A Distribution-Aware Decision Rule for Neural Machine Translation

Bryan Eikema   Wilker Aziz

University of Amsterdam

Neural Machine Translation (NMT)

NMT is trained as a **probabilistic model**:

- Given a sentence x, an NN predicts a distribution over its possible translations

$$Y \mid \theta, x$$

Neural Machine Translation (NMT)

NMT is trained as a **probabilistic model**:

- Given a sentence x, an NN predicts a distribution over its possible translations

$$Y \mid \theta, x$$

- This distribution is factorized into **locally normalized** Categorical distributions

$$Y_j \mid \theta, x, y_{<j} \sim \mathrm{Cat}(\mathrm{NN}(x, y_{<j}; \theta))$$

Neural Machine Translation (NMT)

NMT is trained as a **probabilistic model**:

- Given a sentence x, an NN predicts a distribution over its possible translations

$$Y \mid \theta, x$$

- This distribution is factorized into **locally normalized** Categorical distributions

$$Y_j \mid \theta, x, y_{<j} \sim \mathrm{Cat}(\mathrm{NN}(x, y_{<j}; \theta))$$

- And its parameters are chosen via **maximum likelihood estimation** (MLE)

$$\theta_{MLE} = \mathrm{argmax}_\theta \sum_{(x,y)} \log P(y|x,\theta)$$

Decision Rules in NMT

At test-time we need to map from a probability distribution to a single ‘preferred’ translation, this requires a **decision rule**.

In NMT, the most commonly employed decision rule is **maximum-a-posteriori (MAP) decoding**.

MAP Decoding

MAP predicts the translation that has maximum probability under the model

$$y^{\mathrm{mode}} = \mathrm{argmax}_y \log P(y|x, \theta_{\mathrm{MLE}})$$

that is, the **mode** of the model distribution.

MAP Decoding

MAP predicts the translation that has maximum probability under the model

$$y^{mode} = \mathrm{argmax}_y \log P(y|x, \theta_{MLE})$$

that is, the **mode** of the model distribution.

Finding the exact mode is **intractable**, so we use an approximation: **beam search.** Larger beams approximate the MAP objective better.

Pathologies and Biases of NMT

- Length bias
- Beam search curse
- Empty mode
- Word frequency bias
- Susceptibility to copy noise
- Hallucination under domain shift

Many works blame NMT as a **model or its training algorithm**

But note: all these observations are using **approximate MAP decoding**

[Sountsov and Sarawagi, 2016; Huang et al ,2017; Koehn and Knowles, 2017; Murray and Chiang, 2018; Ott et al., 2018; Khayrallah and Koehn, 2018; Kumar and Sarawagi, 2019; Stahlberg and Byrne, 2019; Müller et al., 2020]

Biased Statistics & The Inadequacy of the Mode

We use the mode for **model criticism**, but:

- The mode is **no unbiased statistic** of the learnt distribution
    - e.g. a short mode does not imply that the model underestimates average sequence length!

Biased Statistics & The Inadequacy of the Mode

We use the mode for **model criticism**, but:

- The mode is **no unbiased statistic** of the learnt distribution
  - e.g. a short mode does not imply that the model underestimates average sequence length!

We target the mode for **making predictions**, but:

- The mode could still be a **very rare outcome**
- Focusing on the mode alone **throws away a lot of valuable information** learnt by the model

Biased Statistics & The Inadequacy of the Mode

We use the mode for **model criticism**, but:

- The mode is **no unbiased statistic** of the learnt distribution
  - e.g. a short mode does not imply that the model underestimates average sequence length!

We target the mode for **making predictions**, but:

- The mode could still be a **very rare outcome**
- Focusing on the mode alone **throws away a lot of valuable information** learnt by the model

A common misconception is that MAP is the only logical choice for an MLE-trained model.

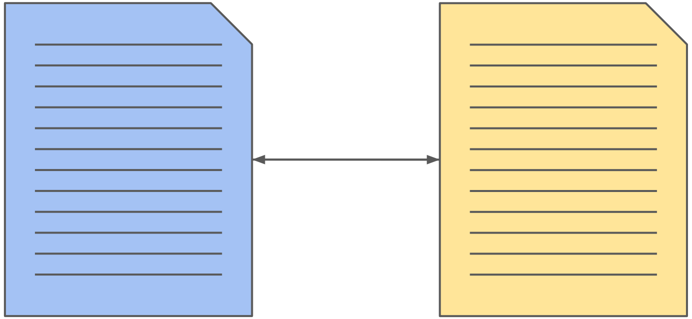
Experiments

We will be answering:

1. Does the NMT model fit the data well?
2. What do the learnt distributions look like?
3. Can we make predictions using all of the information available?

Experiments

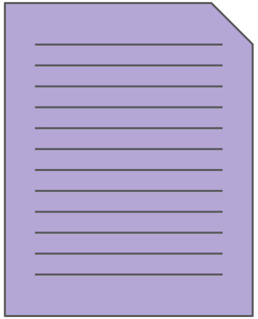
Train on:



English

German (5.9M)
Nepali (573k)
Sinhala (235k)

Test on:



newstest2018
Flores
Flores

Model:

Assessing Data Fit

Unbiased Samples

Generative story $\quad Y_j \mid \theta, x, y_{<j} \sim \mathrm{Cat}(\mathrm{NN}(x, y_{<j}; \theta))$

Unbiased Samples

Generative story $Y_j \mid \theta, x, y_{<j} \sim \mathrm{Cat}(\mathrm{NN}(x, y_{<j}; \theta))$

Ancestral sampling

1. Start with empty string, predict a Categorical distribution in context draw the next word following the predicted distribution $y_{<1} = \text{“<s>”}$

$$Y_1 \mid \theta, x, \text{“<s>”} \sim \mathrm{Cat}(\mathrm{NN}(x, \text{“<s>”}; \theta))$$

Unbiased Samples

Generative story $Y_j \mid \theta, x, y_{<j} \sim \mathrm{Cat}(\mathrm{NN}(x, y_{<j}; \theta))$

Ancestral sampling

1. Start with empty string, predict a Categorical distribution in context draw the next word following the predicted distribution $y_{<1} = \text{"<s>"}$

$$Y_1 \mid \theta, x, \text{"<s>"} \sim \mathrm{Cat}(\mathrm{NN}(x, \text{"<s>"}; \theta))$$

