Going Cross-Lingual: Computational Methods for Multilingual Text Analysis

Hauke Licht and Fabienne Lind

Gesis Training Course

Cologne

6-8.12.2023

Contents (smaller changes possible)

Day	Morning Session	Afternoon Session
1	Intro & Overview about applications, problems, and solutions, validation	Corpus and Input Selection
2	Obtaining Measures: machine translation and supervised classifications	Multilingual supervised classification and measurement validation
3	Multilingual topic modeling with BERTopic and input on tokenization and pre-processing	Individual consultations on your projects

Workshop repository

https://github.com/fabiennelind/Going-Cross-Lingual_Course

Today

09:30 - 11:00	Machine translation
11:00 - 11:15	*Coffee break*
11:15 - 12:30	Multilingual transformers for supervised text classification (input and guided exercise)
12:30 - 13:30	*Lunch break*
13:30 - 15:00	Valid outputs in multilingual & multi-context scenarios (input and guided exercise)
15:00 - 15:15	*Coffee break*
15:15 - 16:30	Individual practice with with own/prepared data

Machine translation

transferring multilingual corpora to a “common” denominator language

History

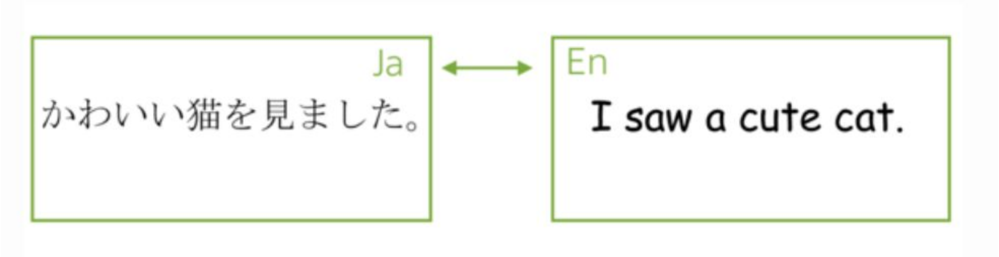
until recently, **machine translation** (MT) has been the default in multilingual quantitative text analysis

- several methods papers evaluate its reliability and validity for bag-of-words analysis (e.g., Lucas et al., 2015; de Vries et al., 2018, Mate et al., 2023) and embedding-based analysis (e.g., Mate et al., 2023)
- lots of applications in substantive comparative research

How it works (note: all illustrations by Lena voita)

MT is a **sequence-to-sequence** NLP task: we want to transfer

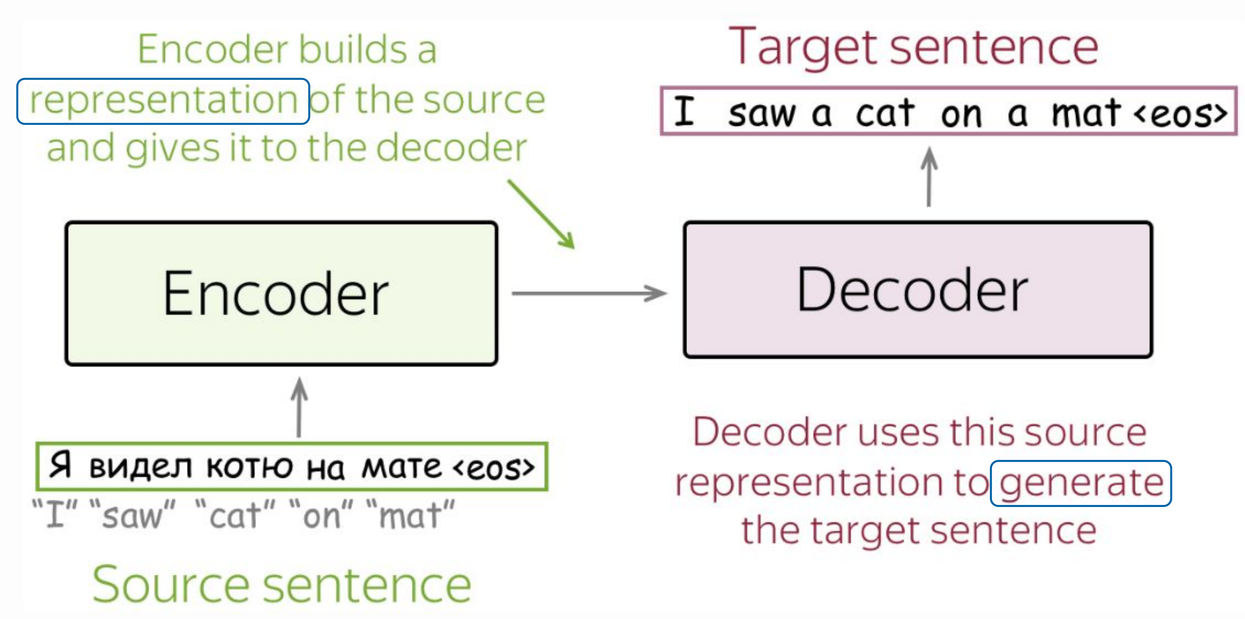
- a sequence of words in one language (the source)
  into
- a sequence of words in another language (the target)



