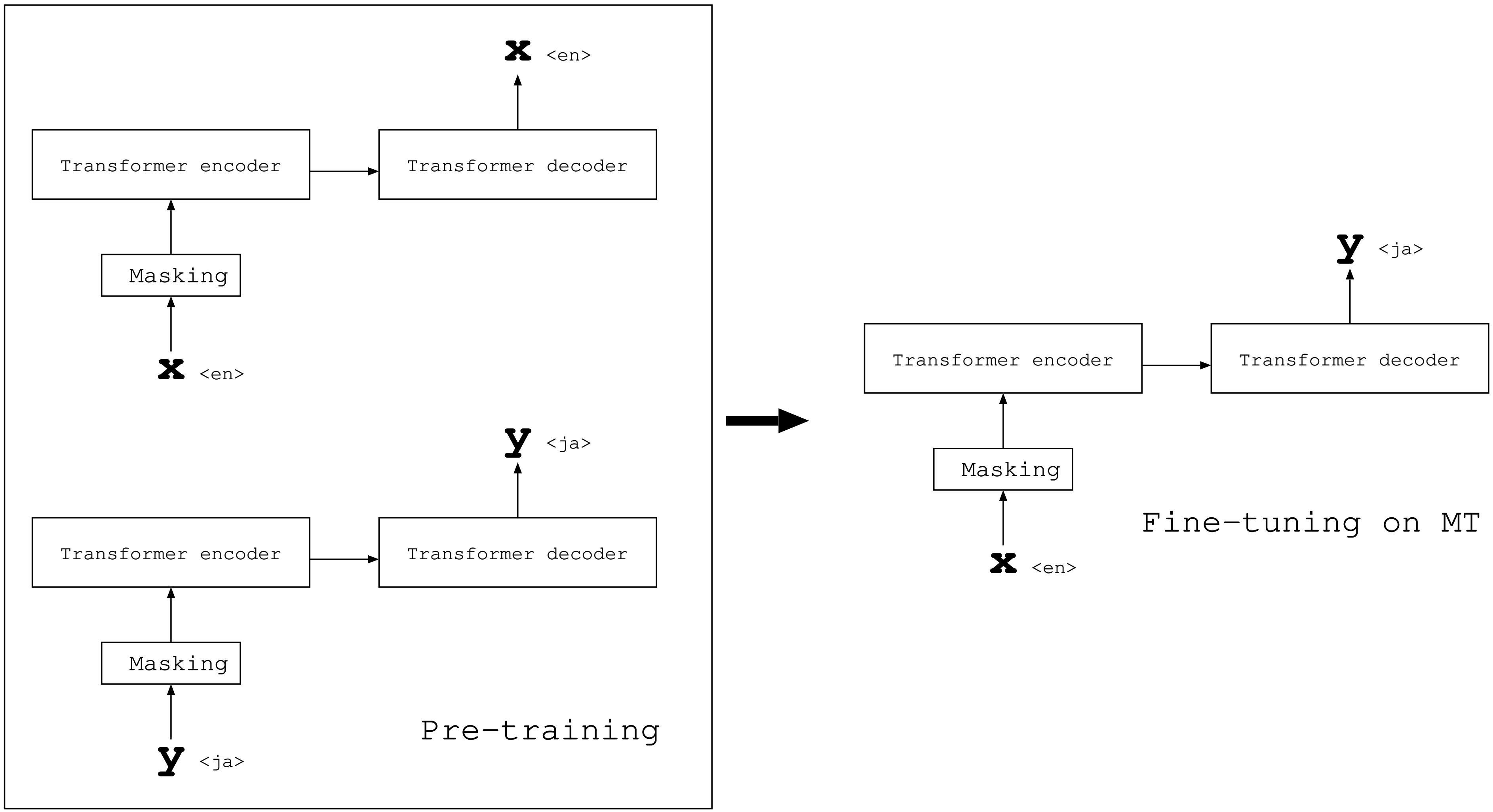
Why multilingual pre-trained models? [Lewis ACL 2020]

