On the Diversity of Beam Search

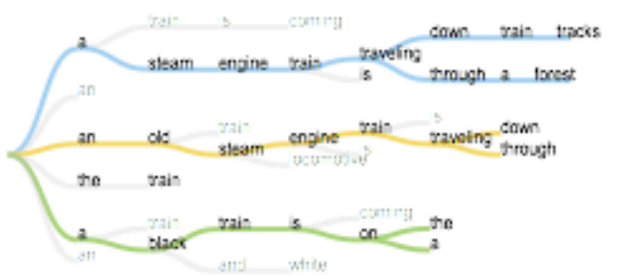
Yilin Yang

Table of content

Motivation for DBS

Beam Search

A steam engine train travelling down train tracks.
A steam engine train travelling down tracks.
A steam engine train travelling through a forest.
A steam engine train travelling through a lush green forest.
A steam engine train travelling through a lush green countryside
A train on a train track with a sky background.

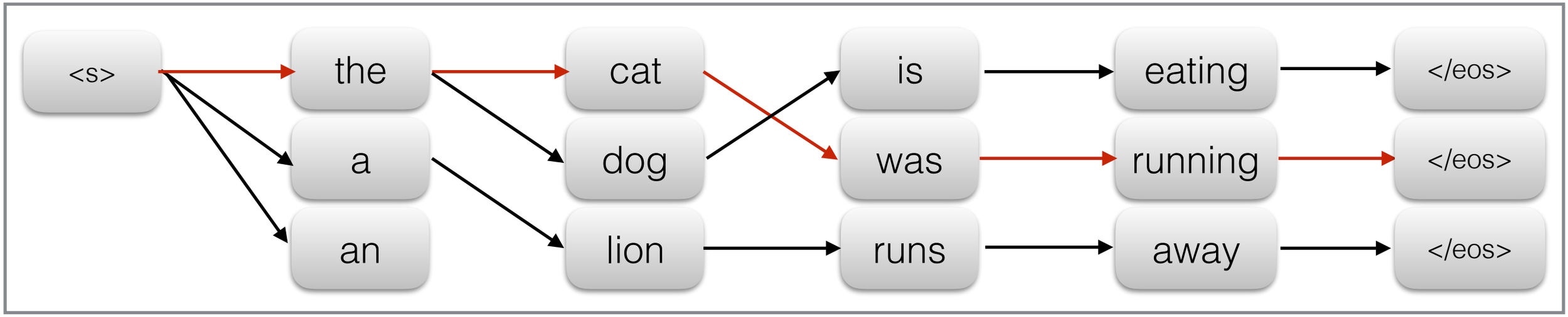


Diverse Beam Search

A steam engine travelling down train tracks.
A steam engine train travelling through a forest.
An old steam engine train travelling down train tracks.
An old steam engine train travelling through a forest.
A black train is on the tracks in a wooded area.
A black train is on the tracks in a rural area.

Beam search suffers from diversity!