How it works

The state-of-the-art is training a **decoder-encoder** neural network for this

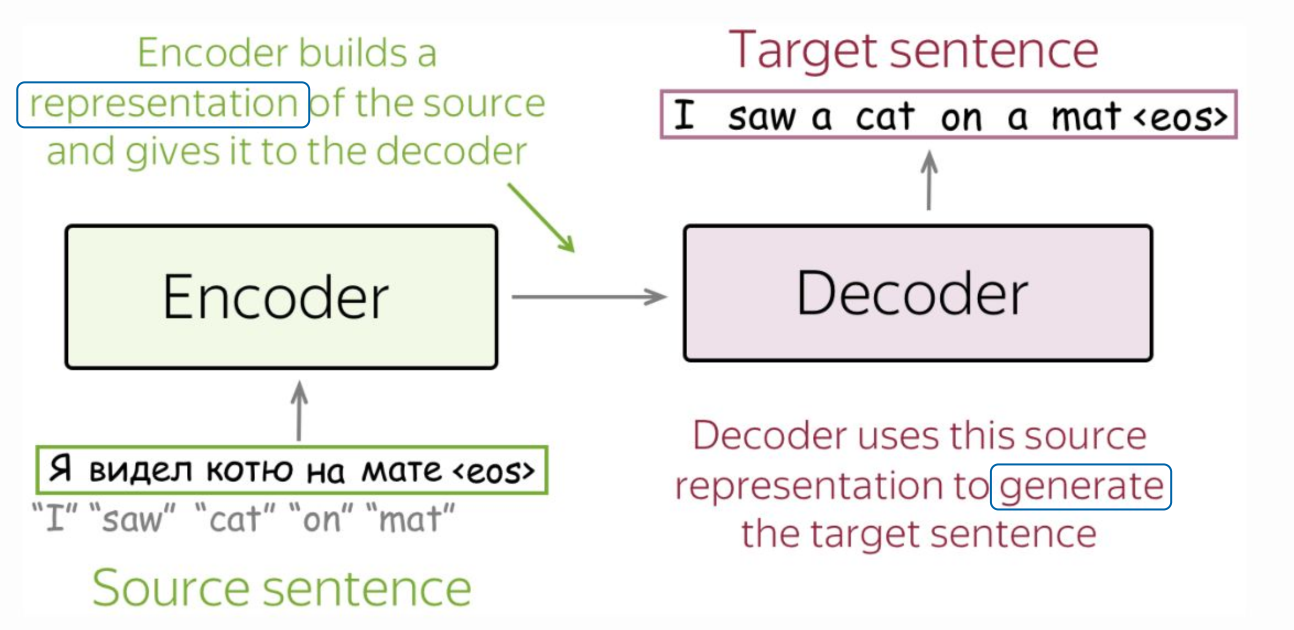
- using many sentence pairs
- covering many languages



How it works

The **encoder** is an embedding model

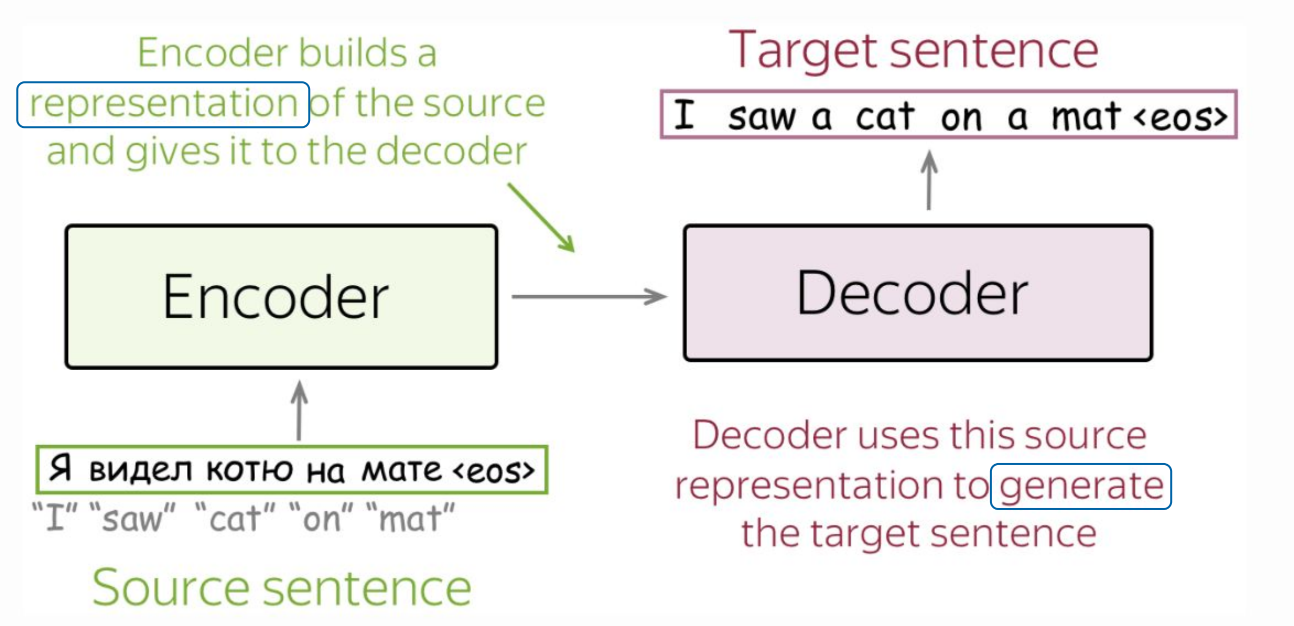
- it inputs the source sentence
- it represents it with an embedding (numeric vector)



