Simultaneous Translation and Paraphrase for Language Education

Stephen Mayhew, Klinton Bicknell, Chris Brust,
Bill McDowell, **Will Monroe**, and **Burr Settles**
Duolingo
Pittsburgh, PA, USA
{stephen, klinton, chrisb, mcdowell, monroe, burr}@duolingo.com

Abstract

We present the task of *Simultaneous Translation and Paraphrasing for Language Education (STAPLE)*. Given a prompt in one language, the goal is to generate a diverse set of correct translations that language learners are likely to produce. This is motivated by the need to create and maintain large, high-quality sets of acceptable translations for exercises in a language-learning application, and synthesizes work spanning machine translation, MT evaluation, automatic paraphrasing, and language education technology.

We developed a novel corpus with unique properties for five languages (Hungarian, Japanese, Korean, Portuguese, and Vietnamese), and report on the results of a shared task challenge which attracted 20 teams to solve the task. In our meta-analysis, we focus on three aspects of the resulting systems: external training corpus selection, model architecture and training decisions, and decoding and filtering strategies. We find that strong systems start with a large amount of generic training data, and then fine-tune with in-domain data, sampled according to our provided learner response frequencies.

1 Introduction

Machine translation systems are typically trained to produce a single output, but in certain cases, it is desirable to have many possible translations of a given input text. For example, Duolingo—the world's largest language-learning platform—uses translation-based exercises for some of its lessons. For any given translation prompt there may be hundreds or thousands of valid responses, so we use a set of human-curated translations in order to grade learner responses. The manual process of maintaining these sets is laborious, and we believe it can be improved with the aid of rich multi-output translation and paraphrase systems.

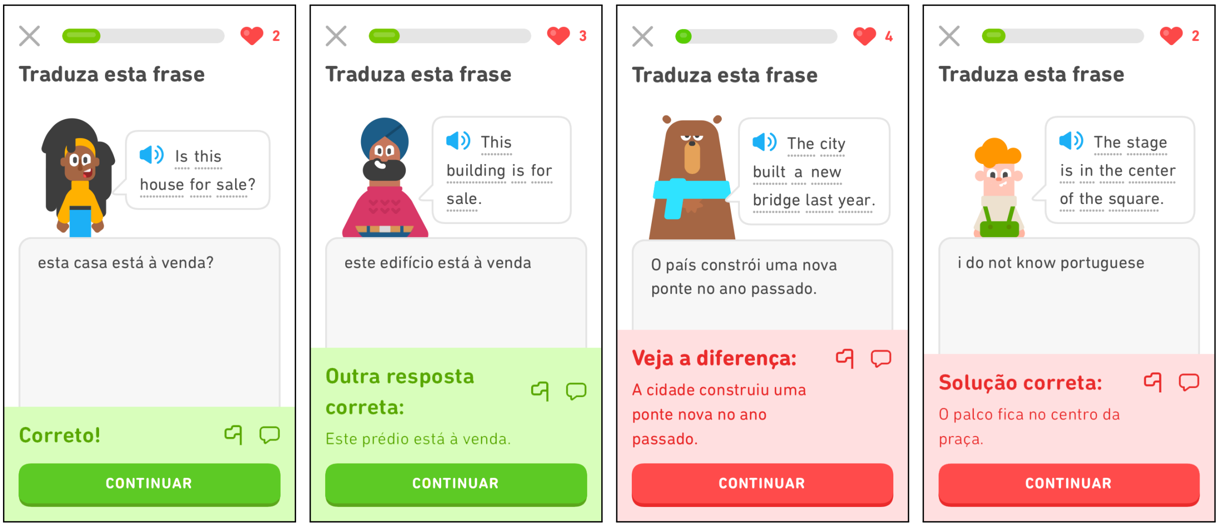
Prompt	
is my explanation clear?	
Reference Translation	
a minha explicação está clara?	
Accepted Translations	**Weight**
minha explicação está clara?	.267
minha explicação é clara?	.162
a minha explicação está clara?	.111
a minha explicação é clara?	.088
minha explanação está clara?	.057
está clara minha explicação?	.044
minha explanação é clara?	.039
a minha explanação está clara?	.036
...	...

Table 1: An example from the Portuguese dataset. In this task, teams are given an English prompt and a reference translation, and are required to produce as many variants in the accepted translations as possible. The evaluation favors translations with higher weight, which is a measure of learner response frequency.

To this end, we introduce a new task called *Simultaneous Translation and Paraphrasing for Language Education (STAPLE)*. From the perspective of the research community, we believe this poses an interesting exercise that is similar to machine translation (MT), but also provides data with new and unique properties that we expect to be of interest to researchers in MT evaluation, multilingual paraphrasing, and even language education technology. It is our hope that this new task can help synthesize efforts from these various subfields to further the state of the art, and broaden their applications.

2 Shared Task Description

For the STAPLE task, participants begin with English *prompts* and generate high-coverage sets of plausible translations in five different languages. For training and evaluation, each prompt is paired with a relatively comprehensive set of handcrafted,



Traduza esta frase

Is this house for sale?

esta casa está à venda?

Correto!

Traduza esta frase

This building is for sale.

este edifício está à venda

Outra resposta correta:

Este prédio está à venda.

Traduza esta frase

The city built a new bridge last year.

O país constrói uma nova ponte no ano passado.

