Human-assisted Neural Machine Translation: Harnessing Human Feedback for Machine Translation

Paige Finkelstein

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2020

Committee:

Shane Steinert-Threlkeld

Julia Kreutzer

Program Authorized to Offer Degree:
Linguistics

University of Washington

Abstract

Human-assisted Neural Machine Translation:
Harnessing Human Feedback for Machine Translation

Paige Finkelstein

Chair of the Supervisory Committee:
Shane Steinert-Threlkeld
Department of Linguistics

Neural machine translation (NMT) is a promising approach to the task of machine translation that has led to state-of-the-art results in many settings. However, NMT translations are still far from sufficient for many practical purposes. For this reason, there is a robust body of ongoing research on how to improve NMT systems with human feedback. This feedback can take many forms, including interactive-predictive NMT, post-editing of NMT output, and soliciting ratings or corrections of translations for the purposes of online learning. While these approaches are often effective, they are also often very time consuming and expensive. For that reason, there is important research into the question of how best to ensure that any human effort is used optimally. In this thesis, we contribute to this line of work by proposing a system that *learns* when it should ask for human feedback on a translation. This system makes use of an existing pre-trained NMT model, and introduces an additional *feedback-requester* model that learns to selectively solicit feedback from a human translator on the NMT translations. This system reduces human effort by directing attention to the most problematic sentences in a document, and the feedback-requester model itself is updated according to the translator's feedback. We also experiment with two active learning (AL) strategies for the feedback-requester model, and present a range of experiments simulating human translator use of the system and show the results over time.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

While the first attempts at using neural networks for the task of machine translation date back several decades, neural machine translation (NMT) was not generally considered a viable approach until deep learning started gaining rapid traction in the 2010s (e.g. Krizhevsky et al. [13]; Sutskever et al. [29]). Rather than being composed of many sub-components that are separately tuned, as occurs in statistical phrase-based machine translation, NMT instead "uses a single, large neural network to model the entire translation process," often referred to as direct "end-to-end" learning (Tu et al. [30]). Over the past few years, NMT models—most of which conform to an encoder-decoder architecture (Bahdanau et al. [2])—have been shown to outperform statistical machine translation approaches (Barraultet al. [4]). Papers presenting new state of the art results using NMT methods are now published frequently, and the yearly expansion of the Conference on Machine Translation (WMT)—in terms of number of submissions as well as number of tasks and language resource pairs—and the overwhelming predominance of NMT approaches in the submissions underscores the current popularity and advancement of NMT research (Barraultet al. [4]). However, NMT systems still have a particularly difficult time in certain scenarios, such as with long sequences of text (Yang et al. [33]). Even relatively straightforward samples can yield garbled and incorrect translations.

As NMT approaches to automatic machine translation have become widespread in both academia and industry, research focusing on how to best utilize human effort in conjunction with and to improve NMT models has also shown promising results. In particular, there has been significant work focusing on how to best utilize human effort towards improving NMT output by updating NMT models themselves. Various approaches incorporating interactive-predictive NMT, the use of reinforcement learning with different levels of human feedback,

and active learning for selecting the best samples for further improving an NMT model have shown promising results. However, one research question that remains under-explored, to the best of our knowledge, is the question of how to harness human effort most efficiently when the NMT model itself is not available to be updated and improved. In other words, of how to combine NMT model output with human feedback when the NMT model cannot be changed or updated itself but we can use the outputs of the model, specifically the predicted tokens as well as the probabilities of those tokens, to inform whether and how to solicit human feedback.

In this thesis, we contribute to NMT-related research by presenting an approach that works to optimize human translator feedback on NMT output, even when the NMT model itself cannot be updated. We present a system that takes an existing pre-trained NMT model and trains an additional *feeback-requester* model to determine which sentences from an NMT-translated output document to solicit human translator feedback for. This system reduces human effort by directing attention to the most problematic sentences in a document. Additionally, the feedback-requester model itself is updated according to the translator's feedback, i.e. whether they do decide to make changes on a sentence when prompted and how much typing effort those changes take as well as whether they post-edit sentences that they were not prompted to provide feedback on at the end.

We also experiment with two active learning (AL) strategies for the feedback-requester model: one that is that incorporates an entropy-based approach to help select sentences that will be most useful for the feedback-requester model to see, measured by the entropy of the feedback-requester model's predictions, and a second strategy in which we attempt to *learn* the AL sampling strategy that will provide the most informative examples for the feedback-requester. Finally, we present a range of experiments simulating human translator use of the system, using two different behavior *policies* to represent different levels of engagement translators might have with the system, and we present the results of simulated usage of the system over time.