2. Extend the context with sampled outcome repeat until the end-of-sentence symbol is drawn $y_{<2} = \text{"<s> } y_1\text{"}$

$$Y_2 \mid \theta, x, \text{"<s> } y_1\text{"} \sim \mathrm{Cat}(\mathrm{NN}(x, \text{"<s> } y_1\text{"}; \theta))$$

Unbiased Samples

Generative story $\quad Y_j \mid \theta, x, y_{<j} \sim \text{Cat}(\text{NN}(x, y_{<j}; \theta))$

Ancestral sampling

1. Start with empty string, predict a Categorical distribution in context draw the next word following the predicted distribution $\quad y_{<1} = \text{"<s>"}$

$$Y_1 \mid \theta, x, \text{"<s>"} \sim \text{Cat}(\text{NN}(x, \text{"<s>"}; \theta))$$

2. Extend the context with sampled outcome repeat until the end-of-sentence symbol is drawn $\quad y_{<2} = \text{"<s> } y_1\text{"}$

$$Y_2 \mid \theta, x, \text{"<s> } y_1\text{"} \sim \text{Cat}(\text{NN}(x, \text{"<s> } y_1\text{"}; \theta))$$

A summary of the model's beliefs that is not biased towards an external criterion

Assessing Data Fit: Methodology

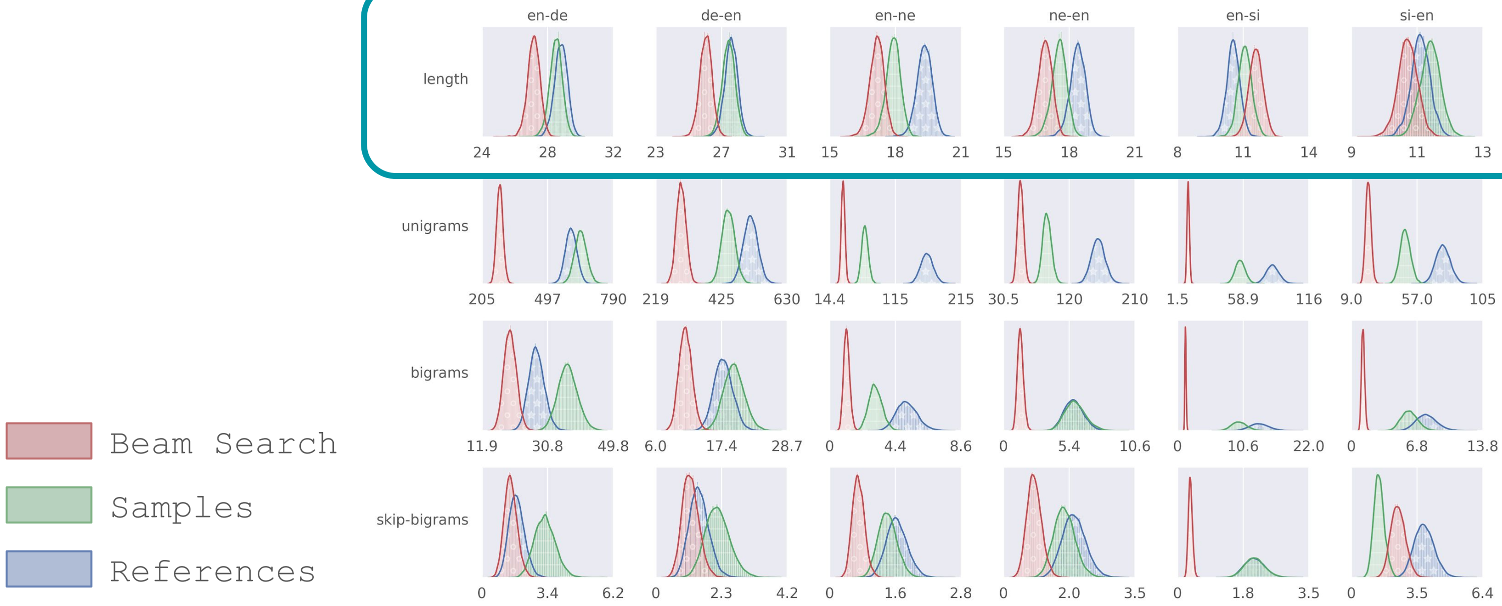
1. Gather statistics from **human references**, **unbiased samples**, and **beam search** outputs
2. Model all data in a hierarchical Bayesian model
3. Compare posteriors between human references and model outputs

We compare:

- Length
- Lexical properties: unigram and bigram counts
- Word order: skip-bigram counts

