Jam or Cream First?[1] Modeling Ambiguity in Neural Machine Translation with SCONES

Felix Stahlberg and **Shankar Kumar**
Google Research
{fstahlberg,shankarkumar}@google.com

Abstract

The softmax layer in neural machine translation is designed to model the distribution over mutually exclusive tokens. Machine translation, however, is intrinsically uncertain: the same source sentence can have multiple semantically equivalent translations. Therefore, we propose to replace the softmax activation with a multi-label classification layer that can model ambiguity more effectively. We call our loss function Single-label Contrastive Objective for Non-Exclusive Sequences (SCONES). We show that the multi-label output layer can still be trained on single reference training data using the SCONES loss function. SCONES yields consistent BLEU score gains across six translation directions, particularly for medium-resource language pairs and small beam sizes. By using smaller beam sizes we can speed up inference by a factor of 3.9x and still match or improve the BLEU score obtained using softmax. Furthermore, we demonstrate that SCONES can be used to train NMT models that assign the highest probability to adequate translations, thus mitigating the "beam search curse". Additional experiments on synthetic language pairs with varying levels of uncertainty suggest that the improvements from SCONES can be attributed to better handling of ambiguity.

1 Introduction

Conventional neural machine translation (NMT) models learn the probability $P(\mathbf{y}|\mathbf{x})$ of the target sentence $\mathbf{y}$ given the source sentence $\mathbf{x}$ (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014). This framework implies that there is a single best translation for a given source sentence: if there were multiple valid translations $\mathbf{y}_1$ and $\mathbf{y}_2$ they would need to share probability mass (e.g. $P(\mathbf{y}_1|\mathbf{x}) = 0.5$ and $P(\mathbf{y}_2|\mathbf{x}) = 0.5$), but such a distribution could also represent *model* uncertainty, i.e. the case when *either* $\mathbf{y}_1$ *or* $\mathbf{y}_2$ are correct translations. Therefore, learning a single distribution over all target language sentences does not allow the model to naturally express *intrinsic uncertainty*[2] (Padó et al., 2009; Dreyer and Marcu, 2012; Ott et al., 2018; Stahlberg et al., 2022), the nature of the translation task to allow multiple semantically equivalent translations for a given source sentence. A single distribution over all sequences represents uncertainty by assigning probabilities, but it cannot distinguish between different kinds of uncertainty (e.g. model uncertainty versus intrinsic uncertainty).

Therefore, in this work we frame machine translation as a multi-label classification task (Tsoumakas and Katakis, 2007; Zhang and Zhou, 2014). Rather than learning a single distribution $P(\mathbf{y}|\mathbf{x})$ over all target sentences $\mathbf{y}$ for a source sentence $\mathbf{x}$, we learn binary classifiers for each sentence pair $(\mathbf{x}, \mathbf{y})$ that indicate whether or not $\mathbf{y}$ is a valid translation of $\mathbf{x}$. In this framework, intrinsic uncertainty can be represented by setting the probabilities of two (or more) correct translations $\mathbf{y}_1$ and $\mathbf{y}_2$ to 1 simultaneously. The probabilities for each translation are computed using separate binary classifiers, and thus there is no requirement that the probabilities sum to one over all translations. In practice, the probability of a complete translation is decomposed into a product of the token-level probabilities. Thus we replace the softmax output layer in Transformer models (Vaswani et al., 2017) with sigmoid activations that assign a probability between 0 and 1 to each token in the vocabulary at each time step. We propose a loss function, ***S**ingle-label **C**ontrastive **O**bjective for **N**on-**E**xclusive **S**equences* (**SCONES**) that allows us to train our models on single reference training data. Our work is inspired by noise-contrastive es-

[1] https://en.wikipedia.org/wiki/Cream_tea#Variations

[2] This is sometimes referred to as *aleatoric* uncertainty in the literature (Der Kiureghian and Ditlevsen, 2009).

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4950 - 4961

timation (NCE) (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012). Unlike NCE, whose primary goal was to efficiently train models over large vocabularies, our motivation for SCONES is to model non-exclusive outputs.

We demonstrate multiple benefits of training NMT models using SCONES when compared to standard cross-entropy with regular softmax. We report consistent BLEU score gains between 1%-9% across six different translation directions. SCONES with greedy search typically outperforms softmax with beam search, resulting in inference speed-ups of up to 3.9x compared to softmax without any degradation in BLEU score.

SCONES can be tuned to mitigate some of the pathologies of traditional NMT models. Softmax-based models have been shown to assign the highest probability to either empty or inadequate translations (modes) (Stahlberg and Byrne, 2019; Eikema and Aziz, 2020). This behavior manifests itself as the "beam search curse" (Koehn and Knowles, 2017): increasing the beam size may lead to worse translation quality. We show that SCONES can be used to train models that a) assign the highest probability to adequate translations and b) do not suffer from the beam search curse.

Finally, we use SCONES to train models on synthetic translation pairs that we generate by sampling from the IBM Model 3 (Brown et al., 1993). By varying the sampling temperature, we control the level of ambiguity in the language pair. We show that SCONES is effective in improving the adequacy of the highest probability translation for highly ambiguous translation pairs, confirming our intuition that SCONES can handle intrinsic uncertainty well.