The next chapter offers a review of recent work on relevant topics, including interactive

translation prediction, reinforcement learning for improving NMT models, and active learning with NMT. Chapter 3 will provide a detailed explanation of the system approach and implementation, as well as a description of the data used to train the feedback-requester model and some explanation about how the system could be adapted to use different pre-trained NMT models. Chapter 4 describes the various experiments simulating human translator interaction with the system and presents results for outcomes over time as the system is used, primarily focusing on chrF and BLEU score improvements for varying levels of (simulated) human effort. Finally, Chapter 5 discusses some key observations and implications of the system and experiments as well as lays a plan for future work. All of the code for the sytem as well as for running the experiments is publicly available on Github. [1]

[1] https://github.com/bolducp/human-assisted-nmt

Chapter 2
LITERATURE REVIEW

As NMT approaches to automatic machine translation have gained predominance in recent years, there has been concurrent productive research working to apply a variety of machine and deep learning strategies to further improve NMT models and maximize human effort. This literature review summarizes recent research that is most closely related and/or has inspired the approach in our system.

2.1 NMT and Human-in-the-Loop Learning: Industry and Academia

Because human translations—typically needed for training NMT models as well as for post-editing corrections of NMT output—are generally quite expensive to obtain both in terms of time and human effort, recent work on NMT in both academia and industry often focuses on the goal of reducing the level of human effort required to reach a satisfactory translation. Toward this end, one promising subarea that has gained increasing exploration over the past few years is the application of human-in-the-loop learning for NMT. Broadly speaking, human-in-the-loop (HITL) learning is the process by which a machine learning model receives and utilizes human intervention or feedback. Two examples of this approach applied to NMT systems in industry are the translation companies ModernMT[1] and Lilt,[2] both of which integrate human-in-the-loop learning into their platforms to increase efficiency for translators. The user interfaces for their software products are different; however, in both cases the companies use NMT systems that update according to human feedback to improve their models so that less human post-editing effort is required on subsequent translations.

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

[2] https://lilt.com/

Both Lilt and ModernNMT do not only market to companies seeking cost effective translation options, but also to translators themselves who are interested in using machine-assisted translation in their own work in order to translate more quickly.

Another company harnessing the power of NMT models for an industry translation product is Unbabel.[3] Unbabel partners with non-experts who post-edit NMT output to provide clients with translated content. Unbabel also frequently publishes NMT-related research, especially focusing developing frameworks for evaluating MT output and post-editing (e.g. Kepler et al. [7]). Because Unbabel works with non-professional users who post-edit the NMT output, they have a particular need for developing effective and efficient evaluating systems for ensuring post-editing quality.

From a business perspective, it is not surprising that there would be a broad market for the sort of translation products being offer by companies such as Lilt, ModernMT, and Unbabel. Many individuals and companies today need to make content available in multiple languages. While some use cases may require perfect, highly-skilled translations that can still only be produced by human experts, there are other cases where less precision and artfulness is required. For the latter, it is likely that the baseline result currently produced by automatic machine translation systems without post-editing corrections is often not sufficient, but the cost, in terms of money or time, is too much to warrant pure human translation. Thus, enterprise systems that harness NMT to increase speed for, but still rely upon input or correction from, human translators are able to provide a more cost-efficient option.

Human-in-the-loop learning applied to NMT is also gaining increasing attention in academic research. One of the main approaches used by both Lilt and ModernMT in the translator-facing products, a method known as interactive translation prediction (ITP), has also recently received fruitful attention in academic research. Recent studies have examined user experience and productivity in an ITP context as well as experimenting with reinforcement learning to improve ITP systems. Over the past two years, there have also been several

[3]https://unbabel.com/

studies published that examine active learning (AL) strategies for improving NMT models (e.g. Liu et al. [16]; Zeng et al. [34]). While there are a range of different methods for how AL can be incorporated into any given NMT system, the general idea of AL is that it introduces a learning process whereby the learner system is able to request additional labeled data to use for training. Because procuring labeled data is typically expensive, a large motivation for AL is for the system to be able to estimate which data points will be most beneficial for training the model so that it can request data annotations in the most cost-effective way possible. Thus, one of the crucial considerations of recent work on AL for NMT is determining the most effective sampling strategy for selecting further annotated data. Furthermore, AL is often used in conjunction with human-in-the-loop learning with the goal of optimally requesting intervention or feedback from the human, as determined by the specific AL strategy.

2.2 Interactive Translation Prediction (ITP)

The process in which a human translator corrects the output of a machine translation system, either sequentially as the system translates or at the end once the entire document has been translated, is commonly referred to as "post-editing." Post-editing, however, is generally considered not to be cost-effective when compared to full human translation (Simard et al. [28]). There has been research on automatic post-editing that attempts to address this shortcoming (e.g. Isabelle et al, 2007; Pal et al. [22]), including a WMT 2019 shared task. A different approach to improving the output of machine translation systems, however, is interactive translation prediction (ITP).

