Machine Translation

Guillaume Wisniewski

(with the help of Rachel Bawden)

guillaume.wisniewski@u-paris.fr

UPCité Master in Computational Linguistics

November, 2025

Part I

An historical perspective

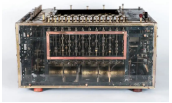
What a long strange trip it's been! (1)

1934 2 patents for "translation machines": Georges Artsrouni Peter Petrovitch Trojanskii

1940 MT as one of the 1st applications of informatics. Pioneers: Weaver, Booth & Richens

1949 Warren's Weaver's memorandum Translation

outlining 4 propositions, incl. (i) need for context to resolve ambiguity, (ii) formal methods can be used, (iii) use of cryptography and (iv) meaning independent of language



Electromechanical translation machine known as the "mechanical brain", Musée des arts et métiers – ©CNAM / Photo Hélène Mauri.

What a long strange trip it's been! (2)

1954 IBM-Georgetown experiment: 60 Russian sentences translated into English from (250 words and 6 syntactic rules)

"five, perhaps three, years hence, interlingual meaning conversion by electronic process in important functional areas of several languages may well be an accomplished fact." (IBM press release)

1960 Bar Hillel's report on MT

"The unreasonableness of aiming at fully-automatic high-quality translation is stressed"

1966 ALPAC report

A russian sentence to be translated by the IBM-Georgetown experiment source

The 4 Machine Translation Paradigms

Rule-Based approaches rules to describe how to "transform" a sentence in a source language into a sentence in target languages

↪ symbolic IA

↪ main approach until 1990 / 2000

The 4 Machine Translation Paradigms

Statistical MT word- and phrase-based systems

↪ rely on counts of words & (vaguely) on machine learning

↪ both monolingual (language model) & bilingual (translation model)

↪ main approach between 2000–2015

The 4 Machine Translation Paradigms

Neural Machine Translation

↪ the deep learning revolution in NLP

The 4 Machine Translation Paradigms

Large Language Models

↪ rely on transformers & multilingual language models

↪ the future...

First Things First: Parallel Corpora

- Parallel data: (generally) sentences in one language aligned with translations
- Generally, translations produced by humans

↪ Translations exhibit different properties from original texts ("translationese")

↪ The quality of translations is variable

- Training data and evaluation data

Some Corpora

- Institutional Corpora
  - ↪ Europarl (22 languages)
  - ↪ MultiUN (7 languages)
- Subtitles OpenSubtitles (1,782 languages pairs)
- Mined Corpora
  - ↪ ParaCrawl (42 language pairs)
  - ↪ CCMatrix (1,200 language pairs)
  - ↪ NLLB (1,600 language pairs)

Parallel Corpora

Opus

- `https://opus.nlpl.eu/`
- 1,210 corpora
- 45,945,946,108 total sentence pairs
- 744 languages available

OPUS
OPEN PARALLEL CORPORA

Creating Parallel Corpora

- creating parallel data is hard & tedious
- many heuristics and know-how
- pipeline sketch
  - identify documents that are translations of each other
  - find sentences that are translation of each other
  - ↪ lot of filtering (e.g.: keep only 1:1 translation, filter out bad translation, ...)
- tradeoff between precision and recall

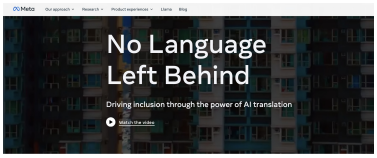
↪ for the moment: improve precision as much as possible

A clear message

- mining for parallel resources is still an open issue
- mining for supervised data is still the best way to improve a ML system

↪ look at `Whisper`

Example: The No Language Left Behind project (I)



ABOUT NO LANGUAGE LEFT BEHIND

No Language Left Behind (NLLB) is a first-of-its-kind, AI breakthrough project that open-sources models capable of delivering evaluated, high-quality translations directly between 200 languages—including low-resource languages like Asturian, Luganda, Urdu and more. It aims to give people the opportunity to access and share web content in their native language, and communicate with anyone, anywhere, regardless of their language preferences.

- `https://ai.meta.com/research/no-language-left-behind/`

Example: The No Language Left Behind project (I)