2 Training NMT models with SCONES

We denote the (subword) vocabulary as $\mathcal{V} = \{w_1, \ldots, w_{|\mathcal{V}|}\}$, the special end-of-sentence symbol as $w_1 =$ </s>, the source sentence as $\mathbf{x} = \langle x_1, \ldots, x_{|\mathbf{x}|} \rangle \in \mathcal{V}^*$, a translation as $\mathbf{y} = \langle y_1, \ldots, y_{|\mathbf{y}|} \rangle \in \mathcal{V}^*$, and a translation prefix as $\mathbf{y}_{\leq i} = \langle y_1, \ldots, y_i \rangle$. We use a center dot "$\cdot$" for string concatenations. Unlike conventional NMT that models a single distribution $P(\mathbf{y}|\mathbf{x})$ over all target language sentences, SCONES learns a separate binary classifier for each sentence pair $(\mathbf{x}, \mathbf{y})$. We define a Boolean function $t(\cdot, \cdot)$ that indicates whether $\mathbf{y}$ is a valid translation of $\mathbf{x}$:

$$t(\mathbf{x}, \mathbf{y}) := \begin{cases} \texttt{true} & \text{if } \mathbf{y} \text{ is a translation of } \mathbf{x} \\ \texttt{false} & \text{otherwise} \end{cases}. \quad (1)$$

We do not model $t(\cdot, \cdot)$ directly. To guide decoding, we learn variables $z_{\mathbf{x},\mathbf{y}}$ which generalize $t(\cdot, \cdot)$ to translation *prefixes*:

$$z_{\mathbf{x},\mathbf{y}} := \begin{cases} 1 & \exists \mathbf{y}' \in \mathcal{V}^* : t(\mathbf{x}, \mathbf{y} \cdot \mathbf{y}') = \texttt{true} \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

i.e. $z_{\mathbf{x},\mathbf{y}}$ is a binary label for the pair $(\mathbf{x}, \mathbf{y})$ consisting of source sentence $\mathbf{x}$ and the translation prefix $\mathbf{y}$: $z_{\mathbf{x},\mathbf{y}} = 1$ iff. $\mathbf{y}$ is a prefix of a valid translation of $\mathbf{x}$. We decompose its probability as a product of conditionals to facilitate left-to-right beam decoding:[3]

$$\begin{aligned} P(z_{\mathbf{x},\mathbf{y}} = 1|\mathbf{x}) &:= \prod_{i=1}^{|\mathbf{y}|} P(z_{\mathbf{x},\mathbf{y}_{\leq i}} = 1 | z_{\mathbf{x},\mathbf{y}_{<i}} = 1, \mathbf{x}) \\ &= \prod_{i=1}^{|\mathbf{y}|} P(z_{\mathbf{x},\mathbf{y}_{\leq i}} = 1 | \mathbf{x}, \mathbf{y}_{<i}). \end{aligned} \quad (3)$$

We assign the conditional probabilities by applying the sigmoid activation function $\sigma(\cdot)$ to the logits:

$$P(z_{\mathbf{x},\mathbf{y}_{<i} \cdot w} = 1 | \mathbf{x}, \mathbf{y}_{<i}) = \sigma(f(\mathbf{x}, \mathbf{y}_{<i})_w), \quad (4)$$

where $w \in \mathcal{V}$ is a single token, $f(\mathbf{x}, \mathbf{y}_{<i}) \in \mathbb{R}^{|\mathcal{V}|}$ are the logits at time step i, and $f(\mathbf{x}, \mathbf{y}_{<i})_w$ is the logit corresponding to token w. The only architectural difference to a standard NMT model is the output activation: instead of the softmax function that yields a single distribution over the full vocabulary, we use multiple sigmoid activations in each logit component to define separate Bernoulli distributions for each item in the vocabulary (Fig. 1). However, using such a multi-label classification view requires a different training loss function because, unlike the probabilities from a softmax, the probabilities in Eq. 4 do not provide a normalized distribution over the vocabulary. An additional challenge is that existing MT training datasets typically do not provide more than one reference translation. Our SCONES loss function aims to balance two token-level objectives using a scaling factor $\alpha \in \mathbb{R}^+$:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{y}|} \sum_{i=1}^{|\mathbf{y}|} \mathcal{L}_{\text{SCONES}}(\mathbf{x}, \mathbf{y}, i), \quad (5)$$

[3] As a base case we define $P(z_{\mathbf{x},\epsilon} = 1|\mathbf{x}) = 1$ for the empty translation prefix.

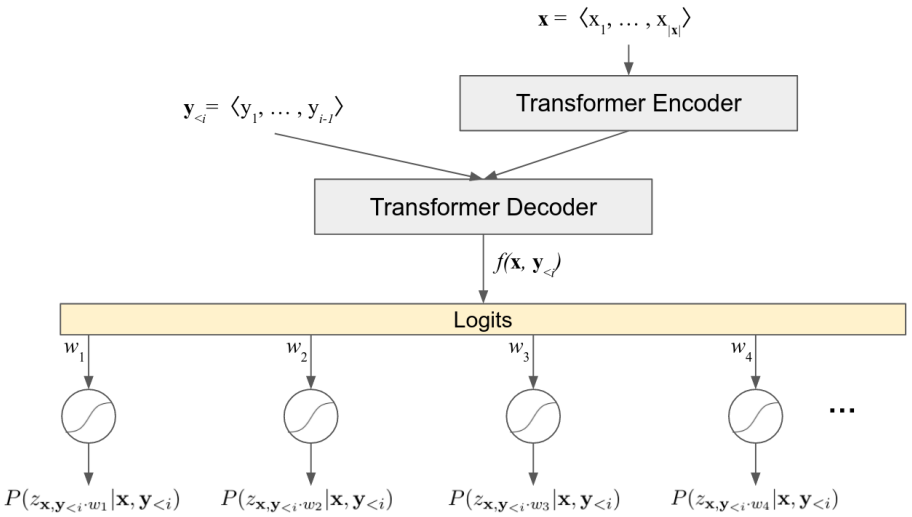

Figure 1: Multi-way NMT Transformer architecture for non-exclusive target sequences.

where

$$\mathcal{L}_{\text{SCONES}}(\mathbf{x}, \mathbf{y}, i) = \mathcal{L}_{+}(\mathbf{x}, \mathbf{y}, i) + \alpha \mathcal{L}_{-}(\mathbf{x}, \mathbf{y}, i). \tag{6}$$

$\mathcal{L}_{+}(\cdot)$ aims to increase the probability $P(z_{\mathbf{x},\mathbf{y}_{\leq i}} = 1|\mathbf{x}, \mathbf{y}_{<i})$ of the gold label y_i since it is a valid extension of the translation prefix $\mathbf{y}_{<i}$:

$$\begin{aligned} \mathcal{L}_{+}(\mathbf{x}, \mathbf{y}, i) = & \quad -\log P(z_{\mathbf{x},\mathbf{y}_{\leq i}} = 1|\mathbf{x}, \mathbf{y}_{<i}) \\ = & \quad -\log \sigma(f(\mathbf{x}, \mathbf{y}_{<i})_{y_i}). \end{aligned} \tag{7}$$