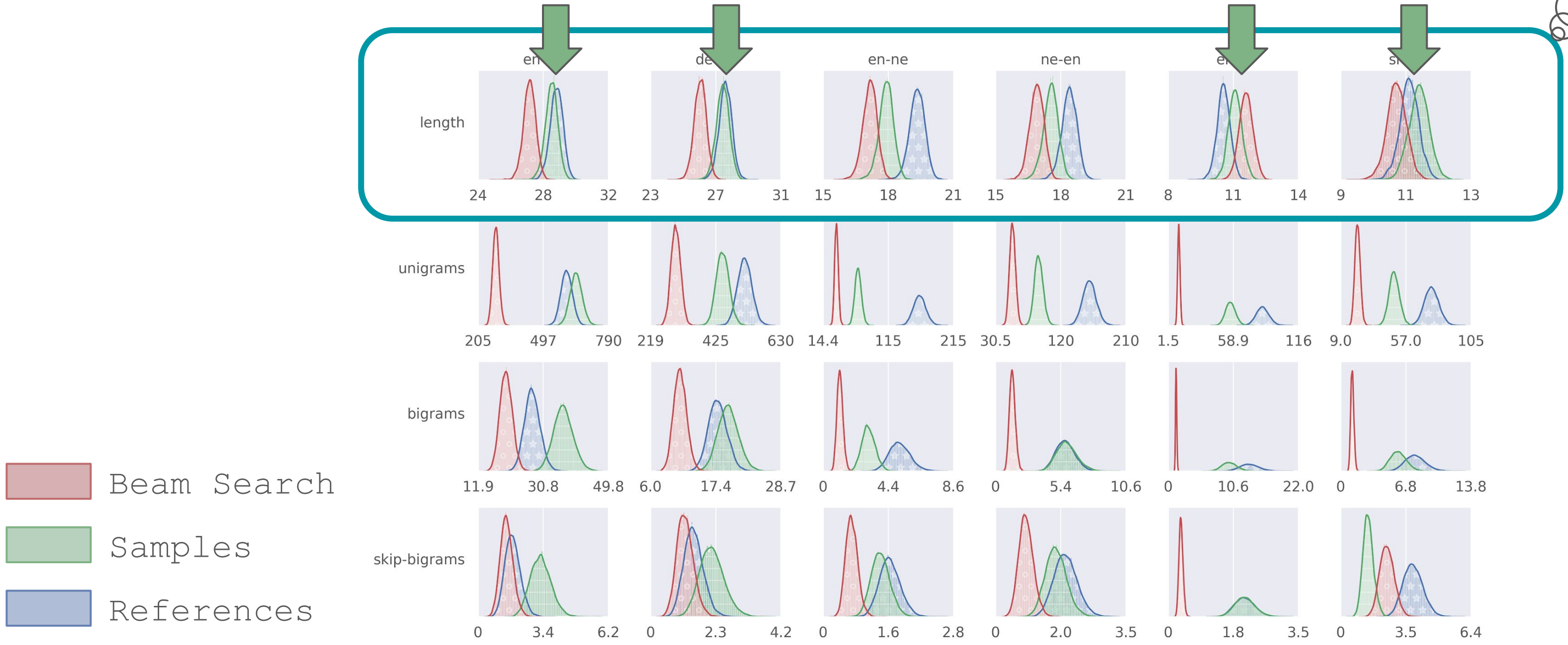
Assessing Data Fit: Length

x-axis shows
“average length”



Assessing Data Fit: Length

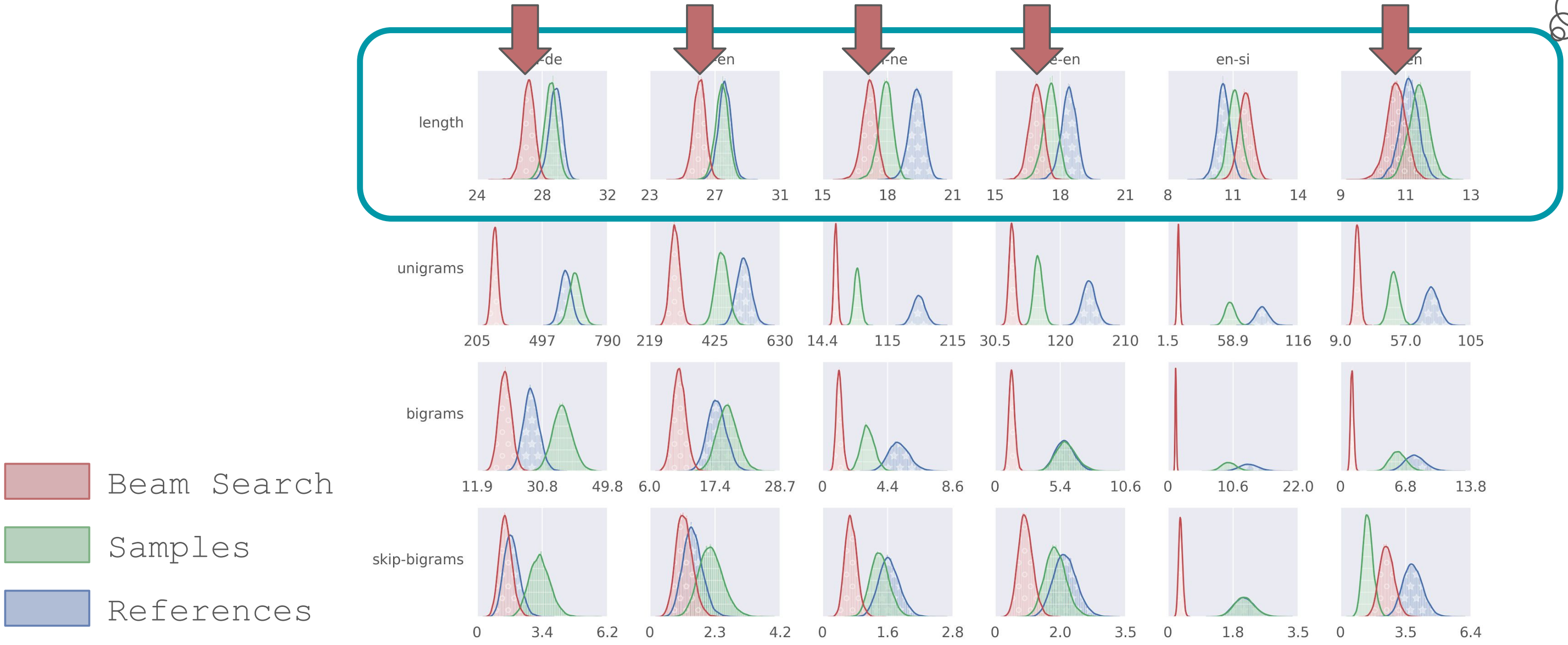
x-axis shows
“average length”



