Decoding Algorithms

Deciding under Uncertainty in Machine Translation

Wilker Aziz

w.aziz@uva.nl

https://probabll.github.io

UTTER

This class is about the *(decoding) algorithms* that turn input text in one language into output text in another, with the help of a (language) model to handle the many choices along the way.

This class is for those

- developing new algorithms
- choosing amongst existing algorithms
- using decoding algorithms

Table of contents

NMT

The *autoregressive language model* API

Throughout the talk, I assume that one's preferred MT engine is powered by an *autoregressive language model*.

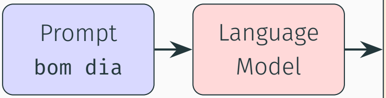
This choice implies access to a specific API that makes various crucial operations (incl. those needed for training and decoding) feasible to varying degrees of approximation.

This API allows us to regard an LM as a means to predict *conditional* (that is, input-specific) *probability distributions* (cpds).[1]

[1] You may also reason the other way around: LMs are designed to predict input-specific probability distributions, when they are designed to comply with a certain API, they are regarded as *autoregressive*.

Prompt → Language Model → Distribution over Responses

From sufficiently far away, we can regard an LM as machine that maps any one prompt to a prompt-specific *probability distribution* whose outcome space is the set of all complete token sequences.



Response Distribution

Complete token sequence	Prob
`morn- -ing EOS`	0.0060
`good morn- -ing EOS`	0.0021
`hello EOS`	0.0015
`hi EOS`	0.0007
`good day EOS`	0.0005
...	

*table-like structure for intuition only

Short Digression: Statistical Learning

Training algorithms that approximate maximum likelihood estimation (e.g., supervised tuning or fine tuning using translation data) will make these cpds 'more coherent' with statistics of observed *translation data*.

That's because LMs trained like that learn to predict distributions from *data samples* (not from those samples' 'probabilities').

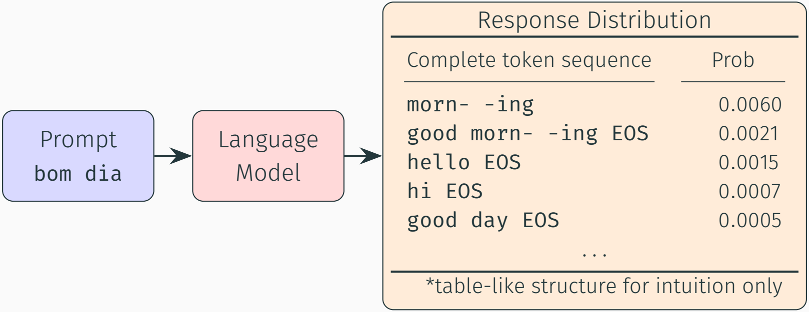
Roughly, the more training data you observe, the less you can tell *data samples* from *model samples* apart.[2]

Prompt	LM-Sampled Responses	Responses by Human Translators
`bom dia`	`good morning`	`morning!`
	`hello`	`good morning`
	`morning!`	`hello`
	`hi there!`	`good morning`
	`morning`	`hey there!`

[2]The notion of 'sample' here is a rather specific one, we will talk about it later.

Not quite the whole story...

As we zoom in, we realise that an LM does not really build anything like this 'tabular' representation of the cpd:



rather, it parameterises a special kind of iterative process, which *implicitly* identifies one such object.[3]

[3]Then, certain ways of interacting with that iterative process is statistically equivalent to interacting with the table-like thing.

Prompt and Prefix → LM Primitive → Next-Token Distribution

With an empty prefix (represented by a sequence containing BOS only)

Prompt	Prefix
`bom dia`	`BOS`

→ LM Primitive →

Token	Prob
`a`	0.15
`day`	0.05
`good`	0.1
`-ing`	0.001
`morn-`	0.1
...	
`EOS`	0.01

With a longer prefix sequence:

Prompt	Prefix
`bom dia`	`BOS good`

→ LM Primitive →

Token	Prob
`a`	0.01
`day`	0.25
`good`	0.02
`-ing`	0.001
`morn-`	0.15
...	
`EOS`	0.01

Prompt `bom dia` and Outcome `good morn- -ing EOS`

*prompt omitted from input for space

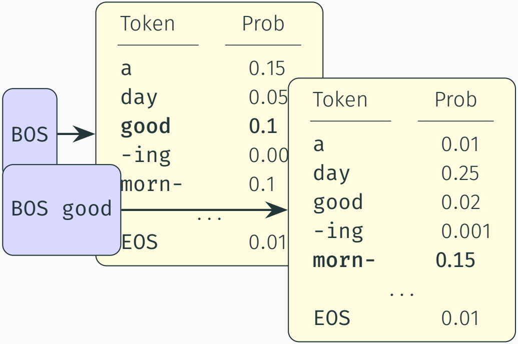
BOS →

Token	Prob
`a`	0.15
`day`	0.05
`good`	**0.1**
`-ing`	0.001
`morn-`	0.1
...	
`EOS`	0.01