- Data scarcity for low/zero resource languages.
- Transfer knowledge from some languages to anothers.
- Consequence: Adding more languages improves performance on low-resource languages due to positive knowledge transfer.
- Adding a special token to identify the language.
- Fine-tuning

Pre-trained models and MT

- mBART. Multilingual BART. Full encoder-decoder (monolingual and multilingual) Transformer [Liu 2020] -Facebook AI-
- mT5. Multilingual T5: Text-to-Text Transfer Transformer (Trained with Colossal Clean Crawled orpus (C4)). Complete Transformer. [Raffel 2020] -Google-
- No Language Left Behind (NLLB). 200 different languages. Complete Transformer. [NLLB Team arXiv 2022] -Meta-
- BLOOM (BigScience Large Open-science Open-access Multilingual Language Model). Decoder of Transformer, 70 layers. [Le Scao arXiv 2022] -BigScience, Microsoft, NVIDIA, IDRIS/GENCI and BigScience-
- XLM: cross-lingual language models. BERT-based architecture [Conneau NIPS 2019] -Meta-

mBART [Lewis ACL 2020]



Index

Why document-based translation?

[Maruf+ arXiv 2019]

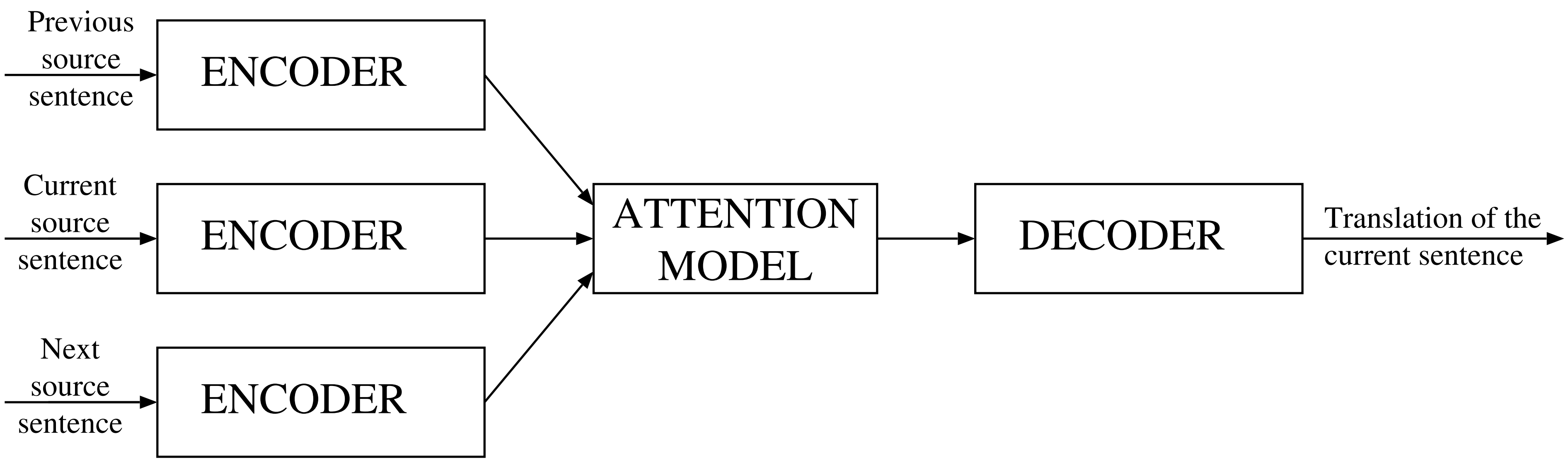
- In most of the applications, the goal is to translate a whole document and sometimes a paragraph.
- The common approach is to translate the document sentence by sentence as independent facts.
- However, sentence-based constraints do not deal with long-term dependencies as anaphora, ellipsis, word ambiguity, ...
- Therefore, the translation of a document sentence by sentence can suffer of a lack of coherence at document level.

