Generating Diverse Translations via Weighted Fine-tuning and Hypotheses Filtering for the Duolingo STAPLE Task

Sweta Agrawal
Department of Computer Science
University of Maryland
sweagraw@cs.umd.edu

Marine Carpuat
Department of Computer Science
University of Maryland
marine@cs.umd.edu

Abstract

This paper describes the University of Maryland's submission to the Duolingo Shared Task on Simultaneous Translation And Paraphrase for Language Education (STAPLE). Unlike the standard machine translation task, STAPLE requires generating a set of outputs for a given input sequence, aiming to cover the space of translations produced by language learners. We adapt neural machine translation models to this requirement by (a) generating n-best translation hypotheses from a model fine-tuned on learner translations, oversampled to reflect the distribution of learner responses, and (b) filtering hypotheses using a feature-rich binary classifier that directly optimizes a close approximation of the official evaluation metric. Combination of systems that use these two strategies achieves F1 scores of 53.9% and 52.5% on Vietnamese and Portuguese, respectively ranking 2nd and 4th on the leaderboard.

1 Introduction

While machine translation (MT) typically produces a single output for each input, scoring and generation for second language learning applications might benefit from systems whose outputs better capture the diversity of translations produced by language learners. The Duolingo Simultaneous Translation And Paraphrase for Language Education (STAPLE) shared task (Mayhew et al., 2020) provides a framework for developing and testing such systems, grounded in real translations produced by English learners into five native languages (Portuguese, Vietnamese, Hungarian, Japanese, Korean). In this task, given an English sentence prompt, systems are asked to produce a set of translations for that prompt, and are scored based on how well their outputs cover human-curated acceptable translations, weighted by the likelihood that an English learner would respond with each translation (Table 1).

Prompt	is my explanation clear?	
Output	minha explicação está clara?	0.267
	minha explicação é clara?	0.161
	a minha explicação está clara?	0.111
	a minha explicação é clara?	0.088
	minha explanação está clara?	0.057
	está clara minha explicação?	0.044
	minha explanação é clara?	0.039

Table 1: STAPLE data: given a prompt in English, translation alternatives are weighted according to Learner Response Frequency (LRF)

While the multiple translations can be viewed as paraphrases, we propose to address the STAPLE task primarily as a MT task to better understand the strengths and weaknesses of neural MT architectures for generating multiple learner-relevant translations. Given a Transformer model for the language pair of interest, we use beam search to generate n-best translation candidates. However, since n-best lists are known to lack diversity, we propose to generate hypotheses that better match the requirements of the STAPLE task via:

1. **Frequency-Aware n-Best Lists**: We encourage hypotheses to reflect the diversity and frequency of learner responses by fine-tuning models on STAPLE data, oversampling translation options to reflect learner preferences.

2. **Hypothesis Filtering**: We filter the resulting n-best lists using a binary classifier which identifies good translations that are likely to be produced by a learner.

Controlled experiments and analysis show the benefits of both strategies. Our final submission which includes both techniques achieves F1 scores of 53.9% and 52.5% for en-vi and en-pt respec-

tively, reaching a rank of 2nd and 4th on the leaderboard, only 2 points below the top scoring system. For completeness, we also submitted systems for the remaining language pairs using Frequency-Aware n-best lists: our system ranked 2nd for Japanese and 3rd for Korean and Hungarian.

2 Background

Unlike in the STAPLE task, recent attempts at generating multiple translations for a single source have targeted output variability along specific stylistic dimensions (Sennrich et al., 2016b; Rabinovich et al., 2016; Niu et al., 2018; Agrawal and Carpuat, 2019) or produce diverse outputs without a specific use case (Kikuchi et al., 2016; Shu et al., 2019). The techniques used can be divided in three categories: (a) constrain the decoding process to generate diverse candidates (Li and Jurafsky, 2016; Li et al., 2015; Cho, 2016); (b) optimize via a diversity promoting loss function (Li et al., 2015); (c) expose the model to different translation candidates with side-constraints (Rabinovich et al., 2016; Sennrich et al., 2016a; Niu et al., 2018; Agrawal and Carpuat, 2019; Shu et al., 2019) or without (Shen et al., 2019). Since it is unclear what dimensions of variations are captured in the STAPLE translation, we focus instead on improving n-best lists generated by a standard neural MT model.

Source texts with multiple references have mostly been used to evaluate rather than train MT systems (Papineni et al., 2002; Banerjee and Lavie, 2005; Qin and Specia, 2015). Evaluation sets with 4 or 5 references have been converted to single-reference training samples (Zheng et al., 2018) to improve MT training, but reference translations vary in arbitrary ways and often exhibit poor diversity, mostly limited to translationese effects. The STAPLE data presents an opportunity to explore multiple translations generated in a more comprehensive fashion.

3 Approach

3.1 Frequency-Aware Hypotheses Generation

While neural MT systems can generate multiple translation candidates per source using beam search, the n-best translations often lack diversity. One issue is that systems are trained on single-translation training samples. We propose to tailor MT to the STAPLE task by fine-tuning models on LRF-weighted multi-reference samples to obtain more diverse translations and a ranking that better reflect learner preferences.

Given the STAPLE data for a language pair, where the i-th training example, $(\mathbf{e}_i, \mathbf{F}_i, \mathbf{W}_i)$ includes a source sentence in English, a reference set $\mathbf{F}_i = \{\mathbf{f}_i^1, \mathbf{f}_i^2, ..., \mathbf{f}_i^K\}$ of $\mathbf{K}$ translations and corresponding LRF weights $\mathbf{W}_i = \{\mathbf{w}_i^1, \mathbf{w}_i^2, ..., \mathbf{w}_i^K\}$, we create MT training samples by copying the translation pair $(\mathbf{e}_i, \mathbf{f}_i^j)$, $\mathbf{w_i^j} \times O$ times.[1] Given model parameters θ, this yields a weighted cross-entropy loss:

$$\mathcal{L}_{lrf}(\theta) = \sum_{i=1}^{M} \sum_{j=1}^{K} (w_i^j \times O) \log Pr(f_i^j | e_i; \theta) \quad (1)$$

3.2 Hypothesis Filtering as Binary Classification

Even when informed by STAPLE data and LRF scores, n-best lists might include translations that are not in the reference set, due to translation errors or selecting paraphrases that do not match language learners' preferences. We design a binary classifier that further filters the n-best lists by predicting for each hypothesis whether or not it should be included in the final set. This lets us define features based on the complete prompt and hypothesis sequence pairs, while the MT model generates the hypothesis incrementally.

Let $D = \{(\mathbf{e}_i, \hat{\mathbf{f}}_i^1, \hat{\mathbf{f}}_i^2, ..., \hat{\mathbf{f}}_i^N)\}_1^M$ represent the n-best list generated via beam search for all the source prompts in the training dataset: $\mathbf{e}_i$ corresponds to the i-th source prompt and $\hat{\mathbf{f}}_i^j$ corresponds to the j-th candidate hypothesis extracted via beam search. $\mathbf{x}_j^i$ represents the feature vector extracted from the source ($\mathbf{e_i}$) and j-th candidate hypothesis ($\hat{\mathbf{f}}_\mathbf{i}^\mathbf{j}$) and $\mathbf{y}_i^j$ is a binary label indicative of whether the candidate hypothesis, $\hat{\mathbf{f}}_i^j$, is found in the gold standard data. The classification model $f : X \rightarrow R$ maps the feature vector to a real value, where, f is a two-layer Neural Network (NN) to enable learning feature combinations.

Features We aim to capture the quality of a source-hypothesis pair using multiple sentence-level features:

- Length features $|\hat{f}|$, $|e|$, $\frac{|\hat{f}|}{|e|}$, $\frac{|e|}{|\hat{f}|}$ might indicate mismatches between source and target content.
- Word alignment features have proved useful to identify semantic divergences in bitext

[1] We set $O = 1000$ in practice.

(Munteanu and Marcu, 2005; Vyas et al., 2018). We use the Forward and Reverse Alignment score, the count of unaligned words for source and target, and the top three largest fertilities for source and target.

- Scores from various MT models as often done when reranking n-best lists (Cherry and Foster, 2012; Neubig et al., 2015; Hassan et al., 2018) including a left-to-right model, a right-to-left model, and a target-to-source model, which provide different views of the example and might better estimate the adequacy of the translation than the original MT model score.
- Target 5-gram language model score to estimate the fluency of the hypothesis.

Loss We optimize a Soft Macro-F1 objective (Hsieh et al., 2018) function to approximate the official evaluation metric.[2] The true positive (tp), false positive(fp), and true negative (tn) rate for each source prompt e_i are estimated as:

$$\mathrm{tp}_{\mathrm{e_i}} = \sum_{t=1}^{N} \hat{y}_i \times y_i$$

$$\mathrm{fp}_{\mathrm{e_i}} = \sum_{t=1}^{N} \hat{y}_i \times (1 - y_i)$$

$$\mathrm{tn}_{\mathrm{e_i}} = \sum_{t=1}^{N} (1 - \hat{y}_i) \times y_i$$

Then, the precision, recall, F1 for a source e_i, and the loss are defined as:

$$\mathrm{P}_{\mathrm{e_i}} = \frac{\mathrm{tp}_{\mathrm{e_i}}}{\mathrm{tp}_{\mathrm{e_i}} + \mathrm{fp}_{\mathrm{e_i}} + \epsilon}$$

$$\mathrm{R}_{\mathrm{e_i}} = \frac{\mathrm{tp}_{\mathrm{e_i}}}{\mathrm{tp}_{\mathrm{e_i}} + \mathrm{fn}_{\mathrm{e_i}} + \epsilon}$$

$$\mathrm{F1}_{\mathrm{Macro}_{\mathrm{e_i}}} = \frac{2 \times \mathrm{P}_{\mathrm{e_i}} \times \mathrm{R}_{\mathrm{e_i}}}{\mathrm{P}_{\mathrm{e_i}} + \mathrm{R}_{\mathrm{e_i}} + \epsilon}$$

$$\mathrm{Loss} = \sum_{i=1}^{M} (1 - \mathrm{F1}_{\mathrm{Macro}e_i})$$

4 Experiment Settings

4.1 Data

STAPLE Data The shared task provides English source prompts, associated with high-coverage sets of plausible translations in five other languages. These translations are weighted and ranked according to LRF scores indicating which translations are more likely. About 3000 prompts per language are available (see Table 2 for details) and the number of reference translations available per prompt vary across languages (mean: 174.2, variance: 116). Figure 1 illustrates the differences in LRF distributions across languages: for languages with many references per prompt (e.g. en-ja, en-ko), the gap between the top-1 and the mean LRF value is small, indicating an almost uniform distribution. Average top-1 LRF scores also vary across languages (e.g en-vi: 0.25, en-ja: 0.05) depending upon the number of references available per prompt.

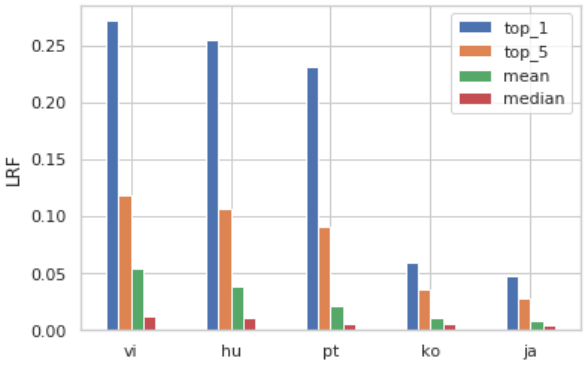


Figure 1: Average of the top-1, top-5, mean and median LRF values across source prompts: the LRF distribution is more uniform for languages with many more references per prompt (e.g. en-ja).

For system development, we divide the STAPLE dataset into train, development and test datasets using 72%, 8%, and 20% of source prompts respectively. We refer to these subsets as **STAPLE train**, **internal dev** and **internal test**. Note that the last two differ from the official blind development and test sets available to participants on codalab.

Other Bitexts We use bitext from OpenSubtitles (Tiedemann, 2012) and Tatoeba (Tiedemann, 2012) as described in Table 3. The Tatoeba corpus provides multiple reference translations for some sources (with 2 translation per source on average), but unlike in the STAPLE data, these translations are not weighted by frequency of usage.

Preprocessing All datasets are pre-processed using Moses tools for normalization, tokenization and lowercasing. We further segment tokens into subwords using a joint source-target Byte Pair Encoding (Sennrich et al., 2016c) model with 32,000

[2] Preliminary experiments showed that a LRF-weighted version of this loss resulted in unstable training and inconsistent results depending on initialization.

Language	Source					Target					T/S
	Train	Dev	Test	Types	Tokens	Train	Dev	Test	Types	Tokens	
en-pt	2.8K	300	800	2.3K	3.8M	380K	42K	104K	8.7K	4M	131
en-vi	2.5K	280	700	2.3K	950K	142K	14K	38K	1.7K	1.3M	56
en-ja	1.8K	200	500	1.3K	3.8M	600K	65K	166K	4K	6.8M	342
en-ko	1.8K	200	500	1.3K	3M	500K	57K	137K	17K	2.6M	280
en-hu	2.8K	320	800	1.5K	1.1M	182K	21K	47K	11K	1M	62

Table 2: STAPLE data statistics: segments in our train/dev/test split, overall vocabulary statistics and average translations per source prompt (T/S).

Language	OpenSubtitles	Tatoeba
en-pt	47.2M	196K
en-vi	3M	5.3K
en-ja	1.8M	200K
en-ko	1.2M	2.7K
en-hu	34.5M	102K

Table 3: Additional bitext used for training and fine-tuning MT models

operations. For Japanese, we use kytea [3] toolkit for word tokenization.

4.2 MT configurations

Model Architecture We use the Transformer model implemented in the Sockeye toolkit[4] as a baseline MT system. Both encoder and decoder are 6-layer Transformer models with model size of $1,024$, feed-forward network size of $4,096$, and 16 attention heads. We adopt label smoothing and weight tying. We tie the output weight matrix with the target embeddings. We use Adam optimizer with initial learning rate of 0.0002.

Experimental Conditions We train several models with the above configuration:

- **OpenSubs** a baseline model trained and validated on the OpenSubtitles bitext.
- **Unweighted** builds on the baseline by fine-tuning on multi-reference samples including the Tatoeba bitext and STAPLE train. We create one training sample per source-reference pair, and the resulting samples are not weighted. We use the internal dev set (1-best reference only) as a validation set.
- **Frequency-Aware** is fine-tuned as the unweighted model except that STAPLE train is oversampled as described in § 3.1.

We generate n-best list of translations for various models by running beam search with a beam size corresponding to the desired n.

4.3 Filtering configurations

Classifier The 2-layer feed-forward NN has 5 hidden units and 2 output units. It is trained with the Adam optimizer with an initial learning rate of 0.001 and runs for 2000 epochs on the internal dev set. The best model is selected based on internal test set performance. We consider two losses: the soft macro F1 loss which approximates the official evaluation metric (§ 3.2) and the standard cross-entropy loss as a baseline.

Reranking Baseline We compare our NN based classifer with a standard MT n-best list reranker trained on the internal dev set. We use the n-best batch MIRA ranker (Cherry and Foster, 2012) included in Moses. A threshold to filter candidates in the reranked list is selected by maximizing the Weighted Macro F1 on the internal dev dataset.

Features We use eflomal [5] trained on the Opensubtitles dataset to obtain word alignment between source and translation hypotheses. The language model is trained with the kenlm (Heafield, 2011) toolkit with default hyper-parameters[6] on the target side of the Opensubtitles and the STAPLE dataset. The Right-to-left and Target-to-source MT models were trained on OpenSubtitles (same configuration as in § 4.2).

[3] https://github.com/neubig/kytea
[4] https://github.com/awslabs/sockeye
[5] https://github.com/robertostling/eflomal
[6] https://github.com/kpu/kenlm

5 Evaluation

We evaluate the lowercased detokenized output of the systems on our internal test dataset using:

Weighted Macro F1 This is the official scoring metric which quantifies how the set of system outputs covers the human-curated acceptable translations, weighted by the LRF of each translation. It is defined as the harmonic mean of unweighted precision (P) and weighted recall (WR) calculated for each prompt $\mathbf{e_i}$, and averaged over all the prompts in the corpus. Specifically, using the same notation as introduced in § 3.1, for each translation $\mathbf{T_i}$ generated by the MT model, we have:

$$\mathrm{WTP}_{\mathrm{e_i}} = \sum_{t \in T_i} \sum_{f_i^j \in F_i} 1[t == f_i^j] w_i^j$$

$$\mathrm{WFN}_{\mathrm{e_i}} = \sum_{f_i^j \notin T_i} w_i^j$$

$$\mathrm{WR}_{\mathrm{e_i}} = \frac{\mathrm{WTP}_{\mathrm{e_i}}}{\mathrm{WTP}_{\mathrm{e_i}} + \mathrm{WFN}_{\mathrm{e_i}}}$$

The weighted Macro F1 (WMF1) is then given by:

$$\mathrm{WMF1}_{\mathrm{e_i}} = \frac{2 \times \mathrm{P}_{\mathrm{e_i}} \times \mathrm{WR}_{\mathrm{e_i}}}{\mathrm{P}_{\mathrm{e_i}} + \mathrm{WR}_{\mathrm{e_i}}}$$

$$\mathrm{WMF1} = \frac{1}{M} \sum_{i}^{M} \mathrm{WMF1}_{\mathrm{e_i}}$$

BLEU@1 We also report the translation quality of the 1-best neural MT output compared against the highest LRF reference translation using the standard BLEU metric (Papineni et al., 2002).

6 Experiment Results

6.1 Impact of Frequency-Aware Fine-Tuning

Table 4 summarizes the evaluation of n-best lists obtained with our neural MT systems.

Baselines We confirm that the neural MT configuration is sound by comparing our neural MT baseline to the provided AWS system. Our baseline ("OpenSubs") improves the BLEU@1 score by 2 points for en-pt, and remains 6 points lower for en-vi, as can be expected given the smaller size of the OpenSubtitles training set. However, the "OpenSubs" n-best lists improve over the AWS baseline according to the official task metric (WMF1), establishing that this system is a good starting point for fine-tuning.

Fine-Tuning The Frequency-Aware n-best hypotheses consistently yield the best Weighted Recall and Weighted Macro-F1 scores for all languages. The improvement in recall and therefore F1 score is largest for en-ja and en-ko which have larger translation reference sets (Table 4). Frequency-Aware oversampling also improves precision over the Unweighted model for all but one language (en-pt). The impact on the auxiliary BLEU@1 metric is less consistent: the Frequency-Aware system achieves the best BLEU@1 in 3 out of 5 languages, but outperforms the OpenSubs baseline in 4 out of 5. BLEU@1 drops when fine-tuning on all the samples without weighting (Unweighted) which we attribute to the increased uncertainty during training as the model is exposed to many different translations for the same source English text.

Overall, these results show the benefits of fine-tuning on task-relevant data and shows that incorporating LRF weights via oversampling improves the ranking of n-best hypotheses. This is further illustrated in Table 5, which shows the top 10 Vietnamese translations for two randomly sampled source prompts: the Frequency-Aware n-best list yields Weighted Recall of 81% at a Precision of 60% and 76% at a Precision of 100% for the two source prompts respectively, illustrating that the model generates high-quality candidates that cover reference translations well, but not perfectly.

N-Best List Quality How well do n-best translations cover the space of reference learner translations? Figure 2 shows the impact of increasing the decoding beam (and resulting n-best list size) from 10 to 500 for the Frequency-Aware model. For en-pt, while weighted recall increases up to 66%, the drop in precision hurts the weighted F1 score. The oracle F1 score, which represents the Weighted Macro F1 at a Precision of 100%, also increases gradually, reaching a score of 76%. This suggests that the raw n-best lists contain many useful translation candidates but need to be filtered down to better match translations preferred by language learners.

6.2 Impact of Hypothesis Filtering

Due to time constraints, we explore the impact of hypothesis filtering only for en-pt and en-vi.

Filtering consistently improves Precision and Weighted Macro F1 (Table 6). The binary classifier that optimizes Soft Macro-F1 performs best,

Language	Method	BLEU@1	n-best size	P	WR	WMF1
en-pt	AWS	68.9	1	**86.67**	14.47	21.60
	OpenSubs	70.9	10	49.66	39.18	37.39
	Unweighted	61.5	10	72.69	40.58	46.11
	Frequency-Aware	**76.6**	10	67.31	**44.34**	**47.4**
en-vi	AWS	61.4	1	**65.09**	13.32	19.57
	OpenSubs	55.2	10	29.10	31.38	25.76
	Unweighted	49.8	10	56.43	42.91	41.00
	Frequency-Aware	**71.9**	10	61.61	**54.37**	**51.87**
en-ja	AWS	**50.6**	1	**67.68**	2.18	4.01
	OpenSubs	32.7	50	2.94	3.47	2.52
	Unweighted	30.1	50	45.71	21.21	24.88
	Frequency-Aware	42.4	50	47.29	**22.83**	**26.57**
en-hu	AWS	63.4	1	**83.70**	18.12	27.12
	OpenSubs	**64.4**	10	41.51	42.6	37.83
	Unweighted	26.2	10	47.11	29.7	31.62
	Frequency-Aware	51.4	10	52.22	**41.05**	**41.69**
en-ko	AWS	27.9	1	**60.68**	2.26	4.11
	OpenSubs	9.2	50	12.53	7.41	7.20
	Unweighted	14.8	50	33.82	18.8	19.78
	Frequency-Aware	**30.2**	50	35.31	**20.92**	**21.94**

Table 4: Frequency-Aware systems outperform both OpenSubs and Unweighted models for all languages. The size of the n-best list for each model was selected based on the WMF1 score on the internal test set.

as the loss leads to a better balance between Precision and Weighted Recall than cross-entropy. The classifier outperforms the MIRA reranker. Since the reranker is trained to maximize BLEU@1, it tends to prefer candidates that are lexically similar to the top reference translation and misses some of the more diverse learner translations. This confirms the benefits of framing the selection of candidate hypothesis as binary classification.

Ablation Experiments show that the MT scores are the most useful of the features used, as they capture not only the generation probability of a candidate hypothesis but estimate adequacy via the Target-to-source neural MT model (Table 8). Length features help precision but not recall, while the alignment and language model scores have little impact overall. This suggests that the classifier could benefit from improved feature design and selection in future work.

6.3 Analysis of Translation Diversity

How diverse are the translations returned by various system configurations? Following Zhang et al. (2018), we quantify diversity using the entropy of k-gram distributions within a translation set:

$$\text{Ent-k (V)} = -\frac{1}{\sum_w F(w)} \sum_{w \in V} F(w) \log \frac{F(w)}{\sum_w F(w)}$$

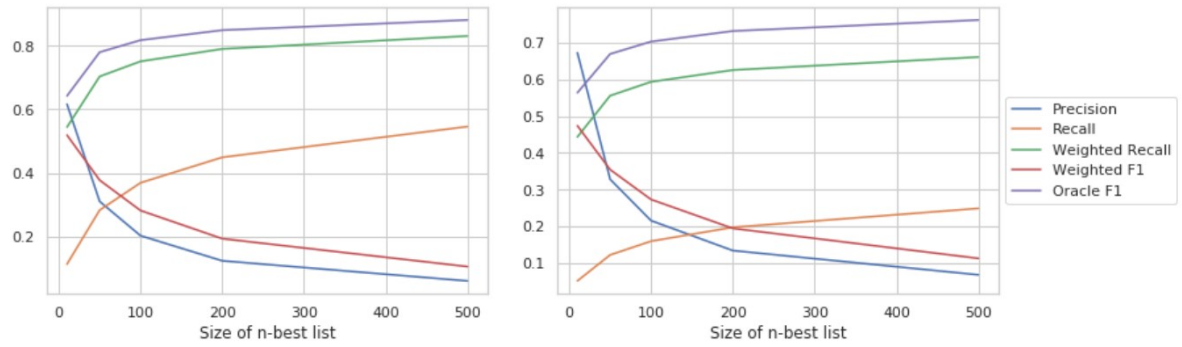
where V is the set of all k-grams that appear in the translation set, and $F(w)$ denotes the frequency of w in the translations. The higher the Ent-k score, the greater the diversity.

Fine-tuned models improve the diversity of 10-best lists compared to the "OpenSubs" baseline for both en-vi and en-pt (Table 9). Overall filtering bridges 40% and 25% of the gap between baseline and reference learner translations for en-pt and en-vi respectively.

6.4 System Combinations

A manual examination of translation sets returned by different models suggest that they make complementary errors. We therefore consider combining system outputs by taking the union of the set of translations they return. We evaluate the following combinations (Table 7):

C1 Frequency-aware (10-best) + Unweighted (10-best)



(a) en-vi (b) en-pt

Figure 2: Increasing the size of n-best list with the Frequency-Aware system improves the coverage of learner translations for en-pt and en-vi. Oracle F1 is the Weighted Macro F1 at a Precision of 100% and represents the upper bound on WMF1 that can be achieved for a given n-best list.

Input: We live near the border.	**LRF**
chúng tôi sống gần biên giới.	0.250
chúng tôi sống ở gần biên giới.	0.053
chúng tôi sống gần đường biên giới.	0.267
chúng tôi sống bên cạnh biên giới.	0.018
chúng tôi sống ở cạnh biên giới.	0.013
chúng ta sống gần biên giới.	0.036
chúng tôi sống cạnh biên giới.	0.061
chúng tôi sống ở bên cạnh biên giới.	0.004
chúng tôi sống ở gần đường biên giới.	0.052
chúng ta sống ở gần biên giới.	0.004
Precision: 100%, Weighted Recall: 76%	
Input: My family lives in the south.	**LRF**
gia đình tôi sống ở miền nam.	0.285
gia đình của tôi sống ở miền nam.	0.134
gia đình tôi sống ở phương nam.	0.061
gia đình của tôi sống ở phương nam.	0.019
gia đình tôi sống ở phía nam.	0.208
gia đình của tôi sống ở phía nam.	0.099
nhà của tôi sống ở miền nam.	-
nhà tôi sống ở miền nam.	-
nhà của tôi sống ở phương nam.	-
gia đình tôi sống ở nam.	-
Precision: 60%, Weighted Recall: 81%	

Table 5: Frequency-Aware 10-best Vietnamese output for two randomly selected English prompts. LRF values are given for translations found in the reference set.

C2 Frequency-aware (10-best) + Frequency-aware (filtered 50-best)

C3 Unweighted (10-best) + Unweighted (filtered 50-best)

C4 Union of all of the above.

For en-pt and en-vi, it helps to combine higher precision unfiltered 10-best lists, and higher recall filtered 50-best lists. For en-pt, the union of all outputs (C4) performs best overall. Recall increases when combining the Frequency-Aware and the Unweighted model (C1) compared to individual lists (Unweighted: +1.6, Frequency-Aware: +2) without compromising Precision. Similar trends are observed when adding the filtered 50-best list to unfiltered 10-best lists (C2: +2.2, C3: +4.8). For en-vi, a different combination (C2) yields the best result, perhaps due to the smaller set of reference translations per source prompt (en-vi: 56, en-pt: 131) and high Precision of the "Unweighted" model for en-pt.

7 Submitted Systems

We tested our systems on the official blind development set to select the best performing models for final evaluation on the test set. For Portuguese and Vietnamese, our official submissions include frequency-aware hypothesis generation and hypothesis filtering:

en-vi C2: Frequency-aware (10-best) + Frequency-aware (filtered 50-best)

Method	en-vi				en-pt			
	P	WR	WMF1	K	P	WR	WMF1	K
No filtering	31.00	**70.31**	37.69	50	44.75	**57.21**	42.84	50
Reranker	**69.71**	46.85	50.67	9	67.44	41.82	45.51	14
Classifier with CE loss	69.70	47.74	48.77	12	69.26	42.70	46.60	10
Classifier with F1 loss	65.15	55.21	**53.69**	12	67.81	45.71	**48.68**	13
Oracle	100	70.31	77.90	15	100	70.31	66.9	16

Table 6: Filtering n-best lists consistently improves WMF1 and substantially reduces the size of the output set ($\mathbf{K}$)

Method	en-pt				en-vi			
	P	R	WR	WMF1	P	R	WR	WMF1
Unweighted (10-best)	**72.69**	5.53	40.58	46.11	56.43	10.32	42.19	41.00
Unweighted (filtered 50-best)	67.81	9.68	45.71	48.17	63.14	15.23	54.35	51.48
Frequency-Aware (10-best)	67.31	5.07	44.34	47.40	61.61	11.28	54.37	51.87
Frequency-Aware (filtered 50-best)	64.33	6.40	36.94	41.44	**65.15**	15.33	55.21	53.69
C1	65.09	7.13	47.52	49.31	55.04	14.82	57.57	50.73
C2	64.33	7.30	48.67	48.81	60.41	16.07	60.19	**53.57**
C3	66.41	10.32	50.18	50.17	56.19	15.93	54.89	48.05
C4	59.76	**11.60**	**53.56**	**50.79**	53.75	**18.31**	**61.04**	50.78

Table 7: Combination of unfiltered 10-best lists (with better precision) and filtered 50-best lists (with better recall) improves Weighted Macro F1. See § 6.4 for details on combinations.

Features	P	R	WR	WMF1
All	63.71	15.97	55.91	**54.04**
- LM score	65.10	15.35	55.62	53.92
- Alignment	**65.21**	15.27	55.08	53.86
- Length	58.44	**16.49**	**55.98**	52.53
- MT Scores	43.77	10.88	31.02	28.06
Oracle	100	28.28	70.31	77.90

Table 8: Impact of dropping one feature type (§ 3.2) at a time from the "All" configuration for en-vi classifier.

Translations	en-pt		en-vi	
	Ent-4	n	Ent-4	n
OpenSubs	2.34	10	2.53	10
Unweighted	2.60	10	2.65	10
Frequency-Aware	2.59	10	2.67	10
Filtered	**2.95**	13	**2.71**	11
Reference	3.93	131	3.23	56

Table 9: Diversity in translation sets: Filtered sets are more diverse, bridging 40% of the gap between baseline and reference translations for en-pt.

en-pt C4: Frequency-aware (10-best) + Frequency-aware (filtered 50-best) + Unweighted (10-best) + Unweighted (filtered 50-best)

We did not build hypothesis filtering models for the other languages, and submitted systems based only on unfiltered models:

en-ja Frequency-aware (50-best) + Unweighted (50-best)

en-hu Frequency-aware (10-best) + Unweighted (10-best)

en-ko Frequency-aware (50-best) + Unweighted (50-best)

Table 10 and 11 compares our submissions to baselines, as well as top and median submissions across participants, for all the languages. On our focus languages (en-pt and en-vi), where systems benefitted from both frequency-aware generation and filtering models, our submissions obtain a Weighted Macro F1 score of 0.539 for en-vi and 0.525 for en-pt on the official test set, achieving a rank of 2nd and 4th on the leader-board, within 2% of the top performing submission. On the other language pairs, where our submissions did not use

any filtering, Weighted Macro F1 outperform the baselines and median submission consistently. Interestingly on the en-ja task, our system ranks second amongst all the submissions despite not using any filtering.

Method	en-vi	en-pt	en-ja	en-hu	en-ko
AWS	0.210	0.211	0.042	0.298	0.040
Fairseq	0.267	0.151	0.031	0.130	0.054
Median	0.382	0.451	0.214	0.298	0.047
Top	0.547	0.557	0.316	0.598	0.413
Ours	0.537	0.538	0.283	0.492	0.254

Table 10: Excerpt from official results: weighted Macro F1 on the STAPLE dev set

Method	en-vi	en-pt	en-ja	en-hu	en-ko
AWS	0.198	0.213	0.043	0.281	0.041
Fairseq	0.254	0.136	0.033	0.124	0.049
Median	0.377	0.436	0.239	0.452	0.230
Top	0.558	0.552	0.318	0.555	0.404
Ours	0.539	0.525	0.294	0.469	0.255
Rank	2nd	4th	2nd	3rd	3rd

Table 11: Excerpt from official results: weighted Macro F1 on the STAPLE test set

8 Conclusion

We proposed two strategies to obtain multiple outputs that mimic translations by produced by language learners from a standard neural MT model. Our experiments showed that (1) finetuning MT models using all reference translations and their weight yields more diverse n-best hypotheses that better reflect learner preferences, and (2) filtering these n-best lists using a feature-rich classifier trained to maximize an approximation of the STAPLE evaluation metric yields further improvements. Combinations of systems that use these two strategies approach the top scoring submission in the official evaluation.

While these results suggest that some degree of output diversity can be achieved with little change to core neural MT models, oracle scores obtained with unfiltered n-best lists indicate that better modeling the space of learner translations might benefit both candidate generation and the filtering model in future work.

References

Sweta Agrawal and Marine Carpuat. 2019. Controlling text complexity in neural machine translation. 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 1549–1564, Hong Kong, China. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics.

Kyunghyun Cho. 2016. Noisy parallel approximate decoding for conditional recurrent language model. *CoRR*, abs/1605.03835.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Cheng-Yu Hsieh, Yi-An Lin, and Hsuan-Tien Lin. 2018. A deep model with local surrogate loss for general cost-sensitive multi-label learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *CoRR*, abs/1510.03055.

Jiwei Li and Dan Jurafsky. 2016. Mutual information and diverse decoding improve neural machine translation. *CoRR*, abs/1601.00372.

Stephen Mayhew, Klinton Bicknell, Chris Brust, Bill McDowell, Will Monroe, and Burr Settles. 2020. Simultaneous translation and paraphrase for language education. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: Naist at wat2015. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 35–41.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018. Multi-Task Neural Models for Translating Between Styles Within and Across Languages. In *27th International Conference on Computational Linguistics (COLING 2018)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ying Qin and Lucia Specia. 2015. Truly exploring multiple references for machine translation evaluation. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, pages 113–120, Antalya, Turkey.

Ella Rabinovich, Shachar Mirkin, Raj Nath Patel, Lucia Specia, and Shuly Wintner. 2016. Personalized Machine Translation: Preserving Original Author Traits. *arXiv:1610.05461 [cs]*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling Politeness in Neural Machine Translation via Side Constraints. pages 35–40. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. *CoRR*, abs/1902.07816.

Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating diverse translations with sentence codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1827.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218.

Yogarshi Vyas, Xing Niu, and Marine Carpuat. 2018. Identifying semantic divergences in parallel text without annotations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1503–1515. Association for Computational Linguistics.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems*, pages 1810–1820.

Renjie Zheng, Mingbo Ma, and Liang Huang. 2018. Multi-reference training with pseudo-references for neural translation and text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3188–3197, Brussels, Belgium. Association for Computational Linguistics.