With probability 0.1, draw **`good`**

Prompt `bom dia` and Outcome `good morn- -ing EOS`

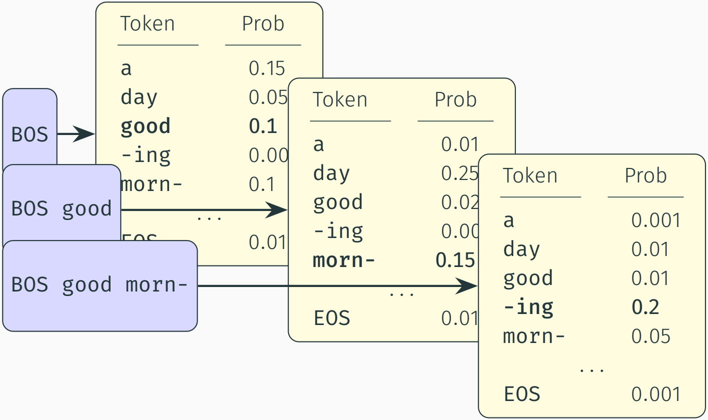
*prompt omitted from input for space



With probability 0.15, draw `morn-`

Prompt bom dia and Outcome good morn- -ing EOS

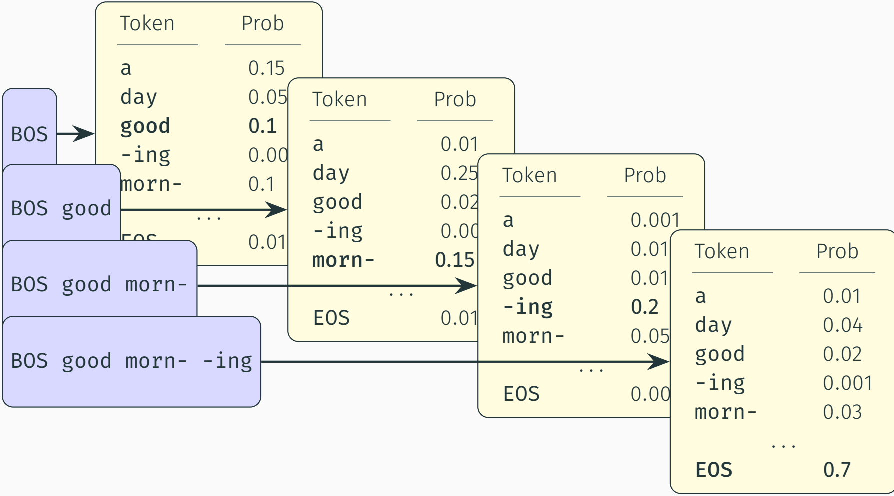
*prompt omitted from input for space



With probability 0.2, draw -ing

Prompt bom dia and Outcome good morn- -ing EOS

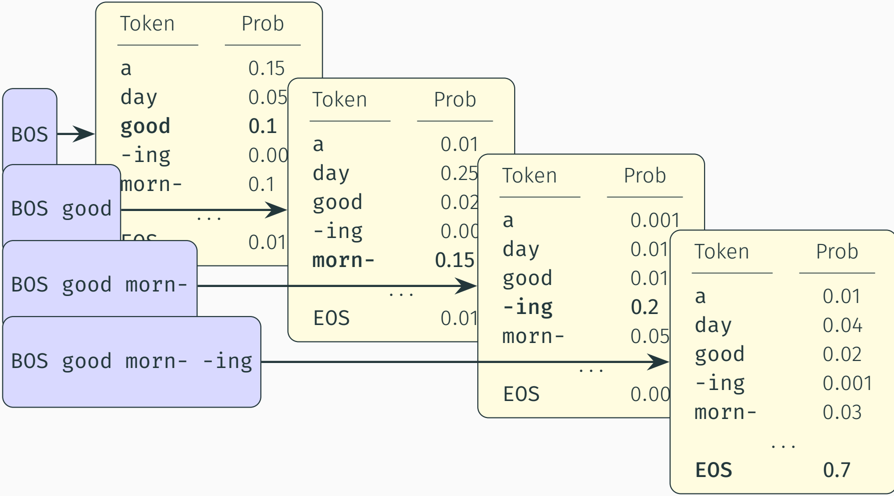
*prompt omitted from input for space



With probability 0.7, draw **EOS**

Prompt `bom dia` and Outcome `good morn- -ing EOS`

*prompt omitted from input for space



$p_\theta(\texttt{good morn- -ing EOS}|\texttt{bom dia}) = 0.1 \times 0.15 \times 0.2 \times 0.7 = 0.0021$

Factorised Probabilities

Given a prompt x, an autoregressive LM factorises the probability it assigns to any one outcome sequence $y = \langle y_1, \ldots, y_\ell \rangle$ along the ℓ tokens that make up the outcome, as follows:

$$p_\theta(y|x) = \prod_{i=1}^{\ell} p_\theta(y_i|x, y_{<i}) \ . \tag{1}$$

[5] Mapping from $(x, y_{<i})$ to one such vector is a task easily accomplished by architectures like RNNs and Transformers.

Factorised Probabilities

Given a prompt x, an autoregressive LM factorises the probability it assigns to any one outcome sequence $y = \langle y_1, \ldots, y_\ell \rangle$ along the ℓ tokens that make up the outcome, as follows:

$$p_\theta(y|x) = \prod_{i=1}^{\ell} p_\theta(y_i|x, y_{<i}) \ . \tag{1}$$

Under the assumption that the vocabulary is finite, with V symbols, and independent of the position i, any one of the next-token distributions is specifiable by a V-dimensional probability vector.[5]

[5] Mapping from $(x, y_{<i})$ to one such vector is a task easily accomplished by architectures like RNNs and Transformers.

Why are LMs so often Designed this Way?

There are various answers, here are some

1. there are infinitely many responses, but only finitely many tokens at each step;
2. this allows us to assess the probability mass of a response efficiently;
3. this allows us to ‘draw’ outcomes from the model, often with useful statistical guarantees.

(1) is about feasibility, (2) is useful for supervised training (but also some forms of decoding), (3) is particularly useful for decoding (but also some forms of training).

Summary

We can regard an LM as a mechanism trained to predict entire input-specific probability distributions over the space of responses.

The most common such mechanisms (incl. encoder-decoder and decoder-only Transformer models) are built upon a chain-rule factorisation of the probability of sequences. This allows us to regard LMs as offering 4 features (the first 2 being the primitives):

1. given prompt x and prefix r, assign probability $p(t|x, r)$ to token t
2. given x and r, draw a token t with probability $p(t|x, r)$
3. assign probability $p(y|x)$ to a response y given x
4. with probability $p(y|x)$, draw a response y given x

There are interesting designs that violate this API (e.g., EBMs), but I am not covering those today.

Translating

Do Translation Models Translate?

“By what built-in mechanism may the model autonomously decide that a response y is to be regarded as the translation of x?”

Do Translation Models Translate?

"By what built-in mechanism may the model autonomously decide that a response y is to be regarded as the translation of x?"

Our models do not have the agency do decide. But we can design recipes—which our models do parameterise—to automate decision making. Those recipes are called *decoding algorithms*.

Principles

Let's outline basic principles we would like a decoding algorithm to observe

Principles

Let's outline basic principles we would like a decoding algorithm to observe

1. translations should be in some sense *preferred* by the model (else, what is the difference between using one model or another?);
2. translations are ideally good for their prompts, let's say they ought to be *adequate*;

Do Translation Models Prefer Some Translations to Others?

In one sense, our models are not very picky. So long as the individual tokens are known ‘a what the cat xxx ? EOS’ is *in the outcome space* of any model no matter the prompt.

That said, given the prompt `` `olha, um gato!` ``, two models that share the same vocabulary may easily differ in the probabilities they assign to ‘look, a cat! EOS’ and ‘a what the cat xxx ? EOS’.

We can regard probabilities as expressing a notion of preference that’s ‘native’ to the model.[6]

[6] This notion is in fact coherent with most forms of training, where model parameters are chosen to assign high probability to observed data.

Translating

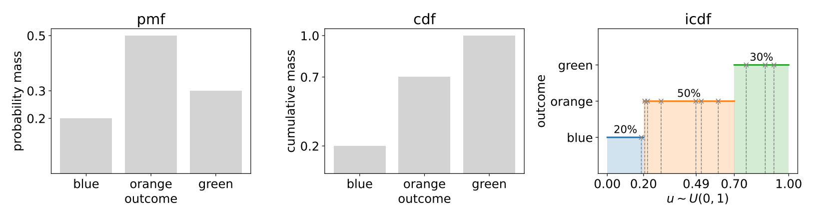
Samplers

Unbiased Sampling

A sampler draws realisations of a random variable. For us, this means drawing responses (i.e., token sequences that end in EOS) from the distribution that an LM (implicitly) predicts when given a prompt x.

An unbiased sampler is one where, if we draw N samples independently of one another, the relative frequency of any of the sampled responses is an unbiased estimator of that response's probability under the model, and the estimation variance decays as N increases.

Example: sampling from a distribution over 3 categories



From the probability mass function (pmf) we obtain the cumulative distribution function (cdf), we then characterise the cdf's inverse (icdf). The icdf associates each outcome with a line segment whose length equals the outcome's probability mass.

The icdf transforms a uniform random generator into an unbiased sampler for this distribution. The example shows 10 samples (e.g., 0.48 maps to `orange`, as do all numbers between 0.2 and 0.7).