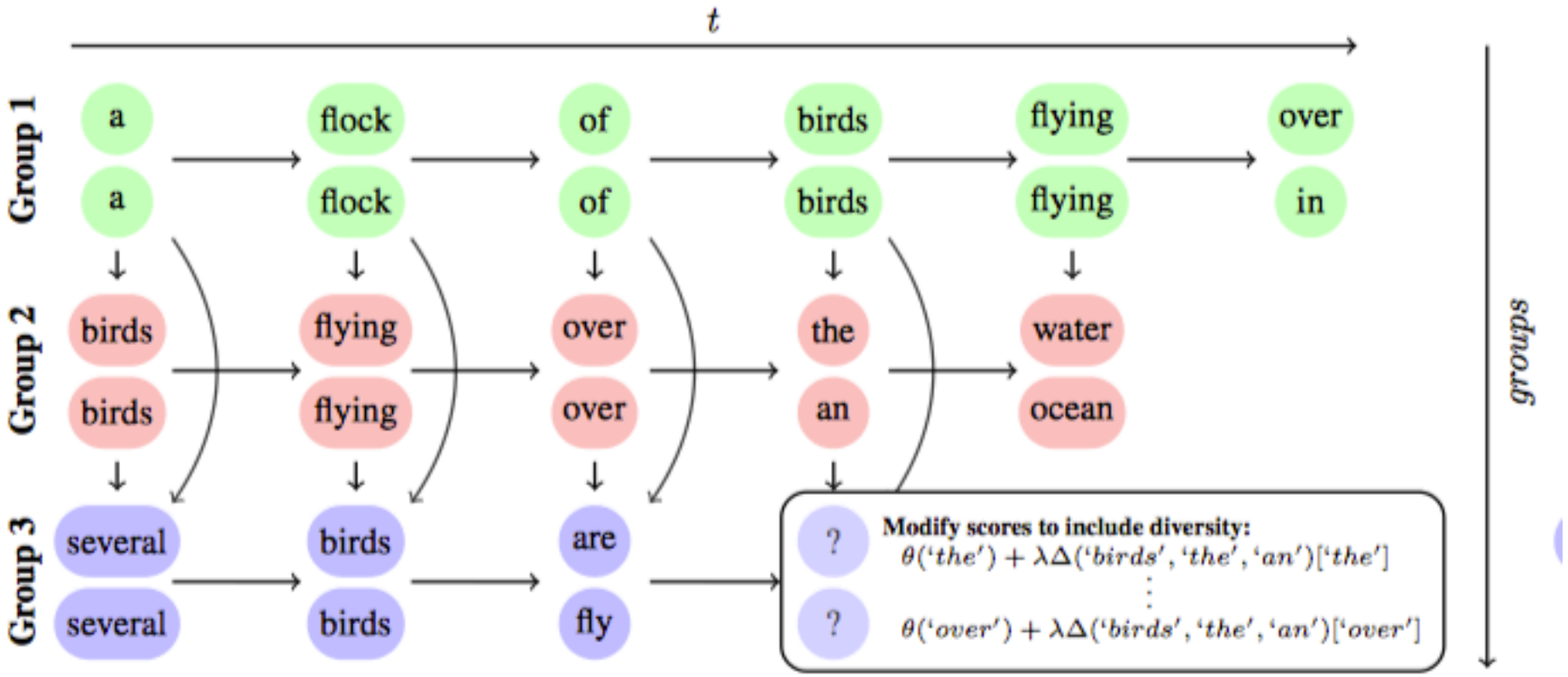
Preliminary on Beam search



Beam Search. The most prevalent method for approximate decoding is BS, which stores the top-B highly scoring candidates at each time step; where B is known as the *beam width*. Let us denote the set of B solutions held by BS at the start of time t as $Y_{[t-1]} = \{\mathbf{y}_{1,[t-1]}, \ldots, \mathbf{y}_{B,[t-1]}\}$. At each time step, BS considers all possible single token extensions of these beams given by the set $\mathcal{Y}_t = Y_{[t-1]} \times \mathcal{V}$ and selects the B most likely extensions. More formally, at each step,

$$Y_{[t]} = \underset{\mathbf{y}_{1,[t]},\cdots,\mathbf{y}_{B,[t]} \in \mathcal{Y}_t}{\operatorname{argmax}} \sum_{b \in [B]} \Theta(\mathbf{y}_{b,[t]}) \quad s.t. \;\; \mathbf{y}_{i,[t]} \neq \mathbf{y}_{j,[t]} \tag{1}$$

Diverse Beam Search



Diverse Beam Search. Let $Y_{[t]}$, the set of all B beams at time t be partitioned into G non-empty, disjoint subsets $Y^g_{[t]},\ g \in [G]$.

$$Y^g_{[t]} = \underset{\mathbf{y}^g_{1,[t]},\cdots,\mathbf{y}^g_{B',[t]} \in \mathcal{Y}^g_t}{\operatorname{argmax}} \sum_{b \in [B']} \Theta(\mathbf{y}^g_{b,[t]}) + \sum_{h=1}^{g-1} \lambda_g \Delta\left(\mathbf{y}^g_{b,[t]}, Y^h_{[t]}\right) \quad \textbf{where} \quad \Delta(\mathbf{y}_{[t]}, Y^g_{[t]}) = \sum_{b=1}^{B'} \delta\left(\mathbf{y}_{[t]}, \mathbf{y}^g_{b,[t]}\right)$$
$$s.t.\ \mathbf{y}^g_{i,[t]} \neq \mathbf{y}^g_{j,[t]},\ \lambda_g \geq 0$$

Diverse Beam Search

Algorithm 1: Diverse Beam Search

Perform a diverse beam search with G groups using a beam width of B

for $t = 1, \ldots T$ **do**

  // perform one step of beam search for first group without diversity

  $Y^1_{[t]} \leftarrow \text{argmax}_{(\mathbf{y}^1_{1,[t]},\ldots,\mathbf{y}^1_{B',[t]})} \sum_{b\in[B']} \Theta(\mathbf{y}^1_{b,[t]})$

  for $g = 2, \ldots G$ **do**

    // augment log-probabilities with diversity penalty

    $\Theta(\mathbf{y}^g_{b,[t]}) \leftarrow \Theta(\mathbf{y}^g_{b,[t]}) + \sum_h \lambda_g \Delta(\mathbf{y}^g_{b,[t]}, Y^h_{[t]}) \quad b \in [B'], \mathbf{y}^g_{b,[t]} \in \mathcal{Y}^g \text{ and } \lambda_g > 0$

    // perform one step of beam search for the group

    $Y^g_{[t]} \leftarrow \text{argmax}_{(\mathbf{y}^g_{1,[t]},\ldots,\mathbf{y}^g_{B',[t]})} \sum_{b\in[B']} \Theta(\mathbf{y}^g_{b,[t]})$

Return set of B solutions, $Y_{[T]} = \bigcup_{g=1}^{G} Y^g_{[T]}$

Choice of Diversity Function

- *Hamming Diversity.* This form penalizes the selection of tokens used in previous groups proportional to the number of times it was selected before.
- *Cumulative Diversity.* Once two sequences have diverged sufficiently, it seems unnecessary and perhaps harmful to restrict that they cannot use the same words at the same time. To encode this 'backing-off' of the diversity penalty we introduce cumulative diversity which keeps a count of identical words used at every time step, indicative of overall dissimilarity. Specifically, $\delta(\mathbf{y}_{[t]}, \mathbf{y}^g_{b,[t]}) = \exp\{-\left(\sum_{\tau \in t} \sum_{b \in B'} I[y^h_{b,\tau} \neq y^g_{b,\tau}]\right)/\Gamma\}$ where Γ is a temperature parameter controlling the strength of the cumulative diversity term and $I[\cdot]$ is the indicator function.
- *n-gram Diversity.* The current group is penalized for producing the same n-grams as previous groups, regardless of alignment in time – similar to Gimpel et al. (2013). This is proportional to the number of times each n-gram in a candidate occurred in previous groups. Unlike hamming diversity, n-grams capture higher order structures in the sequences.
- *Neural-embedding Diversity.* While all the previous diversity functions discussed above perform exact matches, neural embeddings such as word2vec (Mikolov et al., 2013) can penalize semantically similar words like synonyms. This is incorporated in each of the previous diversity functions by replacing the hamming similarity with a soft version obtained by computing the cosine similarity between word2vec representations. When using with n-gram diversity, the representation of the n-gram is obtained by summing the vectors of the constituent words.