Veja a diferença:

A cidade construiu uma ponte nova no ano passado.

Traduza esta frase

The stage is in the center of the square.

i do not know portuguese

Solução correta:

O palco fica no centro da praça.

Figure 1: Screenshots from the Duolingo app (iOS, circa 2020), showing translation exercises for English prompts into Portuguese. The first two examples show correct student translations, with Duolingo suggesting an alternate, preferred translation in the second case. The third and fourth responses show incorrect translations.

field-tested *accepted translations*, each weighted and ranked according to their empirical frequency among Duolingo learners. We also provide a high-quality automatic *reference translation* of each prompt that may (optionally) be used as a reference or anchor point, in the event that researchers want to explore paraphrase-only approaches (this also serves as a strong baseline). See Table 1 for an example from the Portuguese dataset.

2.1 Corpus Collection

Data for the task are derived from Duolingo, a free, award-winning, online language-learning platform. Since launching in 2012, hundreds of millions of learners worldwide have enrolled in Duolingo's game-like courses via the website[1] or mobile apps. Learning happens through a variety of interactive exercise types, combining reading, writing, listening, and speaking activities.

One such format is a translation exercise—shown in Figure 1—in which the learner is shown a prompt in one language, and asked to translate it into the other. Since English is by far the most popular language to learn on Duolingo, we created a task corpus by sampling prompts from English courses, in which users are shown an English sentence, and then asked to translate it into a language they already know. For instance, the examples in Figure 1 come from the course for Portuguese speakers learning English.

Naturally, some prompts have more accepted translations (valid learner responses) than others, depending on such factors as polysemy, synonymy, or prompt length. We filtered out prompts for which the number of accepted translations was in the top or bottom deciles of a course, to avoid outliers. Although each accepted translation is technically correct, usually a small number of them are considered most fluent or idiomatic. To estimate this distribution empirically, we gathered learner response data from October–November 2019. For each translation, we counted the number of times that learners produced that translation (with some allowances for punctuation and capitalization).

This provided a count c_t for each translation t in the set of accepted translations A. Since many translations were never attested in learner data, we then smoothed and normalized these counts to produce a learner response frequency (LRF) weight w_t for each translation, such that they sum to 1 for each prompt:

$$w_t = \frac{\sqrt{c_t + 1}}{\sum_{t' \in A} \sqrt{c_{t'} + 1}}$$

These weights are a unique feature of the STAPLE corpus, and found in almost no other datasets.

Having gathered prompts from each course, we shuffled the prompt set and selected 500 prompts for development and 500 for test. Of the remaining prompts for each course, we created a training set by sampling according to course size, so smaller courses (e.g., Vietnamese) have fewer prompts. Statistics on the datasets can be found in Table 2.

[1] https://www.duolingo.com

	Train			Dev			Test		
Language	prompts	trans.	ratio	prompts	trans.	ratio	prompts	trans.	ratio
Hungarian	4,000	251,442	62.9	500	27,647	55.3	500	33,578	67.2
Japanese	2,500	855,941	342.4	500	172,817	345.6	500	165,095	330.2
Korean	2,500	700,410	280.2	500	140,353	280.7	500	150,477	301.0
Portuguese	4,000	526,466	131.6	500	60,294	120.6	500	67,865	135.7
Vietnamese	3,500	194,720	55.6	500	29,637	59.3	500	28,242	56.5

Table 2: Dataset sizes by number of prompt sentences, and total number of accepted translations.

2.2 Five Language Tracks

We provide data for translating English prompts into five languages: Hungarian, Japanese, Korean, Portuguese (Brazilian), and Vietnamese. These span five different language families, three different writing systems, and represent a wide variety of popular Duolingo courses. For example, as of this writing, English from Portuguese is the fourth-largest Duolingo course overall, whereas English from Korean is median-sized, with the others falling in between. As such, much effort has gone into developing their accepted translation sets, but there is probably still room for improvement. These five languages also vary widely in their status as high-to-low-resource languages in NLP research.

For the shared task, participants were allowed to submit results to any or all of these language tracks. Furthermore, there were no restrictions on the use of external data; teams were encouraged to use any available monolingual or parallel corpora.

2.3 Evaluation

The main scoring metric is (macro) weighted F1 with respect to the accepted translations. In short, systems are scored based on how well they can return all human-curated accepted translations, but with lower penalties on recall for failing to produce translations that learners rarely submit anyway.

For each prompt sentence s with accepted translation set $\mathcal{A}_s$ in the corpus, we evaluate the weighted recall of a system's predicted translation set $\mathcal{P}_s$ as follows:

$$\text{Weighted Recall}(\mathcal{P}_s) = \sum_{t \in |\mathcal{P}_s \cap \mathcal{A}_s|} w_t \Bigg/ \sum_{t \in |\mathcal{A}_s|} w_t$$

Precision is calculated in an unweighted fashion (as there is no weight for false positives), and weighted F1 for each $\mathcal{P}_s$ is simply the usual harmonic mean of precision and weighted recall. These weighted F1s for each prompt are then averaged over the entire evaluation dataset $\mathcal{D}$:

$$\text{(Macro) Weighted F1} = \sum_{s \in \mathcal{D}} \frac{\text{Weighted F1}(\mathcal{P}_s)}{|\mathcal{D}|}$$

Since evaluation is done by matching predictions with accepted translations, we ignore any differences due to punctuation, capitalization, or multiple whitespaces.