Ancestral (or Forward) Sampling

As a consequence of the API we agreed upon, a simple iterative algorithm can be shown to result in unbiased samples from the distribution over *responses*:[7]

1. Reset the sampler state (i.e., condition on prompt and start an empty generation prefix).
2. Use the LM to obtain the next-token distribution, draw the next token from it (via the icdf method) and extend the generation prefix with it.
3. If the token was EOS, terminate the algorithm returning the sampled sequence, else repeat from (2).

[7] [1, 19]

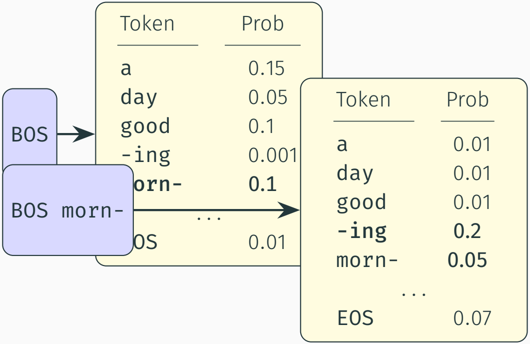
Ancestral Sampling - Prompt `bom dia`

BOS →

Token	Prob
`a`	0.15
`day`	0.05
`good`	0.1
`-ing`	0.001
`morn-`	**0.1**
...	
`EOS`	0.01

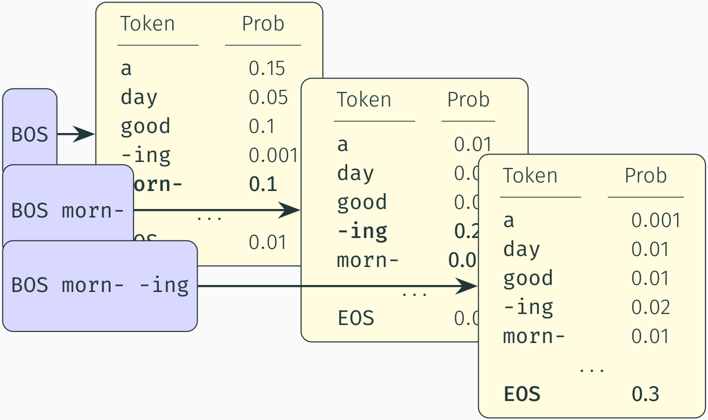
With probability 0.1, draw `morn-`

Ancestral Sampling - Prompt `bom dia`



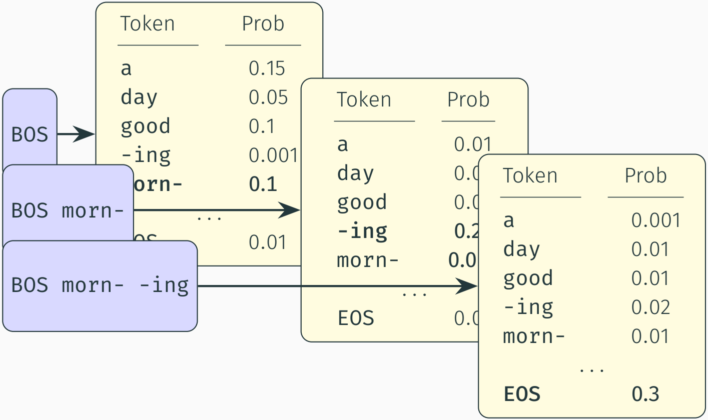
With probability 0.2, draw `-ing`

Ancestral Sampling - Prompt `bom dia`



With probability 0.3, draw **EOS**

Ancestral Sampling - Prompt `bom dia`



Return `morn- -ing EOS` with

$p_\theta(\texttt{morn- -ing EOS}|\texttt{bom dia}) = 0.1 \times 0.2 \times 0.3 = 0.006$

A Critical Eye

Unbiased sampling operationalises a notion of ‘preferred under the model’, but this notion is a ‘statistical’ one: the decisions it can support get increasingly risky the less samples we draw.

If we collect many samples, we expect a fraction to come from the red group (1 in 10, on average).

Complete token sequence	Prob
`hello EOS`	0.35
`hi EOS`	0.25
`hey EOS`	0.2
`morn- -ing EOS`	0.1
`EOS`	0.01
...	

But, if we draw one sample, it might well be `EOS` or one of the outcomes in that group.

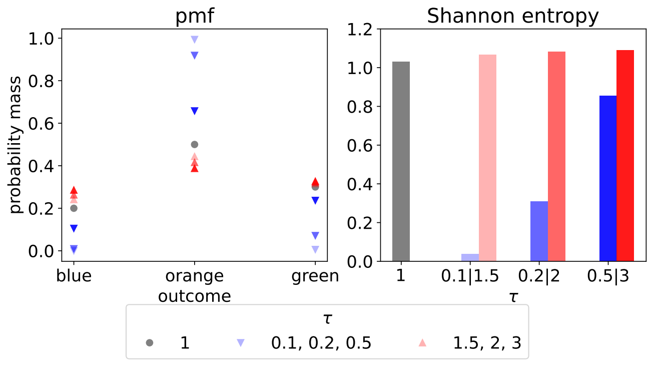
Biased Samplers

A *biased* sampler deviates from the model's native probabilistic interpretation.

But, if we invest so much in model training, are there good reasons for deviating from the model?

Biasing a sampler with a ‘temperature’

Let’s use the 3 categories example. Say the probabilities are p_1, p_2, p_3. We can define alternative distributions by introducing a ‘temperature parameter’ $\tau > 0$: then new pmfs can be obtained via $\frac{p_k^{1/\tau}}{p_1^{1/\tau}+p_2^{1/\tau}+p_3^{1/\tau}}$



Temperature Sampling

A modification of ancestral sampling, where we transform next-token distributions by exponentiation and renormalisation *before* sampling.

Critical Eye - Choosing the Temperature

Should we aim for more or less entropy?

Critical Eye - Choosing the Temperature

Should we aim for more or less entropy?

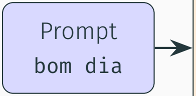
In the literature, you will find advices such as

- *more entropy*, as to promote diversity amongst samples;
- *less entropy*, as to discourage/prune low-probability outcomes.

There is no reason to believe that we can motivate a choice of τ from introspection alone. The best we can do is to treat τ as a hyperparameter and pick it experimentally (under the assumption that we can simulate the test conditions reasonably well in the lab).

Critical Eye - Understanding the Effect

By applying a temperature (say $\tau = 0.5$) to each next-token distribution from left-to-right, are we essentially applying a temperature κ to the distribution over responses?



Complete token sequence	Prob	$\text{Prob}^{1/\kappa}$
`morn- -ing EOS`	0.0060	0.8276
`good morn- -ing EOS`	0.0021	0.1014
`hello EOS`	0.0015	0.0517
`hi EOS`	0.0007	0.0113
`good day EOS`	0.0005	0.0057
...		

*table-like structure for intuition only

Critical Eye - Understanding the Effect

By applying a temperature (say $\tau = 0.5$) to each next-token distribution from left-to-right, are we essentially applying a temperature κ to the distribution over responses?

Prompt
`bom dia`

Complete token sequence	Prob	Prob$^{1/\kappa}$
`morn- -ing EOS`	0.0060	0.8276
`good morn- -ing EOS`	0.0021	0.1014
`hello EOS`	0.0015	0.0517
`hi EOS`	0.0007	0.0113
`good day EOS`	0.0005	0.0057
...		

