The JHU Submission to the 2020 Duolingo Shared Task on Simultaneous Translation and Paraphrase for Language Education

Huda Khayrallah[‡] **Jacob Bremerman**[§] **Arya D. McCarthy**[‡]
Kenton Murray[‡] **Winston Wu**[‡] and **Matt Post**[‡,†]
[‡]Center for Language and Speech Processing, Johns Hopkins University
[†]Human Language Technology Center of Excellence, Johns Hopkins University
[§]University of Maryland, College Park

Abstract

This paper presents the Johns Hopkins University submission to the 2020 Duolingo Shared Task on Simultaneous Translation and Paraphrase for Language Education (STAPLE). We participated in all five language tasks, placing first in each. Our approach involved a language-agnostic pipeline of three components: (1) building strong machine translation systems on general-domain data, (2) fine-tuning on Duolingo-provided data, and (3) generating n-best lists which are then filtered with various score-based techniques. In addition to the language-agnostic pipeline, we attempted a number of linguistically-motivated approaches, with, unfortunately, little success. We also find that improving BLEU performance of the beam-search generated translation does not necessarily improve on the task metric—weighted macro F1 of an n-best list.

1 Introduction

The Duolingo 2020 STAPLE Shared Task (Mayhew et al., 2020) focuses on generating a comprehensive set of translations for a given sentence, translating from English into Hungarian, Japanese, Korean, Portuguese, and Vietnamese. The formulation of this task (§2) differs from the conventional machine translation setup: instead of the n-gram match (BLEU) against a single reference, sentence-level exact match is computed between a list of proposed candidates and a weighted list of references (as in Figure 1). The set of references is drawn from Duolingo's language-teaching app. Any auxiliary data is allowed for building systems, including existing very-large parallel corpora for translation.

Our approach begins with strong MT systems (§3) which are fine-tuned on Duolingo-provided data (§4). We then generate large n-best lists, from which we select our final candidate list (§5). Our entries outperform baseline weighted F1 scores by a factor of 2 to 10 and are ranked first in the official evaluation for every language pair (§6.2).

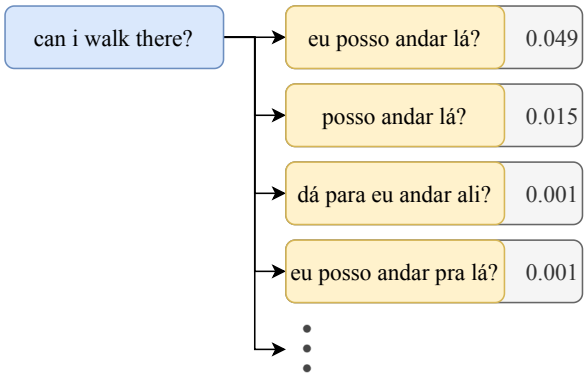

Figure 1: An example English source sentence with its weighted Portuguese target translations. The objective of the task is to recover the list of references, and performance is measured by a weighted F-score.

In addition to our system description, we perform additional analysis (§7). We find that stronger BLEU performance of the beam-search generated translation is not indicative of improvements on the task metric—weighted macro F1 of a set of hypotheses—and suggest this should encourage further research on how to train NMT models when n-best lists are needed (§7.1). We perform detailed analysis on our output (§7.2), which led to additional development on English–Portuguese (§8.1). We also present additional linguistically-informed methods which we experimented with but which ultimately did not improve performance (§8).

2 Task Description

Data We use data provided by the STAPLE shared task (Mayhew et al., 2020). This data consists of a single English prompt sentence or phrase paired with multiple translations in the target lan-

Proceedings of the 4th Workshop on Neural Generation and Translation (WNGT 2020), pages 188–197
Online, July 10, 2020. ©2020 Association for Computational Linguistics
www.aclweb.org/anthology/D19-56%2d

	hu	ja	ko	pt	vi
total prompts	4,000	2,500	2,500	4,000	3,500
mean translations	63	342	280	132	56
median translations	36	192	154	68	30
STD. translations	66	362	311	150	62

Table 1: Statistics over the Duolingo-provided data.

guage. These translations come from courses intended to teach English to speakers of other languages; the references are initially generated by trained translators, and augmented by verified user translations. Each translation is associated with a relative frequency denoting how often it is selected by Duolingo users. Table 1 shows the total number of prompts provided as well as the mean, median, and standard deviation of the number of translations per training prompt. All of the provided task data is lower-cased.