Interactive translation prediction, also known as interactive machine translation or interactive predictive machine translation, is an alternative to post-editing in which a human translator accepts or rejects incremental translation suggestions from the machine translation system in an auto-complete or drop-down option fashion, rather than having to rewrite or correct translation output from scratch. Early ITP systems used phrase-based machine translation systems (e.g. Och et al. [20]; Barrachina et al. [3]), but in recent years has been

applied to NMT as well.

Green et al. [6] introduce a new computer aided translation interface that has both post-editing and ITP modes and offer a comparative analysis of the human effort required and translation accuracy of the two approaches. They find that the post-editing approach is faster but that ITP approach has higher quality translation when the baseline machine translation quality is high. They also re-tune the MT system using the translator post-editing/ITP feedback and report that re-tuning on the ITP feedback leads to larger gains in terms of human translation edit rate (HTER), which has been shown to correlate with human judgments of fluency and is also a common measure of human effort. Their work on ITP is also notable for using prefix decoding for regenerating translation suggestions based on a user-inputted prefix.

Knowles and Koehn [8] use ITP with neural machine translation, finding that its use with NMT yields higher word prediction accuracy than the phrase-based machine translation method based on search graphs, even when the two baseline MT systems had the same translation quality. They explain how ITP fits naturally into an NMT system during the decoding process when, rather than using the model's own predictions as the conditioning context for the next prediction step, the prefix selected by the translator is used. They further compare an approach that uses beam search and one that does not, ultimately finding that beam search does improve BLEU scores but is too slow for a live system. Knowles and Koehn also demonstrate that their system is better able to recover from errors, i.e. predicts a sequence that is rejected by the translator, than the phrase-based model. Their experiments are based on a simulation of translator feedback in which preexisting human translations are used as input/feedback for the ITP system.

In later work, A User Study of Neural Interactive Translation Prediction, Knowles et al. [9] further their analysis of NMT with ITP using experiments with eight professional English-Spanish translators. Rather than using translated reference text to simulate translator input, the NMT system receives the actual translators' token(s) as conditioning context for the subsequent translation. They find that over half of the translators worked faster and

claimed to prefer the ITP method to post-editing, but did not find a significant difference between translation time between the two methods overall. They also provide some qualitative analysis of the translators' reactions to the ITP tool and analyze the types of errors (e.g. spelling, style, awkward language, grammar, etc.) that occurred most often with ITP versus post-editing.

2.3 Reinforcement Learning with ITP

There have been several recent studies that utilize reinforcement learning with IMT to improve NMT models while also maximizing for low human effort. Lam et al. [14] use human judgments on partial translations as reward signals to train an NMT model through reinforcement learning, where the entropy of word predictions are used as an uncertainty metric to trigger requests for feedback. One of the main focuses of this work is on reducing, and evaluating the consequent success, of human effort required in the translation process. Rather than requiring a translator to edit, delete, or select segments of NMT output, they only require judgments on the quality of partial translations. In their experiments, which simulate translator interaction, bandit feedback is generated by evaluating the predicted partial translation against reference translations using a character F-score metric (Popovic [25]). They also update the NMT model parameters after every interaction, i.e. after each partial translation, by means of an actor-critic reinforcement learning strategy. Lam et al. conclude that segment-wise reward signals from partial translations improve translation quality (BLEU score and character F-score) compared to rewards provided only for full sentences. They also find that this method is able to reduce the average number of feedback requests, decreasing human effort.

Lam et al. [15] use an approach that combines imitation learning and reinforcement learning with ITP for "model personalization" that solicits user feedback in the form of keep, delete, and substitute edits that incorporate model updates based on partial translations and generates alternative translations using constrained beam search. They use an "uncertainty criterion" corresponding to where the entropy of the policy distribution is highest to limit

the number of feedback requests, and also allow model updates based on partial translations. Their experiments simulate a human translator by comparing the partial NMT translations with gold translations. For the reinforcement learning setting, the simulated feedback is weak feedback in the form of keep and delete edits. For the imitation learning setting, the simulated feedback additionally injects substitute edits. They report the average BLEU score increases for 1. the weak feedback of keep and delete, 2. this week feedback plus substitute edits, and 3. full post-edits. Ultimately, they conclude that their results indicate that online learning from edits of partial translations with high uncertainty can achieve performance gains similar to supervised learning on in-domain data while requiring much less human effort to obtain feedback.

2.4 Reinforcement Learning with Real Application Data

Kreutzer et al. [10] explore using different methods of weak user feedback and offline bandit learning to update an NMT model. They offer the first analysis of using real-world human reinforcement to improve NMT (based on explicit and implicit user feedback from the eBay e-commerce platform) through offline bandit learning. Ultimately, they find that learning from the explicit feedback of five-point Likert scale (five star) ratings did not result in BLEU score improvements but that using implicit task-based feedback (implemented as part of a cross-lingual product search) as a reward signal for NMT optimization resulted in BLEU score and individual word translation gains. In a subsequent study, however, Kreutzer et al. [12] show that it is possible to obtain BLEU score improvements when the five-point ratings are reliable indicators of translation quality.