*table-like structure for intuition only

By chain rule of probabilities, we know that there exists an autoregressive decomposition of Prob$^{1/\kappa}$ along the token sequence, but that factorisation **is not** of the form $\propto \prod_{i=1}^{\ell} p_\theta^\tau(y_i|x, y_{<i})$, where we simply exponentiate and normalise the original next-token cpds.

Critical Eye - Summary

We cannot expect a temperature to serve all prompts alike.

We need to treat temperature as a hyperparameter.

The intuition we developed in the simple case of distributions over categories does not transfer to distributions over sequences.

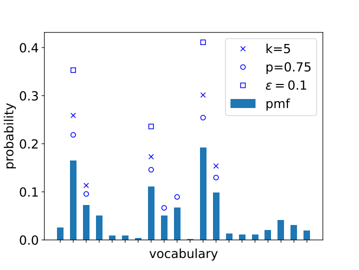
Probability- and Mode-Seeking Samplers

When we sample with low temperature, we not only discourage the outcomes with less mass, we exaggerate differences throughout the whole probability spectrum, distorting the shape of the distribution.

Other ideas are formulated more directly as a form of pruning and tend to better preserve the relative merits of the outcomes that are not pruned.

Truncation Sampling: top-k, top-p, and ϵ-sampling.

Choose a criterion, prune outcomes that do not meet it, renormalise the next-token cpd, sample.



The top-k sampler [7] prunes all but the k most probable tokens, the next-token cpd is then renormalised over this reduced outcome space.

The top-p sampler [aka nucleus sampler; 14] also prunes all but the most probable tokens, but it keeps as many tokens as needed to cover a pre-specified amount of probability mass.

The ϵ-sampler [13] prunes any outcome whose mass is less than some $\epsilon > 0$.

Critical Eye

These biased samplers operationalise a clearer bet: we are betting that good sequences will have few, if any, low-probability tokens.

The question is, why should that be the case?

Critical Eye

These biased samplers operationalise a clearer bet: we are betting that good sequences will have few, if any, low-probability tokens.

The question is, why should that be the case?

- there is no obvious reason why we should expect a good sequence to have no (or even very few) low-probability tokens;
- we may like models that exhibit such a property, but ours were not designed and trained to meet it.

Locally Typical Sampling

Meister et al. [26] motivate a different criterion to sort the tokens for a nucleus sampler (think of it as defining the nucleus differently).

Keep enough tokens to cover at least a probability mass p, but sort tokens on the absolute difference between their individual 'surprisal'

$-\log p(t|x, r)$ for a token t, prompt x and generated prefix r

and the expected surprisal (aka Shannon entropy):

$$-\frac{1}{V}\sum_{w\in\mathcal{W}} \log p(w|x, r)\ .$$

Critical Eye

The original paper motivated locally typical sampling from i) findings in psycholinguistics, and ii) a remarkable property of certain Markov processes (MPs) concerning how surprisal values distribute.

To my understanding, there are at least two points of contention:

- The psycholinguistic finding need not transfer to any one model (we may wish that to be true, but it need not be)
- Autoregressive LMs are not guaranteed to meet the necessary formal properties of MPs that exhibit strong regularities in how surprisals distribute.

Nonetheless, typical sampling offers an interesting, 'non-mode-seeking' way to truncate next-token distributions.

Stochastic Processes

When we pair a model and a choice of sampler, we *induce* a distribution over responses [4].

If this sampler is unbiased, the distribution is precisely the one the model predicts.

When the sampler is biased, we cannot say much.

But we can say one thing: relative to that distribution (unless it happens to have a remarkably low entropy), a single sample conveys very little information.

Summary

Unbiased samplers operationalise a notion of 'preferred by the model': they allow us to interact with the prompt-specific probability distribution that is coherent with our model.

Biased samplers capture preferences that we motivate ourselves (such as more or less entropy, avoiding low-probability transitions, avoiding too-low or too-high token surprisal relative to the entropy of the next-token cpd, etc.).

Samplers induce stochastic processes and it's hard to imagine a property that a single sample is guaranteed to satisfy.

Translating

Decision Rules: Searching for a Specific Translation

From Random (but not arbitrary) Exploration to Search

Suppose we could assign a notion of of quality $\mu(c; x)$ to any candidate translation c of a prompt x.

Example: ask a person to give it a mark, from 0 to 100.

From Random (but not arbitrary) Exploration to Search

Suppose we could assign a notion of of quality $\mu(c; x)$ to any candidate translation c of a prompt x.

Example: ask a person to give it a mark, from 0 to 100.

Wouldn't a good translation be one that maximises that score?

$$y^{\text{decision}} = \mathsf{argmax}_{c \in \mathcal{Y}} \; \mu(c; x) \tag{2}$$

This is what we call a *decision rule*, where we search for a specific response, using an explicitly stated criterion.

From Random (but not arbitrary) Exploration to Search

Suppose we could assign a notion of of quality $\mu(c; x)$ to any candidate translation c of a prompt x.

Example: ask a person to give it a mark, from 0 to 100.

Wouldn't a good translation be one that maximises that score?

$$y^{\text{decision}} = \text{argmax}_{c \in \mathcal{Y}} \; \mu(c; x) \tag{2}$$

This is what we call a *decision rule*, where we search for a specific response, using an explicitly stated criterion.

How can our translation model be of any use for this?

From Random (but not arbitrary) Exploration to Search

Suppose we could assign a notion of of quality $\mu(c; x)$ to any candidate translation c of a prompt x.

Example: ask a person to give it a mark, from 0 to 100.

Wouldn't a good translation be one that maximises that score?

$$y^{\text{decision}} = \mathsf{argmax}_{c \in \mathcal{Y}} \ \mu(c; x) \tag{2}$$

This is what we call a *decision rule*, where we search for a specific response, using an explicitly stated criterion.

How can our translation model be of any use for this?
One or both of the following:

- it can contribute to the definition of μ;
- it can prioritises subsets of the search space;

Most Probable Response

Here's a line of argumentation: *"if there is one outcome that my model prefers, that outcome ought to be the mode of the conditional distribution over responses."*

$$y^{\text{mode}} = \text{argmax}_{c \in \mathcal{Y}} \; p_\theta(c|x) \tag{3}$$

Unlike a single sample from any sampler, this outcome satisfies a clear criterion: its probability is larger than that of any other outcome.

Do you see any problems?

This has come to be known as maximum-a-posteriori (MAP) decoding.

Most Probable Response

Here's a line of argumentation: *"if there is one outcome that my model prefers, that outcome ought to be the mode of the conditional distribution over responses."*

$$y^{\text{mode}} = \text{argmax}_{c \in \mathcal{Y}} \; p_\theta(c|x) \tag{3}$$

Unlike a single sample from any sampler, this outcome satisfies a clear criterion: its probability is larger than that of any other outcome.

Do you see any problems? I see two: i) go about finding it, and ii) what if the mode is of no special significance?

This has come to be known as maximum-a-posteriori (MAP) decoding.

Intractable Search

The search space is unbounded and due to the chain-rule factorisation (no Markov assumptions) dynamic programming isn't possible.