How it works

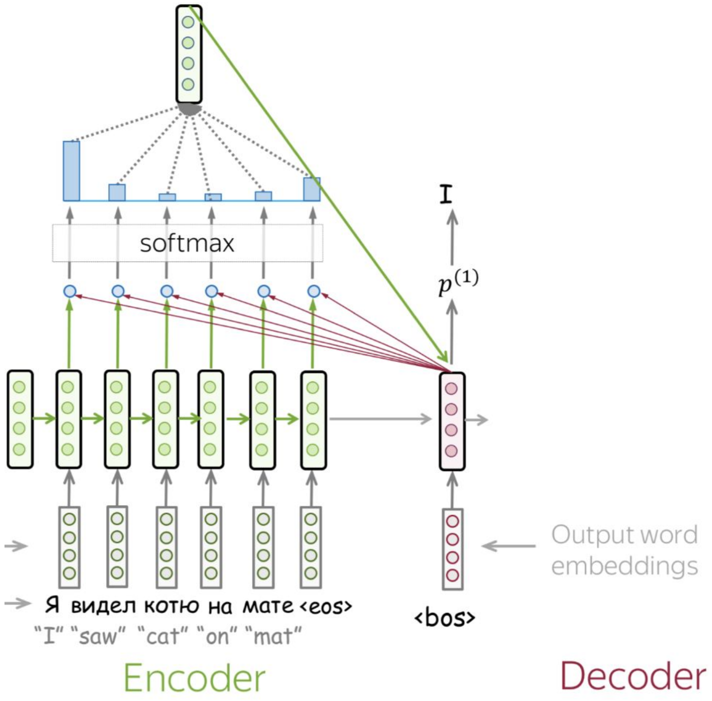
The **decoder** is a generative model

- it generates the output sentence *conditional* on the input embedding
- with the “correct” target sentence as a cue