Contents

1 Introduction 4
2 Human-Centered Low-Resource Language Translation 7
2.1 Exploratory Interview Study Research Design . . . 8
2.2 No Language Left Behind: Guiding Principles . . . 11
3 Languages 12
4 Creating Professionally Translated Datasets: FLORES-200 and NLLB-Seed 18
4.1 FLORES-200 . . . 18
4.2 NLLB Seed Dataset . . . 23
4.3 NLLB Multi-Domain Dataset . . . 24
4.4 Conclusion . . . 25
5 Automatically Creating Translation Training Data for Hundreds of Languages 26
5.1 Language Identification . . . 26
5.2 Gathering and Cleaning Monolingual Data at Scale . . . 34
5.3 Mining Bitexts for Low-Resource Languages . . . 37
5.4 Conclusion . . . 46
6 Modeling 47
6.1 Preliminaries . . . 48
6.2 Conditional Compute for Massively Multilingual Machine Translation . . . 50
6.3 Self-Supervision Strategies on Large-scale Monolingual Corpora . . . 59
6.4 Data Augmentation . . . 64
6.5 Bootstrapping models with NLLB-Seed . . . 67
6.6 Human Evaluation . . . 70
6.7 Conclusion . . . 71
7 Evaluation 71
7.1 Automatic Evaluation . . . 72
7.2 Human Evaluation . . . 73
7.3 Toxicity . . . 79
7.4 Conclusion . . . 89
8 Bringing it All Together 90
8.1 Preparing the Data . . . 91
8.2 Preparing the Model . . . 97
8.3 Results on FLORES-200 . . . 102
8.4 Out-of-domain Generalization: Performance on non-FLORES-200 Domains . . . 108
8.5 Analysis of NLLB-200 . . . 114
8.6 Making Large Models More Accessible through Distillation . . . 119

- a 190-pages paper ! (NLLB Team et al. 2022)
- most of the paper/work = collecting data
  1. seed data = parallel data
  2. data augmentation method

2nd Example: Glot500

- (Imani et al. 2023)

↪ largest LLM (in terms of number of languages)

↪ main question: how to find & filter out data for enough languages

Part II

Statistical MT

Statistical MT

- Main idea: find the translation candidate $\hat{t}$ that maximizes the probability of being the translation of a source sentence s

$$\hat{t} = \arg\max_{t \in \mathcal{T}} P(t|s) \tag{1}$$

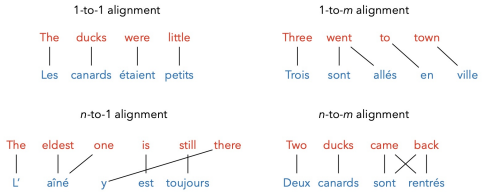
$\Rightarrow$ noisy channel model

- with Bayes rule:

$$\begin{aligned}\hat{t} &= \arg\max_{t \in \mathcal{T}} P(t|s) \\ &= \arg\max_{t \in \mathcal{T}} \frac{P(s|t) \times P(t)}{P(s)} \\ &= \arg\max_{t \in \mathcal{T}} \underbrace{P(s|t)}_{\text{translation model}} \times \underbrace{P(t)}_{\text{language model}}\end{aligned}$$

Word-Based Models

- Translation probabilities at the word-level: decompose $P(s|t)$ into lexical translation probabilities (i.e. probability of translating/generating a single word)
- Requires the notion of word alignments
- IBM models 1-6 of increasing complexity (Brown et al. 1993) (first success of statistical model in NLP)



Formalization of the word alignment task

Given a source sentence $s = s_1, s_2, ..., s_{|s|}$ and a target sentence $t = t_1, t_2, ..., t_{|t|}$, find which target word is translated by which source word.

↪ the choice of the source and the target sentence is arbitrary

↪ we are only interested in the alignment

Learning Alignments

What we have...

- parallel corpora: alignment at the sentence level
- but almost no alignment at the word level
  - not always possible
  - tedious

⇒ unsupervised learning

The alignment function (1)

Source: das_1 $Haus_2$ ist_3 $klein_4$

Target: the_1 $house_2$ is_3 $small_4$

↪ formally, an alignment is a function from $[\![1, |t|]\!]$ to $[\![1, |s|]\!]$:

$$a = \{1 \to 1; 2 \to 2; 3 \to 3; 4 \to 4\}$$

↪ Warning: a is not symmetrical!

The alignment function (2)

$klein_1$ ist_2 das_3 $Haus_4$

the_1 $house_2$ is_3 $small_4$

$$a = \{1 \to 3; 2 \to 4; 3 \to 2; 4 \to 1\}$$

The alignment function (3)

das_1 $Haus_2$ ist_3 $klitzeklein_4$

the_1 $house_2$ is_3 $very_4$ $small_5$

$$a = \{1 \to 1; 2 \to 2; 3 \to 3; 4 \to 4; 5 \to 4\}$$

The alignment function (4)

the_1 $poor_2$ $don't_3$ $have_4$ any_5 $money_6$

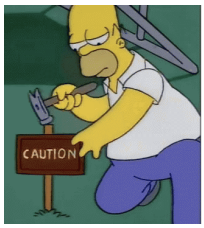
les_1 $pauvres_2$ $sont_3$ $démunis_4$

Can no longer be represented!