Prompt
`bom dia`

Complete token sequence	Prob	Prob$^{1/\kappa}$
`morn- -ing EOS`	0.0060	0.8276
`good morn- -ing EOS`	0.0021	0.1014
`hello EOS`	0.0015	0.0517
`hi EOS`	0.0007	0.0113
`good day EOS`	0.0005	0.0057
...		

*table-like structure for intuition only

The Greedy Approximation

At each step i, we find the token that is assigned maximum probability given the prompt and the generated prefix $y_{<i}$:

$$y_i \leftarrow \text{argmax}_{w \in \mathcal{W}} \; p_\theta(w|x, y_{<i}) \tag{4}$$

At each step i, we find the token that is assigned maximum probability given the prompt and the generated prefix $y_{<i}$:

$$y_i \leftarrow \mathrm{argmax}_{w \in \mathcal{W}} \ p_\theta(w|x, y_{<i}) \tag{4}$$

BOS →

Token	Prob
a	**0.15**
day	0.05
good	0.1
-ing	0.001
morn-	0.1
...	
EOS	0.01

The Greedy Approximation

At each step i, we find the token that is assigned maximum probability given the prompt and the generated prefix $y_{<i}$:

$$y_i \leftarrow \text{argmax}_{w \in \mathcal{W}} \; p_\theta(w|x, y_{<i}) \tag{4}$$

`BOS` →

Token	Prob
`a`	**0.15**
`day`	0.05
`good`	0.1
`-ing`	0.001
`morn-`	0.1
...	
`EOS`	0.01

`BOS a` →

Token	Prob
`a`	0.0008
`day`	0.0002
`good`	0.0002
`-ing`	0.0001
`morn-`	0.0005
...	
`EOS`	**0.0010**

Return `a EOS` with $p_\theta(\texttt{a EOS}|\texttt{bom dia}) = 0.15 \times 0.01 = 0.00015$

This strategy is simple but makes a lot of *search errors* (i.e., fails to find the mode).

Better Approximate Search: Beam Search

At each step, we keep refining a small set of candidates (for example, $k = 5$ candidates). We could then,

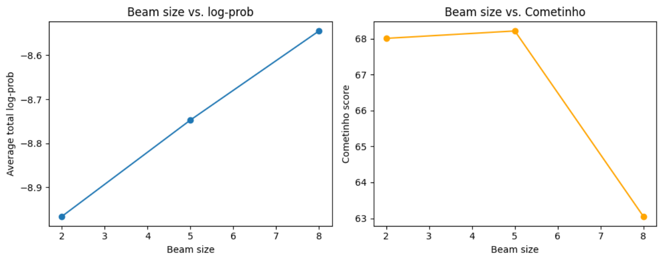
- consider all $k \times V$ ways in which these k candidates can be extended by one token each;
- rank these on an estimate of their future success as complete responses, and retain again only k.

The simplest estimate of future success is the probability of the (incomplete) sequence as it stands.

Implementations vary (see for example [24]), but that's the general idea.

Beam Search Curse

With more computation (i.e., larger *k*), beam search reduces search errors (i.e., it finds responses with higher probabilities than greedy search does), but this does not always translate to better translations [aka 'the beam search curse'; 18].



A spoiler for this afternoon's lab

The Length ‘Culprit’

As beam size increases, and quality deteriorates, we often observe that the MAP decoder returns **shorter sequences** [33].

This observation led to various attempts at identifying a built-in bias towards short sequences and correct for it [15, 28, 36].

Controlling Length

We can augment the MAP decoder with the ability to judge outcomes on their *length* besides their probabilities:

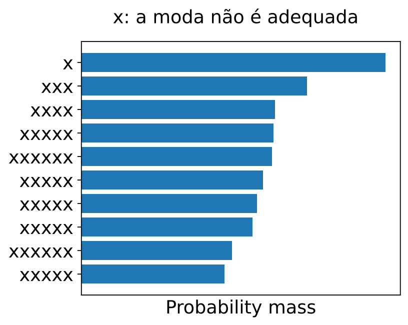
- length normalisation

$$\text{argmax}_{c \in \mathcal{Y}} \ \frac{1}{|c|} \log p(c|x)$$

- length penalty

$$\text{argmax}_{c \in \mathcal{Y}} \ \log p(c|x) - |c|\lambda$$

- amongst others
  [2, 12, 15, 17, 28, 36]



Regularised Beam Search

Meister et al. [23] views the ‘search errors’ of beam search as implicit (but interpretable) biases in search. They then express these biases explicitly as ‘regularisers’ on the original objective

$$\operatorname{argmax}_{c\in\mathcal{Y}} \; \log p(c|x) - \lambda\mathcal{R}(c, p_\theta(.|x)) \tag{5}$$

and use this framework to propose novel decoding strategies.

Critical Eye

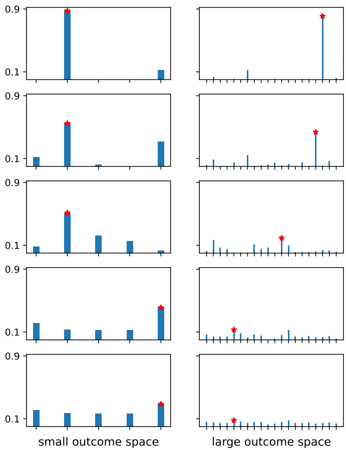
Remember my two contentions: i) go about finding the mode, and ii) **what if the mode is of no special significance?**

In relation to (ii)

- Stahlberg and Byrne [34] show that modes are often inadequate translations (such as the empty sequence);
- Eikema and Aziz [5] show that the mode is indeed often simply rare;
- adequate samples (e.g., references) tend not to be modes [5, 25].

There's growing evidence that 'typically realisable' samples from autoregressive models exhibit a concentration of surprisal. Roughly, if models were efficient data stores, they would store adequate responses in samples of 'average surprisal'.

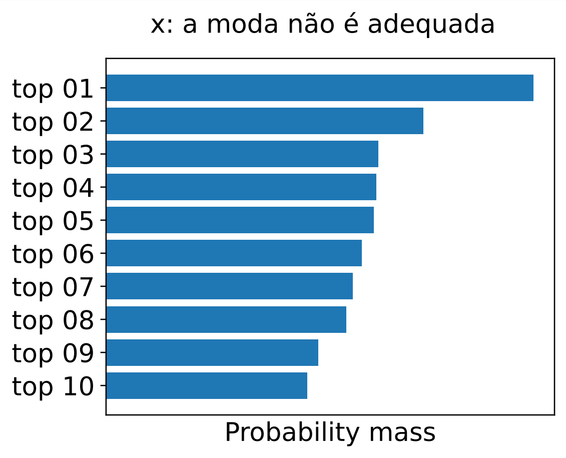
Intuitions about the Mode



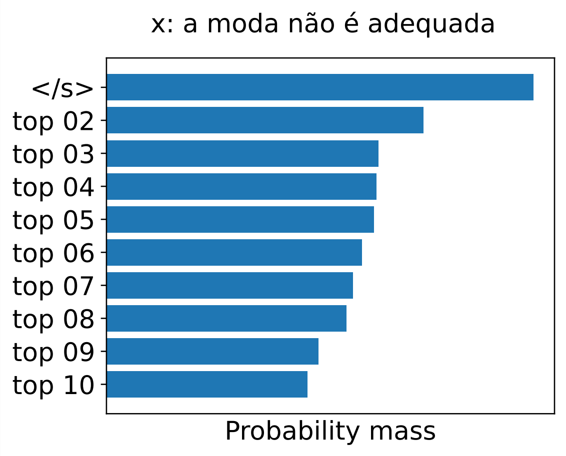
Our intuitions about modes quickly fall apart as outcome spaces grow very large.