$\mathcal{L}_{-}(\cdot)$ is designed to reduce the probability $P(z_{\mathbf{x},\mathbf{y}_{<i}\cdot w} = 1|\mathbf{x}, \mathbf{y}_{<i})$ for all labels w except for the gold label y_i:

$$\begin{aligned} \mathcal{L}_{-}(\mathbf{x}, \mathbf{y}, i) &= -\sum_{w \in \mathcal{V} \setminus \{y_i\}} \log P(z_{\mathbf{x},\mathbf{y}_{<i}\cdot w} = 0|\mathbf{x}, \mathbf{y}_{<i}) \\ &= -\sum_{w \in \mathcal{V} \setminus \{y_i\}} \log(1 - \sigma(f(\mathbf{x}, \mathbf{y}_{<i})_w)). \end{aligned} \tag{8}$$

Appendix C provides an implementation of SCONES in JAX (Bradbury et al., 2018).

During inference we search for the translation $\mathbf{y}^*$ that ends with </s> and has the highest probability of being a translation of $\mathbf{x}$:

$$\begin{aligned} \mathbf{y}^* = & \quad \underset{\mathbf{y} \in \{\mathbf{w}\cdot\text{</s>}|\mathbf{w} \in \mathcal{V}^*\}}{\arg\max} \; P(z_{\mathbf{x},\mathbf{y}} = 1|\mathbf{x}) \\ \overset{\text{Eqs. 3, 4}}{=} & \quad \underset{\mathbf{y} \in \{\mathbf{w}\cdot\text{</s>}|\mathbf{w} \in \mathcal{V}^*\}}{\arg\max} \sum_{i=1}^{|\mathbf{y}|} \log \sigma(f(\mathbf{x}, \mathbf{y}_{<i})_{y_i}). \end{aligned} \tag{9}$$

We approximate this decision rule with vanilla beam search. The same inference code is used for both our softmax baselines and the SCONES-trained models. The only difference is that the logits from SCONES models are transformed by a sigmoid instead of a softmax activation, i.e. no summation over the full vocabulary is necessary.

Parameter	Value
Attention dropout rate	0.1
Attention layer size	512
Dropout rate	0.1
Embedding size	512
MLP dimension	2,048
Number of attention heads	8
Number of layers	6
Training batch size	256
Total number of parameters	121M

Table 1: Transformer hyper-parameters.

Language pair	#Training sentence pairs	
	Unfiltered	Filtered
German-English	39M	33M
Finnish-English	6.6M	5.5M
Lithuanian-English	2.3M	2.0M

Table 2: MT training set sizes.

Relation to noise-contrastive estimation Our SCONES loss function is related to noise-contrastive estimation (NCE) (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012) because both methods reformulate next word prediction as a multi-label classification problem, and both losses have a "positive" component for the gold label, and a "negative" component for other labels.[4] Unlike NCE, the negative loss component ($\mathcal{L}_{-}(\cdot)$) in SCONES does not require sampling from a noise distribution as it makes use of *all* tokens in the

[4]Technically, SCONES could be written as an instance of NCE with a scaling factor α and an exhaustive enumeration of negative NCE samples.

	Greedy search						Beam search (beam size = 4)					
	de-en	**en-de**	**fi-en**	**en-fi**	**lt-en**	**en-lt**	**de-en**	**en-de**	**fi-en**	**en-fi**	**lt-en**	**en-lt**
Softmax	38.8	38.7	26.9	18.5	26.3	11.5	39.6	39.4	27.7	19.0	26.9	12.0
SCONES	39.9	39.1	27.6	19.5	27.7	12.5	40.3	39.8	28.4	20.0	28.9	12.6
Rel. improvement	**+2.7**‡	**+1.2**	**+2.8**†	**+5.4**‡	**+5.3**‡	**+8.5**‡	**+1.7**†	**+0.9**	**+2.7**†	**+5.5**‡	**+7.4**‡	**+5.7**

Table 3: BLEU score gains from SCONES over our NMT softmax baselines with tuned α-values (Table 5). Using a paired bootstrap method (Koehn, 2004), we highlight improvements that are statistically significant either at a .05 level (†) or a .01 level (‡).

	Greedy search						Beam search (beam size = 4)					
	de-en	**en-de**	**fi-en**	**en-fi**	**lt-en**	**en-lt**	**de-en**	**en-de**	**fi-en**	**en-fi**	**lt-en**	**en-lt**
Softmax	70.44	68.08	68.93	66.16	68.52	56.68	70.78	68.48	69.56	66.44	69.20	57.61
SCONES	70.69	67.55	69.28	67.32	68.96	58.68	70.88	67.99	69.72	67.91	69.95	59.48

Table 4: BLEURT (Sellam et al., 2020) scores (`BLEURT-20` checkpoint) for SCONES and our NMT softmax baselines with tuned α-values (Table 5).

Language pair	α
de-en	0.5
en-de	0.5
fi-en	0.7
en-fi	1.0
lt-en	0.7
en-lt	0.9

Table 5: Values of α that yield the best greedy BLEU scores on the respective development sets.

vocabulary besides the gold token. This is possible because we operate on a limited 32K subword vocabulary whereas NCE is typically used to efficiently train language models with much larger word-level vocabularies (Mnih and Teh, 2012). NCE has a "self-normalization" property (Gutmann and Hyvärinen, 2010; Pihlaja et al., 2010; Mnih and Teh, 2012; Goldberger and Melamud, 2018) which can reduce computation by avoiding the expensive partition function for distributions over the full vocabulary. To do so, NCE uses the multi-label classification task as a proxy problem. By contrast, in SCONES, the multi-label classification perspective is used to express the intrinsic uncertainty in MT and is not simply a proxy for the full softmax. Thus the primary motivation for SCONES is not self-normalization over the full vocabulary.