2.5 Active Learning for NMT

2.5.1 Active Learning Sampling Strategies for NMT

González-Rubio et al. [5] explore different active learning strategies for ITP with statistical machine translation (SMT) and introduce an AL framework for ITP that splits the data

stream into blocks of sentences and applies AL techniques individually for each sentence block. They test three sampling strategies, namely, random sampling, n-gram coverage sampling, and dynamic coverage sampling. They conclude that dynamic coverage sampling is the most effective and can be used to greatly reduce human effort required to translate sentences in the stream.

Peris and Casacuberta [23] perform an analysis of active learning strategies in a prefix-based IMT framework for unbounded streams of data for NMT. They examine four uncertainty sampling strategies: 1. quality estimation, which estimates the quality of a translation according to the confidence scores for each word; 2. coverage sampling, which uses the translation coverage to calculate uncertainty; 3. attention distraction sampling, which looks at the attention probability distribution to determine translation confidence; 4. random sampling, which is considered as a baseline and also used in conjunction with other sampling methods as a means of introducing noise to prevent overfitting. These four sampling strategies are then incorporated in a query-by-committee (QBC) method that selects samples that have the highest level of disagreement among the four strategies. In their experiments, they simulated human translator feedback using gold translations, and compared the four sampling strategies separately as well as the QBC method combining them in terms of translation accuracy (BLEU score) and human effort required (keystroke mouse-action ratio (Barrachina et al. [3])). They conclude that incorporating AL, even the baseline of random sampling, improves BLEU score and decreases human effort by approximately 25%.

Zeng et al. [34] describe sampling strategies, or "scoring functions" in their AL framework, as being divisible into two main classes: model-driven and data-driven approaches. Model-driven approaches rely upon the model, labeled dataset, and unlabeled dataset to sample instances where the model is most uncertain about the translation, while data-driven approaches use only the labeled and unlabeled data. They design an experiment to test several sampling strategies that fall into each of these categories for use with a current state-of-the-art Transformer architecture based NMT system, and their experiments are based on simulated human feedback via gold translation in a batch setup. The model-driven class of

methods they explore include 1. Least Confidence, which estimate the model uncertainty of a sense by averaging token-level log probability of the decoded translation; 2. N-best Sequence Entropy (NSE), which computes the entropy of the n-best hypotheses; 3. Coverage Sampling, which calculates the extent of coverage of the attention weights over the source sentence; and 4. Round Trip Translation Likelihood (RTTL), a new method that offers a "neural extension" of an existing sampling strategy used in phrase-based MT. In RTTL, two models are trained—one that translates from the source language to the target language, and another in the opposite direction-—and during training, a sample sentence is first translated into the target language and then that output is translated back into the source language, but "instead of decoding, [they] compute the probability of the original source sentence x and use it as a measure of uncertainty" (86). The data-driven methods they explore are n-gram overlap and cosine similarity (computed in several different ways including a bag of words representation, contextual embedding, and paraphrastic embedding). Their results show that all of the sampling methods except for coverage sampling outperform a random baseline. They hypothesize that coverage sampling's low performance is due to the fact that it relies on the attention mechanism and the Transformer's more complex multi-headed and multi-layered attention architecture leads to attention scores that are "not reliable enough to be used with the AL methods" (92).

2.5.2 Learning an AL Selection Strategy for NMT

Liu et al. [16] experiment with learning a selection policy to use for active learning selection with NMT, rather than using a predetermined heuristic or policy as typically done. Their main focus is on using a high-resource language pair to train the active learner and then transferring the AL selection policy to a low-resource language pair that shares some characteristics. They also use a pool of monolingual source text (assuming a high resource language), making use of multilingual word embeddings trained on monolingual text and bilingual dictionaries. They use imitation learning to train the query policy and conduct experiments on three language pairs, namely, Finnish-English, German-English, and Czech-

English, and their approach for learning the AL strategy is based on a Hierarchical Markov Decision Process (HMDP) formulation of pool-based AL, used in Bachman et al. [1] and Liu et al. [16]. They imitate low-resource language scenarios, using varying amounts of held out translation data to simulate human feedback.They compare their learned policy with random, length-based, and total token entropy sentence selection strategies and conclude that the learned selection policy is more successful than dominant heuristic-based policies across the three language pairs in their experiments.

Chapter 3

METHODOLOGY AND IMPLEMENTATION

3.1 Overview: Human-in-the-loop NMT System

In this thesis, we build a human-assisted NMT system that aims to maximize the quality of translation, while reducing the total amount of human translator effort involved. At a high level, we construct a system in which we endeavor to use the NMT translation directly when it is of sufficiently high quality, and ask for human feedback to improve the translation only when it is poor.