Remember, the distribution over *responses* has infinitely many outcomes in it.

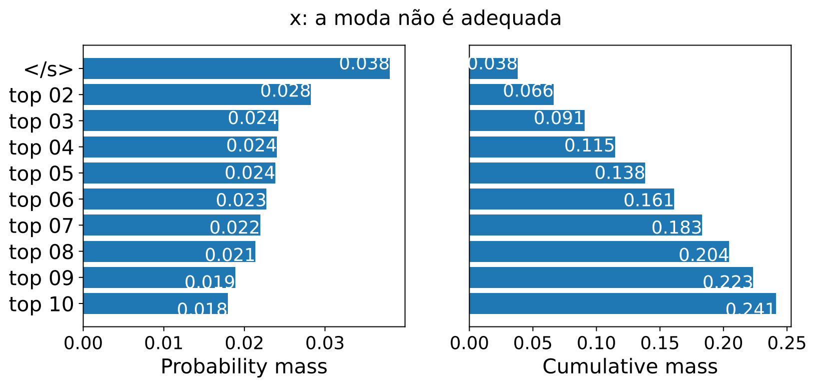
This is how a MAP decoder makes decisions: it judges outcomes on probability alone.



But then the mode can be clearly inadequate

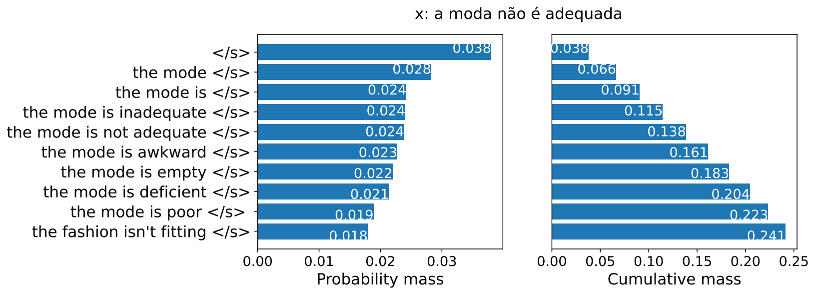


But empty modes are often *rare*



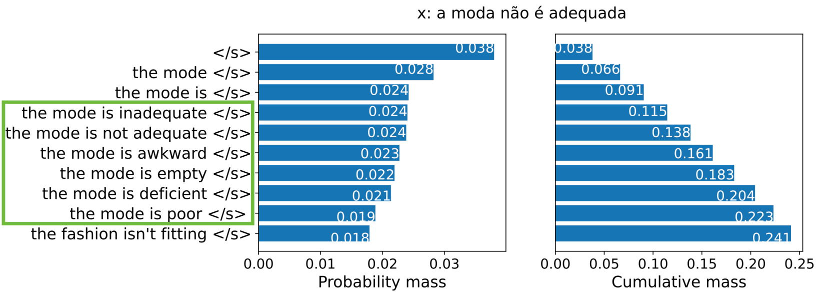
Outcomes Matter!

We have been neglecting the actual outcomes



Equivalence Classes

The fact that every single outcome is rare does not mean the distribution codes no useful knowledge.



For example, Ilia and Aziz [16] use an external classifier to form such a class.

Principles Recap

It's been a while... do you still remember the principles we outlined for decoding algorithms?

1. translations should be in some sense *preferred* by the model (else, what is the difference between using one model or another?);
2. translations are ideally good for their prompts, let's say they ought to be *adequate*;

We came up with a number of ways to operationalise (1), but, with the exception of some pressure against awkwardly short outcomes, we barely considered (2).

Quality Estimate

A 'quality estimate' $\mu(c; x)$ quantifies the goodness of fit of a candidate translation c to the prompt x.

Examples: CometKiwi [31]; average next-token surprisal (TP), average entropy (Softmax-Ent), inter alia [9].

Quality Estimate

A ‘quality estimate’ $\mu(c; x)$ quantifies the goodness of fit of a candidate translation c to the prompt x.

Examples: CometKiwi [31]; average next-token surprisal (TP), average entropy (Softmax-Ent), inter alia [9].

Outside MT, a function of this kind is better known as a *reward model*.

Re-Ranking

Quality-aware approaches [8] are re-rankers, which typically work like this:

1. enumerate a candidate set (e.g., using beam search or a sampler);
2. rank the candidates using a quality estimate (e.g., COMETKIWI);

$$y^{\text{decision}} = \text{argmax}_{c \in \mathcal{Y}} \; \mu(c; x) \tag{6}$$

Critical Eye

The problem here is that, unless we are very careful, we violate principle 1. As the candidate list grows, the quality estimate will render the model less and less relevant.

In practice, this appears to be of no importance, after all, we are unlikely to enumerate too many candidates anyway (it's a costly operation). But, how so?

- If we were sampling, small sample size means riskier decisions;
- If we were already optimising a robust criterion, then why bother with quality estimation?

Utility

Let's get back to quality of a translation, but we call it *utility*. Unlike quality, utility is a paired judgement.

We say that $u(c, y; x)$ quantifies the benefit in choosing c as the translation of x when y is known to be a plausible translation of it.

Examples: human judgement, ChrF [29], BLEURT [32], COMET [30], etc.

Utility

Let's get back to quality of a translation, but we call it *utility*. Unlike quality, utility is a paired judgement.

We say that $u(c, y; x)$ quantifies the benefit in choosing c as the translation of x when y is known to be a plausible translation of it.

Examples: human judgement, ChrF [29], BLEURT [32], COMET [30], etc.

Outside MT, the utility $u(c, y; x)$ is known as a *paired reward*.

Expected Utility

We can design a ‘quality estimate’ by combining our LM with a utility function $u(c, r; x)$ that compares a candidate translation c to a reference translation r.

In decoding, we do not have access to references, but in good probabilistic fashion, we can treat it as a *random variable* whose distribution our LM is assumed to predict from x.

We can then associate the merit of a candidate c with its *expected* utility under the model:

$$\mu_\theta(c; x) = \mathbb{E}_{p_\theta}[u(c, Y; x)] = \sum_{y \in \mathcal{Y}} p_\theta(y|x) u(c, y; x) \tag{7}$$