2.4 Challenge Timeline

We announced the shared task on December 20, 2019, with information about the task timeline, data, etc., published on a regular basis to a dedicated website[2]. We released the training data on January 15, blind dev data on March 2, and blind test data on March 30, 2020.

During the blind dev phase, participants were able to submit up to five submissions per day to an online evaluation leaderboard. Originally, we had planned on closing the dev phase at the start of the test phase, but upon request, we kept it open so that teams could continue to experiment and submit to the dev leaderboard even after the test phase opened, without counting against their final submission(s). We allowed up to three submissions in total to the test leaderboard (to account for technical problems, etc.).

3 Results

A total of 20 teams participated during the dev phase, 13 teams during the test phase, and 11 teams submitted system description papers. Of the teams with system descriptions, three of them (**jbrem**, **sweagraw**, **jindra.helcl**) participated in all five language tracks. One team (**rakchada**) submitted to two tracks, and the remaining teams only submitted to a single track, with Japanese and Portuguese being the most popular.

[2] https://sharedtask.duolingo.com

	Hungarian		Japanese		Korean		Portuguese		Vietnamese	
Team	Rank	F1	Rank	F1	Rank	F1	Rank	F1	Rank	F1
jbrem	1	.555	1	.318	1	.404	1	.552	1	.558
nickeilf		–		–		–	1	.551		–
rakchada	1	.552		–		–	1	.544		–
jspak3		–		–	2	.312		–		–
sweagraw	2	.469	2	.294	3	.255	2	.525	2	.539
masahiro		–	2	.283		–		–		–
mzy		–	3	.260		–		–		–
dcu		–		–		–	3	.460		–
jindra.helcl	3	.435	4	.213	4	.206	4	.412	3	.377
darkside		–	5	.194		–		–		–
nagoudi		–		–		–	5	.376		–
baseline_aws	4	.281	6	.043	5	.041	6	.213	4	.198
baseline_fairseq	5	.124	7	.033	5	.049	7	.136	5	.254

Table 3: F1 results for all systems, on all languages. Rank is assigned according to statistical significance (§3).

Official weighted F1 results are shown in Table 3. Ranks are determined using an approximate permutation test with 100,000 samples (Padó, 2006), and adjacent-scoring systems are considered significantly different at $p < .05$. Figure 3 provides additional detail on precision and weighted recall. Overall, teams outperformed our provided baselines by a wide margin, and submissions tended to score higher on precision than weighted recall.

3.1 Baselines

We prepared two very different baselines. For **baseline_aws**, we used Amazon Translate[3] to generate a single "best" machine translation from English into the target language. These were also provided as reference translations at each phase.

For **baseline_fairseq**, we used the fairseq framework (Ott et al., 2019) trained solely on the STAPLE task data. We created bitexts by pairing English prompts with each of their target language translations (making no use of the weights). The baseline employs a convolutional neural network (CNN) using byte-pair encoding (BPE) with a vocabulary size of 20,000, and simply outputs default n-best lists of size 10. While we ensured that the output BLEU scores of this model were sensible, we did not tune any parameters, instead treating this as a baseline that should be attainable by any team with minimal effort. Our baseline code was provided as a starting point for participants, and many chose to derive their systems from it.

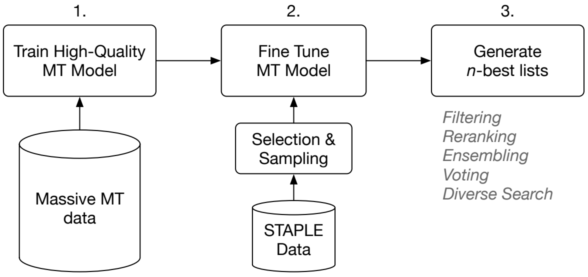


Figure 2: Generalized pipeline used by most systems.

3.2 Submitted Systems

With few exceptions, participating teams followed the generalized pipeline illustrated in Figure 2. This consists of (1) training a high-quality machine translation model using massive but mostly out-of-domain corpora, (2) fine-tuning the model using STAPLE task corpora (and sometimes others), and then (3) employing various tricks for diverse output generation and filtering.

jbrem (Khayrallah et al., 2020) took an approach involving score-based filtering of n-best lists, from a Transformer model pre-trained on large external corpora and then fine-tuned on the STAPLE data. The authors describe benefits from using various pre-training datasets, two different filtering methods, and various ways of upweighting of translations of high frequency (weight). The resulting system was among the strongest in the competition, ranking first in all five tracks.

nickeilf (Li et al., 2020) explored a family of diversification approaches including beam expansion, Monte Carlo random dropout, lexical substi-