Approaches to document-based translation [Huo+ WMT 2020]

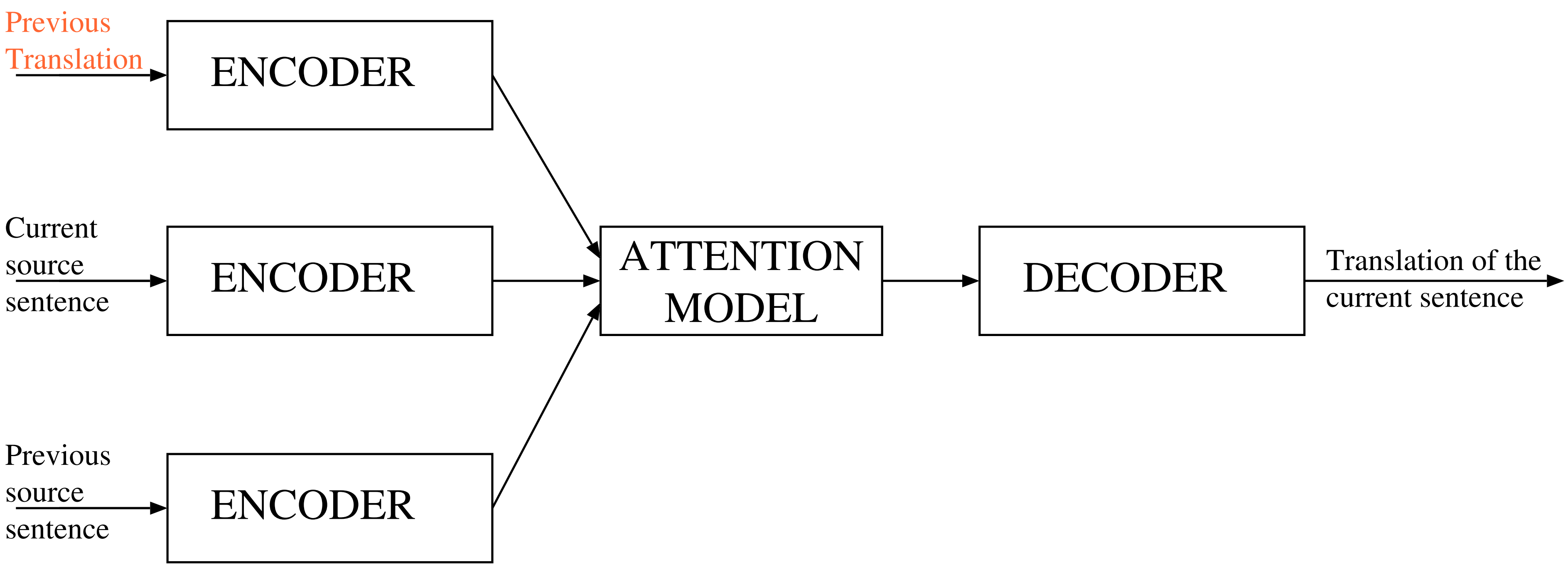
- SMT: Hard problem.

- NMT
  - Context witout a modification of the architecture:
    - ∗ Concatenate the current source and previous sentence.
  - Context via additional components:
    - ∗ Additional context encoder and combining the representations from the current and the previous source sentence to fed the decoder.
    - ∗ Additional context encoder and the current enconder feed to the cross-attention of the decoder in an independent way.
    - ∗ Additional context encoder and the current enconder feed in parallel to two cross-attentions of the decoder and a combination of the output.
    - ∗ A regular transformer for mapping source to target sentences plus a pretrained model BERT that deals with the concatenation of the current source and previous sentence.