Expected Utility - Example

We derive a model-based notion of quality by computing a candidate's ChrF in expectation under the model (that is, using the model in place of a reference generator):

```
c                              y                                p(y|x)    u(c, y;x)    p(y|x) * u(c, y;x)
----------------------------   ------------------------------   --------  -----------  --------------------
</s>                           </s>                             0.0380        100.00                   3.80
                               the mode </s>                    0.0283         29.71                   0.84
                               the mode is </s>                 0.0242         24.93                   0.60
                               the mode is inadequate </s>      0.0240         13.84                   0.33
                               the mode is not adequate </s>    0.0238         13.25                   0.32
                               the mode is awkward </s>         0.0227         15.97                   0.36
                               the mode is empty </s>           0.0220         17.79                   0.39
                               the mode is deficient </s>       0.0214         14.48                   0.31
                               the mode is poor </s>            0.0189         18.87                   0.36
                               the fashion isn't fitting </s>   0.0179         12.21                   0.22
                               [...]
                               [SUM]                                                                  24.68

the mode isn't adequate </s>   </s>                             0.0380         37.93                   1.44
                               the mode </s>                    0.0283         58.62                   1.66
                               the mode is </s>                 0.0242         62.16                   1.51
                               the mode is inadequate </s>      0.0240         77.17                   1.85
                               the mode is not adequate </s>    0.0238         82.98                   1.98
                               the mode is awkward </s>         0.0227         45.80                   1.04
                               the mode is empty </s>           0.0220         49.20                   1.08
                               the mode is deficient </s>       0.0214         44.47                   0.95
                               the mode is poor </s>            0.0189         49.81                   0.94
                               the fashion isn't fitting </s>   0.0179         23.08                   0.41
                               [...]
                               [SUM]                                                                  36.18
```

Maximisation of Expected Utility

Under the assumption that expected utility

$$\mu_\theta(c;x) = \mathbb{E}_{p_\theta}[u(c,Y;x)] = \sum_{y\in\mathcal{Y}} p_\theta(y|x)u(c,y;x) \tag{8}$$

quantifies a reasonable notion of 'the quality of a candidate c in relation to a prompt x', we can use it for decision making:

$$y^{\text{MBR}} = \text{argmax}_{c\in\mathcal{Y}} \; \mu_\theta(c;x) \; . \tag{9}$$

This is known as minimum Bayes risk decoding [21].

Sampling-Based MBR

Eikema and Aziz [6] approximate expected utility using unbiased sampling

$$\mu_\theta(c; x) = \mathbb{E}_{p_\theta}[u(c, Y; x)] \overset{\text{MC}}{\approx} \frac{1}{S} \sum_{s=1}^{S} u(c, y^{(s)}; x) \quad \text{where } y^{(s)} \sim p_\theta(.|x) \tag{10}$$

Then they consider a reduced search space, made of N candidates $c^{(1)}, \ldots, c^{(N)}$ enumerated via sampling (unbiased, biased) and/or beam search.

Sampling-Based MBR Example

```
c                              y ~ p(.|x)                               u(c, y;x)
-----------------------------  -------------------------------------  -----------
</s>                           the mode is a mode </s>                      17.79
                               is </s>                                      58.01
                               uncool </s>                                  32.88
                               the mode is awkward </s>                     15.97
                               well I told you so didn't I ? </s>           12.21
                               fashionable </s>                             21.48
                               the is </s>                                  36.82
                               the mode is poor </s>                        18.87
                               mode is not cool </s>                        18.87
                               the mode is very probable </s>               12.71
                               rare rare rare rare ! </s>                   15.19
                               mode is a mode </s>                          21.48
                               I told you so didn't I ? </s>                14.48
                               nada nada </s>                               27.11
                               mode is not cool </s>                        18.87
                               the mode is inadequate </s>                  13.84
                               aren't adequate </s>                         17.79
                               sometimes NMT does strange things </s>        9.59
                               mode is weird </s>                           21.48
                               the fashion isn't fitting </s>               12.21
                               [AVG]                                        20.88

the mode isn't adequate </s>   mode </s>                                    41.02
                               nada nada nada nada </s>                     13.29
                               modes aren't adequate </s>                   69.07
                               the mode is a mode </s>                      55.00
                               what ? </s>                                  21.01
                               mode mode mode mode </s>                     28.24
                               the mode is actually rare </s>               42.80
                               the mode is a mode </s>                      55.00
                               modes aren't adequate </s>                   69.07
                               what ? </s>                                  21.01
                               the mode is poor </s>                        49.81
                               the mode is deficient </s>                   44.47
                               the the the the the the the </s>             17.52
                               the mode is </s>                             62.16
                               nada nada nada nada </s>                     13.29
                               mode is weird </s>                           35.96
                               is the </s>                                  35.67
                               weird mode </s>                              33.37
                               is </s>                                      25.28
                               is </s>                                      25.28
```

Critical Eye

MBR exploits similarity between responses to redistribute beliefs (can be thought of as a 'soft' way to form equivalence classes).

Less bias towards short translations, robustness to copying noise and hallucination [27]. Surprisal closer to that of references [25]. Improves substantially with modern neural utilities [10].

The search problem is formulated as re-ranking (expected utility does not bias the candidate set).

Learn to Search for MBR

To address the problem of *approximate, incremental search* for MBR, we have to address the problem of predicting *expected rewards* from incomplete responses [Monte Carlo Tree Search; 22].

Tomani et al. [35] formulated an approximation to this by training a model to perform quality estimation in addition to translation.

Origin Story

Consider the 'exact match' utility $u(c, y; x)$, which assigns 1 to c when it is identical to y.

It can be shown that

$$\mu_\theta(c; x) = \mathbb{E}_{p_\theta}[u(c, Y; x)] = p_\theta(c|x) \tag{11}$$

Origin Story

Consider the 'exact match' utility $u(c, y; x)$, which assigns 1 to c when it is identical to y.

It can be shown that

$$\mu_\theta(c; x) = \mathbb{E}_{p_\theta}[u(c, Y; x)] = p_\theta(c|x) \tag{11}$$

and hence

$$\text{argmax}_{y \in \mathcal{Y}} \;\; \mu_\theta(c; x) = \text{argmax}_{c \in \mathcal{Y}} \;\; p_\theta(c|x) \tag{12}$$

which is mode-seeking (MAP) decoding.

Origin Story

Consider the 'exact match' utility $u(c, y; x)$, which assigns 1 to c when it is identical to y.

It can be shown that

$$\mu_\theta(c; x) = \mathbb{E}_{p_\theta}[u(c, Y; x)] = p_\theta(c|x) \tag{11}$$

and hence

$$\text{argmax}_{y \in \mathcal{Y}} \ \mu_\theta(c; x) = \text{argmax}_{c \in \mathcal{Y}} \ p_\theta(c|x) \tag{12}$$

which is mode-seeking (MAP) decoding.

The mode is the MBR solution using an arguably poor (low coverage) notion of utility.

Summary

To obtain some form of 'guarantee' for the one response we want to regard as 'the translation' of x, we turned away from sampling and towards *decision rules*.

The most probable translation (MAP decoding) ruled supreme for years, despite piling evidence against it.

Re-ranking enables the use of complex reward models, but at the expense of integration with the underlying MT model.

To meet both principles (that the output should be informed by the model and adequate) we can combine our model and a (paired) reward model, deriving MBR decoding.

MBR decoding is a class of objectives, and provides a strong rationale against MAP decoding.

Modern Decoding, as I see it

Modern Training has Just Too Many Ingredients

Modern training is a rather heterogenous combination of ideas:

1. we pretrain on ‘all-we-can-eat’ data

Modern Training has Just Too Many Ingredients

Modern training is a rather heterogenous combination of ideas:

1. we pretrain on 'all-we-can-eat' data
2. we then train on translation data, but also many other tasks

Modern Training has Just Too Many Ingredients

Modern training is a rather heterogenous combination of ideas:

1. we pretrain on 'all-we-can-eat' data
2. we then train on translation data, but also many other tasks
3. we use a lot of synthetic data
   for example, to learn certain 'skills' (like in-context learning or chain-of-thought reasoning)

Modern Training has Just Too Many Ingredients

Modern training is a rather heterogenous combination of ideas:

1. we pretrain on 'all-we-can-eat' data
2. we then train on translation data, but also many other tasks
3. we use a lot of synthetic data
   for example, to learn certain 'skills' (like in-context learning or chain-of-thought reasoning)
4. we learn from preference data (using RLHF or DPO, or whatnot)

In some of these steps we rely on 'samples' (e.g., 3 and 4), but this usually means biased samples with heterogenous (possibly undisclosed) hyperparameters.