[3] https://aws.amazon.com/translate/

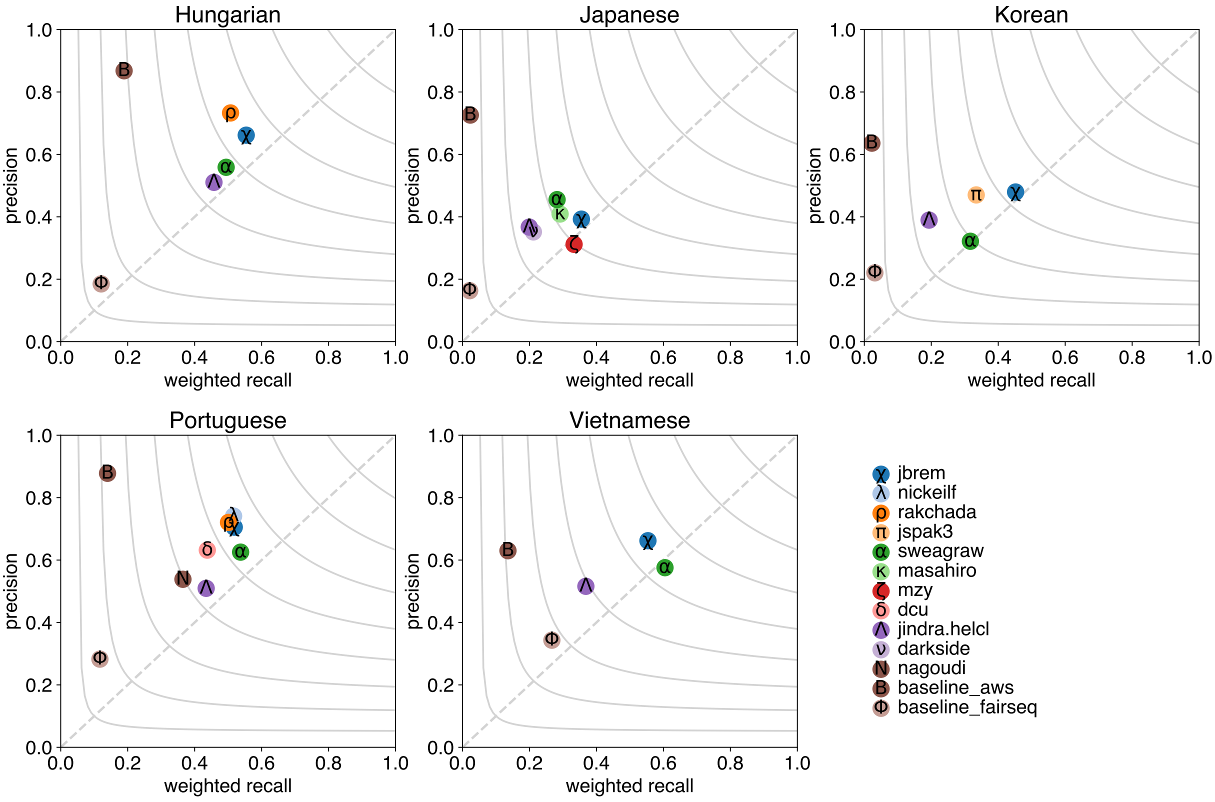


Figure 3: Precision and weighted recall for each system and language. The dashed line represents equal precision and weighted recall. Curved lines represent weighted F1 in increments of 0.1.

tution, and mixture of experts models, combined through ensemble-based consensus voting to generate a high quality set of translation suggestions. This tied for first place in the Portuguese track.

rakchada (Chada, 2020) used pre-trained Transformer models fine-tuned on the STAPLE data with an oversampling trick that afforded more weight to translations with higher frequency. They then used a classifier to filter the n-best lists based on predicted learner frequency. This tied for first place in the Hungarian and Portuguese tracks.

jspak3 (Park et al., 2020) took a similar approach to the original BART setup (Lewis et al., 2019), except they fine-tuned the model not only on larger parallel corpora, but also on the STAPLE data. This ranked second in the Korean track.

sweagraw (Agrawal and Carpuat, 2020) used a Transformer model pre-trained on the OpenSubtitles corpus, then fine-tuned on Tatoeba and the STAPLE data (§4.1), with the STAPLE translations oversampled to capture frequency. Resulting n-best lists were filtered with a two-layer neural classifier optimized for a soft-F1 objective. This ranked second or third in all five language tracks.

masahiro (Kaneko et al., 2020) took a simple ensemble approach that requires no modification to an off-the-shelf NMT system (fairseq). The authors train multiple forward (L2R) and backward (R2L) models using different initial seeds, first by pre-training on general corpora and then fine-tuning on STAPLE data. Their experiments show that combining ensembling forward-backward models yields more diversity and higher F1 than simply using different seeds alone. This tied for second place in the Japanese track.

mzy (Yang et al., 2020) explored three particular strategies: pre-training on larger corpora before fine-tuning on in-domain corpora, using diverse beam search, and finally reranking candidate translations. The authors found that first fine-tuning on a similar intermediate corpus was better than fine-tuning on the STAPLE data alone. Diverse beam search provided modest further gains, although they report no improvement from beam re-ranking. This ranked third in the Japanese track.

dcu (Haque et al., 2020) compared both phrase-based and neural models by extending the STAPLE data with additional corpora (selected for similarity to the task data under a language model), with the neural model performing better. They generate sets of high-scoring predictions according to beam searches, majority voting, and other techniques, and also run these initial translations through an additional paraphrasing model, placing third in the Portuguese track.

jindra.helcl (Libovický et al., 2020) trained a Transformer model by combining STAPLE data with additional parallel corpora and back-translated monolingual corpora. They also employed a filtering classifier that predicts whether their models' beam search outputs within accepted translations. This ranked third or fourth in all five tracks.

darkside (Nomoto, 2020) took a very different approach, treating the task as a paraphrase generation problem and using no data beyond what was provided for the shared task. They took two approaches, both based on autoencoders. The first is a sequence-to-sequence model with Gaussian noise added to the context vector, and the second is based on a conditional Variational Autoencoder, which has seen success in generating variations of input content in the literature (Bowman et al., 2015). This ranked fifth in the Japanese track.

nagoudi (Nagoudi et al., 2020) used a combination of data augmentation and ensembles. They combined STAPLE data with additional parallel corpora to train their models, finding (curiously) that this outperformed the fine-tuning approach employed by many others. They generated multiple translations by passing the source sentence through an ensemble of model training checkpoints, taking the n-best outputs from each and de-duplicating. This ranked fifth in the Portuguese track.

4 Meta-Analyses

In this section, we analyze different facets of the various approaches taken, in an effort to understand which design choices were most impactful on final results. We identified three major areas of variance: use of external training corpora (§4.1), model architecture and training procedures (§4.2), and decoding and filtering strategies (§4.3).

4.1 External Training Corpora

The STAPLE dataset is relatively small compared to many modern machine translation efforts. This is by design: it is challenging to develop a parallel corpus that is complete with many acceptable translations. One of our goals in organizing this task was to see how teams could effectively leverage existing corpora, with a modest amount of in-domain data, to bootstrap high-quality models for the task.

Corpus effects	**Precision**		**W. Recall**		**W. F1**	
(Intercept)	.418	***	.293	**	.283	**
Tatoeba	+.190		+.223	.	+.214	.
ParaCrawl	+.018		+.103		+.071	
Europarl	+.061		+.057		+.063	
QED	+.011		−.004		+.004	
OpenSubtitles	−.098		−.083		−.087	
Wikipedia	−.034		−.213		−.153	
Random effects	**St.Dev.**		**St.Dev.**		**St.Dev.**	
Prompt ID	±.183		±.210		±.173	
Track ID	±.085		±.106		±.103	
Team ID	±.082		±.080		±.075	

Table 4: Mixed-effects analysis of the most commonly-cited external corpora used for training.

Most teams began with a generic MT system pre-trained on massive but out-of-domain parallel corpora, either before or in tandem with the STAPLE task data. These were largely drawn from the Open Parallel Corpus (OPUS) project (Tiedemann, 2012). One natural question is whether the choice to train on a particular dataset from this collection had any meaningful impact on final results.

To answer this question, we coded each team with features variables indicating each corpus they reported using for their final submission, and used a regression analysis to see if these data choices significantly impacted precision, weighted recall, and weighted F1 scores for each prompt in the test set[4]. To analyze this properly, however, we need to distinguish between effects among data choices are actually meaningful versus those that can be explained by sampling error due to random variations among prompts, tracks, or teams. To do this, we use a *linear mixed-effects model* (cf., Baayen, 2008, Ch. 7). In addition to modeling the *fixed* effects of the various corpora, we can also model the *random* effects represented by the prompt ID (some sentences may be longer or harder), the track ID (the languages inherently vary), and the team ID (teams will differ in other aspects not captured by these corpus variables).

Table 4 presents a mixed-effects analysis for the most-cited corpora among participating teams, each used by at least four different systems. The intercepts can be interpreted as "average" metrics, which then go up or down according to fixed and random effects. Only the Tatoeba corpus appears to have a significant positive impact on metrics. In other words, we might expect that pre-training

[4] Thus, a team participating in all five tracks would yield 5 × 500 = 2,500 data points for this regression analysis.

	Feature \ Team	jbrem	nickeilf	rakchada	jspak3	sweagraw	masahiro	mzy	dcu	jindra.helcl	darkside	nagoudi
Model Architecture & Training	Transformer	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	CNN										✓	✓
	LRF Weights	✓	✓	✓		✓					✓	
	Pre-train→Fine-tune	✓	✓	✓	✓	✓	✓	✓	✓			
	Train Combined									✓		✓
Decoding & Filtering	Diverse Beam Search		✓		✓			✓				
	Beam Reranking				✓	✓		✓				
	Beam Filtering	✓		✓		✓				✓	✓	
	Paraphrasing		✓						✓	✓		
	Ensembling	✓	✓				✓		✓			✓
	Backtranslation			✓						✓		✓

Table 5: Table of features used by team. Descriptions of features can be found in §4.2 and §4.3.

with Tatoeba would add +.214 to prompt-specific F1 scores ($p = .088$), all else being equal. Since Tatoeba is a collaborative online database[5] of sentences geared towards foreign language learners (some of which even have multiple translations, although no weights), it is extremely similar to the STAPLE task domain. Thus it makes sense that this corpus would be helpful; in fact, **sweagraw** and **jindra.helcl** included it alongside the STAPLE data in fine-tuning their models.

Other effects are smaller and statistically insignificant, suggesting that the particular choice of supplementary out-of-domain data may not matter as much as simply using a large amount. One notable exception is the parallel Wikipedia corpus (Wołk and Marasek, 2014), which exhibits a large negative trend on recall and F1, possibly due to its noisy, automatically-aligned provenance.

The volume of parallel training data may also impact performance. For example, for the Korean track **jbrem** report internal results using similar datasets to **sweagraw**, and achieving the same score. But further experiments extending the training set yielded improvements of about +.1 F1. However, simply using larger corpora in pre-training does not guarantee higher scores: **nagoudi** apparently trained on all of OPUS, yet had the lowest Portuguese scores among participants.

4.2 Model Architecture & Training

Decisions made on model architecture and training procedures seemed to have more impact on final system performance. We mapped many of these design decisions into high-level system features, summarized at the top of Table 5.

***Transformer* vs. *CNN*.** The **baseline_fairseq** we provided is based on a convolutional neural network (CNN) architecture, and a few teams also went this route. However, top-ranking teams largely opted for a Transformer-based architecture (Vaswani et al., 2017) instead. **jspak3** notably used the BART architecture (Lewis et al., 2019) to pre-train a decoder in particular, and **dcu** also compared a phrase-based statistical MT approach (Koehn et al., 2007) to a Transformer-based neural MT system, with the latter performing better.

***LRF Weights*.** When training on STAPLE task data, teams had to decide how to convert the one-to-many relationship of prompts and accepted translations into standard bitext for more conventional MT training. Some teams simply repeated the English prompt for each target translation (as we did for **baseline_fairseq**), while others used only the highest-weighted translation. Some of the more successful teams took advantage of the weights associated with each accepted translation. In particular, **jbrem** included multiple copies of the highest-weighted translation, **nickeilf** used only the top k, and **sweagraw** and **rakchada** both sampled each translation in proportion to its weight.

***Pre-train→Fine-tune* vs. *Train Combined*.** Top-performing teams also tended to pre-train a generic MT model (e.g., trained on corpora from §4.1) and fine-tune it using STAPLE task data. This is opposed to pooling all data together for joint training. The latter approach certainly outperformed STAPLE-only baselines, but lagged behind fine-tuned pipeline approaches in most cases.

To measure the impact of these choices, we conducted a second mixed-effects regression analysis, coding each team with the model architecture and

[5] https://tatoeba.org

Model effects	**Precision**	**W. Recall**	**W. F1**
(Intercept)	.351 ***	.232 **	.221 **
Transformer	+.107	+.098 .	+.107 .
LRF Weights	+.097 *	+.060 *	+.075 *
Pre-train→Fine-tune	+.050	+.080	+.065
Random effects	**St.Dev.**	**St.Dev.**	**St.Dev.**
Prompt ID	±.183	±.210	±.173
Track ID	±.085	±.105	±.102
Team ID	±.070	±.049	±.044

Table 6: Mixed-effects analysis of various model architecture and training procedure choices.

training decisions that describe their final submissions. Results are presented in Table 6. Here we see empirical confirmation that Transformer-based systems tended to perform $+.1$ points better for all three metrics, although only marginally statistically significant (perhaps because it was also the most common choice).

Incorporating LRF weights in the fine-tuning strategy also appears to have a robust positive effect ($p < .05$ across all metrics). The importance of the weighting strategy can be further illustrated by comparing **jbrem** with **jindra.helcl**. Both systems submitted to all five tracks, and otherwise used similar approaches. However, **jbrem** reports on an ablation experiment using only the top-weighted translation, the results of which are similar to those of **jindra.helcl**, who used this very strategy.

Finally, there is also a positive trend favoring pre-training on external corpora before fine-tuning, as opposed to training on all data combined.

4.3 Decoding & Filtering

Since the STAPLE task requires multiple translations for each input prompt, all teams generated n-best lists, and employed various strategies for pruning them to contain only desirable translations. The feature group at the bottom of Table 5 represent these decoding and filtering steps.

Diverse Beam Search. Multiple teams attempted to use diverse beam search (Vijayakumar et al., 2016) to generate a more varied set of tranlation candidates. However, it proved either to be only marginally helpful (**nickeilf**, **mzy**) or unhelpful (**jspak3**) in various ablation experiments.

Beam Reranking. Two teams tried training an auxiliary model to rank output candidates by predicted learner response frequencies. In both cases, this approach performed poorly.

Beam Filtering. Several teams attempted to filter candidate translations, which were applied to candidate translations to decide if they should be removed from final predictions. Approaches to this varied significantly, from language-model probabilities (**jbrem**) to binary classifiers including gradient-boosted decision trees (**rakchada**), feed-forward neural networks (**sweagraw**), and multilingual transformers (**jindra.helcl**). **nickeilf** showed improvements using consensus voting among an ensemble of MT models, in which only sentences attested by multiple subsystems are retained. Most of these teams reported significant gains from filtering in ablation studies.

Paraphrasing. Three teams implemented monolingual paraphrasing models to increase the size of their n-best list of candidates. **jindra.helcl** reported experiments with a Levenshtein Transformer (Gu et al., 2019), a model that learns to create new paraphrases by editing candidate sentences. However, this produced output too noisy to be useful, and was omitted from their final submission.

Ensembling. A number of teams employed an ensemble of MT models, by combining either different training checkpoints, random initialization seeds, or other training regimes (such as training on reversed sequences, which was the main strategy used by **masahiro**, who tied for second in the Japanese track). Three teams also tried ***Backtranslation*** (Sennrich et al., 2016), with mixed results.

We conducted a mixed-effects analysis of decoding and filtering techniques, however, the effect sizes and p-values were much less significant than those from §4.1 and §4.2. These inconclusive results suggest that decoding and filtering play a smaller role in overall system performance than pre-training and model architecture decisions.

4.4 Scoring the Top-k Test Translations

The learner response frequency weights tend to have a tall head: a few common responses carry most of the weight, and many more responses carry much less weight (e.g., many human-curated accepted translations were not ever attested by learners during our data collection window). Since this distribution determines weighted recall, and therefore our overall evaluation metric, it is instructive to compare against a benchmark "oracle" that is able to return the top-k gold translations. Table 7 shows results of such an oracle for several values of k evaluated over the test set.

	Hun.	**Jap.**	**Kor.**	**Por.**	**Vie.**
$k = *$	1.0	$\underline{1.0}$	$\underline{1.0}$	1.0	1.0
$k = 10$	.735	.302	.350	.655	.789
$k = 5$	$\underline{.643}$	.231	.266	$\underline{.578}$	$\underline{.692}$
$k = 1$	.372	.090	.101	.340	.387
jbrem	.555	.318	.404	.552	.558
sweagraw	.469	.294	.255	.525	539
jindra.helcl	.435	.213	.206	.412	.377
baseline_aws	.281	.043	.041	.213	.198

Table 7: Weighted F1 scores on the test set for an "oracle" that outputs the top k translations from gold data. All translations ($k = *$) gives a perfect score of 1.0. For comparison, we include teams who submitted to all tracks, and one baseline. Underscores show the smallest value of k to outperform **jbrem** (the top system).

At $k = 1$, macro weighted F1 is still relatively low, showing that systems need to return more than a single translation to do well. Comparing $k = 1$ to **baseline_aws** (both output a single translation) shows that this high-quality baseline still does not generally produce the translation favored by Duolingo learners. It is also worth noting that top-ranking systems output the $k = 1$ translation more often than that of **baseline_aws** (83% vs. 69%).

The top-ranking teams performed on par with or better than the $k = 5$ oracle, and much better for languages with a higher translation-to-prompt ratio (see Table 2). This suggests that high-performing models for this task are consistently producing output comparable to the five most commonly-attested translations, and often beyond (at some expense to precision, for which the oracle is perfect).

4.5 Error Analysis

So far we have discussed only quantitative outcomes for the STAPLE task. Here we present a qualitative analysis by inspecting the most common **recall errors** and **precision errors** among participating systems. These help us to get a sense for how important typical errors are for our educational use case, and shed light on what performance gaps need to be closed in future work.

Alternative word order or synonym variations were a challenge for all teams in all tracks. For example, here are the top four accepted translations for a prompt in the Portuguese test dataset:

1. ***please** don't smoke*
   por favor, não fume ($w_1 = .663$)
   não fume, **por favor** ($w_2 = .030$)
   por gentileza, não solte fumaça ($w_3 = .011$)
   não fume, **se faz favor** ($w_4 = .011$)

Most teams produced the top-weighted translation, several more identified other variants of *please*, but few systems generated reorderings that place it after the main clause (which, for this instance, accounts for ≈ .184 of the total LRF weight). This can be partially explained by the use of fixed beam sizes. Since the number of translations grows exponentially with the number of lexical and structural variations, many correct combinations that the system could be capable of generating may still fall off the beam. One possible solution here would be to explore lattice-based decoding strategies that may avoid such bottlenecks.

Korean, Japanese, and Vietnamese have diverse sets of pronouns for use with different registers and relationships to the subject and the listener, as seen in this example from Japanese:

2. ***i** exercise*
   私は運動する (top translation)
   僕は運動する (not in accepted translations)

Here 私 (watashi) is the most common first person pronoun, but about half the submissions instead produced 僕 (boku) which carries with it more youthful or masculine connotations. While the latter is arguably correct, learners (especially beginners) are unlikely to use it, and it was also missing from the human-curated set of translations.

Pronouns were difficult in general, for multiple language tracks. All five languages allow some level of pronoun-dropping, as per these examples from Hungarian and Portuguese:

3. ***we** run to the garden*
   [mi] futunk a kertbe

4. *would **you** like to try on those shoes?*
   [você] gostaria de provar esses sapatos?

This resulted in both over- and under-use of pronouns, both in system outputs and occasionally gold data. While both variations (with or without the pronoun) may be correct, the rules governing which is more fluent or more appropriate for instruction are subtle, and remain challenging.

Systems often produced verb suffixes that convey discourse nuances or speaker attitudes not necessarily present in the English prompt or its accepted translations, as per these Korean and Japanese translations generated by multiple teams:

5. *the woman is pretty*
   그 여자는 예쁘네요
   ("**wow**, that woman is pretty")

6. *you are not a victim*
   あなたは被害者ではないよ
   (“you are not a victim, **you know**”)

One likely explanation for this is the pervasive use of OpenSubtitles data in pre-training; such suffixes are especially common in on-screen dialogue.

Mistranslation of numbers was a common problem for multiple languages, which is unacceptable for education, or indeed most applications:

7. *i have **eighteen** horses*
   tizenhárom lovam van
   (“i have **thirteen** horses”)

8. *she has **sixteen** cats*
   彼女は猫を六匹飼っています
   (“she has **six** cats”)

Correct noun declension was also a struggle for all systems, particularly the allative case in Hungarian (-hoz/-hez/-höz); the following example was not produced by any system:

9. *we run **to the** garden*
   elrohanunk a kert**hez**

Similarly, noun cases and postpositions in Korean led some systems to alter the sentence meaning:

10. *who do **you love**?*
    누가 너를 사랑하니
    (“who **loves you**?”)

For Japanese, many systems frequently used English loanwords in their translations:

11. *she makes me **happy***
    彼女は私をハッピーにしてくれる
    (uses phonetic English loan for **happy**)

These were generally missing from the gold data. Such loanwords are not especially rare, although one could also argue that using them is “cheating” in a language-learning context!

5 Related Work

The STAPLE task is similar to *machine translation* in that one takes input from one language, and produces output in another language. In fact, nearly all of the models used by participating teams were built using standard, off-the-shelf, modern machine translation software. But machine translation systems typically produce only a single output.

Ultimately our goal for Duolingo—a robust system for automatically grading learner translation submissions—is closer to the world of *machine translation evaluation*. Motivated by shortcomings of the BLEU metric (Papineni et al., 2002), some researchers have proposed alternative measures of evaluating MT systems against many references (Qin and Specia, 2015), or even exhaustive translation sets collected by human translators, as with HyTER (Dreyer and Marcu, 2012).

We even considered using these alternatives as official metrics for the STAPLE task. The main challenge is the difficulty of gathering *all* possible translations (the authors of HyTER estimate that creating all translation variants for a single sentence can take two hours or more), or the assumption that the translations are all equally important. To ease the burden of manually collecting references, there have been proposals for automatically generating them (Apidianaki et al., 2018) using paraphrase databases such as PPDB (Pavlick et al., 2015).

This brings us to other areas of research that are very related to our task: automatic *paraphrasing* (Wieting et al., 2015; Witteveen and Andrews, 2019), as well as research in diverse beam search methods (Vijayakumar et al., 2016; Li et al., 2016) for decoding multiple natural language outputs. We are happy that this shared task can serve as a forum for studying the intersection of these problems, and it is our hope that the STAPLE task data will continue to foster research in all of these areas.

6 Conclusion and Future Work

We have presented the STAPLE task, described a new and unique corpus for studying it, and reported on the results of a shared task challenge designed to explore this new domain. The task successfully drew participation from dozens of research teams from all over the world, synthesizing work in machine translation, MT evaluation, and automatic paraphrasing to name a few.

We learned that a pipeline of strong machine translation followed by fine-tuning on learner-weighted STAPLE data produces strong results. While the data for this task are geared toward language learners (and are therefore simpler than more commonly-studied domains such as newswire), it is our hope that the STAPLE task provides a blueprint for ongoing interdisciplinary work in this vein. All task data, including dev and test labels, will remain available at: `https://doi.org/10.7910/DVN/38OJR6`

Acknowledgements

The authors would like to thank Colin Cherry for seeding the idea that ultimately became the STAPLE task. Thanks also to the organizers of the Workshop on Neural Generation and Translation (WNGT) for providing a forum for this work, as well as all the participating teams. Special thanks to Nathan Dalal and Andrew Runge for help reviewing and summarizing the system papers.

References

Sweta Agrawal and Marine Carpuat. 2020. Generating diverse translations via weighted fine-tuning and hypotheses filtering for the Duolingo STAPLE task. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

Marianna Apidianaki, Guillaume Wisniewski, Anne Cocos, and Chris Callison-Burch. 2018. Automated paraphrase lattice creation for HyTER machine translation evaluation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 480–485, New Orleans, Louisiana. Association for Computational Linguistics.

R.H. Baayen. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge University Press.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Rakesh Chada. 2020. Simultaneous paraphrasing and translation by fine-tuning transformer models. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

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

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, pages 11179–11189.

Rejwanul Haque, Yasmin Moslem, and Andy Way. 2020. The ADAPT system description for the STAPLE 2020 English-to-Portuguese translation task. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

Masahiro Kaneko, Aizhan Imankulova, Tosho Hirasawa, and Mamoru Komachi. 2020. English-to-Japanese diverse translation by combining forward and backward outputs. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

Huda Khayrallah, Jacob Bremerman, Arya D. McCarthy, Kenton Murray, Winston Wu, and Matt Post. 2020. The JHU submission to the 2020 Duolingo shared task on simultaneous translation and paraphrase for language education. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.

Zhenhao Li, Marina Fomicheva, and Lucia Specia. 2020. Exploring model consensus to generate translation paraphrases. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

Jindřich Libovický, Zdeněk Kasner, Jindřich Helcl, and Ondřej Dušek. 2020. Expand and filter: CUNI and LMU systems for the WNGT 2020 Duolingo shared task. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, and Hasan Cavusoglu. 2020. Growing together: Modeling human language learning with n-best multi-checkpoint machine translation. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

Tadashi Nomoto. 2020. Meeting the 2020 Duolingo challenge on a shoestring. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

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. *arXiv preprint arXiv:1904.01038*.

Sebastian Padó. 2006. *User's guide to* `sigf`*: Significance testing by approximate randomisation*.

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.

Junsu Park, Hongseok Kwon, and Jong-Hyeok Lee. 2020. POSTECH submission on Duolingo shared task. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *CoRR*, abs/1511.08198.

Sam Witteveen and Martin Andrews. 2019. Paraphrasing with large language models. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 215–220, Hong Kong. Association for Computational Linguistics.

Krzysztof Wołk and Krzysztof Marasek. 2014. Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs. *Procedia Technology*, 18:126–132.

Michael Yang, Yixin Liu, and Rahul Mayuranath. 2020. Multi-step fine-tuning and encouraging diversity of high-coverage neural machine translation. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.