For each language pair, we created an internal split of the Duolingo-provided training data: 100 training prompts for use in validating the MT system (JHU-VALID), another 100 intended for model selection (JHU-DEV),[1] and a 300-prompt test set for candidate selection (JHU-TEST). The remaining data (JHU-TRAIN) was used for training the MT models.

Evaluation metric The official metric is weighted macro F_1. This is defined as:

$$\text{Weighted Macro } F_1 = \sum_{s \in S} \frac{\text{Weighted } F_1(s)}{|S|},$$

where S is all prompts in the test corpus. The weighted F1 is computed with a weighted recall, where TP_s are the true positives for a prompt s, and FN_s are the false negatives for a prompt s:

$$\text{WTP}_s = \sum_{t \in TP_s} \text{weight}(t)$$

$$\text{WFN}_s = \sum_{t \in FN_s} \text{weight}(t)$$

$$\text{Weighted Recall}(s) = \frac{\text{WTP}_s}{\text{WTP}_s + \text{WFN}_s}.$$

Note that recall is weighted (according to weights provided with the gold data), but precision is not.

Evaluation is conducted on lowercased text with the punctuation removed.

[1] However, we discovered that BLEU did not correlate well enough with task performance to be used for this. See §7.1 for more analysis and discussion.

3 Machine Translation Systems

We began by building high-quality state-of-the-art machine translation systems.

Data and preprocessing Additional data for our systems was obtained from Opus (Tiedemann, 2012).[2] We removed duplicate bitext pairs, then reserved 3k random pairs from each dataset to create a validation, development, and test sets of 1k sentence each. The validation dataset is used as held-out data to determine when to stop training the MT system.[3] Table 2 shows the amount of training data used from each source.

The Duolingo data (including the evaluation data) is all lowercased. Since our approach is to overgenerate candidates and filter, we want to avoid glutting the decoder beam with spurious cased variants. For this reason, we lowercase all text on both the source and (where relevant) target sides prior to training. However, it is worth noting that this has a drawback, as source case can provide a signal towards meaning and word-sense disambiguation (e.g., *apple* versus *Apple*).

After lowercasing, we train separate SentencePiece models (Kudo and Richardson, 2018) on the source and target sides of the bitext, for each language. We train a regularized unigram model (Kudo, 2018) with a vocabulary size of 5,000 and a character coverage of 0.995. When applying the model, we set $\alpha = 0.5$. No other preprocessing was applied.

Translation models We used fairseq (Ott et al., 2019) to train standard Transformer (Vaswani et al., 2017) models with 6 encoder and decoder layers, a model size of 512, feed forward layer size of 2048, and 8 attention heads, and a dropout of 0.1. We used an effective batch size of 200k tokens.[4] We concatenated the development data across test sets, and quit training when validation perplexity had failed to improve for 10 consecutive checkpoints.

We trained two sets of models: MODEL1 was trained on just the data above the line in Table 2, while MODEL2 was trained on all the data.

[2] opus.nlpl.eu

[3] The other two were reserved for unanticipated use cases that never materialized.

[4] (batch size 4000) × (2 GPUs) × (update interval 25)

	hu	ja	ko	pt	vi
Europarl (Koehn, 2005)	2,351k	-	-	2,408k	-
GlobalVoices (opus.nlpl.eu/GlobalVoices.php)	194k	822k	37k	1,585k	-
OpenSubtitles (Lison and Tiedemann, 2016)	252,622k	13,097k	8,840k	196,960k	20,298k
Tatoeba (tatoeba.org)	580k	1,537k	-	1,215k	16k
WikiMatrix (Schwenk et al., 2019)	5,682k	9,013k	2,598k	45,147k	17,427k
JW300 (Agić and Vulić, 2019)	19,378k	34,325k	32,356k	39,023k	11,233k
QED (Abdelali et al., 2014)	5,693k	9,064k	9,992k	8,542k	5,482k

Table 2: Number of English word tokens for all datasets used to train the baseline MT models. Just the data above the line was used to train the MODEL1 baseline, all the data was used to train the MODEL2 baseline.