In most cases the **model captures length reasonably well**

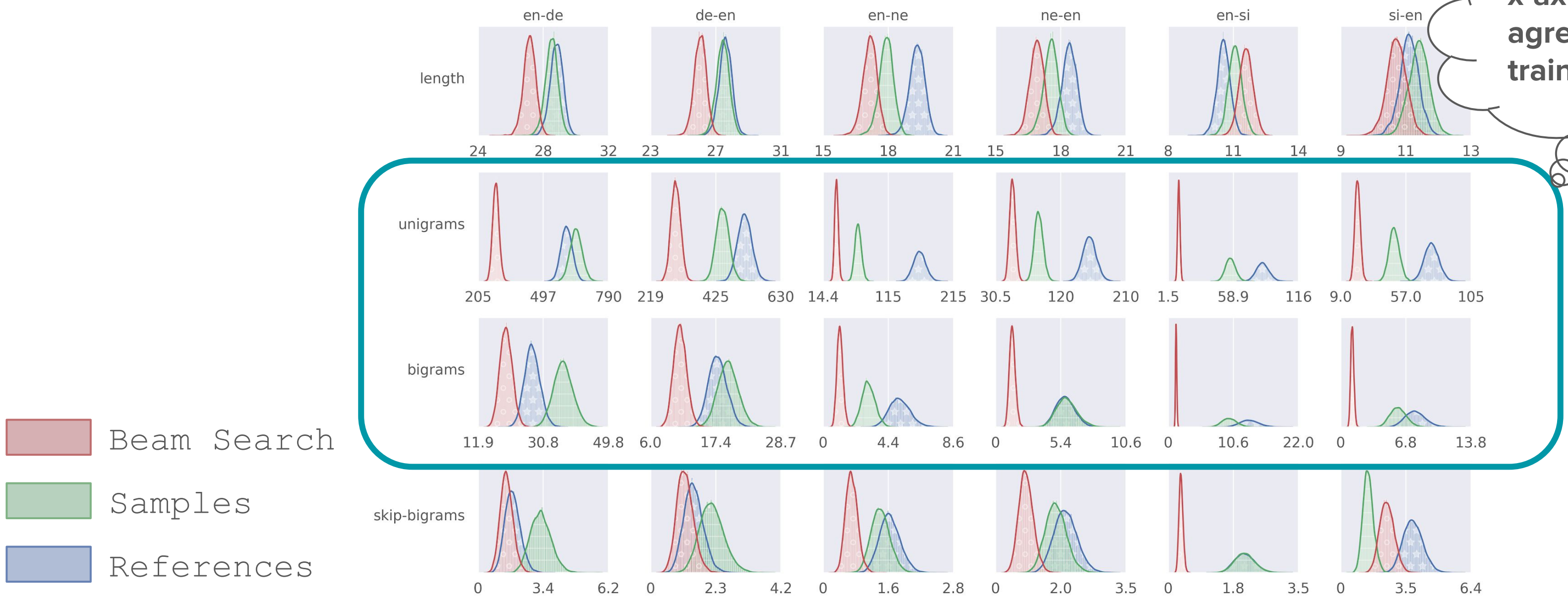
Assessing Data Fit: Length

x-axis shows "average length"



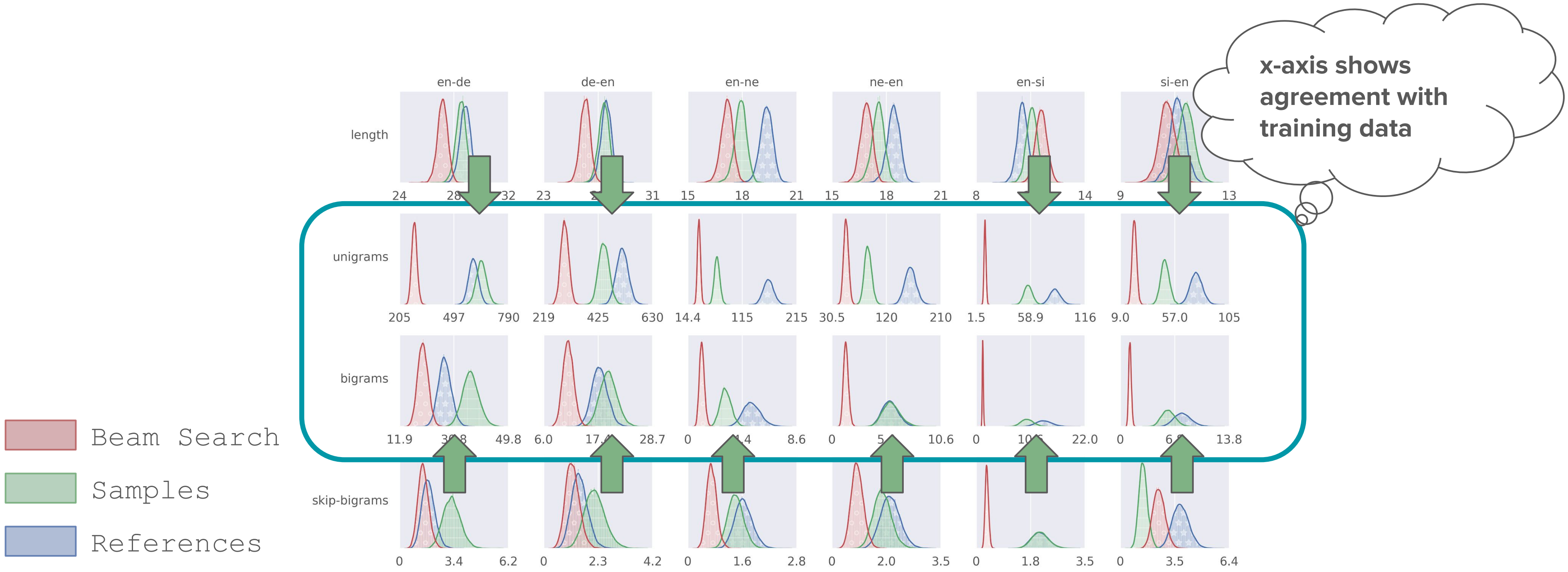
Beam search shifts from data statistics, underestimating length

Assessing Data Fit: Lexical Statistics



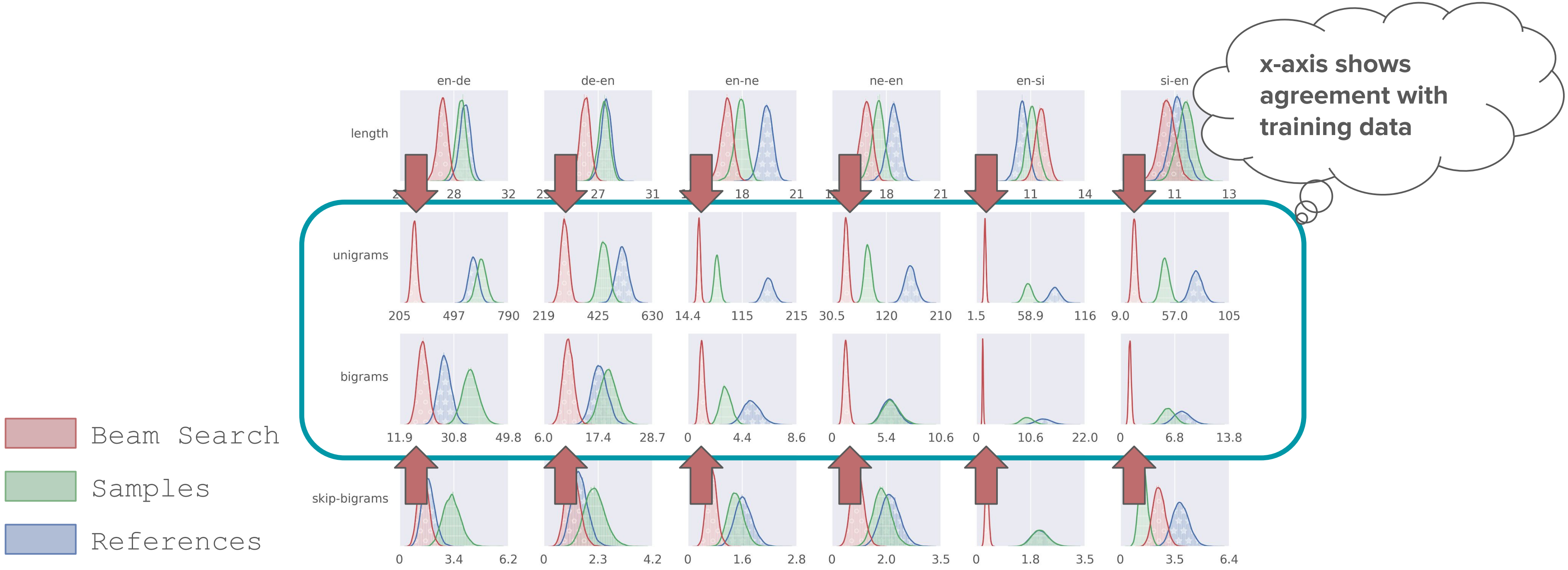
x-axis shows agreement with training data

Assessing Data Fit: Lexical Statistics



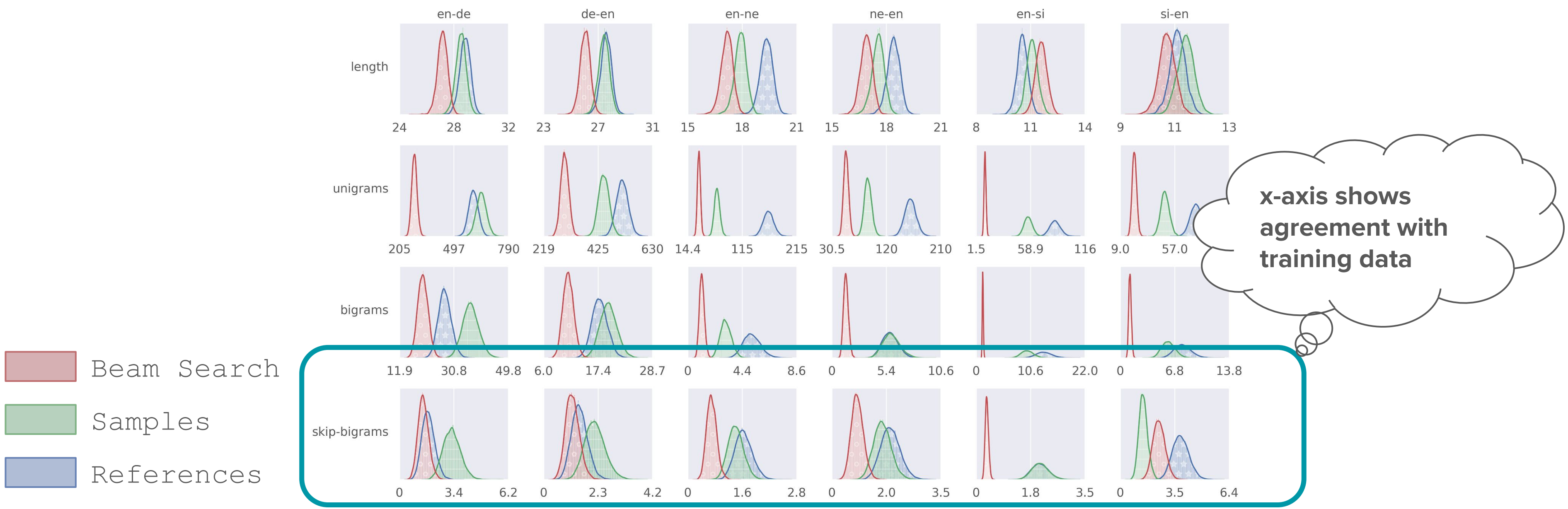
In most cases the **model captures lexical statistics reasonably well**

Assessing Data Fit: Lexical Statistics

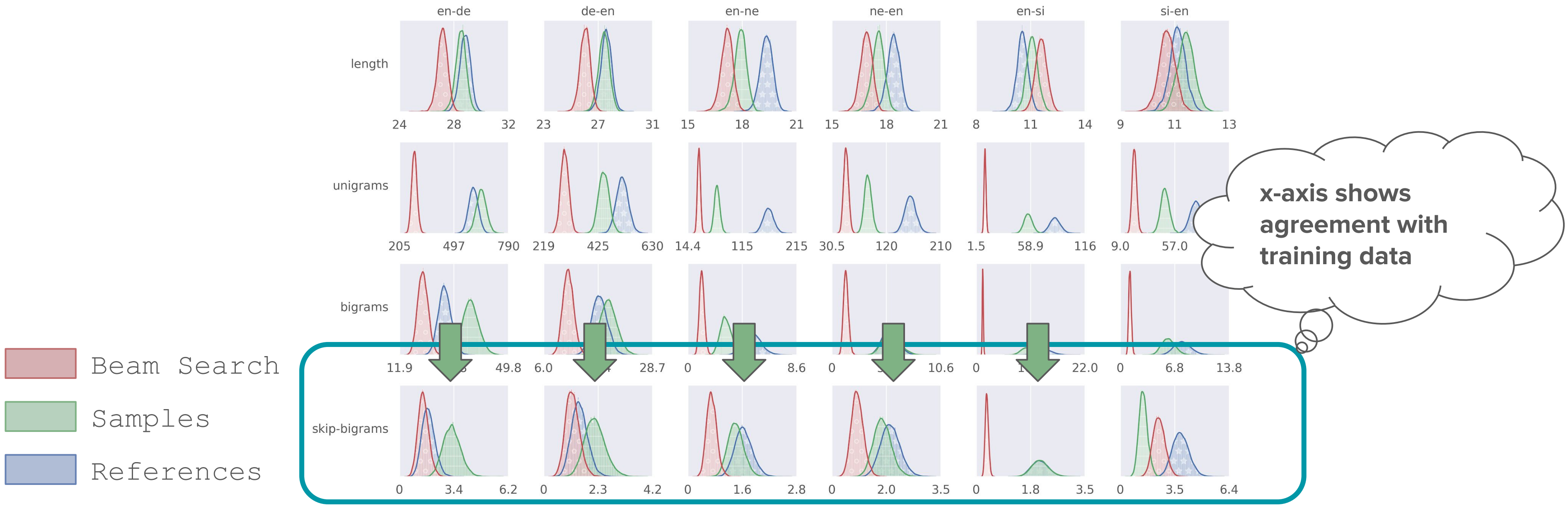


Beam search shifts from data statistics, changing lexical characteristics

Assessing Data Fit: Word Order

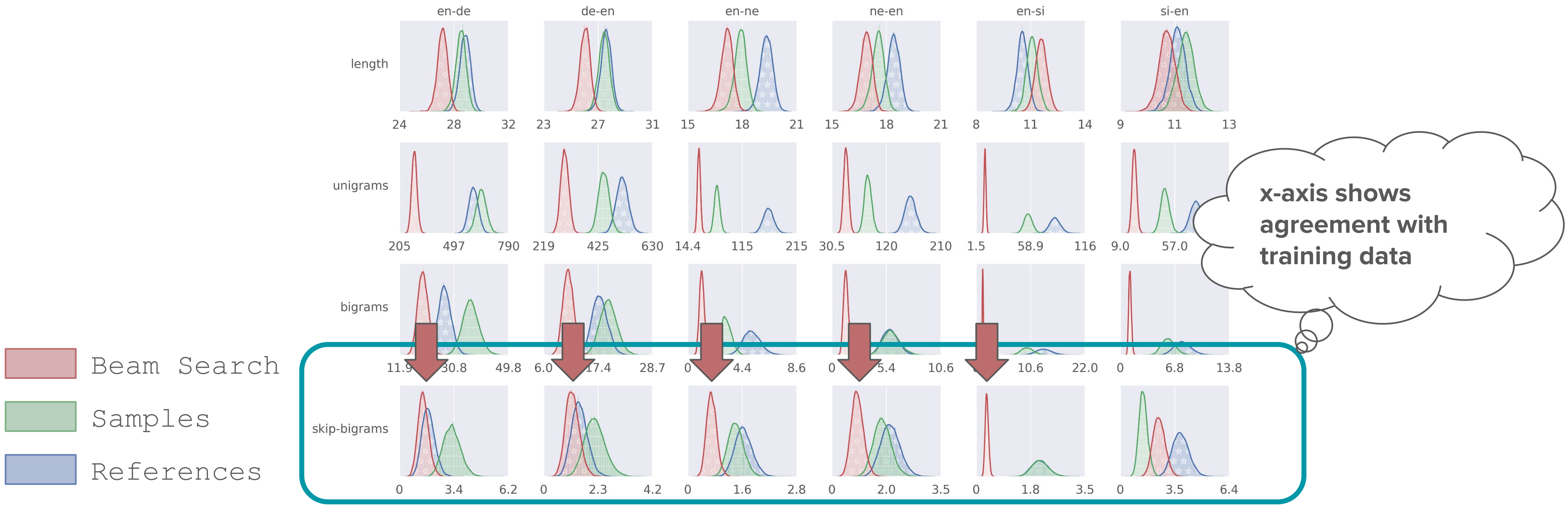


Assessing Data Fit: Word Order



In most cases the **model captures word order statistics reasonably well**

Assessing Data Fit: Word Order



Beam search shifts from data statistics, affecting word order

Q1: Does the NMT model fit the data well?

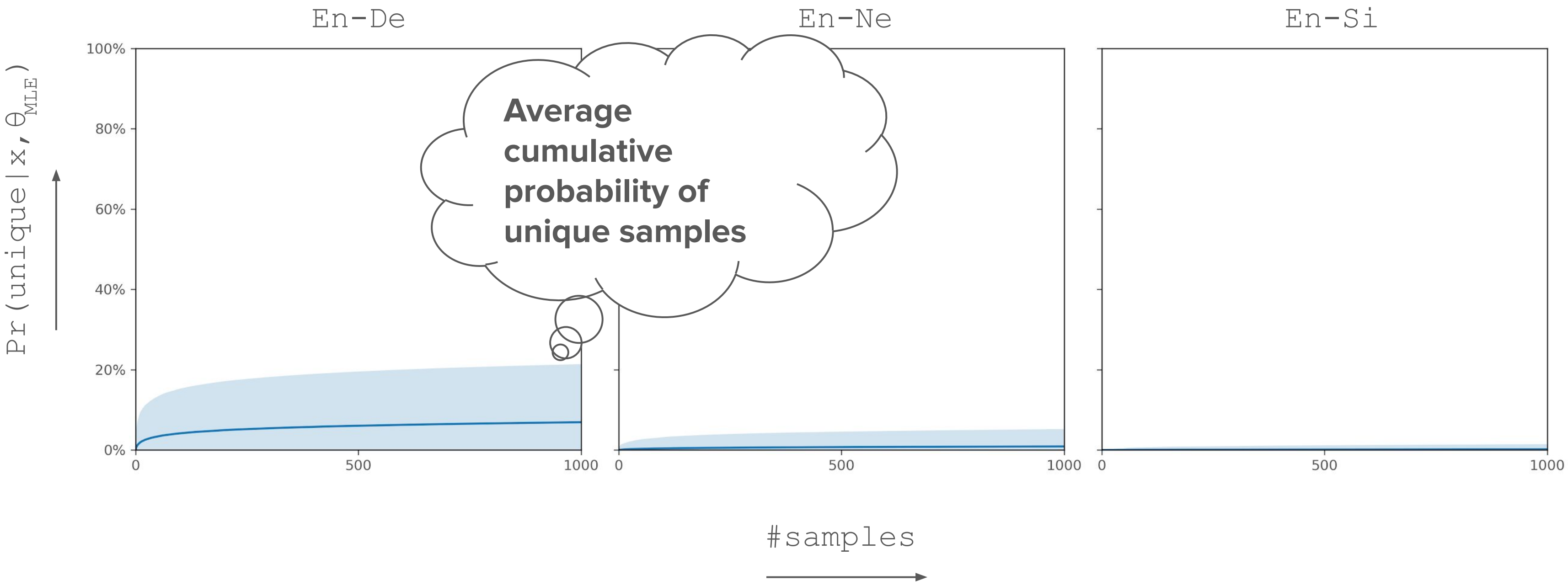
Beam search **shifts** the distribution of statistics such as length, unigram/bigram, and skip-bigram counts away from human references.

Unbiased samples better reproduce those statistics.

The model fits the data better than beam search outputs would have suggested.

Properties of Translation Distributions

Spread of the Translation Distribution



NMT **spreads mass** over many translations

Sampling the Mode

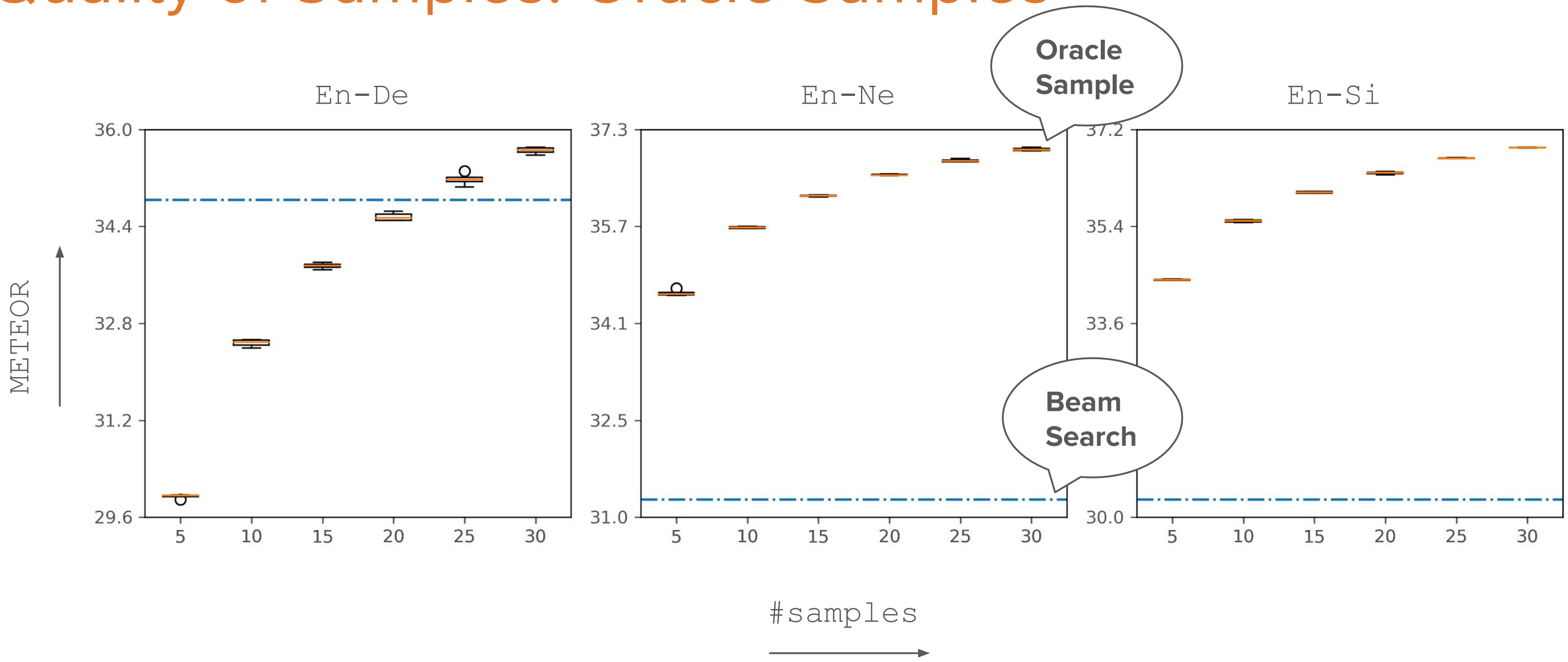
Beam search:

For most input sequences, the beam search output was **not drawn after 1,000 samples** (>50% high-resource, >90% low-resource)

Empty Sequence:

In fewer than 35% of input sequences the empty string is drawn, but if drawn it **only occurs roughly once in 1,000 samples**

Quality of Samples: Oracle Samples



A **small number of samples** contains **good translations**

Q2: What do translation distributions look like?

They are not particularly peaked:

- That is, they do not show a clear preference for any of the translations in a large set (as large as 1000 samples)
- The situation is worse in low-resource settings

They do not emphasize the mode nor the empty sequence *(MAP decoding does)*.

Yet, they support translations of reasonable quality

- A few unbiased samples is enough to come across good translations

A Distribution-Aware Decoding Algorithm

Minimum Bayes Risk (MBR) Decoding

$$y^{\mathrm{MBR}} = \mathrm{argmax}_{h}\ E_{Y|x,\theta}[U(h,\ Y)]$$

Minimum Bayes Risk (MBR) Decoding

$$y^{MBR} = \text{argmax}_h \ E_{Y|x,\theta}[\mathbf{U}(h, Y)]$$

- Find hypothesis h that maximises utility **U**, e.g. METEOR

Minimum Bayes Risk (MBR) Decoding

$$\mathtt{y}^{\mathtt{MBR}} = \mathtt{argmax}_{\mathtt{h}}\ \mathtt{E}_{\mathtt{Y|x,\theta}}[\mathtt{U(h,}\ \mathbf{Y}\mathtt{)}]$$

- Find hypothesis `h` that maximises utility `U`, e.g. METEOR
- But we don't have access to the reference

Minimum Bayes Risk (MBR) Decoding

$$y^{\mathrm{MBR}} = \mathrm{argmax}_h \; \mathbf{E}_{\mathbf{Y}|\mathbf{x},\theta}[U(h, \mathbf{Y})]$$

- Find hypothesis h that maximises utility U, e.g. METEOR
- But we don't have access to the reference
- Use the translation distribution to **fill in the reference** using $\mathbf{Y}|\mathbf{x},\theta$

Minimum Bayes Risk (MBR) Decoding

$$y^{MBR} = \mathbf{argmax}_{\mathbf{h}} \; E_{Y|x,\theta}[U(\mathbf{h}, Y)]$$

- Find hypothesis h that maximises utility U, e.g. METEOR
- But we don’t have access to the reference
- Use the translation distribution to fill in the reference using $Y|x,\theta$
- Pick the hypothesis $\mathbf{h}$ with **highest expected utility**

Minimum Bayes Risk (MBR) Decoding

$$y^{MBR} = \text{argmax}_h \; E_{Y|x,\theta}[U(h, Y)]$$

- Find hypothesis h that maximises utility U, e.g. METEOR
- But we don't have access to the reference
- Use the translation distribution to fill in the reference using $Y|x,\theta$
- Pick the hypothesis h with highest expected utility

Properties:

- Makes use of the translation distribution as a whole
- We can approximate it using unbiased samples
- Doesn't typically suffer from idiosyncratic translations