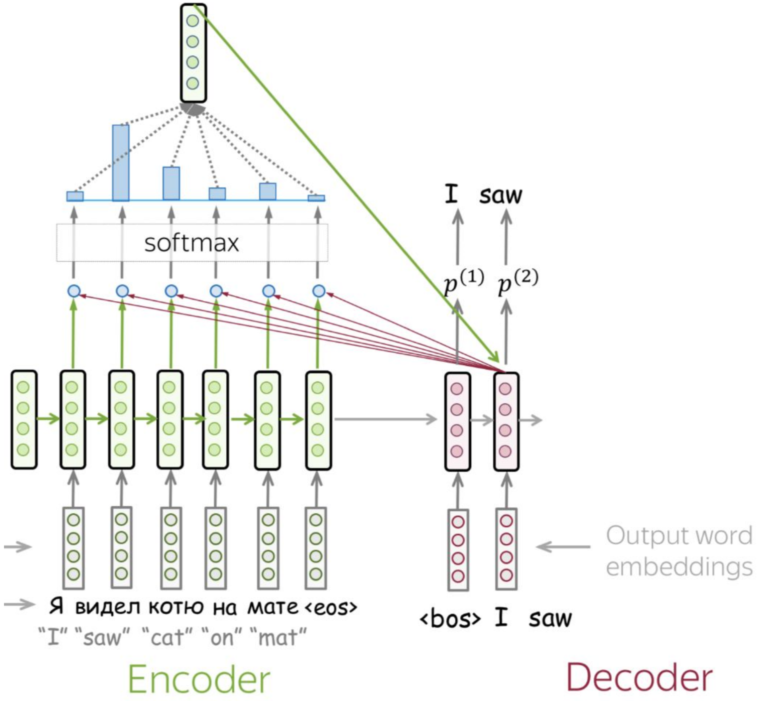
How it works

Text generation is recursive



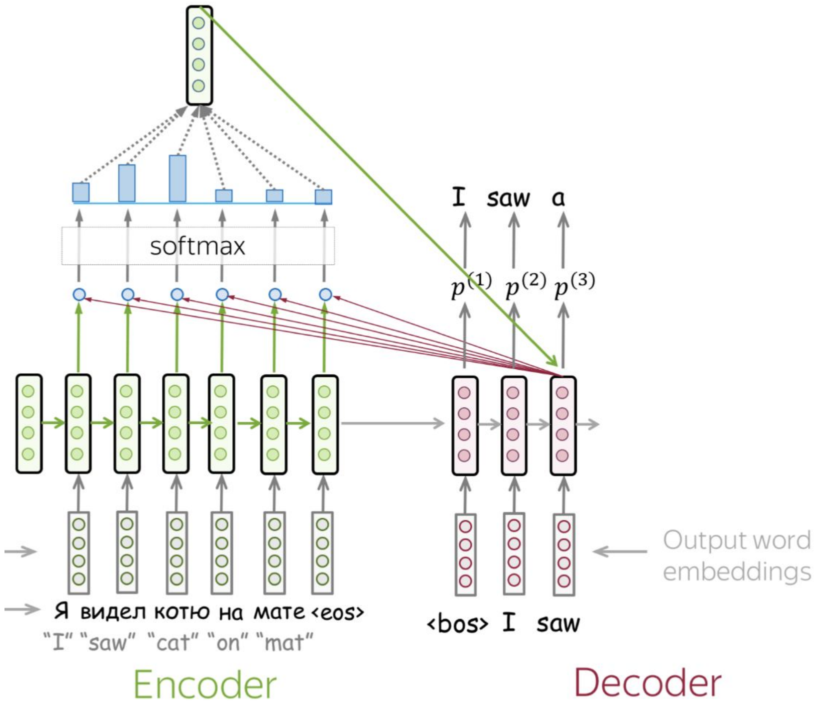
How it works

Text generation is recursive



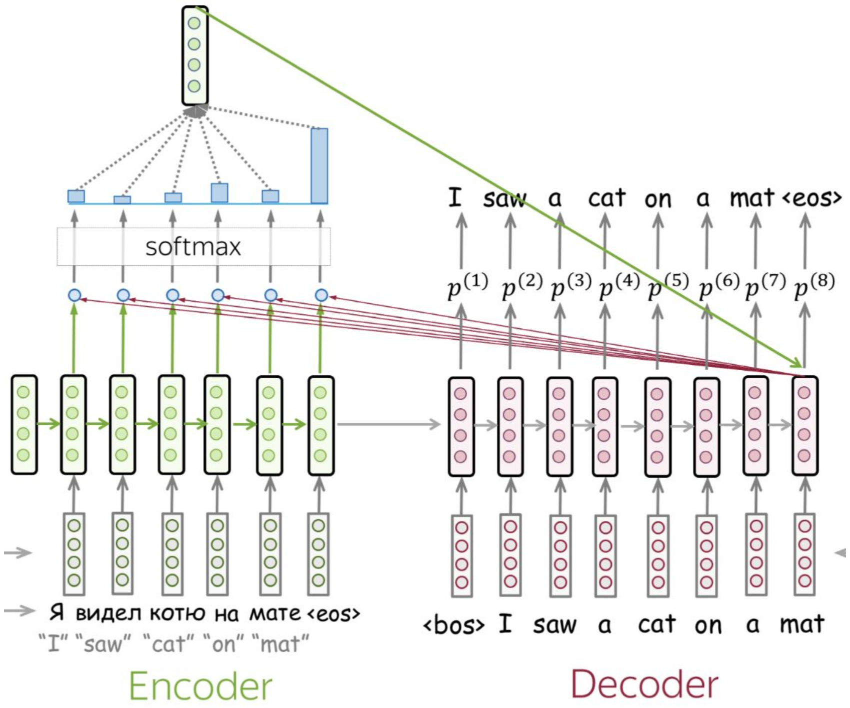
How it works

Text generation is recursive



How it works

At training time, text is generated recursively



How it works

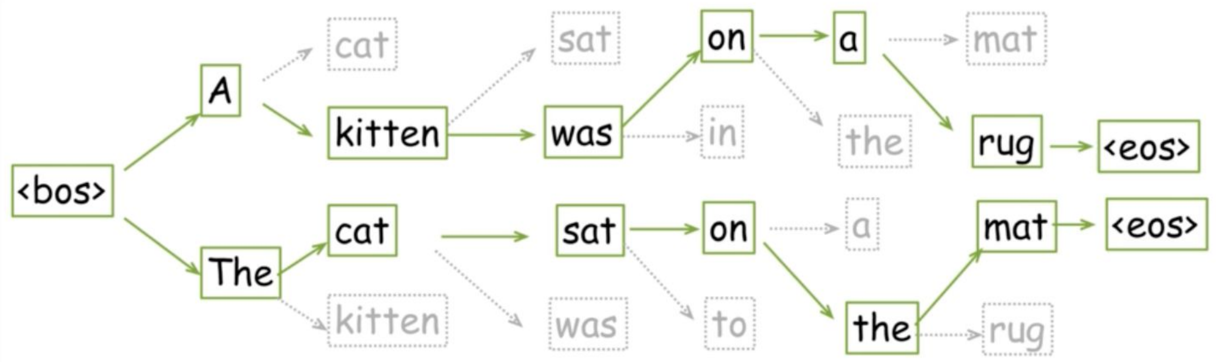
At inference time (i.e., prediction), “optimal” translation is found via *beam search*

tokens predicted at previous decoding steps

$$y' = \arg\max_{y} p(y|x) = \arg\max_{y} \prod_{t=1}^{n} p(y_t|y_{<t}, x)$$

encoder’s output embedding of input sentence

token predicted at current decoding step



Does it work for applied research?

Several pol/comm sci papers have evaluated NMT for bag-of-words analysis

Reference	Task	Domain	Translation service	Source language(s)	Target lang.
Lucas et al. (2015)	Topic modeling (STM)	Citizen-produced social media	Google Translate	Arabic, Chinese	English
Vries et al. (2018)	Topic modeling (LDA)	Parliamentary speech	Expert translations, Google Translate	Danish, French, German, Spanish, Polish	English
Reber (2019)	Topic modeling (LDA)	Web pages of (I)NGOs	Google Translate, DeepL	German	English
Windsor et al. (2019)	Dictionary analysis (LWIC)	UN plenary speeches	Google Translate	English, French, German, Russian, Chinese, Arabic	English
Düpont and Rachuj (2021)	Textual similarity	Party manifestos	Google Translate	12 languages [a]	English
Courtney et al. (2020)	Supervised classification	News article paragraphs	Google Translate	German, Spanish	English
Lind et al. (2021)	Supervised classification	News articles	Google Translate	German, Hungarian, Polish, Romanian, Spanish, Swedish	English
Licht (2023)	Supervised classification	Party manifestos	M2M (Fan et al. 2021)	12 languages [b]	English

[a] Catalan, Danish, Dutch, Finnish, French, Galician, German, Italian, Norwegian, Portuguese, Spanish, and Swedish
[b] same as Düpont and Rachuj (2021) by pooling their and data by Lehmann and Zobel (2018)

Does it work for applied research?

topic modeling:

- on average, document-topic and topic-word representations are very similar when comparing LDA topic models fitted to human- and machine-translated (*Google*) topics, respectively (de Vries *et al.*, 2018)
- Reber (2019) also evaluate DeepL and Google and finds similarly encouraging results

Does it work for applied research?

dictionary analysis:

- english dictionary applied to machine-translated corpus gives similar measurements as if applied to human-translated documents (Windsor *et al.*, 2019)
- but machine-translating of keywords is not a good idea (experts should check them; see Lind *et al.*, 2019; Proksch *et al.*, 2019)

Does it work for applied research?

supervised classification

- language-specific classifiers trained on machine-translated texts (Google) perform as well as classifiers trained on texts in their source languages (holding dataset constant; Courtney *et al.*, 2020)
- Transformers fine-tuned on machine-translated labeled texts perform as well as multilingual Transformers (Mate *et al.*, 2023, Table 3) (but only evaluated for Hungarian)

How to

two options

- commercial services (Google, DeepL, AWS, Microsoft, etc.)
- “free” open-source NMT models:
  - Helsinki NLP’s OPUS-MT
  - Facebook Research’s M2M
  - some others, see here for example

How to

two options

- commercial services (Google, DeepL, AWS, Microsoft, etc.)
- “free” open-source NMT models:

Discuss: what are these options pros and cons?

Two options – pros and cons

Commercial services

pros:

- high-quality translations
- often many translation directions (especially Google)
- usability (API, interface available)
- language coverage

cons:

- costs (you pay per character, 1 million characters ~= 20 US Dollars)
- limit replicability
- Replicability (there might be an update to the model)
- Problematic for sensitivity data

Two options – pros and cons

Open-source models

pros:

- freely available for download and use ⇒ replicable
- transparent: we know what training corpora have been used
- usually good-enough translation quality

cons:

- limited translation directions (problem with "low-resource" languages)
- you need to know how to code (but that's why you're here ;)
- need access to a GPU

So, let’s code

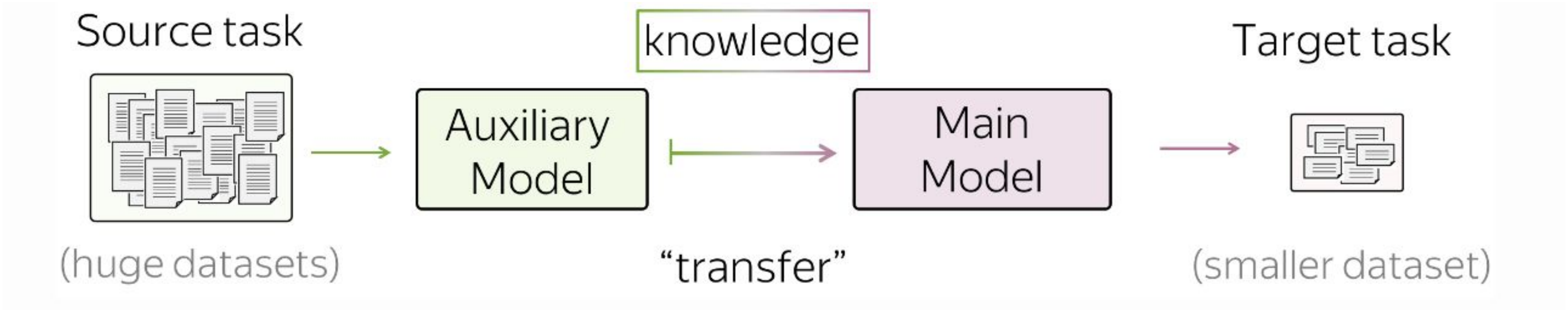
notebook ‘translation_basics.ipynb’ in code/ on github

multilingual Transformers and embeddings

transferring multilingual corpora into a common embedding space

The idea: transfer learning (note: all illustrations by Lena voita)

- Training task-specific models is resource-intensive =(
- Train general-purpose models on large text corpora through self-supervision (next slide) ⇒ “fine-tune” (i.e. adapt) for various tasks

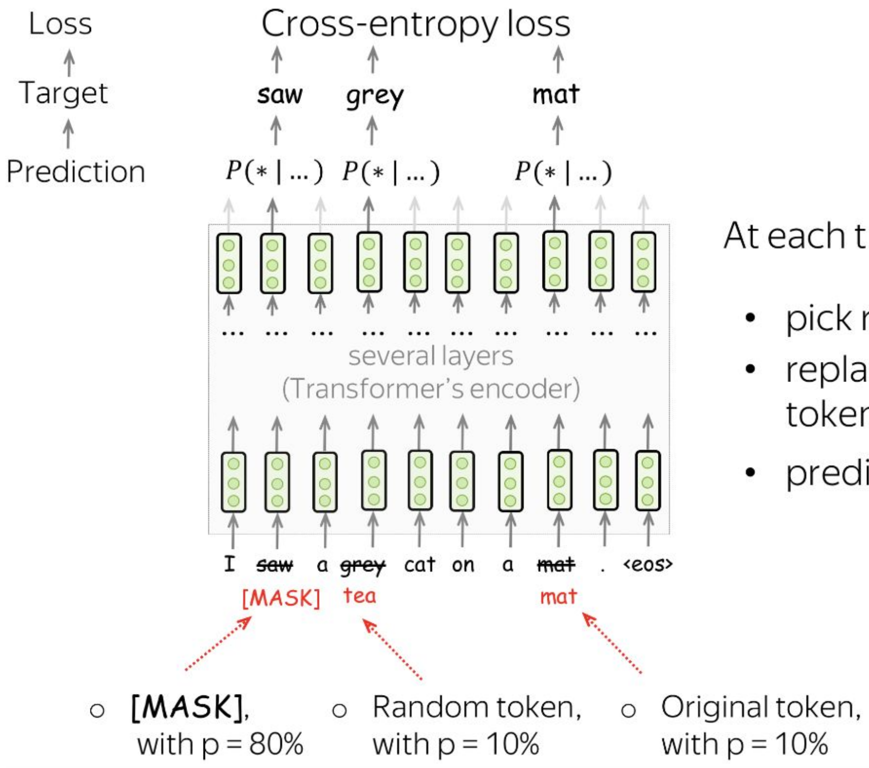


Transformers

Neural networks trained for language modeling (LM) on large (multilingual) text corpora through self-supervision

BERT: masked LM

GPT: “causal” LM



At each training step:

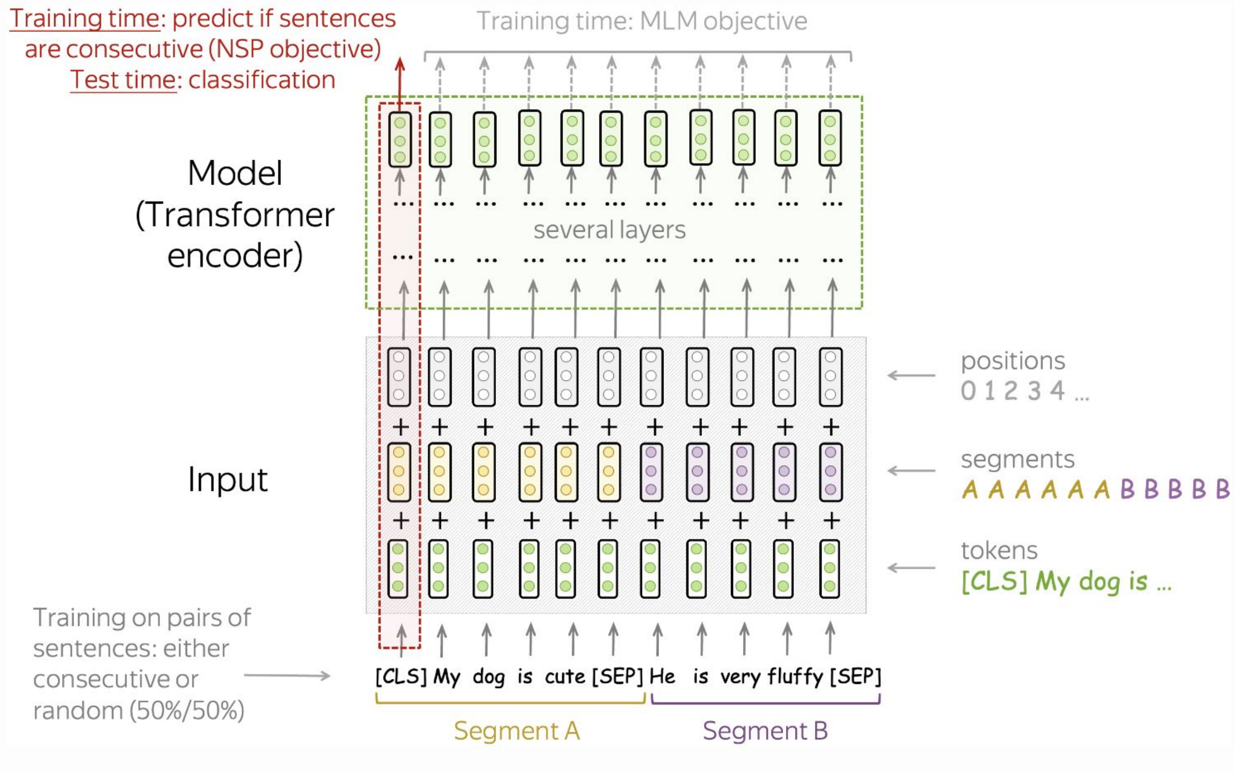
- pick randomly 15% of tokens
- replace each of the chosen tokens with something
- predict original chosen tokens

Transformers

BERT performs trains for two tasks

1. masked LM
2. next sentence prediction (NSP)

Note: other models omit NSP

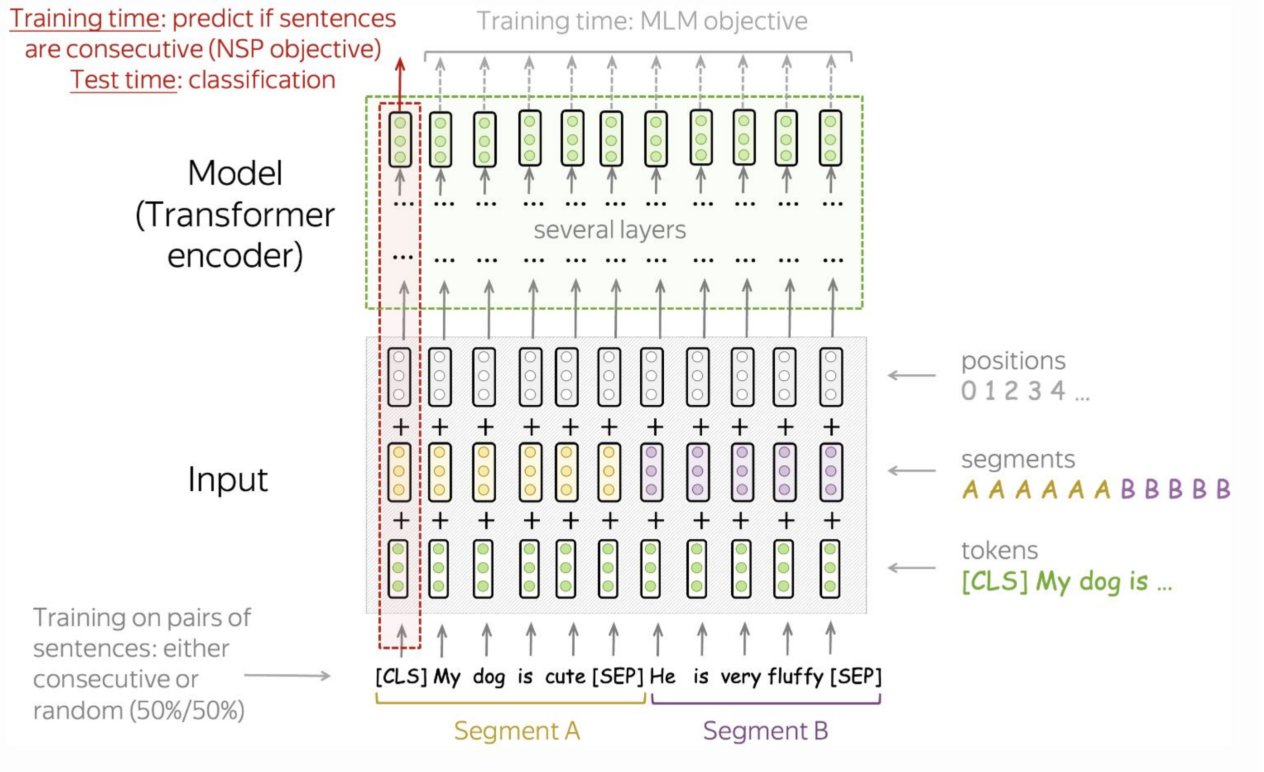


Transformers

each layer of BERT encoder has three components

1. token embeddings
2. segment embeddings
3. position embeddings

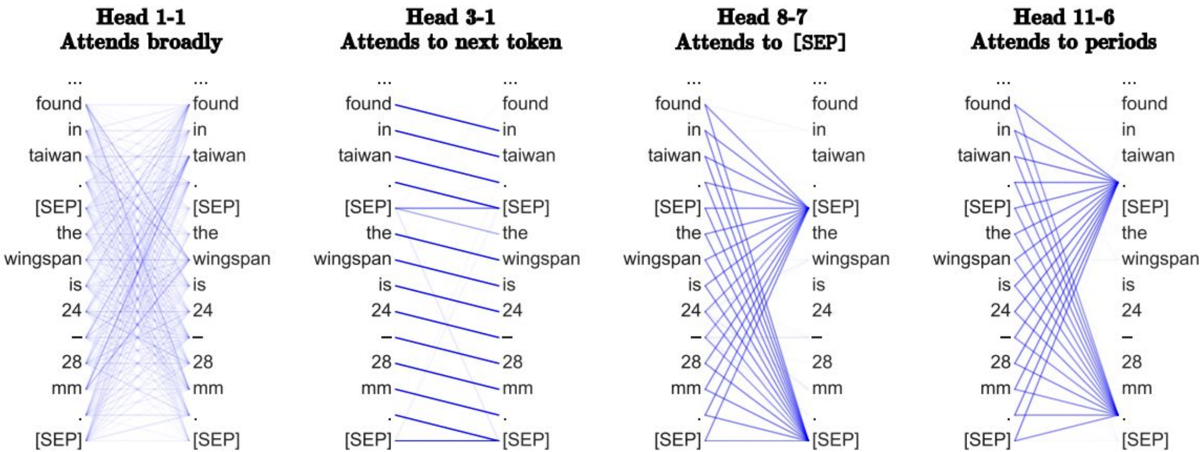
Note: other models dispens with NSP



Transformers

Attention is all you need!

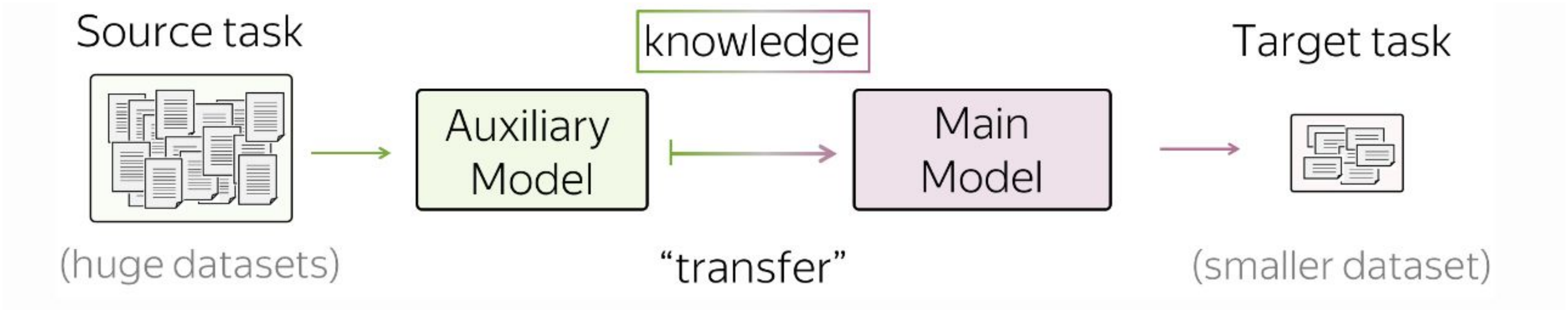
To create token embeddings that reflect a word's context, the attention mechanism is applied



explore it with BERTviz

Illustration from paper "What Does BERT Look At?"

Fine-tuning



1. take pre-trained Transformer
2. add classification layer on top of output embeddings
3. take labeled text dataset to train Transformer+classifier through supervised learning ⇒ fine-tuning

Fine-tuning ingredients

- a *label* text dataset:
  - a corpus of texts (e.g., sentences; could be machine-translated)
  - in which *each* document/text has been assigned to
  - a *single* label from
  - a *fixed set* of label classes (e.g., 'positive', 'neutral', 'negative')
- a pre-trained (multilingual) Transformer model you can fine-tune for sequence classification
- the pre-trained tokenizer (comes with the model!)

Political Sci applications

A very selective list of examples

- identify populist campaign messages (Bonkowski *et al.*, 2022)
- measure expressed emotions in parl. speech (Widmann & Wich, 2023)
- classify stances on political issues (Bestvater & Monroe, 2023)
- measure parties' anti-elite strategies (Licht *et al.*, 2023, multilingual)
- transfer classifications across languages (Ho & Chan, 2023)
- categorize the content of social (and other) media (Kroon et al., 2023)

Let’s code

notebook ‘transformer_finetuning.ipynb’ in code/ on github

What are open questions?

Thank you very much