An important detail

The alignment task

For every target word find the source word (including NULL) that has 'generated' it



An important detail

The alignment task

For every target word find the source word (including NULL) that has 'generated' it

- ↪ the model is inherently asymmetric... you cannot invert source and target
- ↪ ... but alignments are symmetric!
- ↪ symetrize alignment in a second step
- ↪ two step process ⇒ modeling easier

IBM 1 Model

Assumptions

- model the probability of generating the target sentence and the alignment knowing the source sentence
- using only the probability to translate a given source word by a target word: $\theta(t|s)$
- assuming that each target word is generated independently

Model

$$p(\mathrm{t}, a|\mathrm{s}) \propto \prod_{j=1}^{|\mathrm{t}|} \theta(t_j|s_{a(j)})$$

Example

s = la		s = maison		s = est		s =petite	
t	$\theta(t\|s)$	t	$\theta(t\|s)$	t	$\theta(t\|s)$	t	$\theta(t\|s)$
the	0.7	house	0.8	is	0.8	small	0.4
that	0.15	building	0.16	's	0.16	little	0.4
which	0.075	home	0.02	exists	0.02	short	0.1
who	0.05	household	0.015	has	0.015	minor	0.06
this	0.025	shell	0.005	are	0.005	petty	0.04

↪ Probability of aligning `la maison est petite` and `the house is small` monotonically:

$$\begin{aligned} p(\mathrm{t}, a|\mathrm{s}) &\propto \theta(\text{the}|\text{la}) \times \theta(\text{house}|\text{maison}) \times \theta(\text{is}|\text{est}) \times \theta(\text{small}|\text{petite}) \\ &\propto 0.7 \times 0.8 \times 0.8 \times 0.4 \\ &\propto 0.179 \end{aligned}$$

Aligning two sentences (1)

Aim

Given two sentences, s and t and an alignment table Θ, find the word alignment between the two sentences

Aligning two sentences (1)

Aim

Given two sentences, s and t and an alignment table Θ, find the word alignment between the two sentences

Maximum a posteriori:

$$\mathrm{a}^* = \arg\max_{\mathrm{a}} p(\mathrm{a}|\mathrm{s}, \mathrm{t}; \Theta) \tag{2}$$

$$= \arg\max_{\mathrm{a}} \prod_{j=1}^{|\mathrm{t}|} \theta(t_j|s_{a(j)}) \tag{3}$$

- ↪ max of a product of independent terms
- ↪ each term $\in [0, 1]$

⇒ max when each term is maximal

Aligning two sentences (2)

s = la		s = maison		s = est		s =petite	
t	$\theta(t\|s)$	t	$\theta(t\|s)$	t	$\theta(t\|s)$	t	$\theta(t\|s)$
the	0.7	house	0.8	is	0.8	small	0.4
that	0.15	building	0.16	's	0.16	little	0.4
which	0.075	home	0.02	exists	0.02	short	0.1
who	0.05	household	0.015	has	0.015	minor	0.06
this	0.025	shell	0.005	are	0.005	petty	0.04

alignment between `la maison est petite` and `the house is small ?`

Translation Probability

- $\theta(t|s)$:
  - ↪ probability of translating s by t
  - ↪ for each source word s: dictionary mapping target words to probability
- follows the basic rules of probabilities:
  - ↪ $\forall s, t \quad \theta(t|s) \in [0, 1]$
  - ↪ $\forall s \quad \sum_t \theta(t|s) = 1$
- $\theta(t|s)$ = parameters of the model
  - ↪ have to be estimated from the data

Learning Translation Probability

1. if we know the alignments (i.e. supervised learning)
   - ↪ easy to estimate the translation probabilities

Learning Translation Probability

1. if we know the alignments (i.e. supervised learning)
   - ↪ easy to estimate the translation probabilities
2. if we know the translation probabilities
   - ↪ easy to find the alignments

Learning Translation Probability

1. if we know the alignments (i.e. supervised learning)
   - ↪ easy to estimate the translation probabilities
2. if we know the translation probabilities
   - ↪ easy to find the alignments

but we know neither of them!

- ↪ incomplete data
- ↪ latent/hidden variables

Learning Translation Probability

1. if we know the alignments (i.e. supervised learning)
   - ↪ easy to estimate the translation probabilities
2. if we know the translation probabilities
   - ↪ easy to find the alignments

but we know neither of them!

- ↪ incomplete data
- ↪ latent/hidden variables

⇒ EM algorithm

EM algorithm: intuition

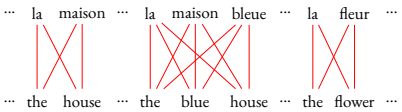
Given a corpus:

… la maison … la maison bleue … la fleur …

… the house … the blue house … the flower …

EM algorithm: intuition

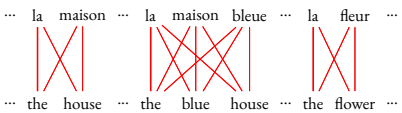
Given a corpus:



1. assume all alignments are equally likely

EM algorithm: intuition

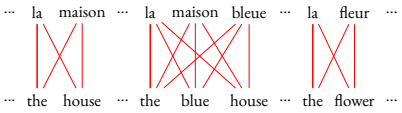
Given a corpus:



1. assume all alignments are equally likely
2. estimate the translation probabilities and predict alignments accordingly

EM algorithm: intuition

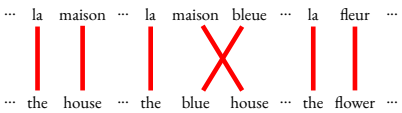
Given a corpus:



1. assume all alignments are equally likely
2. estimate the translation probabilities and predict alignments accordingly
   - ↪ model learns that `la` is often aligned with `the`
   - ↪ the alignment between `la` and `the` is reinforced
   - ↪ the alignments between `la` and other words become weaker

EM algorithm: intuition

Given a corpus:



1. assume all alignments are equally likely
2. estimate the translation probabilities and predict alignments accordingly
3. iterate steps 1 & 2
   - ↪ it becomes apparent that the alignment between `fleur` and `flower` are more likely (pigeon hole principle)

Summary

We can estimate θ with an iterative two steps procedure (EM algorithm):

1. apply the model to the data (E-step)
   - ↪ using the current value of the parameters estimate the value of the hidden variables (here alignment)
2. estimate model from data (M-step)
   - ↪ take assign values as fact
   - ↪ collect counts (weighted by probability)
   - ↪ estimate model from count

These two steps are repeated until convergence.

An historical note: the EM algorithm

Maximum Likelihood from Incomplete Data via the *EM* Algorithm

By A. P. Dempster, N. M. Laird and D. B. Rubin

Harvard University and Educational Testing Service

[Read before the Royal Statistical Society at a meeting organized by the Research Section on Wednesday, December 8th, 1976, Professor S. D. Silvey in the Chair]

Summary

A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

Keywords: maximum likelihood; incomplete data; em algorithm; posterior mode

1. Introduction

This paper presents a general approach to iterative computation of maximum-likelihood estimates when the observations can be viewed as incomplete data. Since each iteration of the algorithm consists of an expectation step followed by a maximization step we call it the em algorithm. The em process is remarkable in part because of the simplicity and generality of the associated theory, and in part because of the wide range of examples which fall under its

↪ A classical paper (1977)

↪ one of the most cited paper in the world (before Transformer & deep learning revolution)...

Predicting Word Alignment

Past

- Giza++ (used to be the *de facto* standard)
- FastAlign (modern best-line, pre-word embeddings)

Today

- `eflomal` (Östling and Tiedemann 2016)
- `SimAlign` (Jalili Sabet et al. 2020)
- `Awesome-Align` (Dou and Neubig 2021)

SimAlign: Embedding-Based Alignment

(Jalili Sabet et al. 2020)

Core idea: Align words using multilingual LM embeddings. **No training**.

Process (short):

- Encode source + target with `mBERT/XLM-R`
- Get token embeddings
- Compute cosine similarity matrix
- Apply heuristic:
  - argmax: direct best match
  - itermax: iterative + symmetrised (best)
  - softmax: prob. filtering

\+ fast; multilingual; strong zero-shot – tokenisation sensitivity; not translation-trained

Awesome-Align: Fine-Tuned Alignment

(Dou and Neubig 2021)

Core idea: Fine-tune `mBERT` on parallel data (TLM) → better alignment.

Process (short):

- Start from `mBERT`
- Fine-tune with TLM: source+target concatenated; mask + predict cross-lingually
- Extract cross-attention weights
- Symmetrise + threshold → alignments

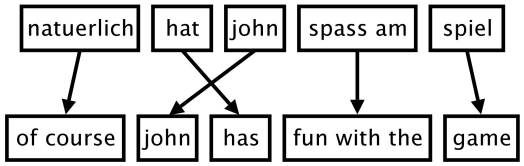
\+ higher accuracy; uses supervision – needs training + GPU; domain-sensitive

Part III

Phrase-Based Machine Translation

Phrase-Based Machine Translation

- another (better) way to estimate $p(s|t)$

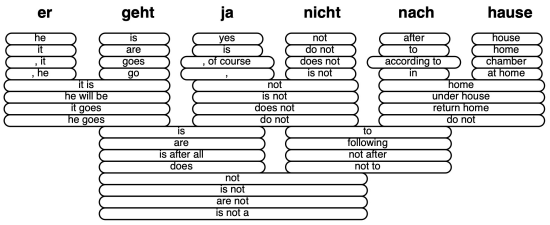


Phrase Table

s	t	$\log P_t(t\|s)$	$\log P_t(s\|t)$	$\log P_{lex}(t\|s)$	$\log P_{lex}(s\|t)$
I like that .	j' aime ça !	0.25	0.03336	0.00689	7.79e-05
I like that .	j' aime ça , tiens .	1	0.01779	0.00344	2.51e-08
I like that .	je préfère ça .	0.0185	0.01348	0.00344	0.000136
I like that	ça , ça me plaît .	0.1	0.00175	0.00344	2.90e-08
I like that	ça me fait plaisir .	0.0277	0.00021	0.00344	4.10e-09
I like that	j' aime cette idée .	0.25	0.04379	0.00344	6.06e-08