Our system is comprised of two modeling components. The first is a pre-existing **NMT model**, with which we can obtain a predicted translation for any input (in the model's accepted source language(s)), but which we are unable to adjust or update. The second is a **feedback-requester model**, which uses the output of the NMT system as well as the untranslated source sentence as input to predict for each sentence whether human translator feedback should be solicited to improve the NMT translation. For the current version of this research, we have created a command line interface tool through which requests for human feedback will be made and used to correct the translation. The interface also displays the translated document upon completion for the human translator to further post-edit if desired. To our knowledge, the effects of presenting the translated sentences in context have not previously been explored in active learning systems for NMT. All human feedback provided at any stage is used to improve the feedback-requester model.

We also explore the use of active learning techniques to maximize the benefit of the feedback-requester model. The feedback-requester can learn from the answers of the human translator in response to a request for feedback. For example, if the translator does not alter an NMT translation after a request for feedback, the feedback-requester may learn that

it need *not* have made the request in the first place. In order to best train the feedback-requester, we take account of this dual purpose of human feedback—first, to improve the translation output and, second, to produce training data for the feedback-requester—when determining whether to request feedback. We do so using active learning techniques, described in the *Active Learning Adaptation* section below.

3.2 Baseline Human-assisted NMT System: Architecture Details

Figure 3.1: System Architecture

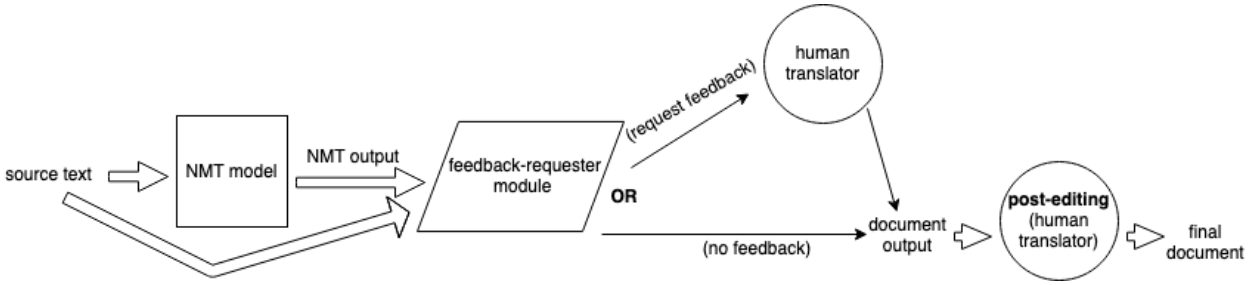


Figure 3.1 above outlines the flow of the system when being used by a human translator. When the feedback-requester determines it necessary to ask for human translator feedback for a sentence, the source text sentence and NMT translation in question will be presented to the translator to accept or edit. Once the system has worked through an entire document, the full translation will be displayed to the user for post-editing. Any interaction with the system by the translator will also be used to update the feedback-requester model once the document translation is completed.

The NMT module used in our system could be any trained NMT model, as it will is treated as incapable of being adjusted or updated and only its final layer (specifically the output tokens and their probabilities) is used. For the purpose of this thesis, we used the recently published JParaCrawl pre-trained Japanese-to-English model for our NMT module. More information on this model will be provided in the *Implementation Details* section below.

The feedback-requester module is a binary classifier neural network. We experimented with the architecture of the model, and have currently settled on an LSTM with one hidden layer. For training the feedback-requester, we used a parallel corpus consisting of Japanese source sentences and corresponding English target translations (further details provided in the *Training Data* section). The feedback-requester model produces a prediction about whether human feedback is needed for each sentence in the document. It accepts a single sentence from the source text, and the final layer of the NMT model run on that sentence, as input. The output prediction of the feedback-requester model is a single value between 0 and 1, where 0 means certainty that no human feedback is required and 1 means certainty that feedback is necessary. The threshold for what prediction value demarcates needing human feedback on a sentence is adjustable, and we experiment with setting the threshold point to different levels for the feedback-requester classifier output (e.g. 0.25, 0.5, 0.75, 0.9) to see how changing this value corresponds to differences in document translation quality and in required human effort, as defined below.

3.2.1 Feedback-requester Loss Function

The feedback-requester module can make two types of errors: 1. It can solicit feedback for an NMT output sentence that is already well translated. 2. It can fail to solicit feedback for a badly translated sentence that needs human correction. The feedback-requester model is trained against a loss function designed to minimize requests for feedback when the NMT translation is sufficient and maximize them when not.

The loss function for training the feedback-requester model penalizes both unnecessary requests and missed requests, taking into account the relative degree of severity for each as represented by the chrF score between the NMT translated sentence and the gold target translation. The intuition is that sentences for which the model thinks feedback would improve the translation, as represented by a prediction near 1, should have low chrF scores. Likewise, sentences that the model thinks are well translated, as represented by a prediction near 0, should have low chrF scores.