4 Fine-Tuning

After training general-domain machine translation models, we fine-tune them on the Duolingo data.[5] The Duolingo data pairs single prompts with up to hundreds of weighted translations; we turned this into bitext in three ways:

- **1-best:** the best translation per prompt.
- **all:** each translation paired with its prompt.
- **up-weighted:** all possible translations with an additional 1, 9, or 99 copies of the 1-best translation (giving the 1-best translation a weight of 2x, 10x, or 100x the others).[6]

We fine-tune with dropout of 0.1, and an effective batch size of 160k tokens. We sweep learning rates of 1×10^{-4} and 5×10^{-4}.

We withhold a relatively high percentage of the Duolingo training data for internal development (500 prompts total, which ranged from to 12.5 to 20% of the provided data), so we also train systems using all the released data (with none withheld), taking hyperparameters learned from our splits (number of fine-tuning epochs, candidate selection parameters, etc).

5 Candidate Generation and Selection

From the models trained on general-domain data (§3) and refined on in-domain data (§4), we generate 1,000-best translations. For each translation, fairseq provides word-level and length-normalized log-probability scores, which all serve as grist for the next stage of our pipeline: candidate selection.

[5] Training on the Duolingo data directly was less effective.

[6] A better method might be to train using the weights to weight the sentences in training as available in Marian (Junczys-Dowmunt et al., 2018) but that was not available in fairseq, so we improvised.

5.1 Ensembling

For Portuguese only, we experimented with ensembling multiple fine-tuned models in two ways: (a) using models from different random seeds, and (b) using different types of systems.

5.2 Selecting top k hypotheses

As a baseline, we extract hypotheses from the n-best list using the provided `my_cands_extract.py` script.[7] which simply extracts the same number of hypotheses, k, per prompt. To determine how many hypotheses to retain from the model's n-best list, we conduct a sweep over k on JHU-TEST and select the best k per language pair based on weighted macro F1.

5.3 Probability score thresholding

We propose to use the log probability scores directly and choose a cutoff point based on the top score for each prompt.

We consider a multiplicative threshold on the probabilities of the hypothesis, relative to the best hypothesis. For example, if the threshold value is -0.40, for a prompt where the top hypothesis log-probability is -1.20, any hypothesis from the top 1000 with a log-probability greater than or equal to -1.60 will be selected.[8] As in §5.2, we sweep over this threshold value for each language pair and choose the value that results in the highest weighted macro F1 score from JHU-TEST.

[7] github.com/duolingo/duolingo-sharedtask-2020/blob/626239b78621af96fbb324e678cca17b3dd4e470/my_cands_extract.py

[8] In other words, we set a threshold of $\exp\{-0.40\}$ on the likelihood ratio.

en → x		hu	ja	ko	pt	vi
MODEL1		44.8	11.8	4.0	32.6	27.2
fine-tune on:	JHU-TRAIN: 1-best	43.4	12.4	11.4	41.6	41.6
	JHU-TRAIN: all	52.1	23.1	23.1	49.3	52.0
	upweighted JHU-TRAIN: all + 1x 1-best	52.1	23.5	24.3	50.1	52.3
	upweighted JHU-TRAIN: all + 9x 1-best	**56.6**	**24.1**	**25.0**	**52.8**	**54.3**
	upweighted JHU-TRAIN: all + 99x 1-best	54.0	23.0	21.9	51.1	52.4

Table 3: The weighted macro F1 on JHU-TEST for MODEL1 and fine-tuned variants. Candidates are extracted from the n-best list using the proposed probability score thresholding (§5.3).

en → x		ja	ko
MODEL2		16.8	12.5
fine-tune on:	JHU-TRAIN: 1-best	18.4	18.7
	JHU-TRAIN: all	31.5	38.0
	upweighted JHU-TRAIN: all + 1x 1-best	30.3	38.0
	upweighted JHU-TRAIN: all + 9x 1-best	**32.1**	**38.8**
	upweighted JHU-TRAIN: all + 99x 1-best	31.0	33.4

Table 4: The weighted macro F1 on JHU-TEST for MODEL2 and fine-tuned variants for Japanese and Korean. Candidates are extracted from the n-best list using the proposed probability score thresholding (§5.3).