↪ extracted from word alignment with heuristics
↪ probability estimated from raw counts

Decoding with a phrase table



⇒ small phrase table: 2,727 matching phrase pairs for this sentence

Is statistical MT still relevant? (1)

- n-gram language models

↪ Are still relevant when you need speed, e.g. filtering large quantities of mined text

- Word alignment

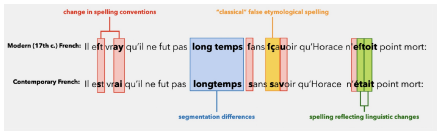
↪ Relevant for a certain number of tasks, including sentence alignment, dictionary creation, ...

- phrase-based models

↪ For certain "translation-like" tasks that do not require long-distance context, e.g. normalization of old texts (with minor spelling errors)

↪ to understand what NMT is doing (see (**<empty citation>**))

Is statistical MT still relevant? (2)

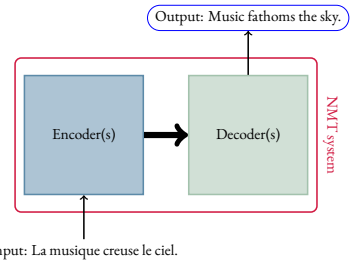


see (Bawden et al. 2022)

Part IV

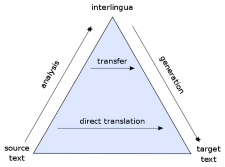
Neural Machine Translation

Overview of a NMT System



↪ encoder (a NN) builds a representation of the source...
↪ ...used by the decoder (a NN) to generate the target

Back to the Future...



↪ same intuition as in Vauquois' pyramid (1968)
⇒ Bernard Vauquois (1929–1985), creator of the *Centre d'Étude pour la Traduction Automatique* (now GETALP) in Grenoble

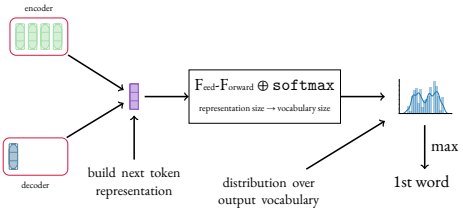
↪ but here all parameters/intermediate representation are discovered automatically

Which NN?

- past: many (many many) choices / alternative
- present: Transformer

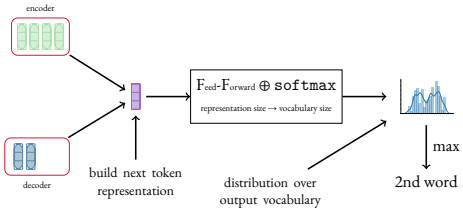
↪ remember: Transformer has been "invented" in the context of MT

Transformer Decoder: a Bird's Eye View



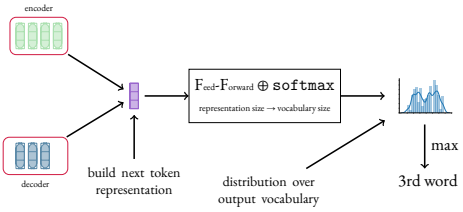
- starting with an empty hypothesis: generates word incrementally
- two sources of information:
  1. representation of the source (always the same)
  2. representation of the prefix (updated at each step)

Transformer Decoder: a Bird's Eye View



- starting with an empty hypothesis: generates word incrementally
- two sources of information:
  1. representation of the source (always the same)
  2. representation of the prefix (updated at each step)

Transformer Decoder: a Bird's Eye View



- starting with an empty hypothesis: generates word incrementally
- two sources of information:
  1. representation of the source (always the same)
  2. representation of the prefix (updated at each step)

JoeyNMT Decoding Method method transformer_greedy in search.py

```
ys = encoder_output.new_full((batch_size, 1), bos_index, dtype=torch.long)

for step in range(max_output_length):
with torch.no_grad():
out, _, att, _ = model(
return_type="decode", trg_input=ys, # <-- prefix
encoder_output=encoder_output, encoder_hidden=None, # <-- technical debt
src_mask=src_mask, unroll_steps=None, decoder_hidden=None)

out = out[:, -1]  # get logits
prob, next_word = torch.max(out, dim=1)
ys = torch.cat([ys, next_word.data.unsqueeze(-1)], dim=1)

is_eos = torch.eq(next_word.data, eos_index)
finished += is_eos
if (finished >= 1).sum() == batch_size: break
```

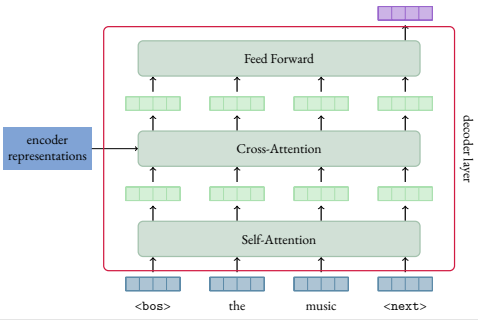
Building the Target Token Representation

Recall:

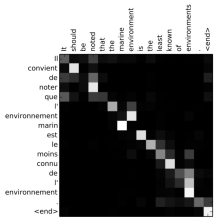
1. old pots made the best soups!
2. attention is all you need!