3 Experimental setup

In this work our focus is to compare NMT models trained with SCONES with well-trained standard softmax-based models. Thus we keep our setup simple, reproducible, and computationally economical. We trained Transformer models (Table 1) in six translation directions – German-English (de-en), Finnish-English (en-fi), Lithuanian-English (lt-en), and the reverse directions – on the WMT19 (Barrault et al., 2019) training sets as provided by TensorFlow Datasets.[5] We selected these language pairs to experiment with different training set sizes (Table 2). The training sets were filtered using language ID and simple length-based heuristics, and split into subwords using joint 32K SentencePiece (Kudo and Richardson, 2018) models. All our models were trained until convergence on the development set (between 100K and 700K training steps) using the LAMB (You et al., 2020) optimizer in JAX (Bradbury et al., 2018). Our softmax baselines are trained by minimizing cross-entropy without label smoothing. Our multi-way NMT models are trained by minimizing the SCONES loss function from Sec. 2, also without label smoothing. We evaluate our models on the WMT19 test sets (Barrault et al., 2019) with SacreBLEU (Post, 2018),[6] using the WMT18 test sets as development sets to tune α.

4 Results

4.1 Translation quality

Table 3 compares our SCONES-based NMT systems with the softmax baselines when α is tuned based on the BLEU score on the development set (Table 5). SCONES yields consistent improvements across the board. For four of six language pairs (all except en-de and fi-en), SCONES with greedy search is even able to outperform the softmax models with beam search. The language pairs with fewer resources (fi↔en, lt↔en) benefit from SCONES training much more than the high-resource language pairs (de↔en). SCONES still yields gains for all language directions except

[5] https://www.tensorflow.org/datasets/catalog/wmt19_translate

[6] Comparable to http://wmt.ufal.cz/.

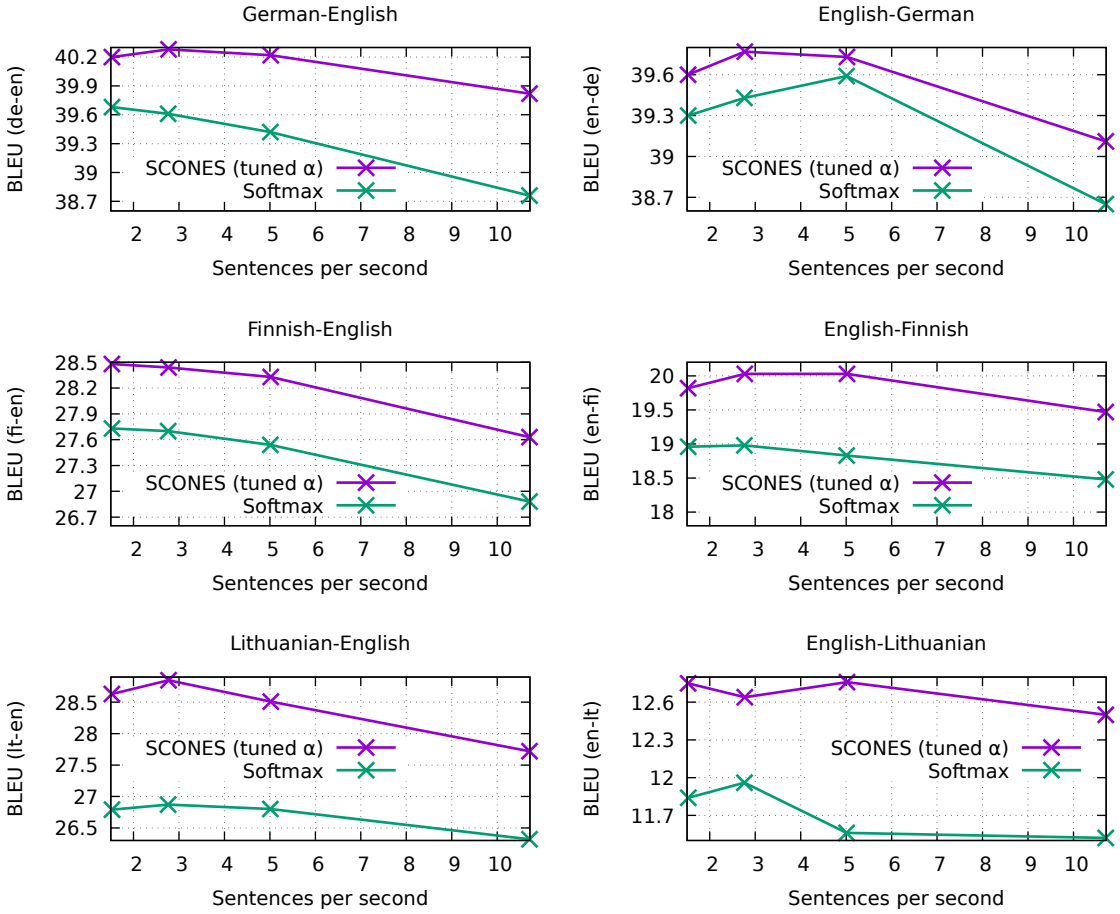

Figure 2: BLEU scores as a function of GPU decoding speeds (median over five runs) for softmax and SCONES with greedy search and beam search with beam sizes 2, 4, and 8 (annotated with $\times$).

English-German when we use BLEURT (Sellam et al., 2020) instead of BLEU as the evaluation measure (Table 4).

4.2 Decoding speed

Our softmax-based models reach their (near) optimum BLEU score with a beam size of around 4. Most of our SCONES models can achieve similar or better BLEU scores with greedy search. Replacing beam-4 search with greedy search corresponds to a 3.9x speed-up (2.76 $\rightarrow$ 10.64 sentences per second) on an entry-level NVIDIA Quadro P1000 GPU with a batch size of 4.[7] Fig. 2 shows the BLEU scores for all six translation directions as a function of decoding speed. Most of the speed-ups are due to choosing a smaller beam size and not due to SCONES avoiding the normalization over the full vocabulary. We expect further speed-ups when comparing models with larger vocabularies.

[7] As an additional optimization, our greedy search implementation operates directly on the logits without applying the output activations.