6 Results

We present results of our different methods on our internal development set in §6.1 and present our official evaluation performance in §6.2.

6.1 Internal evaluation

Table 3 shows the weighted macro F1 performance on JHU-TEST for MODEL1 and fine-tuned variants. Candidates are extracted from the n-best list using the proposed probability score thresholding (§5.3). Fine-tuning improves performance (except for fine-tuning on just the 1-best translation in Hungarian). For all language pairs, the best fine-tuning performance came from training on the up-weighted training data, where we trained on all possible translations with the 1-best up-weighted 10 times. For Japanese and Korean[9] MODEL2 (Table 4), all types of fine-tuning improve weighted F1, but for both language pairs, the best fine-tuning variant matches that of MODEL1.

Table 5 shows the weighted macro F1 on JHU-TEST for two methods of selecting candidates from the n-best list. The first line is the baseline top k hypothesis selection (§5.2), the second is our proposed probability score thresholding (§5.3). The best fine-tuned system is shown with each selection method for each language pair. The proposed probability score thresholding improves performance over the baseline top k candidate selection by 2–3.3 F1 points.

6.2 Official evaluation

In Table 6, we present the final results of our submission on the official test set (DUO-TEST). Our systems ranked first in all language pairs, with improvements of 0.1 to 9.2 over the next best teams. We denote in parenthesis the improvement over the next best team's system on DUO-TEST. We also report the score that our system achieved on our internal test set (JHU-TEST).

For Hungarian and Vietnamese, our winning submission was MODEL1 fine-tuned on the up-weighted Duolingo data (1-best repeated 10x) with a learning rate of 1×10^{-4}. For Japanese, our winning submission was MODEL2 fine-tuned on the up-weighted Duolingo data (1-best repeated 10x) with a learning rate of 5×10^{-4}. For Korean, our winning submission was MODEL2 fine-tuned on the up-weighted Duolingo data (1-best repeated 10x) with a learning rate of 1×10^{-4}, but without

[9]These were the two languages where MODEL2 improved fine-tuning performance compared to MODEL1.

en → x	hu	ja	ko	pt	vi
top k hypothesis selection (§5.2)	54.6	29.5	35.6	50.0	51.0
Probability score thresholding (§5.3)	**56.6**	**32.1**	**38.8**	**52.8**	**54.3**

Table 5: The weighted macro F1 on JHU-TEST for two methods of selecting candidates from the n-best list: baseline top k hypothesis selected (discussed in §5.2), and our proposed probability score thresholding (§5.3). The best fine-tuned system is shown with each selection method for each language pair.

en → x	DUO-TEST	JHU-TEST
hu	55.5 (+0.3)	56.6
ja	31.8 (+2.4)	32.1
ko	40.4 (+9.2)	38.9[10]
pt	55.2 (+0.1)	54.6
vi	55.8 (+1.9)	54.3

Table 6: The weighted macro F1 of our final submitted systems on the official shared task test set (DUO-TEST) on our internal test set (JHU-TEST). We denote in parenthesis the improvement over the next best team's system on DUO-TEST.

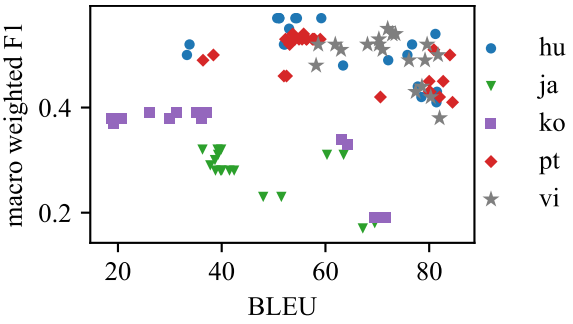

Figure 2: Macro Weighted F1 (JHU-TEST) vs. BLEU (JHU-DEV) for a variety of fine-tuned systems for each language pair. The two metrics are not well correlated within a language pair.

any internal development data withheld.[10]

For Portuguese, our winning submission was an ensemble of 3 systems. We began with MODEL1 fine-tuned on the up-weighted Duolingo data with a learning rate of 1×10^{-4}. We used fairseq's default ensembling to ensemble 3 systems trained on all the translations of each Duolingo prompt, with the 1-best data repeated a total of 2x, 10x, and 100x for each system.

While we submitted slightly different systems for each language pair, the following worked well overall: Fine-tuning on the Duolingo data was crucial. This is a domain adaptation task—the Duolingo data differs greatly from the standard MT bitext we pretrain on, such as Europarl proceedings, GlobalVoices news, Subtitles, or Wikipedia text.[11] Taking advantage of the relative weights of the training translations and up-weighting the best one was also helpful across the board. We suspect that using the weights in training directly (as opposed to our hack of upweighting the best translation) would likely improve performance further.[12]

7 Analysis

We perform qualitative and quantitative analyses of our output, which informed our own work and will motivate future work.

7.1 BLEU vs. Macro Weighted F1

In Figure 2, we plot macro weighted F1 on JHU-TEST against BLEU score[13] on JHU-DEV for fine-tuned systems for each language. It is clear that this BLEU score did not identify the best performing system according to the macro weighted F1 metric. For example, performance on beam search BLEU could be improved by further fine-tuning systems that had already been fine-tuned on all translations of each prompt on just the 1-best translation of each prompt, but that degraded the task performance. In fact, the systems that performed best on macro weighted F1 in Hungarian and Korean were over 20 BLEU behind the highest BLEU score for those languages (and the top BLEU scoring systems did poorly on the task metric).

[10] As described in §4, we first fine-tune a system and use our internal splits for model selection from checkpoints and threshold selection. Then we apply all the same parameters to fine-tune a system with no data withheld. This was better than with holding data only for en-ko (on DUO-DEV). Since this en-ko system was trained on JHU-TEST, Table 6 reports the JHU-TEST results on the corresponding system that withheld that data.

[11] In addition to style differences, the Duolingo sentences are much shorter on average.

[12] This feature does exist in Marian (Junczys-Dowmunt et al., 2018) but not in Fairseq.

[13] Computed against the 1-best translation of each prompt.

While this phenomenon may be an artifact of these particular metrics, we suspect this is indicative of an interesting topic for further research. MT models trained with NLL are trained to match a 1-hot prediction, which may make their output distributions poorly calibrated (Ott et al., 2018; Kumar and Sarawagi, 2019; Desai and Durrett, 2020). More research is needed for strong conclusions, but our initial analysis suggests that training on the more diverse data improves quality of a deep n-best list of translations at the expense of the top beam search output. This may be important in cases where an n-best list of translations is being generated for a downstream NLP task.

The data for this task was unique in that it provided diverse translations for a given prompt. In most cases where this type of data is not available, training towards a distribution (rather than a single target word), as is done in word-level knowledge distillation (Buciluundefined et al., 2006; Hinton et al., 2015; Kim and Rush, 2016) may prove useful to introduce the diversity needed for a strong n-best list of translations. This can be done either towards a distribution of the base model when fine-tuning (Dakwale and Monz, 2017; Khayrallah et al., 2018) or towards the distribution of an auxiliary model, such as a paraphraser (Khayrallah et al., 2020).

7.2 Qualitative error analysis

In each language, we performed a qualitative error analysis by manually inspecting the difference between the gold and system translations for prompts with lowest weighted recall on JHU-TEST.

Our systems were often incapable of expressing target language *nuance* absent from the source language. For example, for the prompt "we have asked many times.", a gold translation was '私たちは何度も尋ねてしまった' whereas our system output '私たちは何度も尋ねました'. The gold translations often included the てしまった verb ending, which conveys a nuance similar to perfect aspect. The prompt's scenario would lead many Japanese users to use this nuanced ending when translating, but our system produces valid but less natural translations that do not appear in the references.