The Probabilistic View

It’s getting hard to insist in ‘coherence with a certain probabilistic view’ of the model, because this view is itself losing coherence.

That is okay, all this means is that ‘the principled choice’ argument, which was already weak, is now practically void of meaning.

This is good, it forces us to seek stronger rationales for our choices.

General Advice

If I am pressed to choose, here are some of my choices

- seek to establish equivalence classes (that is, exploit the fact that outcomes aren't linguistically unrelated to one another [20])
- or to, at least, incorporate similarity in scoring (e.g., softly like MBR and others [3] do)
- use a sampler to parameterise a decision rule
  but realise that due to heterogenous training, no sampler is privileged (we need to validate their properties in each model/data setting [11])

Open Problems

Efficient ways to search with non-factorised objectives (we mostly use re-ranking-type algorithms because it's hard to search efficiently, but re-ranking isn't that efficient either).

Decision rules for long-form generation (samplers claim most territory because we lack good decision rules).

Thanks!

References

[1] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[2] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Audio chord recognition with recurrent neural networks. In *ISMIR*, pages 335–340. Curitiba, 2013.

[3] Julius Cheng and Andreas Vlachos. Measuring uncertainty in neural machine translation with similarity-sensitive entropy. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2115–2128, St. Julian's, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.129. URL `https://aclanthology.org/2024.eacl-long.129/`.

[4] Li Du, Holden Lee, Jason Eisner, and Ryan Cotterell. When is a language process a language model? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11083–11094, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.659. URL `https://aclanthology.org/2024.findings-acl.659/`.

[5] Bryan Eikema and Wilker Aziz. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online), December 2020. International Committee on Computational

Linguistics. doi: 10.18653/v1/2020.coling-main.398. URL https://aclanthology.org/2020.coling-main.398/.

[6] Bryan Eikema and Wilker Aziz. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10978–10993, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.754. URL https://aclanthology.org/2022.emnlp-main.754/.

[7] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL https://aclanthology.org/P19-1346/.

[8] Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. Quality-aware decoding for neural machine translation. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, pages 1396–1412, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.100. URL https://aclanthology.org/2022.naacl-main.100/.

[9] Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555, 2020. doi: 10.1162/tacl_a_00330. URL https://aclanthology.org/2020.tacl-1.35/.

[10] Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825, 2022. doi: 10.1162/tacl_a_00491. URL `https://aclanthology.org/2022.tacl-1.47/`.

[11] Mario Giulianelli, Joris Baan, Wilker Aziz, Raquel Fernández, and Barbara Plank. What comes next? evaluating uncertainty in neural text generators against human production variability. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14349–14371, Singapore, December 2023. Association for Computational Linguistics. doi:

10.18653/v1/2023.emnlp-main.887. URL https://aclanthology.org/2023.emnlp-main.887/.

[12] Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. Improved neural machine translation with smt features. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[13] John Hewitt, Christopher Manning, and Percy Liang. Truncation sampling as language model desmoothing. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.249. URL https://aclanthology.org/2022.findings-emnlp.249/.

[14] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rygGQyrFvH`.

[15] Liang Huang, Kai Zhao, and Mingbo Ma. When to finish? optimal beam search for neural text generation (modulo beam size). In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2134–2139, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1227. URL `https://aclanthology.org/D17-1227/`.

[16] Evgenia Ilia and Wilker Aziz. Variability need not imply error: The case of adequate but semantically distinct responses. *arXiv preprint arXiv:2412.15683*, 2024.

[17] Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. Montreal neural machine translation systems for WMT'15. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors, *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3014. URL `https://aclanthology.org/W15-3014/`.

[18] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In Thang Luong, Alexandra Birch, Graham Neubig, and Andrew Finch, editors, *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3204. URL `https://aclanthology.org/W17-3204/`.

[19] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[20] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=VD-AYtP0dve`.

[21] Shankar Kumar and William Byrne. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics. URL `https://aclanthology.org/N04-1022/`.

[22] Rémi Leblond, Jean-Baptiste Alayrac, Laurent Sifre, Miruna Pislar, Lespiau Jean-Baptiste, Ioannis Antonoglou, Karen Simonyan, and Oriol Vinyals. Machine translation decoding beyond beam search. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language*

Processing, pages 8410–8434, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.662. URL https://aclanthology.org/2021.emnlp-main.662/.

[23] Clara Meister, Ryan Cotterell, and Tim Vieira. If beam search is the answer, what was the question? In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.170. URL https://aclanthology.org/2020.emnlp-main.170/.

[24] Clara Meister, Tim Vieira, and Ryan Cotterell. Best-first beam search. *Transactions of the Association for Computational Linguistics*, 8:795–809, 2020. doi: 10.1162/tacl_a_00346. URL https://aclanthology.org/2020.tacl-1.51/.

[25] Clara Meister, Gian Wiher, Tiago Pimentel, and Ryan Cotterell. On the probability–quality paradox in language generation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 36–45, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.5. URL https://aclanthology.org/2022.acl-short.5/.

[26] Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121, 2023. doi: 10.1162/tacl_a_00536. URL https://aclanthology.org/2023.tacl-1.7/.

[27] Mathias Müller and Rico Sennrich. Understanding the properties of minimum Bayes risk decoding in neural machine translation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 259–272, Online, August 2021. Association for Computational Linguistics. doi:

10.18653/v1/2021.acl-long.22. URL
https://aclanthology.org/2021.acl-long.22/.

[28] Kenton Murray and David Chiang. Correcting length bias in neural machine translation. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6322. URL https://aclanthology.org/W18-6322/.

[29] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors, *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL `https://aclanthology.org/W15-3049/`.

[30] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. Association for Computational Linguistics. doi:

10.18653/v1/2020.emnlp-main.213. URL https://aclanthology.org/2020.emnlp-main.213/.

[31] Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In Philipp Koehn, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri, editors,

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.wmt-1.60/`.

[32] Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.704. URL `https://aclanthology.org/2020.acl-main.704/`.

[33] Pavel Sountsov and Sunita Sarawagi. Length bias in encoder decoder models and a case for global conditioning. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1525, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1158. URL https://aclanthology.org/D16-1158/.

[34] Felix Stahlberg and Bill Byrne. On NMT search errors and model errors: Cat got your tongue? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362,

Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1331. URL `https://aclanthology.org/D19-1331/`.

[35] Christian Tomani, David Vilar, Markus Freitag, Colin Cherry, Subhajit Naskar, Mara Finkelstein, Xavier Garcia, and Daniel Cremers. Quality-aware translation models: Efficient generation and quality estimation in a single model. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15660–15679, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.836. URL `https://aclanthology.org/2024.acl-long.836/`.

[36] Yilin Yang, Liang Huang, and Mingbo Ma. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1342. URL `https://aclanthology.org/D18-1342/`.