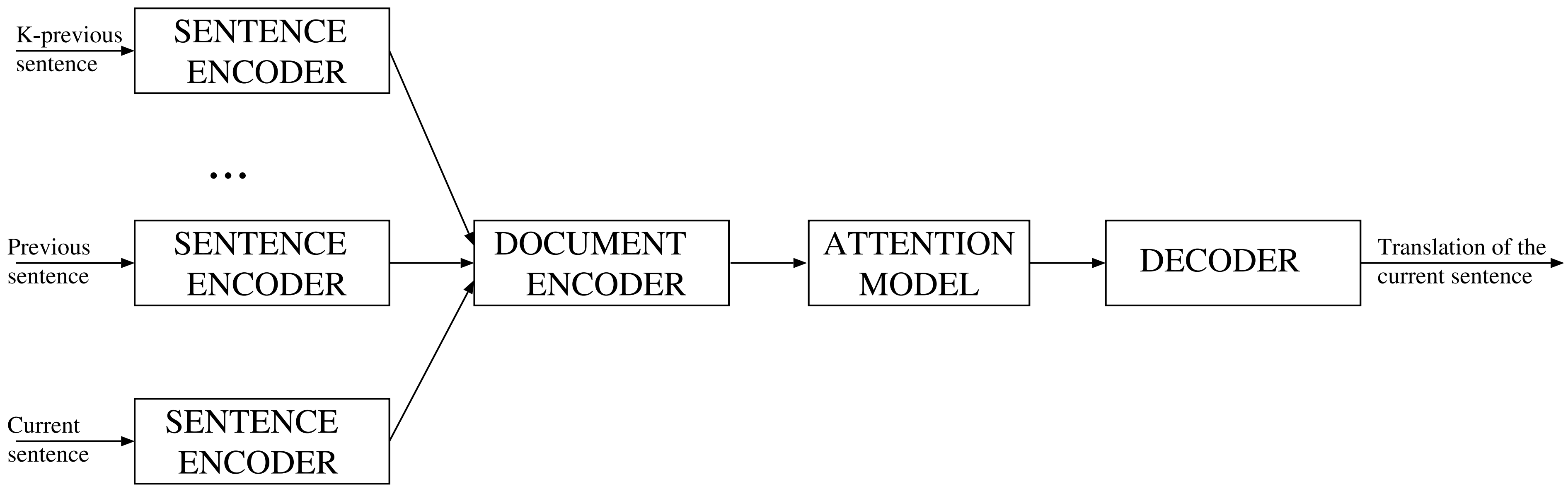
Additional components: source context in DNMT



Additional components: source & target context in DNMT



Additional components: two-level source context in DNMT



Some experimental results [Andújar TFM 2021]

- Multi-encoder architecture based on Transformer.
- 2 encoders.
- Toolkit: Keras and Tensorflow.
- Assesement: BLEU.
- Corpora:
  - TED Talks (Spanish-English and Spanish-German)
  - News Commentary (Spanish-English and Spanish-German)

Some experimental reults [Andújar TFM 2021]

System	TED		News	
	Sp-En	Es-Ge	Sp-En	Es-Ge
Baseline	34.3	18.6	11.8	11.2
Context	34.8	18.9	12.3	11.3

Index

Why monolingual corpora?

- Low resource languages:
  - Small bilingual corpora.
  - Lack of bilingual corpora.
- Large availability of monolingual corpus in many languages.

On the use of monolingual corpora for NMT [Gibadullin arXiv 2019]