- we want to build a contextualized representation of the next target tokens
- 2 "kinds" of context:
  - ↪ information from the source
  - ↪ information from the prefix
- we can use to kind of attentions:
  - ↪ self-attention (between target token)
  - ↪ cross-attention (between source and target token)

The Decoder Layer



Cross-Attention



from (Bahdanau, Cho, and Bengio 2015)

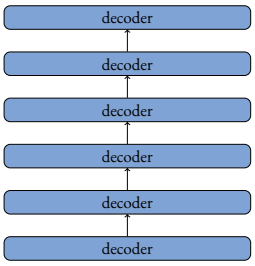
↪ attention in RNN, not transformers

- direct generalization of self-attention
- often described as "soft-word alignment"
- often corresponds to intuition but does not always correspond exactly
- Is attention explanation? Interesting debate:

↪ Attention is not Explanation (Jain and Wallace 2019)

↪ Attention is not not Explanation (Wiegreffe and Pinter 2019)

As Usual...



- stack of decoder layers
- build/predict the representation of the next token
- FF ⊕ softmax → probability distribution over target vocabulary

Translation is reduced to multi-class classification

Training a NMT model

- Loss function

↪ Token-wise cross-entropy: compare true label distribution (1-hot vector) to predicted distribution

- Teacher forcing during training

↪ use the ground truth labels as decoder input rather than previous predictions

↪ Allows parallel processing, but exposure bias (mismatch between training and prediction) (Wang and Sennrich 2020)

- A number of important parameters (not to underestimate):

↪ Architecture: vocabulary size, number of layers, embedding dimensions, etc.

↪ Training parameters: Batch size (+ accumulation of gradients), learning rate, scheduler, etc.

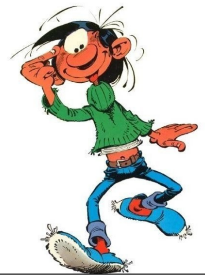
- Model selection — when to stop training?

↪ Loss converges? Performance on a downstream task stops going up (e.g. BLEU)?

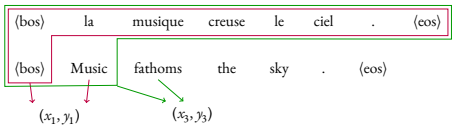
Teacher Forcing: Context

The problem

- NMT = multi-class classification
- history ⊕ source sentence ↦ word
- How training examples can be "created"



Teacher Forcing: definition



- consider the reference translation to create the history ⇒ "gold history"
- 👍 easy to create many examples (no need to predict the history)
- 👍 avoid "bad" history at the beginning of training
- 👍 easy to parallelize: send a parallel sentence to the GPU ⊕ use masks
- 👎 discrepancy between training (gold history) and test (predicted history)

Teacher Forcing: consequence

Exposure Bias

- prediction error have "catastrophic" consequences
- ↪ "huge" impact on the next predictions ⇒ the system has not learned to "correct" errors
- see Nivre & Goldberg dynamic oracle (Goldberg and Nivre 2012; Goldberg and Nivre 2013)

But...

- only way to parallelize training
- ↪ only things that matters ⇒ we need to consider as many data as possible
- everyone has moved on 😟

Label Smoothing

- another unexpected consequence of teacher forcing
- learning criterion: reproduce exactly the gold translation
- ↪ but there are always several way to translate a sentence!
- rather than considering a one-hot distribution for the reference:

$$q'(k) = \begin{cases} 1-\epsilon & \text{if } k = y \\ \frac{\epsilon}{K-1} & \text{otherwise} \end{cases} \quad (4)$$

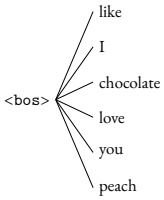
- ↪ K: number of classes
- ↪ label smoothing (Szegedy et al. 2016)
- used in the very first transformer paper ⊕ (Müller, Kornblith, and Hinton 2019) for a full analysis

Part V

NMT: Decoding

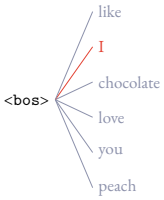
Visualizing Decoding

- vocabulary: I, love, like, you, chocolate, peach



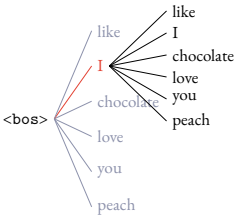
Visualizing Decoding

- vocabulary: I, love, like, you, chocolate, peach



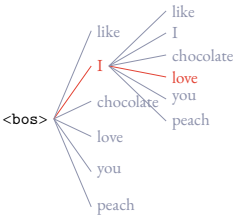
Visualizing Decoding

- vocabulary: I, love, like, you, chocolate, peach



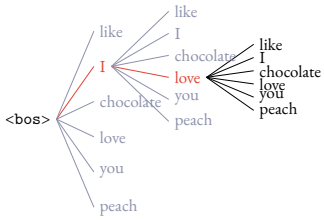
Visualizing Decoding

- vocabulary: I, love, like, you, chocolate, peach

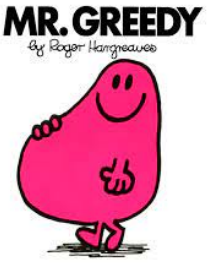


Visualizing Decoding

- vocabulary: I, love, like, you, chocolate, peach



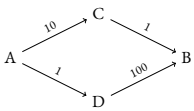
To put things in a nutshell



- greedy decoding
- best decision at each step
- easy to implement
- very efficient
- but no guarantee that the solution is optimal!

The Problem of Greedy Decoding

- shortest path problem: smallest cost to go from A to B



- brute force search: enumerate all paths (exponential!)
- greedy search: very fast but not optimal
- exact search (dynamic programming) fast but optimal

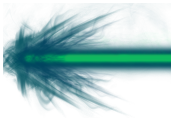
The "Magic" Solution: Beam Search

Principle

- at each step keep the n best solutions
  1. "expand" these solutions $\rightarrow n \times \#\mathcal{Y}$ solutions
  2. prune these solutions to keep only the n "best"

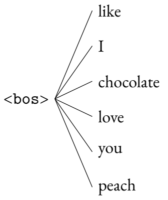
Why?

- allows you to "reconsider" past decisions
- explore a large part of the search space while controlling decoding complexity
- classic solution to reduce the impact of search error when the exact solution can not be found



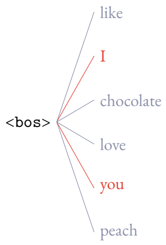
With a Picture

- with a beam of size 2



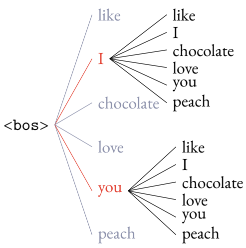
With a Picture

- with a beam of size 2



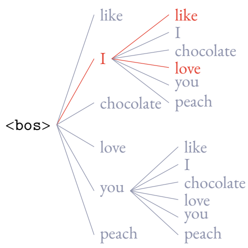
With a Picture

- with a beam of size 2



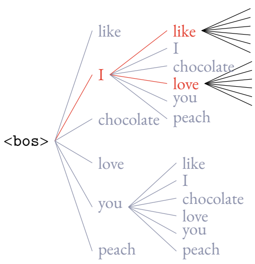
With a Picture

- with a beam of size 2

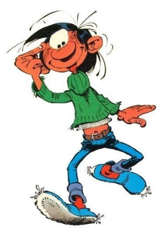


With a Picture

- with a beam of size 2



Does It Work?



Of course...

- large gains (several BLEU points)...
- ...but only for small beams (usually 5)

But...

- much more difficult to implement
- much slower than greedy decoding
- performance decreases with beam size !!!!

The Beam Size Curse

Observation

- experimentally: performance decreases with beam size
  - ↪ counter-intuitive
  - ↪ contrary to the results/observation obtained on many tasks/models
- still no explaination
  - ↪ consensus: we do not know what is going on

Can the situation be any worse?

- Yes!
- if you do exact decoding: best solution = empty hypothesis

↪ we are not modeling what we think we are modeling

↪ no clue how/why the system is working

Other decoding strategy

- Sampling: like greedy, but sampling from the distribution (more diverse outputs)
  - Top-k sampling: a token amongst the k top tokens
  - Top-p (nucleus) sampling: a token amongst those with the highest probabilities (that sum to p)
  - Temperature sampling: flattens or sharpens the probability distribution (0=original, higher=more flat)

important in text generation by LLMs, less so in translation

Part VI

NMT Systems

Context

- difficult part ≠ translation model

↪ "simple" transformer

↪ tutorial to implement your own model in `pytorch` and `tensorflow`

- difficult part = everything around

↪ data loading / tokenization

↪ evaluation

↪ scaling training

Some Names Dropping

- OpenNMT

↪ Harvard + Systran

↪ historical toolkit ⊕ well-documented

- Marian NMT

↪ C++ framework, very efficient

↪ production ready (is/was used in Microsoft Translator)

- Sockeye

↪ developped by Amazon

- JoeyNMT

↪ "pedagogical" project

Part VII

Pre-Trained Models

Main "pretrained" Model of Translation

- all models are available on `Hugging Face` 🤗
- `MarianMT / OPUS-MT (Helsinki-NLP)`

↪ `marian` (transformer 6 × 6) trained on `opus` data

- `mBART-50`

↪ multilingual model

↪ fine-tuned on parallel data

- `M2M100`

↪ model many-to-many, without pivot

- `NLLB-200`

↪ ≃ 200 languages, focus on under-resourced languages

- `T5/mT5`

↪ "independant" encoder/decoder models → need to be fine-tuned for MT

Part VIII

Where are we now?

Evolution of MT performance over time

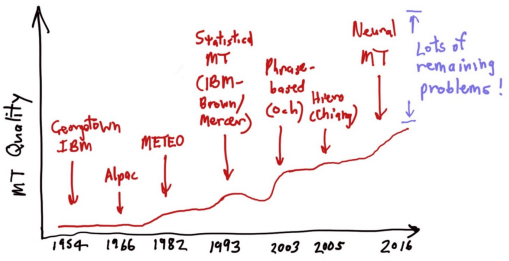
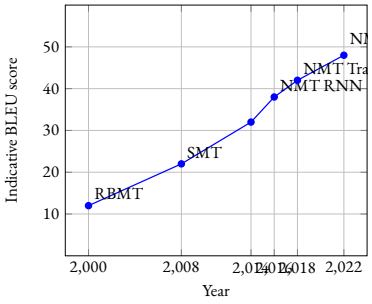


Figure 1.1: **Machine translation progress** – from the 1950s, the starting of modern MT research, until the time of this thesis, 2016, in which neural MT becomes a dominant approach. Image courtesy of Christopher D. Manning

Evolution of MT performance over time



Core Challenges in NMT

- **Data Scarcity and Domain Mismatch**
  - Many language pairs lack large-scale parallel corpora.
  - Performance drops significantly when translating out-of-domain text.
- **Handling Morphologically Rich Languages**
  - Agglutination, inflection, and complex morphology lead to sparse representations.
  - Tokenisation and subword approaches only partly mitigate these issues.
- **Long-Range Dependencies**
  - Difficulty modelling global sentence structure and discourse-level context.
  - Errors accumulate when processing long or syntactically complex sentences.

Linguistic and Semantic Challenges

- **Ambiguity and Polysemy**
  - NMT systems may choose incorrect word senses when context is limited.
- **Idioms, Figurative Language, and Cultural Expressions**
  - Non-literal constructions are often translated word-for-word.
- **Faithfulness vs. Fluency**
  - Models may generate fluent but semantically incorrect output ("hallucinations").

Part IX

References

- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.
- Bawden, Rachel et al. (June 2022). "Automatic Normalisation of Early Modern French". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Ed. by Nicoletta Calzolari et al. Marseille, France: European Language Resources Association, pp. 3354–3366.
- Brown, Peter F. et al. (1993). "The Mathematics of Statistical Machine Translation: Parameter Estimation". In: *Computational Linguistics* 19.2. Ed. by Julia Hirschberg, pp. 263–311.

- Dou, Zi-Yi and Graham Neubig (Apr. 2021). "Word Alignment by Fine-tuning Embeddings on Parallel Corpora". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Ed. by Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty. Online: Association for Computational Linguistics, pp. 2112–2128.
- Goldberg, Yoav and Joakim Nivre (Dec. 2012). "A Dynamic Oracle for Arc-Eager Dependency Parsing". In: *Proceedings of COLING 2012*. Ed. by Martin Kay and Christian Boitet. Mumbai, India: The COLING 2012 Organizing Committee, pp. 959–976.
- — (2013). "Training Deterministic Parsers with Non-Deterministic Oracles". In: *Transactions of the Association for Computational Linguistics* 1. Ed. by Dekang Lin and Michael Collins, pp. 403–414.

- Imani, Ayyoob et al. (July 2023). "Glot500: Scaling Multilingual Corpora and Language Models to 500 Languages". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 1082–1117.
- Jain, Sarthak and Byron C. Wallace (June 2019). "Attention is not Explanation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 3543–3556.
- Jalili Sabet, Masoud et al. (Nov. 2020). "SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 1627–1643.

- Müller, Rafael, Simon Kornblith, and Geoffrey Hinton (2019). "When does label smoothing help?" In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.
- NLLB Team et al. (2022). *No Language Left Behind: Scaling Human-Centered Machine Translation*.
- Östling, Robert and Jörg Tiedemann (Oct. 2016). "Efficient word alignment with Markov Chain Monte Carlo". English. In: *The Prague Bulletin of Mathematical Linguistics* 106, pp. 125–146.
- Szegedy, Christian et al. (2016). "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826.

- Wang, Chaojun and Rico Sennrich (July 2020). "On Exposure Bias, Hallucination and Domain Shift in Neural Machine Translation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, pp. 3544–3552.
- Wiegreffe, Sarah and Yuval Pinter (Nov. 2019). "Attention is not not Explanation". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, pp. 11–20.