Another issue is *vocabulary choice* on a more general level. Often there are several ways to translate certain words or phrases, but our systems prefer the less common version. For example, a common translation of 'please' in Portuguese is 'por favor', which appears in the high-weighted gold translations. Another possible translation, 'por obséquio', which our system seemed to prefer, appears in much lower-weighted translations. Another example is the translation of 'battery' in Korean. The high-weighted references include the common word for battery ('건 전 지') but only lower-weighted references include '배터리', which was preferred by our system.

Our system also struggled with *polysemous prompt words*. For example, for the prompt "cups are better than glasses.", our system output translations like '컵이 안경들보다 낫다' , using 안경 (eyeglasses), instead of translations like '컵이 유리잔보다 낫다' , using 유리잔 (drinking glasses). The systems seem to be incapable of considering the context, "cups" in this case, for the ambiguity resolution.

A final class of our system's errors is *grammatical errors*. For example, for the prompt "every night, the little sheep dreams about surfing.", the gold translations included sentences like 'toda noite a pequena ovelha sonha com surfe' whereas our system output sentences like 'toda noite as ovelhas pequenas sonham com surfe'. The error was that our output included 'ovelhas' (plural sheep), but the gold translations all used 'ovelha' (single sheep).

7.3 Missing paradigm slots in Duolingo data

We also find cases where our system produces valid translations but is penalized because these are not among the gold translations. We consider these cases as a result of an "incomplete" gold set with missing paradigms.[14]

For example, the Vietnamese pronouns for 'he' and 'she' can vary according to age (in relation to the speaker). From youngest to oldest, some pronouns for 'she' are 'chị ấy', 'cô ấy', and 'bà ấy'. For several of the prompts, the gold outputs only include some of these pronouns despite all being valid. In the prompt "she has bread", only the first two pronouns are present even though a translation representing the sentence as an older woman having bread should be equally valid. We also find this missing pronoun slot problem in Portuguese (references only using 'você' and not 'tu' for translations of 'you') and Japanese (only using 'あな

[14]The task website notes this phenomenon. It calls the set of targets 'comprehensive', though not 'exhaustive'.

た' and not '君' for translations of 'you').

We could not easily predict when slots would be missing. Because the data comes from Duolingo courses, we believe this may depend on the prompt's depth in the learning tree. As earlier lessons are studied by more users, we suspect they are also more likely to contain more complete gold translation sets due to more users submitting additional valid translations. This makes it difficult to assess the success of our models and distinguish "true errors" from valid hypotheses that are marked incorrect.

8 What Didn't Work

We explored additional methods both for selecting candidates from an n-best lists and for generating additional candidates based on an n-best list. While they did not improve performance and were not included in our final submission, we discuss the methods and the analyses learned from them.

8.1 Moore–Lewis filtering

Our error analysis revealed that our systems often output sentences that were not incorrect, but not optimized for the Duolingo task. For example, many of our top candidates for translations of "please" in Portuguese used *por obséquio*, which is a very formal version, instead of the more common *por favor*. While both versions were valid for the prompts, the gold translations with *por favor* were weighted higher, so we would desire models to prefer this translation. We interpret this as domain mismatch between the STAPLE data and our MT training data.

To filter out such bad candidates, we experimented with cross-entropy language model filtering (Moore and Lewis, 2010). This takes two language models: a (generally large) out-of-domain language model (OD), and a (typically small) in-domain language model (ID), and uses the difference in normalized cross-entropy from these two models to score sentences. Sentences with good OD scores and poor ID scores are likely out-of-domain and can be discarded based on a score threshold.

Experimenting on Portuguese, we used KenLM (Heafield, 2011) to train a Kneser–Ney-smoothed 5-gram model on the Portuguese side of the MT training data (Table 2) as the OD model and a 3-gram model on the Duolingo Portuguese data (ID). These were used to score all candidates t as

	JHU-TEST	DUO-TEST
Baseline	53.30	**55.16**
Baseline + Moore-Lewis	**53.70**	53.83

Table 7: Moore–Lewis filtering for Pt (macro F1).

$\text{score}(t) = p_{\text{ID}}(t) - p_{\text{OD}}(t)$. We swept thresholds and minimum prompt lengths on our JHU-TEST data, and found with a threshold of -1.50 on 7-word prompts and longer performed the best.

Moore–Lewis filtering was originally designed for more coarse-grained selection of training data. We suspect (but did not have time to test) that a better idea is therefore to apply this upstream, using it to help select data used to train the general-domain MT system (Axelrod et al., 2011).

8.2 Dual conditional thresholding

Extending the probability score thresholding (§5.3), we consider incorporating a score from a reverse model that represents the probability that the original prompt was generated by the candidate. The reverse model score is also used in Dual Conditional Cross-Entropy Filtering when selecting clean data from noisy corpora (Junczys-Dowmunt, 2018), and for re-scoring n-best lists in MMI decoding (Li et al., 2016)

We train base and fine-tuned reverse systems for the five language pairs and use them to score the output translations. We compute the combined score of a hypothesis given a prompt as the arithmetic mean of the forward and backward log probability scores and use them in the probability score thresholding algorithm from §5.3. We find that after sweeping across threshold values, incorporating the reverse score performs slightly worse overall than the standard thresholding method for every language.

8.3 N-gram filtering

The Duolingo data generally consists of simple language, which means we did not expect to see novel phrases in the references that were not in our training corpora. We used this idea to filter hypotheses that had any n-grams that didn't appear in our training data. Our hope was that this would catch rare formulations or ungrammatical sentences, e.g. *cachorro preta*, which has the wrong gender on the adjective. However, even using bigrams caused this method to filter out too many hypotheses and hurt F1 performance.

None	elas	têm	cinco	meninas	?
Open	elas	V;3;PL	NUM	N;PL;FEM	?
Morph	PRO;3;PL;FEM	V;3;PL	NUM	N;PL;FEM	PUNCT
POS	PRO	V	NUM	N	PUNCT

Table 8: Preprocessing operations for filtering on one Portuguese gold output for the prompt *do they have five girls?*, organized from most specific to most general.

Part-of-speech filtering Although the language used in Duolingo is relatively simple, the number of unique types turned out to be quite large. However the number part-of-speech (POS) tags is small. Instead of filtering based on words, we count n-grams of POS tags, hoping to remove ungrammatical sentences with tags such as `DET DET`. In our experiments, this did not actually exclude any hypotheses.

Open class words and morphology In between the extremes of large number of types using raw lexical forms and few types using POS tags is to leverage open class words or additional morphological information. We morphologically tag the dataset with the Stanford NLP toolkit (Qi et al., 2018), then represent each sentence either by its words, its POS tags, its morphological tags, or words for closed-class items and tags for open-class items, as shown in Table 8. This too resulted in few hypotheses being filtered and did not impact F1 performance.

Filtering by difficulty level As the Duolingo data was generated by language learners, we also considered filtering sentences by the difficulty of the words within. Experimenting with Japanese, we examined the grade level of kanji[15] in each sentence. Ignoring non-kanji characters, the average grade level per sentence on the STAPLE training data was 3.77, indicating a 3rd–4th grade level. Future work could consider filtering by other measures such as the coreness of a word (Wu et al., 2020).

8.4 Generation via post-editing

Inspired by query expansion in information retrieval, we post-edit either by consider morphological variants in situations of underspecification, substituting forms in different scripts (for Japanese), or replacing long-form number names with numerals. We found these ineffective because several acceptable translations were not present in the ground truth dataset (see §7.3).