In formula terms, the loss used during the initial training is:

$$\mathcal{L}_1 = \lambda_1 M(s)F(s) + \lambda_2(1 - M(s))(1 - F(s)) \tag{3.1}$$

where s represents the input sentence, $M(s)$ is the prediction about whether a feedback request would improve the translation and $F(s)$ is the chrF score, and λ_1 and λ_2 are hyper-parameters used for weighting the significance of false positives (i.e. feedback is requested when the sentence is adequate) and false negatives (i.e. feedback should have been solicited but was not).

As previously mentioned, we also update our model based upon all interactions that a translator has with the system. Such "online" learning updates will occur at the completion of each document to avoid model updates during translation that would increase the system's latency. At this stage, we make use of a different loss function that incorporates all the available information about the translator's feedback, including requested feedback sentences and any post-editing. Specifically, our loss function is:

$$\begin{aligned}\mathcal{L}_2 = \frac{1}{|\mathcal{R}|}\sum_{s\in\mathcal{R}}\Big(\lambda_1(M(s)F(s)) + \lambda_2(1 - M(s))(1 - F(s))\Big) \\ + \frac{1}{|\mathcal{P}|}\sum_{s\in\mathcal{P}}\Big(\lambda_3(1 - M(s))(1 - F(s))\Big) \\ + \frac{1}{|\mathcal{N}|}\sum_{s\in\mathcal{N}}\lambda_4 M(s)\end{aligned} \tag{3.2}$$

where $\mathcal{R}$ represents the set of sentences for which feedback was requested and $\mathcal{P}$ represents the set of sentences that were not requested and later post-edited, and $\mathcal{N}$ represents the remaining sentences. During training we batch sentences sequentially, such that each batch will tend to have sentences of each type. As before, $F(s)$ represents the chrF score for a sentence s, and $M(s)$ is the prediction of the feedback-requester about whether asking for feedback would improve the translation. The hyperparameters λ_1, λ_2, λ_3, and λ_4 are used to set the relative importance of each term. Experimentally we found that some settings of these hyperparameters led to an unstable training process, whereby the feedback-requester would

tend to make uniform predictions as the extremes, i.e. would predict 0 or 1 for all sentences. We manually tuned the hyperparameters to yield a more stable training process. In the future, it may be fruitful to automate the tuning process to optimize for the effectiveness of human effort.

This loss function (Equation (3.2)) enables us to experiment by varying the weight placed on sentences that receive feedback requests, that are post-edited (after having not received a feedback request), and that do not receive feedback requests and are left as-is during post-editing.

3.3 Active Learning Adaptation

We also experiment with adding an active learning adaptation to training our feedback-requester model. In this case, the feedback-requester model is jointly optimized for the user objective of obtaining the best translation using the NMT output for minimal human input (as in the baseline human-assisted NMT system setup) and also for the system objective of obtaining the most useful human feedback for training the feedback-requester model itself (i.e. active learning). Above we have described how the feedback-requester model learns to optimize for the user objective. In this section, we describe two approaches to for training the feedback-requester model to optimize for the system objective. In both cases, a decision about requesting feedback incorporates both the user objective and the system objective.

In particular, suppose δ_1 is a measure of whether requesting feedback will improve the translation (the user objective), and δ_2 is a measure of whether requesting feedback will improve the model's ability to evaluate whether feedback is needed (the system objective). Then feedback is requested if a weighted sum of these measures exceeds some threshold, with γ_1 and γ_2 being the user-specified weighting factors as follows:

$$\gamma_1\delta_1 + \gamma_2\delta_2 \geq \text{threshold} \tag{3.3}$$

We explore two different active learning approaches for characterizing the system objec-

tive (i.e. calculating δ_2 in Equation (3.3)), and compare the resulting document translation quality and human effort expenditure of both approaches with the results of the initial baseline system.

3.3.1 AL Approach 1: Entropy-based Uncertainty Sampling

Our first active learning technique is to use a pre-specified sampling policy that incorporates uncertainty sampling (Lewis and Gale, 1994). Under this policy, it is assumed that the feedback-requester model will learn the most from obtaining labeled data on sentences for which it is the least "certain" as defined by the entropy of its output. In this setting, if δ_1 is a value between 0 and 1 predicting the improvement to translation of requesting feedback, δ_2 is defined as the entropy of the Bernoulli distribution with parameter δ_1.

3.3.2 AL Approach 2: Learning the Sampling Policy

Under the entropy-based sampling policy, we assumed that the feedback-requester model will "learn" most when its predictions have high entropy. In our second approach, we learn, rather than assume, which sentences will "teach" the feedback-requester model the most. We still jointly optimize for the user objective of obtaining the best translation using the NMT output for minimal human input and for the system (active learning) objective of obtaining the most informative samples for training the model. In this case, however, rather than specify the system objective as the entropy of the output from the feedback-request model (i.e. δ_2 in the formula above), we instead learn this objective itself.