4.3 Mitigating the beam search curse

One of the most irksome pathologies of traditional softmax-based NMT models is the "beam search curse" (Koehn and Knowles, 2017): larger beam sizes improve the log-probability of the translations, but the translation quality gets worse. This happens because with large beam sizes, the model prefers translations that are too short. This phenomenon has been linked to the local normalization in sequence models (Sountsov and Sarawagi, 2016; Murray and Chiang, 2018) and poor model calibration (Kumar and Sarawagi, 2019). Stahlberg and Byrne (2019) showed that modes are often empty and suggested that the inherent bias of the model towards short translations is often obscured by beam search errors. Stahlberg et al. (2022) provided strong evidence that this length deficiency is due to the intrinsic uncertainty of the MT task. Given that models trained with SCONES explicitly take into account inherent uncertainty, we ran an experiment to determine whether these models are

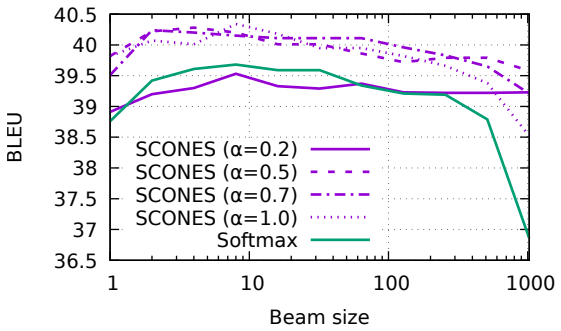

Figure 3: German-English BLEU score as a function of beam size.

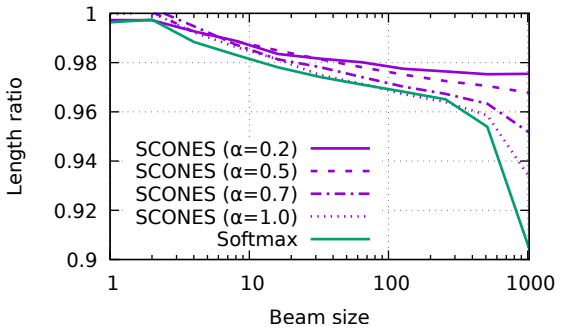

Figure 4: German-English length ratio (hypothesis length / reference length) as a function of beam size.

more robust to the beam search curse compared to softmax trained models.

Fig. 3 plots the BLEU score as a function of the beam size. The sharp decline of the green curve for large beam sizes reflects the *beam search curse* for the softmax baseline. SCONES seems to be less affected at larger beam sizes, particularly for small α-values: the BLEU score for SCONES with $\alpha = 0.2$ (solid purple curve) is stable for beam sizes greater than 100. Fig. 4, which displays the length ratio (the hypothesis length divided by the reference length) versus beam size, suggests that the differences in BLEU trajectories are partly due to translation lengths. Translations obtained using softmax become shorter at higher beam sizes whereas for SCONES with $\alpha = 0.2$, there is no such steep decrease in length. To study the impact of α in the absence of beam search errors we ran the exact depth-first search algorithm of Stahlberg and Byrne (2019) to find the translation with global highest probability.[8] The adequacy of the translations found by exact search depends heavily on α (Fig. 5). With exact search, small α-values yield adequate translations, but $\alpha \approx 1.0$ performs similar to the softmax baseline: the BLEU score drops because hypotheses are too short. Table 6 shows that SCONES with $\alpha = 0.2$ consistently outperforms the softmax baselines by a large margin with exact search. Fig. 6 sheds some light on why SCONES with small α does not prefer empty translations. A small α leads to a larger gap between the log-probabilities of the exact search translation and the empty translation that arises from higher log-probabilities for the exact-search translation along with smaller variances. Intuitively, a small α reduces the importance of the negative loss component $\mathcal{L}_{-}(\cdot)$ in Eq. 6, and thus biases each binary classifier towards predicting the `true` label.

[8]The maximum number of explored states per sentence was set to 1M. This threshold was reached for less than 1.45% of the German-English sentences. See Appendix A for other language directions.

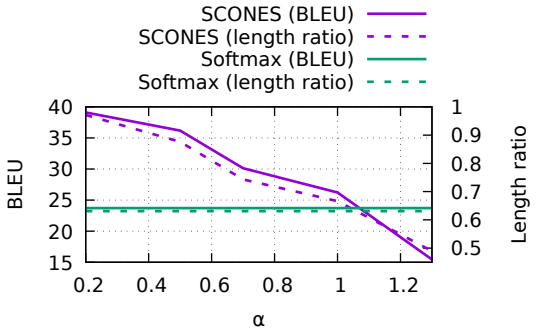

Figure 5: German-English BLEU scores and length ratios (hypothesis length / reference length) for exact search.

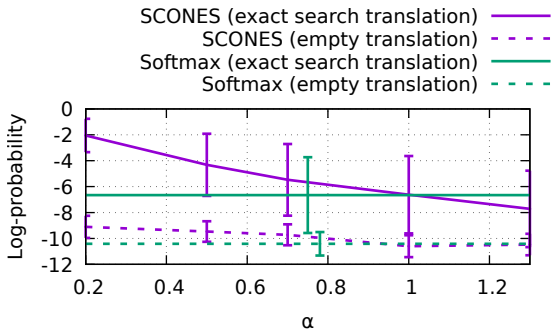

Figure 6: Mean and standard deviation (error bars) of log-probabilities of the global highest probability translations (found using exact search) and the empty translations for German-English.

4.4 Reducing the number of beam search errors

Fig. 7 displays the percentage of beam search errors, the fraction of sentences for which beam

	Beam search (beam size = 4)						Exact search					
	de-en	**en-de**	**fi-en**	**en-fi**	**lt-en**	**en-lt**	**de-en**	**en-de**	**fi-en**	**en-fi**	**lt-en**	**en-lt**
Softmax	39.6	39.4	27.7	19.0	26.9	12.0	23.7	15.6	16.7	10.1	14.2	7.1
SCONES ($\alpha = 0.2$)	39.3	38.9	27.7	19.6	27.9	12.7	39.1	37.2	26.7	18.7	25.6	12.1

Table 6: BLEU scores of beam search and exact search for all six translation directions.