[15] Specified by the Japanese Ministry of Education and annotated in `edrdg.org/wiki/index.php/KANJIDIC_Project`

Strategy	P	RW	WF1macro
Baseline	26.91	69.70	34.49
Add 1;PL	26.62	69.84	34.27
Add 3;SG;MASC	23.97	70.19	33.42
Add 3;SG;FEM	24.69	70.49	33.51
Add 3;PL	22.28	69.75	31.89
Add most frequent 'she'	26.77	69.84	34.38
Swap most common 'he's	26.71	69.82	34.37
Swap 2nd most common 'he's	26.90	69.71	34.47
Swap 3rd most common 'he's	26.88	69.71	34.45

Table 9: Effect of pronoun-based augmentation on metrics in Vietnamese, computed on JHU-TEST. All strategies improve recall and weighted recall, but they cause precision and F1 to decrease.

Morphological expansions English is morphologically poorer than 4 target languages. As an example, the English word 'you' may be translated into Portuguese as 'tu', 'você', 'vocês', or 'vós', to consider only nominative forms. We can thus generate three additional candidates by altering the morphosyntax (and maintaining grammatical concord) while keeping the meaning intact.

Evaluating in Portuguese and Vietnamese, we find that this is ineffective (see §7.3). Consider Vietnamese. It is a morphologically isolating and zero-marking language, so concord between constituents is not overtly marked. This leaves us fairly free to swap out morphological variants of pronouns: there may be difference in age, connotation, or register, but the overt semantics of the English prompt are preserved. All swapping transformations in Table 9 give poorer performance.

Hiragana replacement Japanese has three different writing systems—hiragana, katakana, and kanji—and sometimes a word written in kanji is considered an acceptable translation when written in hiragana. For example, the Japanese word for "child" is 子供 when written with kanji, but an acceptable alternative is the hiragana こども. We experiment with expanding translation candidates by replacing Japanese kanji with pronunciations from a furigana (hiragana pronunciation) dictionary but this method did not improve performance.

Numeral replacement For sentences containing numbers, the list of accepted translations often contains Arabic numbers, in addition to numbers in the native language. For example, 'o senhor

smith virá no dia dez de julho' and 'o senhor smith virá no dia 10 de julho.' are both gold translations of "mr. smith will come on july tenth." We experiment with replacing native numbers with Arabic numerals in Japanese, Portuguese, and Vietnamese. This did not improve weighted F1.

9 Conclusion

Our approach was general, borrowing from best practices in machine translation. We built large, general-domain MT systems that were then fine-tuned on in-domain data. We then followed an "overgenerate and filter" approach that made effective use of the scores from the systems to find a per-prompt truncation of large n-best lists produced from these systems. These techniques performed very well, ranking first in all five language pairs. We expect that further refinement and exploration of standard MT techniques—as well as techniques that we were unsuccessful with (§8)—would bring further improvements that would accrue generally across languages.

At the same time, the Duolingo shared task is distinct from machine translation in subtle but important ways: presenting simpler, shorter sentences and a 0-1 objective. While we were not able to get additional gains from linguistic insights, we don't see these failures as conclusive indictments of those techniques, but instead as invitations to look deeper.

Acknowledgments

We thank Najoung Kim for remarks on Korean, An Nguyen for remarks on Vietnamese and Vinicius C. Costa for remarks on Portuguese, and Doug Oard for general advice.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The AMARA corpus: Building parallel language resources for the educational domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1856–1862, Reykjavik, Iceland. European Language Resources Association (ELRA).

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535–541, New York, NY, USA. Association for Computing Machinery.

Praveen Dakwale and Christof Monz. 2017. Fine-tuning for neural machine translation with limited degradation across in- and out-of-domain data. In *Proceedings of the 16th Machine Translation Summit (MT-Summit 2017)*, pages 156–169.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers.

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.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 36–44, Melbourne, Australia. Association for Computational Linguistics.

Huda Khayrallah, Brian Thompson, Matt Post, and Philipp Koehn. 2020. Simulated multiple reference training improves low-resource machine translation. arXiv preprint arXiv:2004.14524.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. 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. *CoRR*, abs/1903.00802.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

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)*. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3956–3965, Stockholmsmässan, Stockholm Sweden. PMLR.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

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.

Winston Wu, Garrett Nicolai, and David Yarowsky. 2020. Multilingual dictionary based construction of core vocabulary. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4204–4210, Marseille, France. European Language Resources Association.