In order to do this, it is necessary to first define a measure of how much the model has learned. There are various ways one might decide to try to measure the extent to which the model has learned as a result of seeing any given training example, and in Section 5.2 we will discuss at least one alternative to our current approach. For the purposes of this thesis, we use the magnitude of the gradients for the feedback-requester model's weights after observing a training sample as a measure of how much information the model gained

from that example. The following paragraph outlines some of the technical implementation details of this approach.

Because for this approach we are trying to learn both the user and system objectives, our feedback-requester model must be adjusted to provide output predictions for each of these. While the baseline feedback-requester model has a final output layer that provides a single numerical prediction between 0 and 1 corresponding to the likelihood that the sentence should receive human feedback, this learned AL sampling strategy requires two output predictions: one for the likelihood that a sentence needs to receive feedback, and one for predicting how much will be learned by receiving said feedback about a sentence. The second output in our setup is concretely defined as a prediction of the sum of the means of the absolute values of the gradients of the weights with respect to the loss for that sentence. When performing further online learning as the system is used over time, a similar metric is used that takes into account both the user objective and the magnitude of any resulting gradient change from obtaining human feedback. Aside from adding an additional linear layer to handle producing the system objective predictions and returning this value in addition to the user objective prediction, the architecture of the learned AL model is the same as the baseline feedback-requester model.

3.4 Pre-trained NMT Model and Feedback-Requester Model Training Data

3.4.1 JParaCrawl NMT Model

While the high level goal of this thesis is to present a system structure that could be used with any pre-trained NMT model for any pair(s) of source and target languages, for the purpose of maintaining a manageable scope for this current system, we use one specific pre-trained NMT model: the JParaCrawl Japanese-to-English model [18]. This model, published and released in 2019, was trained using fairseq [21], and based on the Transformer architecture described in Vaswani et al. [31]. The base-sized model (which is what we utilize for our experiments in this thesis), uses an encoder/decoder with six layers, an input embedding

size of 512, feed-forward embedding size of 2048, and eight attention heads for both the encoder and the decoder [18]. The model was trained on the JParaCrawl Corpus, which contains over 8.7 million English-Japanese parallel sentence pairs obtained from the web.

As the JParaCrawl model was trained using fairseq [21], there was a robust API for interacting with the pre-trained model. In particular, we use some of the built-in fairseq functionality, as well as a couple small tweaks to the library, in order to obtain not only the predicted tokens output for a given input sentence but also the token probability scores from the final layer of the model. Our project README provides an explanation of how to structure the input data for the feedback-requester so that it will be clear how a different pre-trained NMT model **not** using fairseq could still be used in our system. [1]

3.4.2 Training Data

For initially training the feedback-requester model, we use the Kyoto Free Translation Task (KFTT) corpus [19], which is a corpus of over 500,000 parallel Japanese-English sentences from Wikipedia articles related to Kyoto. To obtain our training data set, we passed the Japanese sentences into our pre-trained (JParaCrawl) NMT model to obtain our "starting" NMT output, and we used the English sentences as the gold standard "human" translations. We also provide a sequence of helper functions to transform the NMT output into the specific form expected as input by the feedback-requester model. We use the 'bert-base-multilingual-cased' model provided by HuggingFace [32] to obtain contextualized word-piece vectors for each token in the NMT output (specifically, we sum the last four hidden state layers from the BERT output for each word-piece token). We concatenate each word-piece vector to the mean of the source sentence word-piece tokens, used to represent the whole source sentence, and the first 50 token likelihood probability tensors from the NMT model output, used to represent prediction uncertainty. If there are fewer than 50 tokens in the NMT output sentence, the remaining values are padded as 0s, and if there are more than 50 tokens, the

[1] https://github.com/bolducp/human-assisted-nmt/blob/master/README.md

list is truncated to 50 to limit the size of the embedding, for computational tractability. Ultimately, this results in input sequences of variable length where each token is represented by a 1586-dimension embedding comprised on the contextualized word-piece embedding, the source sentence embedding, and the NMT output likelihood probability for that token.

3.5 Interactive Command Line Interface Tool

One contribution of this thesis is to provide a prototype of an interface that could be used by translators directly to efficiently utilize NMT output, without needing machine learning knowledge. Towards that end, and to facilitate testing the system for developing the experiments described below, we include an interactive command line tool meant to provide the base functionality that, in future versions, could become a desktop application or browser-based web application.

As with running any iteration of the system, the parameters for the threshold at which to request human feedback and the specific pre-trained NMT model as well as the feedback-requester weights are configurable for running the command line tool. The user can also set the desired approach (i.e. baseline, entropy-based active learning, or learned sampling active learning) for the feedback-requester model. This prototype tool works as follows:

1. The user/translator is prompted to input a document of text.

2. The system generates NMT output per the specified pre-trained NMT model, does the necessary pre-processing, and provides input for the feedback-requester.