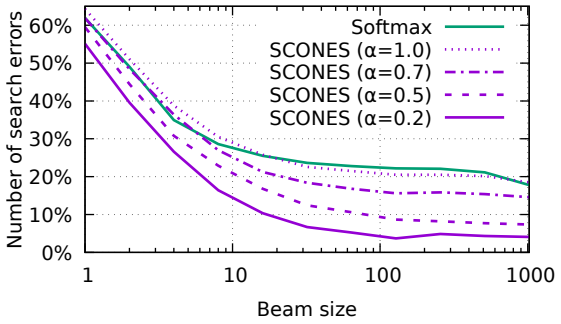

Figure 7: Number of beam search errors for German-English as a function of the beam size.

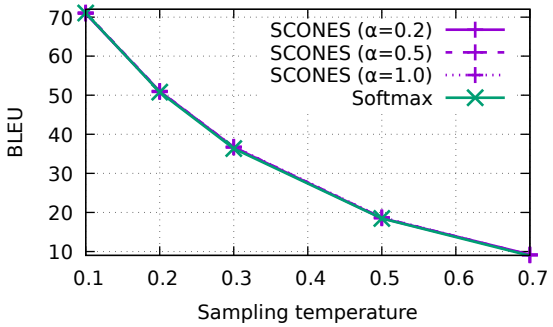

Figure 8: BLEU scores with beam search (beam size of 4) for German-to-synthetic-English translation with different IBM-3 sampling temperatures.

search did not find the global best translation, as a function of beam size. We confirm the findings of Stahlberg and Byrne (2019) for softmax models: the percentage of search errors remains at a relatively high level of around 20% even for very large beam sizes. Increasing the beam size is most effective in reducing the number of search errors for SCONES with a small value of α. However, a small α does not always yield the best overall BLEU score (Fig. 3). Taken together, these observations provide an insight into model errors in NMT: If we describe the "model error" as the mismatch between the global most likely translation and an adequate translation (following Stahlberg and Byrne (2019)), a small α would simultaneously lead to both fewer search errors (Fig. 7) and fewer model errors (Tab. 6). Counter-intuitively, however, BLEU scores peak at slightly higher α-values (Tab. 5). A more sophisticated notion of model errors and search errors is needed to understand the complex inherent biases of beam search for neural sequence-to-sequence models.

5 Experiments with synthetic language pairs

Our main motivation for SCONES is to equip the model to naturally represent intrinsic uncertainty, i.e. the existence of multiple correct target sentences for the same source sentence. To examine the characteristics of SCONES as a function of uncertainty, we generated synthetic language pairs that differ by the level of ambiguity. For this purpose, we trained an IBM-3 model (Brown et al., 1993) on the German-English training data after subword segmentation using MGIZA (Gao and Vogel, 2008). IBM-3 is a generative symbolic model that describes the translation process from one language into another with a generative story, and was popular for finding word alignments for statistical (phrase-based) machine translation (Koehn, 2009). The generative story consists of different steps such as distortion (word reordering), fertility (1:n word mappings), and lexical translation (word-to-word translation) that describe the translation process. The parameters of IBM-3 define probability distributions for each step. In this work we do not use IBM-3 for finding word alignments. Instead, for the original German sentences we sample synthetic English-like translations from the model with different sampling temperatures to control the ambiguity levels of the translation task. A low sampling temperature generates sentence pairs that still capture some of the characteristics of MT such as word reorderings, but the mapping is mostly deterministic (i.e. the same source token is almost always translated to the same target token). A high temperature corresponds to more randomness, i.e. more intrinsic uncertainty. Appendix B contains more details about sampling from IBM-3. We train NMT models using either softmax or SCONES on the synthetic corpora.

Fig. 8 shows that softmax and SCONES perform similarly using beam search: high IBM-3 sampling

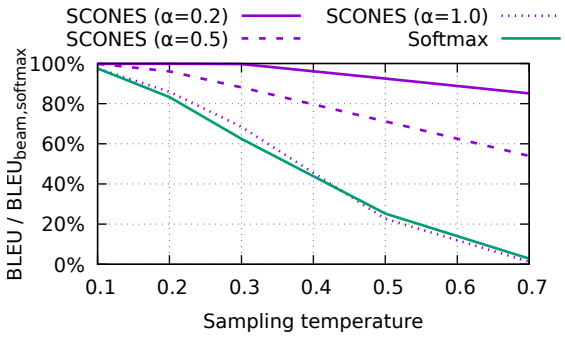

Figure 9: Exact search for German-to-synthetic-English translation with different IBM-3 sampling temperatures. BLEU scores are shown relative to those of the beam search softmax output in Fig. 8 for the respective temperatures.

temperature translation tasks are less predictable, and thus lead to lower BLEU scores. The difference between both approaches becomes clear with exact search (Fig. 9). While the translations with the global highest probability for high IBM-3 sampling temperatures are heavily degraded for softmax and SCONES with $\alpha = 1$, the drop is much less dramatic for SCONES with $\alpha = 0.2$ (solid purple curve). Setting α to a low value enables the model to assign its highest probability to adequate translations, even when the translation task is highly uncertain.

6 Related work

Our approach draws insights from multi-label classification (MLC) (Tsoumakas and Katakis, 2007; Zhang and Zhou, 2006, 2014). One of the earliest approaches for MLC was to transform the problem into multiple binary classification problems while ignoring the correlations between labels (Boutell et al., 2004). More recent work has modeled MLC in the sequence-to-sequence framework with a decoder that generates the labels sequentially, thus preserving the inter-label correlations (Yang et al., 2018). Most prior work in MLC focuses on classification and is not directly applicable to MT. In contrast, our training strategy is tailored for sequence-to-sequence problems. Unlike prior work (Yang et al., 2018), SCONES allows us to perform MLC style training with any underlying NMT architecture by simply changing the loss function. By jointly training all label-specific binary classifiers, our strategy is able to account for label correlations.

Ma et al. (2018) used an MLC objective to improve machine translation. Unlike our approach, they attempted to predict *all* words in the target sentence with a bag-of-words loss function. We formulate the next word prediction at each time step as an MLC problem to handle intrinsic uncertainty, but our models are predicting ordered target sequences, not bags of words.

The speed-ups from SCONES can be partially attributed to avoiding the normalization of the output over the full vocabulary. The same idea motivated earlier work on self-normalized training (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012; Devlin et al., 2014; Goldberger and Melamud, 2018). As described in Sec. 2, unlike work on self-normalization, SCONES does not try to approximate a distribution over the full vocabulary. Rather, its output consists of multiple binary classifiers that do not share probability mass by design to be able to better represent intrinsic uncertainty.

7 Conclusion

Machine translation is a task with high intrinsic uncertainty: a source sentence can have multiple valid translations. We demonstrated that NMT models and specifically Transformers, can learn to model mutually non-exclusive target sentences from single-label training data using our SCONES loss function. Rather than learn a single distribution over all target sentences, SCONES learns multiple binary classifiers that indicate whether or not a target sentence is a valid translation of the source sentence. SCONES yields improved translation quality over conventional softmax-based models for six different translation directions, or (alternatively) speed-ups of up to 3.9x without any degradation in translation performance. We showed that SCONES can be tuned to mitigate the beam search curse and the problem of inadequate and empty modes in standard NMT. Our experiments on synthetic language translation suggest that, unlike softmax-trained models, SCONES models are able to assign their highest probability to adequate translations even when the underlying task is highly ambiguous.

The SCONES loss function is easy to implement. Adapting standard softmax-based sequence-to-sequence architectures such as Transformers requires *only* replacing the cross-entropy loss function with SCONES and the softmax with sigmoid activations. The remaining parts of the training

and inference pipelines can be kept unchanged. SCONES can be potentially useful in handling uncertainty for a variety of ambiguous NLP problems beyond translation, such as generation and dialog. We expect this work to encourage research on modeling techniques that can address ambiguity in much better ways compared to current models.

References

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. 2004. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: Composable transformations of Python+NumPy programs.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? Does it matter? *Structural safety*, 31(2):105–112.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, Maryland. Association for Computational Linguistics.

Markus Dreyer and Daniel Marcu. 2012. HyTER: Meaning-equivalent semantics for translation evaluation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 162–171, Montréal, Canada. Association for Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio. Association for Computational Linguistics.

Jacob Goldberger and Oren Melamud. 2018. Self-normalization properties of language modeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 764–773, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Kevin Knight. 1999. A statistical mt tutorial workbook. In *Prepared for the 1999 JHU Summer Workshop*, pages 1–37.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

Shuming Ma, Xu Sun, Yizhong Wang, and Junyang Lin. 2018. Bag-of-words as target for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 332–338, Melbourne, Australia. Association for Computational Linguistics.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 419–426, Madison, WI, USA. Omnipress.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

Sebastian Padó, Daniel Cer, Michel Galley, Dan Jurafsky, and Christopher D Manning. 2009. Measuring machine translation quality as semantic equivalence: A metric based on entailment features. *Machine Translation*, 23(2-3):181–193.

Miika Pihlaja, Michael Gutmann, and Aapo Hyvärinen. 2010. A family of computationally efficient and simple estimators for unnormalized statistical models. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, page 442–449, Arlington, Virginia, USA. AUAI Press.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Pavel Sountsov and Sunita Sarawagi. 2016. Length bias in encoder decoder models and a case for global conditioning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1525, Austin, Texas. Association for Computational Linguistics.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? 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)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.

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. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 2007:1–13.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*.

Min-Ling Zhang and Zhi-Hua Zhou. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351.

Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837.

A Time complexity of exact search

The exact search algorithm of Stahlberg and Byrne (2019) we used in the paper is guaranteed to find the global best translation. Its runtime, however, varies greatly between language pairs and source sentences. Therefore, we limit the number of explored states per sentence by 1M to keep the decoding time under control. If the 1M threshold is reached, the optimality of the found translation is not guaranteed anymore. Fortunately, for most of our models and test sets, exact search was able to find and verify the global best translation earlier. Table 7 lists the runs for which a fraction of the sentences did not terminate before 1M steps. In these rare cases, we use the best translation found thus far by exact search as an approximation to the global best translation.

B Sampling from IBM-3

The parameters of the IBM-3 model (Brown et al., 1993) are composed of a set of fertility probabilities $n(\cdot|\cdot)$, p_0, p_1, a set of translation probabilities $t(\cdot|\cdot)$, and a set of distortion probabilities $d(\cdot|\cdot)$. According to the IBM Model 3, the following generative process produces the target language sentence $\mathbf{y}$ from a source language sentence $\mathbf{x}$ (Knight, 1999):

1. For each source word x_i indexed by $i = 1, 2, \ldots, |\mathbf{x}|$, choose the fertility ϕ_i with probability $n(\phi_i|x_i)$.
2. Choose the number ϕ_0 of "spurious" target words to be generated from $x_0 = \texttt{NULL}$, using probability p_1 and the sum of fertilities from step 1.
3. Let $m = \sum_{i=0}^{|\mathbf{x}|} \phi_i$.
4. For each $i = 0, 1, 2, \ldots, |\mathbf{x}|$ and each $k = 1, 2, \ldots, \phi_i$, choose a target word τ_{ik} with probability $t(\tau_{ik}|x_i)$.
5. For each $i = 1, 2, \ldots, |\mathbf{x}|$ and each $k = 1, 2, \ldots, \phi_i$, choose a target position π_{ik} with probability $d(\pi_{ik}|i, |\mathbf{x}|, m)$.
6. For each $k = 1, 2, \ldots, \phi_0$, choose a position π_{0k} from the $\phi_0 - k + 1$ remaining vacant positions in $1, 2, \ldots, m$, for a total probability of $\frac{1}{\phi_0!}$.
7. Output the target sentence with words τ_{ik} in positions π_{ik} ($0 \leq i \leq |\mathbf{x}|$, $1 \leq k \leq \phi_i$).

Languages	Run	#incomplete sent.
de-en	SCONES ($\alpha = 0.2$)	1.45%
de-en	SCONES ($\alpha = 0.5$)	0.90%
de-en	SCONES ($\alpha = 0.7$)	0.05%
en-de	SCONES ($\alpha = 0.2$)	4.01%
fi-en	SCONES ($\alpha = 0.2$)	1.30%
en-fi	Softmax	0.10%
en-fi	SCONES ($\alpha = 0.2$)	3.20%
lt-en	Softmax	0.20%
lt-en	SCONES ($\alpha = 0.2$)	5.20%
en-lt	Softmax	0.10%
en-lt	SCONES ($\alpha = 0.2$)	5.31%
synthetic-0.1	Softmax	1.05%
synthetic-0.1	SCONES ($\alpha = 0.2$)	0.55%
synthetic-0.1	SCONES ($\alpha = 0.5$)	1.10%
synthetic-0.1	SCONES ($\alpha = 1.0$)	1.25%
synthetic-0.2	Softmax	1.00%
synthetic-0.2	SCONES ($\alpha = 0.2$)	5.10%
synthetic-0.2	SCONES ($\alpha = 0.5$)	7.65%
synthetic-0.2	SCONES ($\alpha = 1.0$)	2.65%
synthetic-0.3	Softmax	0.10%
synthetic-0.3	SCONES ($\alpha = 0.2$)	12.6%
synthetic-0.3	SCONES ($\alpha = 0.5$)	17.3%
synthetic-0.3	SCONES ($\alpha = 1.0$)	1.80%
synthetic-0.5	SCONES ($\alpha = 0.2$)	25.0%
synthetic-0.5	SCONES ($\alpha = 0.5$)	25.2%
synthetic-0.7	SCONES ($\alpha = 0.2$)	25.9%
synthetic-0.7	SCONES ($\alpha = 0.5$)	20.3%

Table 7: Fraction of sentences for which exact search did not terminate before 1M steps. For runs that are not listed here, exact search terminated within 1M steps for all sentences.

First, we estimate the IBM-3 model parameters using the MGIZA (Gao and Vogel, 2008) word alignment tool. Then, we sample English-like target sentences for the German source sentences following the generative story above. To control the level of uncertainty in the synthetic translation task we alter the entropies of the $n(\cdot|\cdot)$, $t(\cdot|\cdot)$, and $d(\cdot|\cdot)$ distributions by choosing different sampling temperatures $\gamma \in \mathbb{R}^+$. Instead of sampling directly from a categorical distribution $P(\cdot)$ over categories $\mathcal{C}$, temperature sampling uses the following distribution:

$$P_\gamma(c) = \frac{e^{\log P(c)/\gamma}}{\sum_{c' \in \mathcal{C}} e^{\log P(c')/\gamma}} \qquad (10)$$

for each $c \in \mathcal{C}$. A low temperature amplifies large differences in probabilities, and thus leads to a lower entropy and less ambiguity.

C Implementation of SCONES in JAX

Fig. 10 provides an implementation of the SCONES loss function (Sec. 2) in JAX (Bradbury et al., 2018). We bound the inverse model probability (`false_logprob`) by e^{-30} in line 12

```python
from flax import linen as nn
import jax
import jax.numpy as jnp

def compute_scones_loss(
    logits,  # 3D float tensor [batch_size, max_sequence_length, vocab_size]
    targets,  # 2D int tensor [batch_size, max_sequence_length]
    l = 0.0,  # Label smoothing constant (lambda)
    a = 1.0,  # Scaling factor alpha
):
  true_logprob = nn.log_sigmoid(logits)
  false_logprob = jnp.log(jnp.maximum(1.0 - jnp.exp(true_logprob), 1.0e-30))
  gather = jax.vmap(jax.vmap(lambda s, t: s[t]))
  tgt_true_logprob = gather(true_logprob, targets)  # [batch_size, max_seq_length]
  tgt_false_logprob = gather(false_logprob, targets)  # [batch_size, max_seq_length]
  tgt_true_xent = -(1.0 - l) * tgt_true_logprob - l * tgt_false_logprob
  tgt_false_xent = -(1.0 - l) * tgt_false_logprob - l * tgt_true_logprob
  all_false_xent = -(1.0 - l) * false_logprob - l * true_logprob
  loss = a * (jnp.sum(all_false_xent, axis=-1) - tgt_false_xent) + tgt_true_xent
  weights = jnp.where(targets > 0, 1, 0).astype(jnp.float32)  # PAD ID is 0.
  return loss * weights / weights.sum()
```

Figure 10: JAX implementation of the SCONES loss function.

for numerical stability. The JAX implementation generalizes the SCONES loss defined in the main paper in Eq. 6 with a label smoothing (Szegedy et al., 2016) factor $\lambda \in [0, 1]$ (`l` in Fig. 10) such that the positive loss component $\mathcal{L}_+(\cdot)$ becomes the following cross-entropy:

$$\begin{aligned}\mathcal{L}_+(\mathbf{x}, \mathbf{y}, i) = &- (1 - \lambda) \log P(z_{\mathbf{x},\mathbf{y}_{\leq i}} = 1 | \mathbf{x}, \mathbf{y}_{<i}) \\ &- \lambda \log P(z_{\mathbf{x},\mathbf{y}_{\leq i}} = 0 | \mathbf{x}, \mathbf{y}_{<i}).\end{aligned} \tag{11}$$

Similarly, the negative loss component $\mathcal{L}_-(\cdot)$ with label smoothing can be written as:

$$\begin{aligned}\mathcal{L}_-(\mathbf{x}, \mathbf{y}, i) = - \sum_{w \in \mathcal{V} \setminus \{y_i\}} \Big(\\ (1 - \lambda) \log P(z_{\mathbf{x},\mathbf{y}_{<i} \cdot w} = 0 | \mathbf{x}, \mathbf{y}_{<i}) \\ + \lambda \log P(z_{\mathbf{x},\mathbf{y}_{<i} \cdot w} = 1 | \mathbf{x}, \mathbf{y}_{<i}) \Big).\end{aligned} \tag{12}$$

The label smoothing extension is provided for the sake of completeness – we did not use label smoothing in any of the experiments in the main paper since it did not yield improvements in our setups.