Results

	Method	SPICE	Oracle SPICE@k			Distinct n-Grams			
			@5	@10	@20	n = 1	2	3	4
COCO	BS	16.27	22.96	25.14	27.34	0.40	1.51	3.25	5.67
	Li & Jurafsky (2016)	16.35	22.71	25.23	27.59	0.54	2.40	5.69	8.94
	DBS	**16.783**	**23.08**	**26.08**	**28.09**	**0.56**	**2.96**	**7.38**	**13.44**
	Li et al. (2015)	16.74	23.27	26.10	27.94	0.42	1.37	3.46	6.10
PASCAL-50S	BS	4.93	7.04	7.94	8.74	0.12	0.57	1.35	2.50
	Li & Jurafsky (2016)	5.08	7.24	8.09	8.91	0.15	0.97	2.43	5.31
	DBS	**5.357**	**7.357**	**8.269**	**9.293**	**0.18**	**1.26**	**3.67**	**7.33**
	Li et al. (2015)	5.12	7.17	8.16	8.56	0.13	1.15	3.58	8.42

Table 3: Quantitative results on En-Fr machine translation on the newstest-2013 dataset (at $B = 20$). Although we report BLEU-4 values, we find similar trends hold for lower BLEU metrics as well.

Method	Oracle Accuracy (BLEU-4)				Diversity Statistics			
	@1	@5	@10	@20	distinct-1	distinct-2	distinct-3	distinct-4
Beam Search	13.52	16.67	17.63	18.44	0.04	0.75	2.10	3.23
Li & Jurafsky (2016)	13.63	17.11	17.50	18.34	0.04	0.81	2.92	4.61
DBS	**13.69**	**17.51**	**17.80**	**18.77**	**0.06**	**0.95**	**3.67**	**5.54**
Li et al. (2015)	13.40	17.54	17.97	18.86	0.04	0.86	2.76	4.31

Analyze Uncertainty in NMT

- Overall, this is not a normal paper with single clear motivation and methods, rather a collection of exps surrounding one topic “uncertainty in NMT”.

- Thus, this paper doesn’t have clear conclusions or problem-solving methods, but an attempt to “analyze uncertainty” and ends up with a bunch of obeservations.

Throw questions in intro

For instance, according to conventional wisdom neural machine translation (NMT) systems under-estimate rare words (Koehn & Knowles, 2017), why is that? Is the model poorly calibrated? Is this due to exposure bias (Ranzato et al., 2016), i.e., the mismatch between the distribution of words observed at training and test time? Or is this due to the combination of uncertainty in the prediction of the next word and inference being an arg max selection process, which always picks the most likely/frequent word? Similarly, it has been observed (Koehn & Knowles, 2017) that performance degrades with large beams. Is this due to poor fitting of the model which assigns large probability mass to bad sequences? Or is this due to the heuristic nature of this search procedure which fails to work for large beam values? In this paper we will provide answers and solutions to these and other related questions.

Intrinsic Uncertainty

One source of uncertainty is the existence of several *semantically equivalent* translations of the same source sentence. This has been extensively studied in the literature (Dreyer & Marcu, 2012; Padó et al., 2009). Translations can be more or less literal, and even if literal there are many ways to express the same meaning. Sentences can be in the active or passive form and for some languages determiners and prepositions such as 'the', 'of', or 'their' can be optional.

paraphrase

Besides uncertainty due to the existence of distinct, yet semantically equivalent translations, there are also sources of uncertainty due to *under-specification* when translating into a target language more inflected than the source language. Without additional context, it is often impossible to predict the missing gender, tense, or number, and therefore, there are multiple plausible translations of the same source sentence. Simplification or addition of cultural context are also common sources of uncertainty (Venuti, 2008).

Under-specification from source language

Extrinsic Uncertainty

- Noises from web-crawled dataset
- One particular noise: tgt is copies of the src

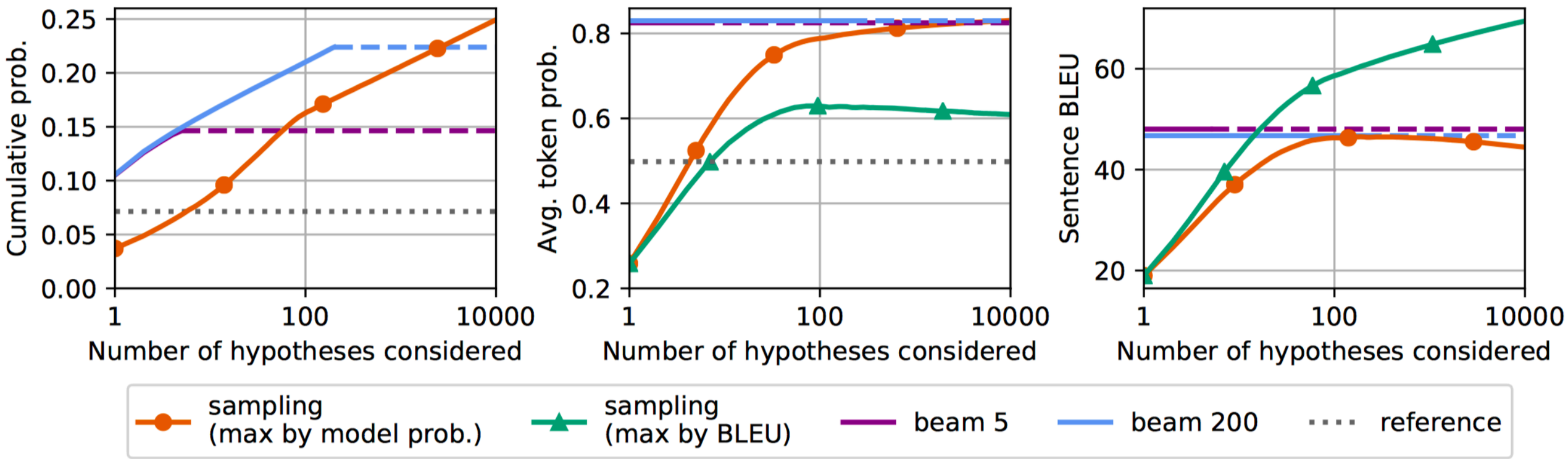
For instance, we found that between 1.1% to 2.0% of *training* examples in the WMT'14 En-De and WMT'14 En-Fr datasets (§4.2) are "copies" of the source sentences.

Sampling & beam search

Figure 1 (*Left*) shows that **the model's output distribution is highly uncertain**

these finding suggest that most of the probability mass is spread elsewhere in the space

BLEU and model probability are imperfectly correlated: while we find more likely translations as we sample more candidates, BLEU over those samples eventually decreases (*Right*, orange curve).[3] Vice versa, hypotheses selected by BLEU have lower likelihood score beyond 80 samples (*Center*, green curve).



Finally, we observe that the model on average assigns much lower scores to the reference translation compared to beam hypotheses (Figure 1, *Center*).

Beam search finds hypotheses with high probs

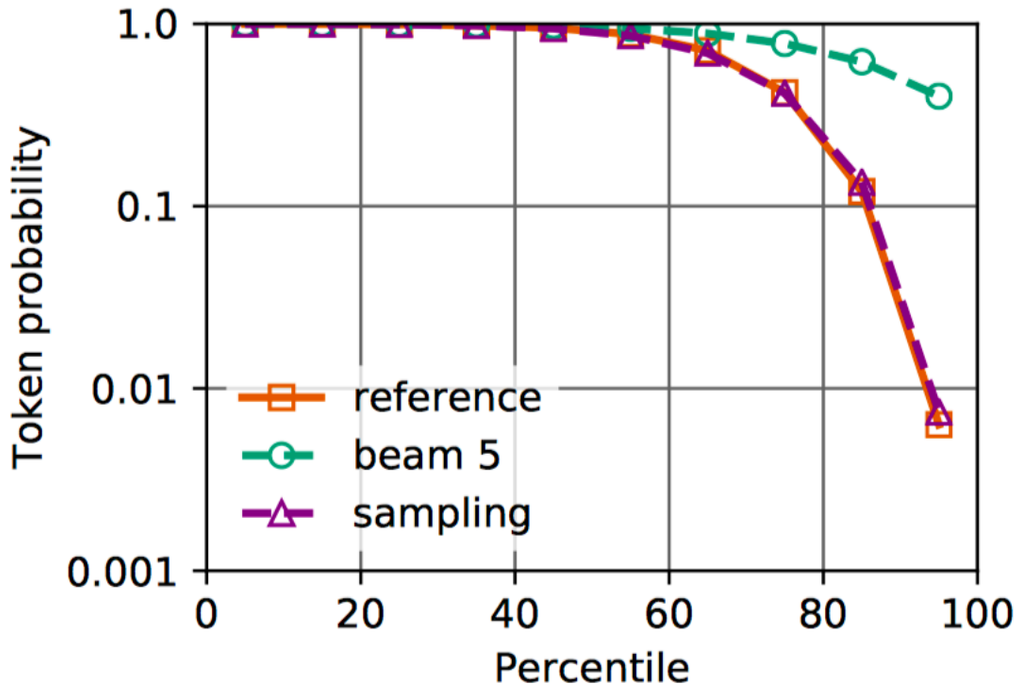


Figure 2. Probability quantiles for tokens in the reference, beam search hypotheses ($k = 5$), and sampled hypotheses for the WMT'14 En-Fr validation set.

beam search is a very effective search strategy, finding hypotheses with very high average token probabilities and rarely leaving high likelihood regions; indeed only 20% of beam tokens have probabilities below 0.7. In contrast, the probabilities for sampling and the human references are much lower.

Copies Causes Large Beam Degradation

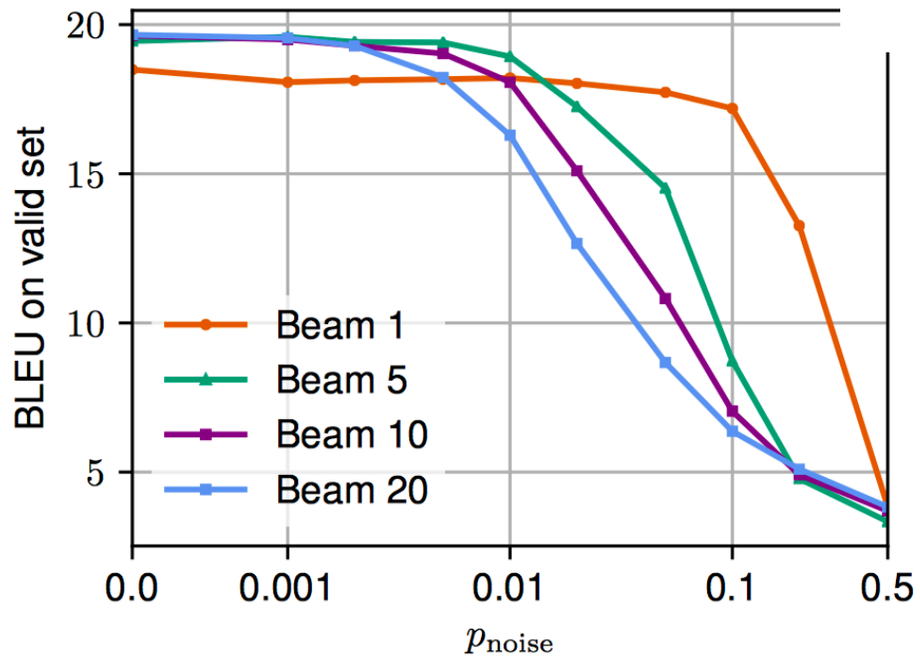


Figure 3. Translation quality of models trained on WMT'17 English-German news-commentary data with added synthetic copy noise in the training data (x-axis) tested with various beam sizes on the validation set.

It turns out that **copies are over-represented in the output of beam search**. On WMT'14 En-Fr, beam search outputs copies at the following rates: 2.6% (beam=1), 2.9% (beam=5), 3.2% (beam=10) and 3.5% (beam=20).

Figure 3 shows that larger beams are much more affected by copy noise. Even just 1% of copy noise can lead to a drop of 3.3 BLEU for a beam of $k = 20$ compared to a model with no added noise. For a 10% noise level, all but greedy search have their accuracy more than halved.

Copies take over with wider beams

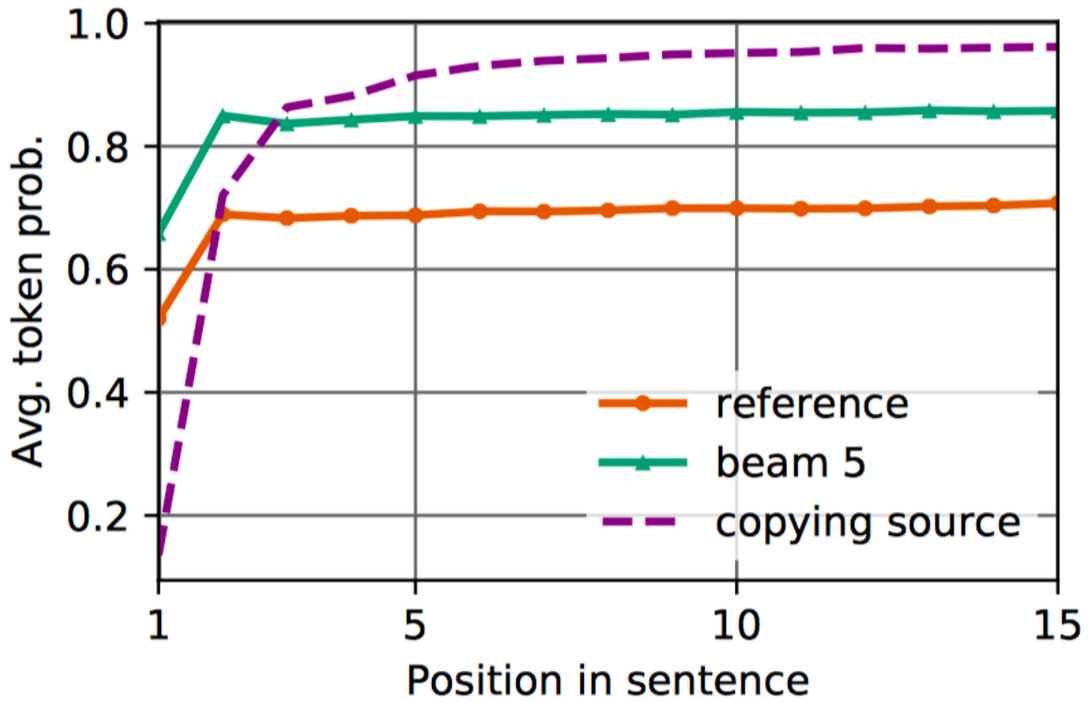


Figure 4. Average probability at each position of the output sequence on the WMT'14 En-Fr validation set, comparing the reference translation, beam search hypothesis ($k = 5$), and copying the source sentence.

Figure 4 shows that the probability of copying the first source token is very unlikely according to the model (and actually matches the ground truth rate of copy noise). However, after three tokens the model switches to almost deterministic transitions. Because beam search proceeds in strict left-to-right manner, the copy mode is only reachable if the beam is wide enough to consider the first source word which has low probability. However, once in the beam, the copy mode quickly takes over.

Solution: filter the dataset

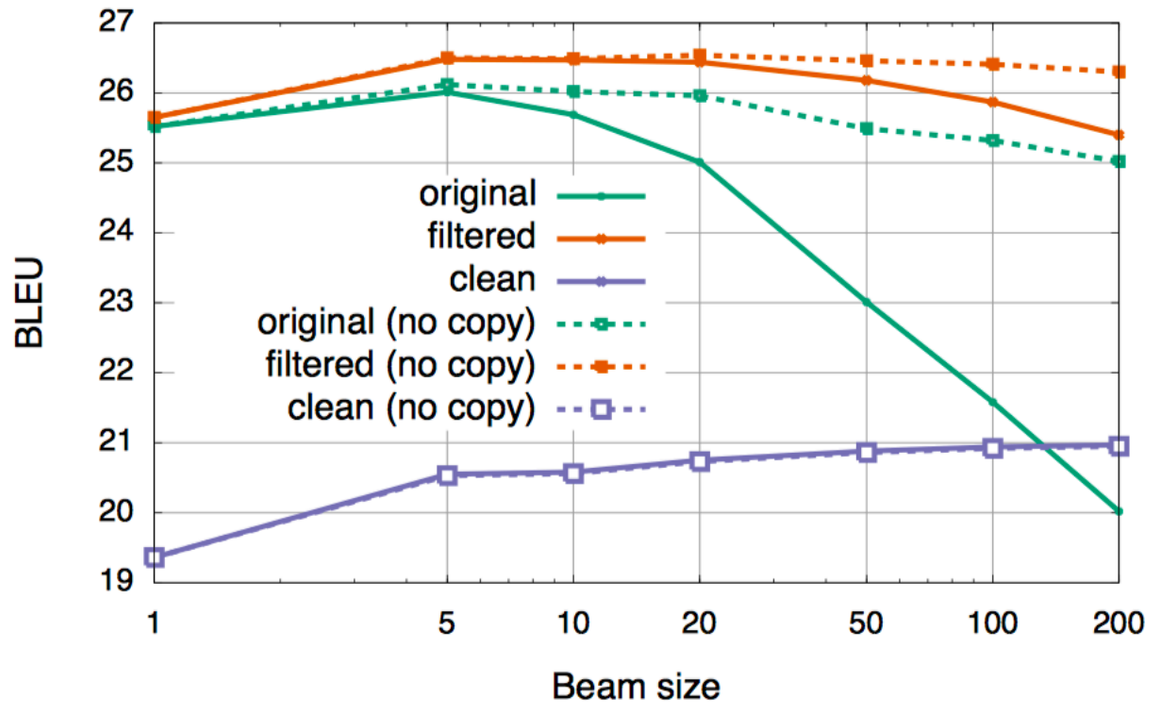


Figure 5. BLEU on newstest2017 as a function of beam width for models trained on all of the WMT'17 En-De training data (*original*), a filtered version of the training data (*filtered*) and a small but clean subset of the training data (*clean*). We also show results when excluding copies as a post-processing step (*no copy*).

extrinsic uncertainty is (at least partially) responsible for the degradation of performance of large beams.

Mixture Models for Diverse Machine Translation: Tricks of the Trade

- This paper is an empirical one to apply “mixture model” to NMT with different tricks to stablize training.

Formally, given a source sentence x and reference translation y, a mixture model introduces a multinomial latent variable $z \in \{1, \cdots, K\}$, and decomposes the marginal likelihood as:

$$p(y|x;\theta) = \sum_{z=1}^{K} p(y, z|x;\theta) = \sum_{z=1}^{K} p(z|x;\theta)p(y|z,x;\theta) \tag{1}$$

where the prior $p(z|x;\theta)$ and likelihood $p(y|z,x;\theta)$ are learned functions parameterized by θ. Each value of z represents an *expert*, and the posterior probability:

$$p(z|x,y;\theta) = \frac{p(z|x;\theta)p(y|z,x;\theta)}{\sum_{z'} p(z'|x;\theta)p(y|z',x;\theta)} \tag{2}$$

can be viewed as the *responsibility* each expert takes for explaining an observation (x, y).

Training & decoding

Training Given a training set $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, we want to find θ that maximizes the log likelihood. To this end, for each training example we compute the gradient:

$$\nabla_\theta \log p(y^{(i)}|x^{(i)};\theta) = \sum_z p(z|x^{(i)}, y^{(i)};\theta) \cdot \nabla_\theta \log p(y^{(i)}, z|x^{(i)};\theta) \quad (3)$$

and train the model with the EM algorithm (Dempster et al., 1977) by iteratively applying the following two steps:

E-step: estimate the responsibilities of each expert $r_z^{(i)} \leftarrow p(z|x^{(i)}, y^{(i)};\theta)$ using the current parameters θ;

M-step: update θ through each expert with gradients $\nabla_\theta \log p(y^{(i)}, z|x^{(i)};\theta)$ weighted by their responsibilities $r_z^{(i)}$.

Decoding All the decoding strategies for $p(y|x;\theta)$ of a baseline model can be applied equally to $p(y|z,x;\theta)$ in a mixture model. We adopt the most straightforward one: generating K hypotheses by first enumerating z and then greedily decoding $\hat{y}_t = \arg\max_y p(y|\hat{y}_{1:t-1}, z, x;\theta)$. Notably, this decoding procedure is efficient and easily parallelizable.

Degeneracy of mixture model

Degeneracies Unfortunately, naïve implementations of mixture models for text generation are prone to two major types of degeneracies:

D1: Only one component gets trained (Eigen et al., 2014; Shazeer et al., 2017) because of the "rich gets richer" effect whereby, once a component is slightly better than others, it is always picked while the other components starve and are eventually never used.

D2: The latent variable is ignored, similar to the collapse of variational auto-encoders where the posterior is always equal to the prior (Bowman et al., 2016).

In both cases, the model operates like a baseline model without any benefit from the latent variable. In practice, however, the chance of these degeneracies is heavily affected by a number of design decisions, which we describe in the following subsections.

Hard assignment in E step

E-step (hard): estimate the responsibilities of each expert $r_z^{(i)} \leftarrow \mathbb{1}[z = \arg\max_{z'} p(y^{(i)}, z'|x^{(i)}; \theta)]$ using the current parameters θ.

The specialization of experts implies that the responsibility $p(z|x, y; \theta)$ for explaining a particular translation y should be large for only one z, i.e., only one element in the sum $\sum_z p(y, z|x; \theta)$ dominates. To encourage this, a *hard mixture model* directly optimizes for $\max_z p(y, z|x; \theta)$ by assigning full responsibility for each training example to the expert with the largest joint probability.

Variants on soft/hard & prior

To encourage all experts to generate good hypotheses for any source sentence, we may set the prior $p(z|x;\theta)$ to be uniform. This can prevent the model from collapsing into only one working expert with extreme $p(z|x;\theta)$ value. This also aligns with our simple decoding strategy that generates a single hypothesis from each expert.

The choices of soft (s) versus hard (h) mixture model (M) and learned prior (lp) versus uniform prior (up) give us four model variants with different loss functions:

$$\mathcal{L}_{\text{sMlp}}(\theta) = \mathbb{E}_{(x,y)\sim\text{data}}\left[-\log\sum_z p(z|x;\theta)p(y|z,x;\theta)\right]$$

$$\mathcal{L}_{\text{sMup}}(\theta) = \mathbb{E}_{(x,y)\sim\text{data}}\left[-\log\sum_z p(y|z,x;\theta)\right]$$

$$\mathcal{L}_{\text{hMlp}}(\theta) = \mathbb{E}_{(x,y)\sim\text{data}}\left[\min_z -\log p(z|x;\theta)p(y|z,x;\theta)\right]$$

$$\mathcal{L}_{\text{hMup}}(\theta) = \mathbb{E}_{(x,y)\sim\text{data}}\left[\min_z -\log p(y|z,x;\theta)\right] \quad (4)$$

where the constant $\log K$ term is omitted for models with a uniform prior.

Models & shared params

Model Architecture All models use a very similar architecture, built using the Transformer (Vaswani et al., 2017) implementation in the Fairseq toolkit (Ott et al., 2019). The encoder and decoder have 6 blocks. The number of attention heads, embedding dimension and inner-layer dimension are 8, 512, 2048 for the "base" configuration and 16, 1024, 4096 for the "big" configuration, respectively. We use the "base" configuration to compare mixture model variants in §5.1, and the "big" configuration for extended experiments in §5.2.

We test different model variants using both independent and shared parameters. With independent parameterization, each expert has a different decoder network. With shared parameterization, experts use the same decoder network but the beginning-of-sentence token at the start of the target sequence is replaced with an embedded representation of the latent variable. This requires a negligible increase in parameters over the baseline model.

Metrics

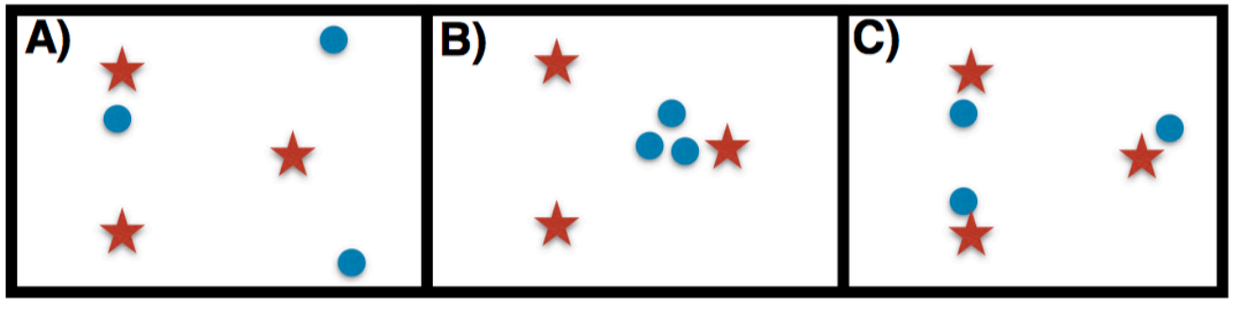
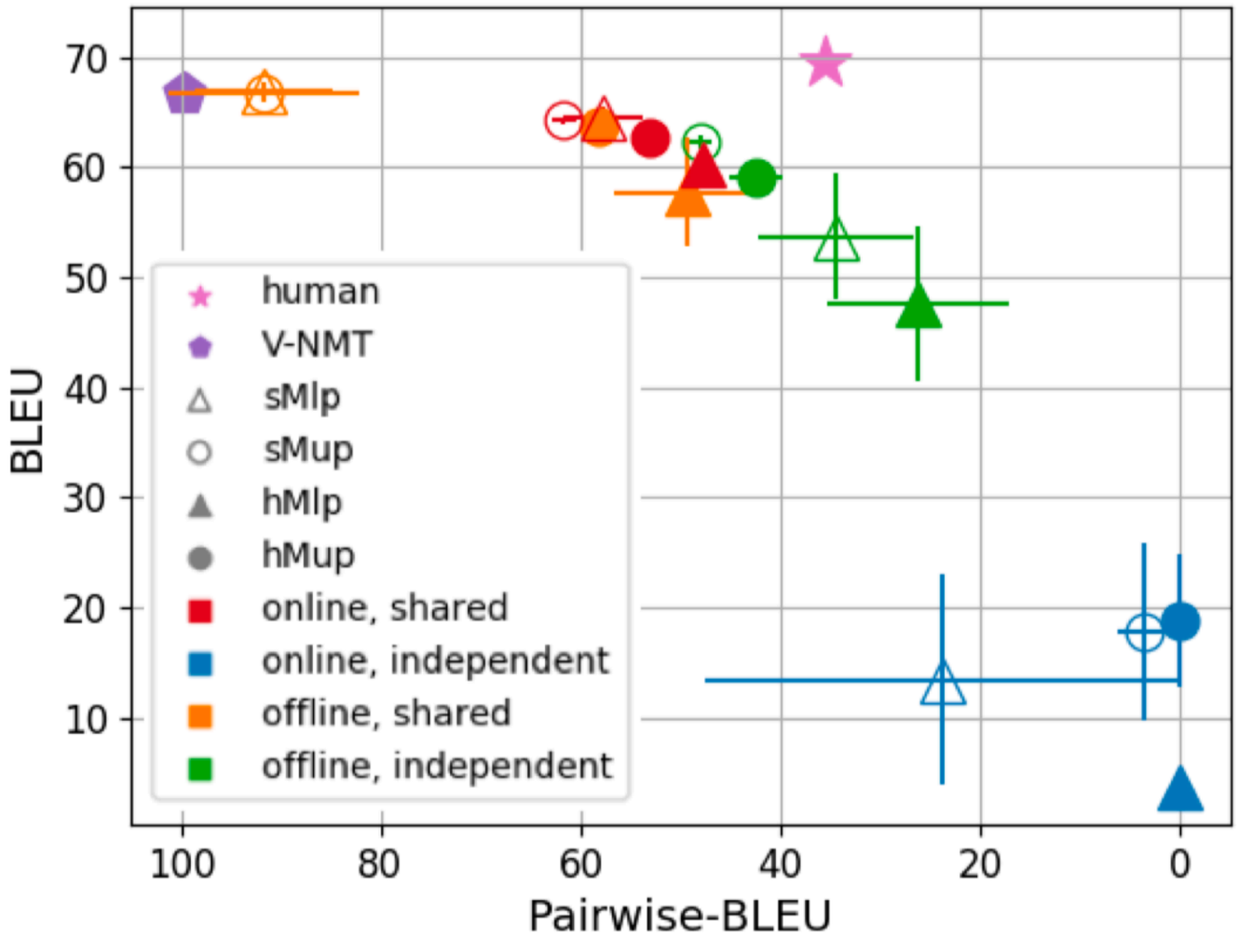


Figure 1. Toy illustration of how the metrics behave in various scenarios. Red stars represent human references, while blue dots represent system hypotheses. **A):** When there are poor hypotheses in the set, the BLEU is low and Pairwise-BLEU is low; this is the typical scenario of sampling hypotheses as well as degeneracy D1 described in §3. **B):** When hypotheses cluster closely together and in vicinity of a reference, the BLEU is high and Pairwise-BLEU is high; this is the typical scenario of beam search hypotheses and also degeneracy D2. **C):** When hypotheses match references (ideal case), the BLEU is high and Pairwise-BLEU matches human Pairwise-BLEU.

- **Pairwise-BLEU:** To measure similarity among the hypotheses, we compare them with each other and compute $\text{BLEU}\{([\hat{y}^j], \hat{y}^k)\}_{x \in \text{data}, j \in [K], k \in [K], j \neq k}$.[3] The more diverse the hypothesis set, the lower the Pairwise-BLEU. Ideally, we would like a model with Pairwise-BLEU matching human Pairwise-BLEU.

- **BLEU:** We calculate human BLEU in a leave-one-out manner by computing $\text{BLEU}\{([y^{-m}], y^m)\}_{x \in \text{data}}$ for $m \in [M]$ and then averaging the M scores. We also use $M - 1$ references when computing system BLEU, to be comparable with human scores, i.e. average $\text{BLEU}\{([y^{-m}], \hat{y}^k)\}_{x \in \text{data}, k \in [K]}$ for $m \in [M]$. This measures the *overall quality* of a hypothesis set. If this metric scores low, it implies that some generated hypotheses have poor quality.

Results



Overall, there are a few variants that work robustly as indicated by the small variance from random initialization, and strike a good balance between generation quality and diversity as indicated by proximity to human performance: all online-shared models, offline-shared hMup, as well as offline-independent sMup and hMup. All of these models perform well, offering slightly different trade-offs between quality and diversity.

If we also account for computational and memory cost, methods using shared parameters and hard EM are preferable, since the extra parameters are negligible and it requires only a single backward pass for the selected expert.

Finally we pick hMup (online-shared)

Results

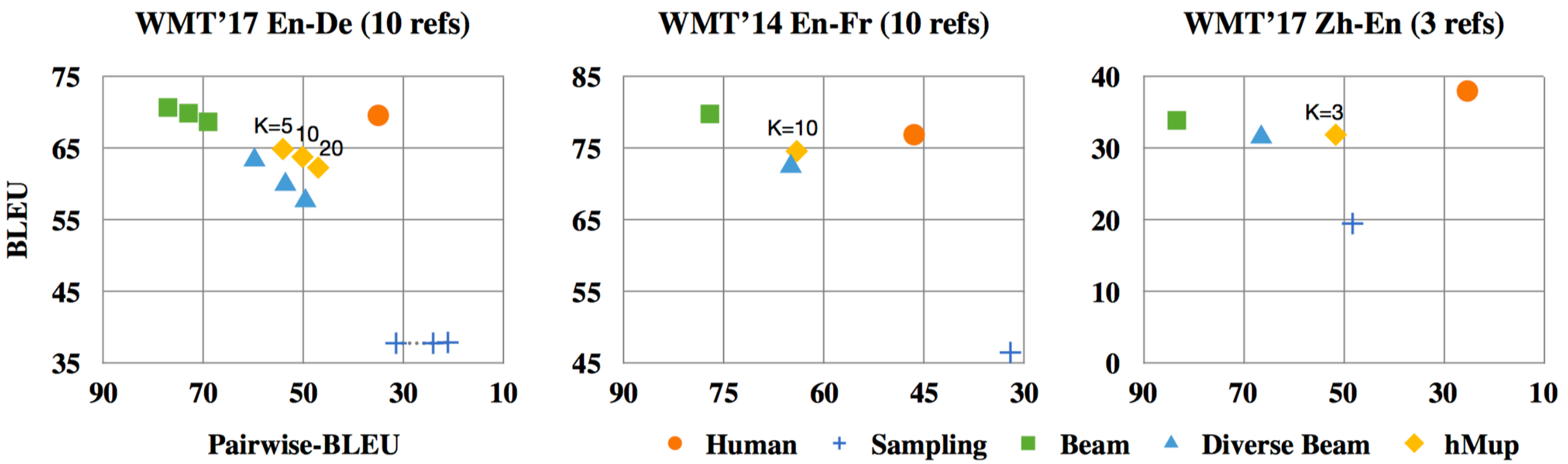


Figure 3. Comparison of the hMup mixture model with baselines on three WMT datasets. The mixture model provides the best trade-off between translation quality and diversity.

	Pairwise-BLEU			BLEU		
	en-de	en-fr	zh-en	en-de	en-fr	zh-en
sampling	24.1	32.0	48.2	37.8	46.5	19.5
beam	73.0	77.1	83.4	69.9	79.8	33.9
div-beam	53.7	64.9	66.5	60.0	72.5	31.6
hMup	50.2	64.0	51.6	63.8	74.6	31.9
human	35.5	46.5	25.3	69.6	76.9	38.0