3. The feedback-requester determines whether or not to solicit the user's feedback for each NMT output sentence one at a time, left to right. For sentences that score above the threshold, the user is shown the source sentence and the NMT output sentence and asked to either edit or accept the sentence, in sequential order.

4. After the user has been prompted to give feedback on all of the sentences meeting the threshold, the fully translated document including any edits they made is presented. At this point, the user can make any additional post-editing changes and submits the final document.

5. The user's responses to prompted feedback requests as well as any post-editing is used to update the feedback-requester model, and the user is prompted asking whether they would like to save the new weights.

Further explanation about running the interactive command line version are available in the Github repository. [2]

[2]https://github.com/bolducp/human-assisted-nmt/blob/master/README.md#command-line-interactive-version

Chapter 4

EXPERIMENTS AND RESULTS

To evaluate the performance of our system, we automate its interaction with a simulated human translator. We simulate two different translation policies, which guide how the simulated translator interacts with the system. These two separate policies are meant to mimic different approaches a translator might take in terms of translation effort. We track both the quality of the translation produced by the system, and the amount of "human" effort involved, demonstrating that our system allows for users to trade off between the two.

4.1 Translator Behavior Policies

For our first policy, referred to as "Full Policy," we will assume that the translator will fully correct each sentence, either when prompted by the feedback-requester during the translation process or afterwards when reviewing the full document (i.e. as post-editing). It should be noted that we do not see this behavior in practice as the most likely use case for our system, which is attempting to provide an efficient middle ground between pure NMT output and full human translation. However, we think it will be a useful baseline for comparing with our other human translator policy.

Our second policy, termed "Efficient Policy," is based on the assumption that the translator is aiming for speed as well as accuracy and that they are at least slightly more likely to correct a sentence if they are prompted specifically about it, rather than just reading it in the entire document output (i.e. more likely to correct when the feedback-requester asks about a sentence than when post-editing). We use the gold translations to represent the ideal translation to compare against the NMT output sentence, and define the Efficient Policy as follows:

- If asked by the feedback-requester and the chrF score is <= 0.75: **fix/replace**
- If not asked by the feedback-requester (i.e. post-editing) and the chrF score is <= 0.60: **fix/replace**
- Otherwise: **don't fix/replace**

The values of 0.75 and 0.60 in the Efficient Policy's rules were selected after several rounds of experimentation, with the goal of selecting values that would marginally adjust (change less than 20% of) the simulated translator's decisions.

4.2 Experiment Details

To compare the different approaches in translator behavior as well as the results of continued online learning and the two active learning strategies, we create a framework for easily running simulated translator interactions of many documents and tracking the results of the system as it processes, and receives feedback for, more documents.

4.2.1 Data

For these experiments, we use a different corpus of Japanese-English parallel sentences, the Japanese English Subtitle Corpus [27]. This corpus contains primarily conversational dialogue from movies and television shows, representing a very different domain than the parallel text Wikipedia articles initially used for training the feedback-requester. This domain is also notoriously difficult for automatic translation, as conversational Japanese often drops subjects, objects, and other entities that can be understood from context. We perform the same pre-processing steps on this corpus to obtain the NMT system output and prepare the feedback-requester input, and add the additional step of grouping sentences into documents (per a user-selected parameter of number of sentences per document). For the experiments in this thesis, we randomly divided the Japanese English Subtitle corpus into documents of varying numbers of sentences, between 25 and 100. For the experiments described below,

we first performed these experiments using 1249 documents of 50 sentences each, and then again using 4679 documents of 32 sentences.

4.3 Experiment Hyper-parameters

For the baseline feedback-requester model loss function used during training, Equation (3.1), λ_1 was set to 1.75 and λ_2 was set to 1.0. For the loss function used when performing online updates after each document (Equation (3.2)), λ_1 is 2.35, λ_2 is 0.5, λ_3 is 0.55, and λ_4 is 0.85. These were manually selected as described in Section 3.2.1.

For the Learned Sampling Active Learning model, the user-objective loss function was the same as that used in the baseline model with λ_1 set to 1.75 and λ_2 and set to 1.0, and for user-objective portion of the loss function used when performing online updates after each document , λ_1 was 2.35, λ_2 was 0.5, λ_3 was 0.15, and λ_4 is 0.5.

For the weighting terms applied to the user objective and system objective in Equation (3.3), for the entropy-based AL approach, γ_1 was set to 0.5 and γ_2 was 0.7. For the learned sampling AL approach, γ_1 was set to 10 and γ_2 was 0.6. These hyperparameters were also manually selected through cross-validation to yield stable training procedures.

4.4 Evaluation Metrics

We measure document translation quality in terms of chrF score as well as BLEU score. We report BLEU score as it is still the predominant measure of evaluation reported in MT research. However, we note that it is problematic as a metric for several reasons. