Tackling Intrinsic Uncertainty with SCONES

Felix Stahlberg
Exact inference in NMT is impossible.
Exact inference in NMT is impossible.

Wrong! (Stahlberg and Byrne, 2019)
Monotonicity of NMT model scores

NMT left-to-right factorization:

$$\log P(y|x) = \sum_{j=1}^{J} \log P(y_j|y_1^{j-1}, x)$$

<0

NMT scores are monotonically decreasing:

$$\forall j \in [2, J] : \log P(y_1^{j-1}|x) > \log P(y_1^j|x)$$
Exact decoding for NMT

1.) Run beam search
   \( \gamma = P(y_{beam}|x) \) is a lower bound on the global best score: \( \gamma \leq \log P(\hat{y}|x) \)

2.) Run depth-first search
   - Prune if a partial hypothesis score exceeds \( \gamma \)
   - Update \( \gamma \) if a better complete hypothesis is found
   - Child nodes are ordered such that EOS is expanded first

\( y_{beam} \): beam hypothesis
\( \hat{y} \): global best hypothesis
\( \gamma \): lower bound
## Empty translations

<table>
<thead>
<tr>
<th>Search</th>
<th>BLEU</th>
<th>Ratio</th>
<th>#Search errors</th>
<th>#Empty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy</td>
<td>29.3</td>
<td>1.02</td>
<td>73.6%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Beam-10</td>
<td>30.3</td>
<td>1.00</td>
<td>57.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Exact</td>
<td>2.1</td>
<td>0.06</td>
<td>0.0%</td>
<td>51.8%</td>
</tr>
</tbody>
</table>

**Search error**: decoder returns a hypothesis with a lower likelihood than that found by exact inference.

In the absence of search errors, NMT often prefers the empty translation.
But Why?

● “Long sentences sum over more log-probabilities (which are negative), so they result in lower scores”
  ○ But: The left-to-right factorization is correct.

● “It doesn’t really matter - we use small beams / length normalization in practice”
  ○ But: Length normalization is a remedy, not a cure
  ○ But: What is the head room? Is beam search obscuring other model errors?

● “Just train bigger models longer and on more data”
  ○ But: Problem is reduced, but not solved
Model uncertainty

- The neural model cannot decide which output is correct
- Example:
  - en: The pixel will receive updates until October 2026.
  - de1: Der Pixel wird bis Oktober 2026 Updates erhalten.
  - de2: Das Pixel wird bis Oktober 2026 Updates erhalten.
Model uncertainty

- The neural model cannot decide which output is correct
- Example:
  - en: The pixel will receive updates until October 2026.
  - de1: Der Pixel wird bis Oktober 2026 Updates erhalten.
  - de2: Das Pixel wird bis Oktober 2026 Updates erhalten.

Intrinsic uncertainty

- The same input has multiple acceptable outputs
- Example:
  - de: Das Pixel wird bis Oktober 2026 Updates erhalten.
  - en1: The pixel will receive updates until October 2026.
  - en2: The pixel phone gets updates till 10/2026.
Model uncertainty

- The neural model cannot decide which output is correct
- Example:
  - en: The pixel will receive updates until October 2026.
  - de1: Der Pixel wird bis Oktober 2026 Updates erhalten.
  - de2: Das Pixel wird bis Oktober 2026 Updates erhalten.

Intrinsic uncertainty

- The same input has multiple acceptable outputs
- Example:
  - de: Das Pixel wird bis Oktober 2026 Updates erhalten.
  - en1: The pixel will receive updates until October 2026.
  - en2: The pixel phone gets updates till 10/2026.

Softmax models spread out probability mass across multiple candidates
(Eikema and Azis, 2020)
Intrinsic uncertainty in softmax models

- Conventional NMT learns a distribution $P(y|x)$ over all translations $y$ given the source sentence $x$.
- Thus, it cannot represent intrinsic uncertainty.
- Intrinsic uncertainty in the training data leads to contradictions since there is a built in assumption that there is exactly one “correct” translation for a source sentence.

**Training data**

<table>
<thead>
<tr>
<th>de1</th>
<th>en1</th>
</tr>
</thead>
<tbody>
<tr>
<td>de2</td>
<td>en2</td>
</tr>
<tr>
<td>de3</td>
<td>en3</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>de1</td>
<td>en243</td>
</tr>
</tbody>
</table>

$P(en1|de1)=1$

$P(en243|de1)=1$
Intrinsic uncertainty of NLP tasks

- Speech recognition
- Grammatical error correction (GEC)
- Optical character recognition
- Machine translation (MT)

Intrinsic uncertainty
Intrinsic uncertainty of NLP tasks

Felix Stahlberg, Ilia Kulikov, and Shankar Kumar. 2022. Uncertainty determines the adequacy of the mode and the tractability of decoding in sequence to-sequence models. ACL
Beam search curse (Koehn and Knowles, 2017)

- MT quality degrades for large beam sizes, but GEC quality saturates.

![Graph showing relative improvement over greedy for different beam sizes and resources]
High number of beam search errors

- MT has a high number of beam search errors, but GEC does not.

![Graph showing beam search errors and resource levels](image)

**High resource:** MT-deen (German-English), MT-ende (English-German)

**Medium resource:** MT-fien (Finnish-English)

**Low resource:** MT-lten (Lithuanian-English)
MT n-best lists cover a tiny fraction of the probability mass, but GEC covers much more.
Sentence-level uncertainty measure $u$

- For an $n$-way annotated source sentence with references $y_1, \ldots, y_n$ we define the uncertainty measure $u$ as the relative edit distance averaged across all the reference pairs.

![Bar chart showing uncertainty $u$ for different reference lengths and tasks.]

Blue: Grammatical error correction (GEC)

Purple: Machine translation (MT)

$u$ increases with sentence length and intrinsic uncertainty of the task.
For both GEC and MT, the probability mass is more spread out for uncertain sentences (in terms of uncertainty measure $u$).
SCONES: Framing seq2seq as a multi-label problem

- Replaces softmax activation with separate sigmoids for each item in the vocabulary
- No summation over the full vocabulary
- Tokens do not share probability mass
- All other parts of the model remain unchanged

Idea: Learn separate binary classifiers for each \((x, y)\) pair that indicate whether \(y\) is a valid translation (prefix) of \(x\):

\[ z_{x,y} = \begin{cases} 1 & \text{iff. there is a continuation of } y \text{ to a valid translation} \\ 0 & \text{otherwise} \end{cases} \]

Decompose into conditionals for left-to-right decoding:

\[
P(z_{x,y} = 1|x) = \prod_{i=1}^{\lvert y \rvert} P(z_{x,y_{\leq i}} = 1|z_{x,y_{<i}} = 1, x) 
\]

\[
P(z_{x,y_{<i}} = 1|w = 1|x, y_{<i}) = \sigma(f(x, y_{<i}, w))
\]

Google
SCONES training loss (token position $i$)

$$
\mathcal{L}_{SCONES}(x, y, i) = \mathcal{L}_+(x, y, i) + \alpha \mathcal{L}_-(x, y, i)
$$

**Increase prob. of reference token**

$$
\mathcal{L}_+(x, y, i) = -\log P(z_{x,y \leq i} = 1|x, y_{<i}) = -\log \sigma(f(x, y_{<i})y_i).
$$

**Decrease prob. of all other tokens**

$$
\mathcal{L}_-(x, y, i) = -\sum_{w \in V \setminus \{y_i\}} \log P(z_{x,y_{<i},w} = 0|x, y_{<i})
$$

$$
= -\sum_{w \in V \setminus \{y_i\}} \log(1 - \sigma(f(x, y_{<i})w)).
$$
### BLEU score improvements (tuned alpha)

<table>
<thead>
<tr>
<th></th>
<th>Greedy search</th>
<th>Beam search (beam size = 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>de-en</td>
<td>en-de</td>
</tr>
<tr>
<td>Softmax</td>
<td>38.8</td>
<td>38.7</td>
</tr>
<tr>
<td>SCONES</td>
<td>39.9</td>
<td>39.1</td>
</tr>
<tr>
<td>Rel. improvement</td>
<td>+2.7‡</td>
<td>+1.2</td>
</tr>
</tbody>
</table>

- SCONES consistently beats the Softmax baselines
- SCONES with greedy search can often outperform softmax with beam search

<table>
<thead>
<tr>
<th>Language pair</th>
<th>α</th>
</tr>
</thead>
<tbody>
<tr>
<td>de-en</td>
<td>0.5</td>
</tr>
<tr>
<td>en-de</td>
<td>0.5</td>
</tr>
<tr>
<td>fi-en</td>
<td>0.7</td>
</tr>
<tr>
<td>en-fi</td>
<td>1.0</td>
</tr>
<tr>
<td>lt-en</td>
<td>0.7</td>
</tr>
<tr>
<td>en-lt</td>
<td>0.9</td>
</tr>
</tbody>
</table>
**BLEURT-20 score improvements (tuned alpha)**