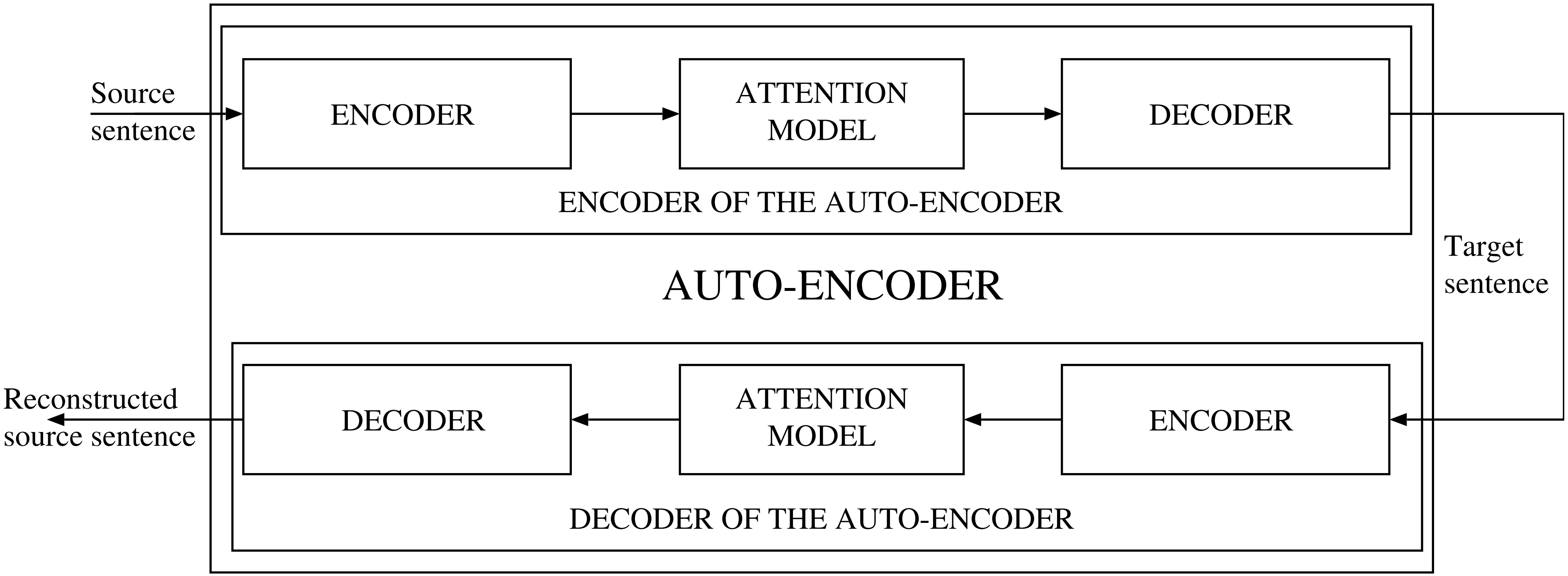
- Architecture independent methods:
  - Generate pseudo-parallel (synthetic) corpus using monolingual corpus.
  - Merge a target language model from monolingual corpora with NMT models.
- Architecture dependent methods:
  - Training with parameters freezing.
  - Integration of language modeling.
  - Pre-training language models.
- Fully unsupervised learning: From unsupervised multilingual word embeddings.

On the use of monolingual corpora for NMT [Gibadullin arXiv 2019]

Pseudo-parallel (synthetic) corpus

- Back-translation:
  - From a small parallel corpus, train a target to source model (T).
  - Translate the monolingual (target) corpus using T: Align source and target sentences (synthetic corpus)
  - True and synthetic corpora are merged to train a translation model.
- Round trip training:
  - Based on an auto-encoder.
  - Source-to-target translator is used as an encoder of auto-encoder and target-to-source as a decoder of the auto-encoder.
  - The whole training objective of the method is to maximize the likelihoods of source-to-target and target-to-source models on parallel corpus, and reconstruction likelihoods of auto-encoders on monolingual corpora.

Round Trip Training



On the use of monolingual corpora for NMT [Gibadullin arXiv 2019]

Merge with a separate language model

- Using monolingual target model to train a language model.
- Merge the target language model with translation models in the inference.
  - Shallow fusion: for each predicted target word, sum the probability of the translation model and the language model. Translation models and language models are trained in a separate way.

Architecture dependent methods [Gibadullin arXiv 2019]

- Training with parameters freezing:
  - Forward-translation: Pseudo-parallel corpus from monolingual source data and freezing decoder parameters when pseudo-parallel data is used.
  - Dummy input: Each target monolingual sentence is associated to a single-word null to produce the pseudo-parallel corpus. The parameters of the encoder and attention model is freezing when that pseudo-parallel corpus is used.
- Integration of language modeling in the training process:
  - Deep fusion: The hidden state of the LM and the hidden states of the decoder are concatenated.
- Pre-training with Language Models: Pre-training of the neural model: A pre-trained source LM is used as the encoder and a pre-trained target LM is used as the decoder. Parallel corpus is used for fine-tuning.

Fully unsupervised learning [Gibadullin arXiv 2019]

- Bilingual dictionary induction, and the pseudo-parallel corpus is obtained by applying the bilingual dictionary to the monolingual corpus.
  - Linear transformation of source word embeddings to target word embeddings. In this common space, translations of words in one language can be found by searching nearest neighbors among the words from another language. The transformation matrix can be found using some small seed dictionary or even without it [Artetxe ACL 2018].

    The linear mapping $\widehat{W}$ between the source W_X^E and the target W_Y^E embeddings:

$$\widehat{W} = \underset{W}{\text{argmin}} \, \|W \, W_X^E - W_Y^E\|$$

Some experimental results [Castellanos TFM 2020]

- Using back-translation to generate pseudo-bilingual corpus.
- Merge bilingual corpus and pseudo-bilingual corpus
- Toolkit: openNMT.
- Corpus: Europarl.
  - Source language: English.
  - Target languages: Spanish.
- Some simulated experimental results:

Experiment	Corpus size Bilingual	Pseudo-parallel	Total	BLEU
Baseline	1,000K		1,000K	20.6
	2,000K		2,000K	22.3
+ back-translations	1,000K	1,000K	2,000K	21.2

Some experimental results [Sanz TFM 2021]

- Using back-translation to generate pseudo-bilingual corpus.
- Use only monolingual corpora.
- Toolkits: Undreamt [Artetxe 2018] & Monoses [Artetxe 2019]
- Corpus: News Crawl.
- Some results:

System	Model	Pair	BLEU
Undreamt	GRU	Fr-En	14.1
Undreamt	GRU	Ge-En	7.3

- Worse results with Transformer.

An example of no-supervised learning for machine translation

El guerrero número 13 (The 13th Warrior - John McTiernan). 1999.

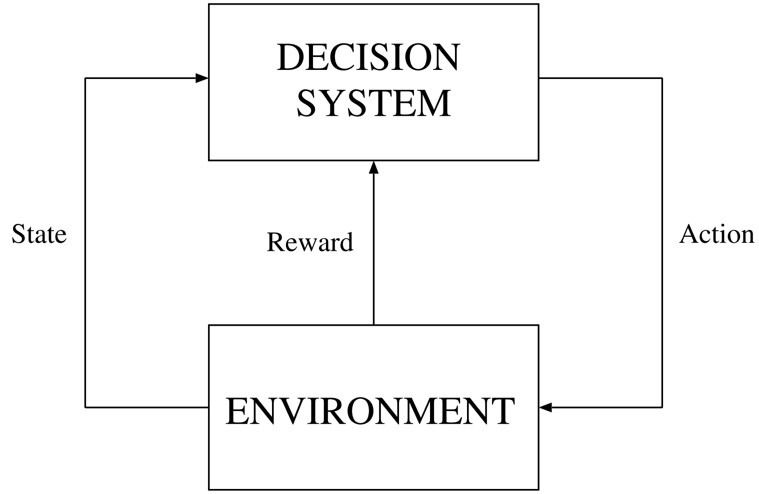
Index

Why reinforcement learning in NMT?

- Fine-tuning for optimizing metrics as BLEU, TER, ...
- A new framework for interactive machine translation

Markov decision process

- Set of states $\mathcal{Q}$.
- Set of actions $\mathcal{A}$.
- Transition probability *(Markovian)*: $\tau(q' \mid q, a),\ \ q, q' \in \mathcal{Q},\ \ a \in \mathcal{A}$.
- Reward function $r : \mathcal{Q} \times \mathcal{A} \times \mathcal{Q} \to \mathbb{R}$.



- An episode h is a sequence of steps (with“finite horizon” or “episodic”):

$$h = [q_1, a_1, q_2, a_2, \ldots, a_T, q_{T+1}]$$

with a gain or accumulated reward for the episode h: $G(h) = \sum_{t=1}^{T} \gamma^{t-1}\, r(q_t, a_t, q_{t+1})$

- A policy is an agent strategy to choose the actions of the successive steps: $\pi(A_t = a \mid Q_t = q)$ with probability: $p_\pi(h) \ = \ p_i(q_1) \prod_{t=1}^{T} \tau(q_{t+1} \mid q_t, a_t)\ \pi(a_t \mid q_t)$
- An optimal policy π^* is: $\pi^* \ = \ \text{argmax}_\pi\, \mathbb{E}_{p_\pi(h)}\big[G(h)\big]$

Policy

- Given a model (τ, r) estimated from a set of episodes, compute the optimal policy π^*:
  - Policy Iteration algorithms (based on states or states-action values)

- If the model (τ, r) is unknown, compute the optimal policy π^* directly from the episodes (model free):
  - Temporal Difference Learning (for computing state and state-action values)
  - Policy gradient $\pi(a \mid q; \theta)$: REINFORCE or ACTOR-CRITIC.
  - Least-Squares Policy Iteration.
  - Deep Q-network: a deep network for computing state-action value.

Reinforcement learning in NMT [Wu EMNLP 2018]

- Training NMT with Reinforcement Learning: directly optimizing the evaluation measure at training time.
  - State: (y_1^{i-1}, x_1^J)
  - Policy: $p(y_i \mid y_1^{i-1}, x_1^J)$
  - Action: choose the next translation word y_i
  - Reward: the BLEU at the end of translation $BLEU(y_1^I, \bar{y}_1^{\bar{I}})$ where $\bar{y}_1^{\bar{I}}$ is the reference translation.
    - Problem: reward at the end of translation.
    - Solution: $r_i(y_i, \bar{y}_1^I) = BLEU(y_1^i, \bar{y}_1^I) - BLEU(y_1^{i-1}, \bar{y}_1^I)$ (Reward shaping)
  - Combine MLE and RL objectives

- Training NMT with Reinforcement Learning and monolingual data

Reinforcement learning for INMT

[Lam EAMT 2018] [Lam MTSUMMIT 2019]

- The user can “DELETE”, “SUBSTITUTE” or “KEEP” a translated word:
  - State: (y_1^{i-1}, x_1^J)
  - Policy: $p(y_i \mid y_1^{i-1}, x_1^J)$
  - Action: choose the next translation word y_i
  - Reward: $r_i(y_i) = \begin{cases} 0.5 & \text{if SUBSTITUTE/KEEP} \\ -0.1 & \text{if DELETE} \end{cases}$

- Learning algorithm for the policy: ACTOR-CRITIC.

- The policy is updated from edits if the entropy (uncertainty) of the action is high enough.

Index

More ...

- Computational aspects: Efficiency, parallelism, multiple GPUs.
- Reduce the model size [Banik+ IEEE Access 2018].
- More explanatory models.
- Linguistic knowledge from continuous representations [Manning+ PNAS 2020].
- Data selection.
- More learning algorithms: incremental learning,
- Automatic/human evaluation.
- Confidence measures.
- Ensemble decoding.
- The use of specific glossaries (place holders)
- Machine translation for machines [Tebbifakhr+ EMNLP 2019].
- Knowledge distillation.
- Multimodality in interactive machine translation and post-editing.
- ...

A future challenge: Alien to English translator

Earth vs. the Flying Saucers
(La Tierra contra los Platillos Volantes - Fred F. Sears). 1956.

A future challenge: Martian to English translator

Mars Attaks! (Tim Burton). 1996.

Index

Bibliography (1)

- Mikel Artetxe, Gorka Labaka, Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. 2018.
- MÃ^{3}nica Castellanos. On The Use of Monolingual Corpus for Training Neural Machine Translation Systems. TFM. 2020.
- A. Conneau, G. Lample, L. Denoyer, MA. Ranzato, H. Jégou. WORD Translation without Parallel Data. Proceedings of the ICLR 2018.
- Jorge Cuevas. Traducción multilingüe neuronal. TFM. 2020.
- Raj Dabre, Chenhui Chu, Anoop Kunchukuttan. A Comprehensive Survey of Multilingual Neural Machine Translation. arXiv:2001.01115v2. 2020.
- Ishat Gibadullin, Aidar Valeev, Albina Khusainova, Adil Khan. A Survey of Methods to Leverage Monolingual Data in Low-resource Neural Machine Translation. arXiv:1910.00373v1, 2019.
- Tsz Kin Lam, Julia Kreutzer, Stefan Riezler. A Reinforcement Learning Approach to Interactive-Predictive Neural Machine Translation. Proceedings of the 21st Annual Conference of the European Association for Machine Translation, 2018.
- Tsz Kin Lam, Shigehiko Schamoni, Stefan Riezler. Interactive-Predictive Neural Machine Translation through Reinforcement and Imitation. Proceedings of MT Summit XVII, 2019.

Bibliography (2)

- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut. ALBERT: A Lite BERT for self-supervised learning o flanguage representations. arXiv:1909.11942v6, 2019.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv:1910.13461v1. 2019.
- Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, Guodong Zhou. Modeling Source Syntax for Neural Machine Translation. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017.
- Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processingand the 9th International Joint Conference on Natural Language Processing, 2019.
- NLLB Team. No Language Left Behind: Scaling Human-Centered Machine Translation. arXiv. 2022.
- Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. A Survey on Document-level Machine Translation: Methods and Evaluation. arXiv:1912.08494v1. 2019.

Bibliography (3)

- Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai & Xuanjing Huang. Pre-trained Models for Natural Language Processing: A Survey. arXiv, 2003.08271v3. 2020.
- Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, Tie-Yan Liu. A Study of Reinforcement Learning for Neural Machine Translation. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018.
- Baosong Yang, Derek F. Wong, Tong Xiao, Lidia S. Chao, Jingbo Zhu. Towards Bidirectional Hierarchical Representations for Attention-Based Neural Machine Translation. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.
- Meishan Zhang, Zhenghua Li, Guohong Fu, Min Zhang. Syntax-Enhanced Neural Machine Translation with Syntax-Aware Word Representations. Proceedings of the North-American Association of Computational Linguistics, 2019.

Bibliography (More ...)

- D. Banik, A. Ekbal and P. Bhattacharyya, ”Machine Learning Based Optimized Pruning Approach for Decoding in Statistical Machine Translation,” in IEEE Access, (7):1736-1751, 2019.
- Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, Omer Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. Proceedings of the National Academy of Sciences (PNAS). 2020.
- Amirhossein Tebbifakhr, Luisa Bentivogli, Matteo Negri, Marco Turchi. Machine Translation for Machines: the Sentiment Classification Use Case. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)- 2019.
- Barret Zoph and Kevin Knight. Multi-Source Neural Translation. arXiv 1601.00710v1. 2016

Pre-trained models: bibliography (I)

- Ethan A. Chi, John Hewitt, Christopher D. Manning. Finding Universal Grammatical Relations in Multilingual BERT. ACL 2020.
- Stéphane Clinchant, Kweon Woo Jung, Vassilina Nikoulina. On the use of BERT for Neural Machine Translation. WNGT. 2019.
- Alexis Conneau, Guillaume Lample. Cross-lingual Language Model Pretraining. NIPS. 2019.
- Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT. 2019.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma. Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. ICLR. 2020.
- Teven Le Scao et al. What Language Model to Train if You Have One Million GPU Hours?. arXiv 2022.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. ACL. 2020.
- Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, Lei Li. Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information. arXiv. 2021.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv. 2019.
- Lewkowycz et al. Solving Quantitative Reasoning Problems with Language Models. arXiv 2022.

Pre-trained models: bibliography (II)

- Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, Guoping Hu. CharBERT: Character-aware Pre-trained Language Model. COLING. 2020.
- Telmo Pires, Eva Schlinger, Dan Garrette. How multilingual is Multilingual BERT?. ACL. 2019.
- Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, Xuanjing Huang. Pre-trained Models for Natural Language Processing: A Survey. Science China Technological Sciences. 2020.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. Report OpenAI. 2018.
- Nils Reimers, Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP. 2019.
- Sebastian Ruder. Recent Advances in Language Model Fine-tuning. `http://ruder.io/recent-advances-lm-fine-tuning`. 2021.
- Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. NIPS. 2019.
- Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, Bryan Catanzaro. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. arXiv 2019.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser. Attention Is All You Need. NIPS. 2017.
- Alex Wang, Kyunghyun Cho. BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model. NeuralGen. 2019.
- Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, Lei Li. Towards Making the Most of BERT in Neural Machine Translatio. AAAI. 2020.

Pre-trained models: bibliography (III)

- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le. NIPS. 2019.
- Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, Tie-Yan Liu. Incorporating BERT into Neural Machine Translation. ICLR. 2020.