Approximate MBR with Unbiased Samples

Given input x, trained model $Y|x,\theta_{MLE}$, utility U, and sample size S

$$y^{MBR} = \operatorname{argmax}_{h \in H} 1/S \sum_s U(h, y^{(s)})$$

Approximate MBR with Unbiased Samples

Given input x, trained model $Y|x, \theta_{MLE}$, utility U, and sample size S

$$y^{MBR} = \text{argmax}_{h \in H}\ 1/S \sum_s U(h, \mathbf{y}^{(s)})$$

1. Sample S unbiased samples: $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(S)} \sim Y|x, \theta_{MLE}$

Approximate MBR with Unbiased Samples

Given input x, trained model $Y|x, \theta_{MLE}$, utility U, and sample size S

$$y^{MBR} = \operatorname{argmax}_{h \in \mathbf{H}} 1/S \sum_s U(h, y^{(s)})$$

1. Sample S unbiased samples: $y^{(1)},\ldots,y^{(S)} \sim Y|x, \theta_{MLE}$
2. Use samples as hypotheses as well: $\mathbf{H} = \text{unique}(y^{(1)},\ldots,y^{(S)})$

Approximate MBR with Unbiased Samples

Given input x, trained model $Y|x, \theta_{MLE}$, utility U, and sample size S

$$y^{MBR} = \text{argmax}_{h \in H}\ 1/S \sum_s \mathbf{U(h,\ y^{(s)})}$$

1. Sample S unbiased samples: $y^{(1)},\ldots,y^{(S)} \sim Y|x,\ \theta_{MLE}$
2. Use samples as hypotheses as well: $H = \text{unique}(y^{(1)},\ldots,y^{(S)})$
3. **Compute a matrix of utilities** between all pairs of hypotheses and samples

Approximate MBR with Unbiased Samples

Given input x, trained model $\mathtt{Y|x,\theta_{MLE}}$, utility $\mathtt{U}$, and sample size $\mathtt{S}$

$$\mathtt{y^{MBR} = argmax_{h \in H}}\ \mathbf{1/S}\ \boldsymbol{\Sigma_s}\ \mathtt{U(h,\ y^{(s)})}$$

1. Sample $\mathtt{S}$ unbiased samples: $\mathtt{y^{(1)},\ldots,y^{(S)} \sim Y|x,\ \theta_{MLE}}$
2. Use samples as hypotheses as well: $\mathtt{H = unique(y^{(1)},\ldots,y^{(S)})}$
3. Compute a matrix of utilities between all pairs of hypotheses and samples
4. Compute the **sample average** utility for each hypothesis

Approximate MBR with Unbiased Samples

Given input x, trained model $Y|x,\theta_{MLE}$, utility U, and sample size S

$$y^{MBR} = \mathbf{argmax}_{h \in H}\ 1/S \sum_s U(h,\ y^{(s)})$$

1. Sample S unbiased samples: $y^{(1)},\ldots,y^{(S)} \sim Y|x,\ \theta_{MLE}$
2. Use samples as hypotheses as well: $H = \mathrm{unique}(y^{(1)},\ldots,y^{(S)})$
3. Compute a matrix of utilities between all pairs of hypotheses and samples
4. Compute the sample average utility for each hypothesis
5. Pick the hypothesis with **highest average utility**

Results of MBR Decoding

Using 30 samples:

	Beam Search	MBR Decoding	Oracle Decoding
High-Resource	**37.1**	34.4	38.3
Low-Resource	24.3	**26.0**	28.9
All	28.6	**28.8**	32.0

Results of MBR Decoding

Using 30 samples:

	Beam Search	MBR Decoding	Oracle Decoding
High-Resource	**37.1**	34.4	38.3
Low-Resource	24.3	**26.0**	28.9
All	28.6	**28.8**	32.0

Beam search outperforms MBR (30) in high-resource setting

Results of MBR Decoding

Using 30 samples:

	Beam Search	MBR Decoding	Oracle Decoding
High-Resource	**37.1**	34.4	38.3
Low-Resource	24.3	**26.0**	28.9
All	28.6	**28.8**	32.0

MBR decoding outperforms beam search in low-resource settings

Results of MBR Decoding

Using 30 samples:

	Beam Search	MBR Decoding	Oracle Decoding
High-Resource	**37.1**	34.4	38.3
Low-Resource	24.3	**26.0**	28.9
All	28.6	**28.8**	32.0

The gap with oracle decoding shows there is a lot of room for improvement

Q3: Can we predict using all information available?

Sure thing

- A few unbiased samples already offer a hypothesis space with great potential.
- Expected utility allows us to select a hypothesis taking into account the translation distribution under the lens of a utility function.

Summary

MAP decoding is not suitable as a decision rule in NMT

MAP decoding **introduces biases** to NMT

Translation distributions **do capture data statistics well**

Distribution-aware decision rules show great potential

Recent Progress

Understanding the Properties of Minimum Bayes Risk Decoding in Neural Machine Translation (Müller and Sennrich, 2021)

Investigate the properties of sampling-based MBR translations

In particular they find:

- MBR to have less word frequency and length bias, but still exhibits some, varying with the utility used
- MBR to be more robust to copy noise in the training data
- MBR to have higher domain robustness, producing fewer hallucinated content
- MBR to empirically not have an equivalent of the beam search curse

[Müller and Sennrich, 2021]

High Quality Rather than High Model Probability: Minimum Bayes Risk Decoding with Neural Metrics (Freitag et al., 2022)

- Explore a number of utilities for sampling-based MBR, most interestingly: BLEURT, a neural automatic evaluation metric
- Scale to large numbers of samples (S=1000)
- Shows MBR with BLEURT:
  - To produce lower probability sequences than beam search and MBR with surface-level utilities.
  - Have lower surface-level metric scores (e.g. BLEU) than beam search (but have higher BLEURT scores)
  - Perform signficantly better than beam search in a human evaluation.

[Freitag et al., 2022]

Other Works

- Using expected utility as analysis tool: Amrhein and Sennrich, 2022
- Linking the inadequacy of the mode to task complexity: Forster et al., 2021, Stahlberg et al., 2022
- Looking at the exact decision rule corresponding to beam search with a small beam: Meister et al., 2021
- Linking the beam search curse to expected information / typicality: Meister et al., 2022

[Forster et al, 2021; Meister et al., 2021; Amrhein and Sennrich, 2022; Meister et al., 2022; Stahlberg et al., 2022]

The Way Forward (on our side)

- More efficient approximations to MBR
  - Bottleneck: too many assessments of external utility
  - Bottleneck: obtaining samples from the NMT model for estimating expected utility is expensive
- Can we explain why NMT models are the way they are?
- Re-evaluating improvements to NMT (e.g. deep generative models) without the bias of assessing it through the lens of beam search alone.

Thanks!