<table>
<thead>
<tr>
<th></th>
<th>Greedy search</th>
<th>Beam search (beam size = 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>de-en</td>
<td>en-de</td>
</tr>
<tr>
<td><strong>Softmax</strong></td>
<td>70.44</td>
<td>68.08</td>
</tr>
<tr>
<td><strong>SCONES</strong></td>
<td>70.69</td>
<td>67.55</td>
</tr>
</tbody>
</table>

- SCONES beats the Softmax baseline BLEURT scores for all language directions except English-German
Quality/runtime trade-off

![Graph showing BLEU scores vs. sentences per second for German-English translation.](image)

- Speed-ups of up to 4x without BLEU degradation
Fixing the beam search curse

A small alpha value mitigates the beam search curse.
Fixing inadequate highest probability translations

Unlike softmax, SCONES with a small alpha value assigns the highest probability to adequate translations.
Fixing empty highest probability translations

Unlike softmax, SCONES with a small alpha value achieves good separation between most likely and empty translation.
Beam search errors

SCONES with a small alpha value reduces the number of beam search errors.
SCONES represents intrinsic uncertainty

French-English example

Input sentence: la stratosphère se trouve à environ 10 à 50 km d’altitude.
Output sentence: The stratosphere is about 10 to 50 km altitude.
SCONES represents intrinsic uncertainty

French-English example

Input sentence:  
la stratosphère se trouve à environ 10 à 50 km d' altitude .

Output sentence:  
The stratosphere is about 10 to 50 km altitude .
Synthetic language experiments

- Idea: Train an IBM-3 word alignment model with GIZA++
- Generate training data by sampling from IBM-3
- Changing the sampling temperature controls the intrinsic uncertainty of the translation task
German-to-synthetic-English translation

SCONES with a small alpha value is more robust against intrinsic uncertainty.
SCONES - Practical advice

- Using the optimizer hyper-parameters from softmax for SCONES training is often a good starting point, but SCONES may need further hyper-parameter search.
- JAX code for SCONES is in the appendix of the paper. A numerically more stable TensorFlow implementation is in Lingvo.
- SCONES gradient norms are huge in the first few training iterations. Using Lamb instead of Adam can help.
- A good alpha value depends on the task and the vocabulary size. We have trained models with alpha between 0.1 and 6.0.
Intrinsic uncertainty causes various issues in conventional seq2seq models

- Beam search curse (degrading performance at higher beam sizes)
- Overly spread out probability mass
- Beam search errors
- Inadequate modes

SCONES is designed to equip sequence models with the ability to represent intrinsic uncertainty

- Has a tunable parameter alpha to balance the importance of correct tokens and incorrect tokens

SCONES with tuned alpha value yields significant BLEU and/or runtime improvements over softmax baselines

SCONES with small alpha value fixes various pathologies of seq2seq models for intrinsically uncertain NLP tasks like machine translation
Thank you!

Your scones connection in Prague:

http://www.articbakehouse.cz/
Length normalization

Length-dependent lower bounds: \( \gamma_k = (k + 1) \frac{\log P(y_{beam}|x)}{|y_{beam}| + 1} \)

<table>
<thead>
<tr>
<th>Search</th>
<th>W/o length norm. BLEU</th>
<th>Ratio</th>
<th>With length norm. BLEU</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beam-10</td>
<td>37.0</td>
<td>1.00</td>
<td>36.3</td>
<td>1.03</td>
</tr>
<tr>
<td>Beam-30</td>
<td>36.7</td>
<td>0.98</td>
<td>36.3</td>
<td>1.04</td>
</tr>
<tr>
<td>Exact</td>
<td>27.2</td>
<td>0.74</td>
<td>36.4</td>
<td>1.03</td>
</tr>
</tbody>
</table>

Length normalization fixes translation lengths but prevents exact search from matching the BLEU score of Beam-10.
Long source sentences are more affected by search errors and empty translations.
### Architectures

<table>
<thead>
<tr>
<th>Model</th>
<th>Beam-10</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>#Search err.</td>
</tr>
<tr>
<td>LSTM*</td>
<td>28.6</td>
<td>58.4%</td>
</tr>
<tr>
<td>SliceNet*</td>
<td>28.8</td>
<td>46.0%</td>
</tr>
<tr>
<td>Transformer-Base</td>
<td>30.3</td>
<td>57.7%</td>
</tr>
<tr>
<td>Transformer-Big*</td>
<td>31.7</td>
<td>32.1%</td>
</tr>
</tbody>
</table>

The problem of search errors and empty translations is not specific to transformer_base.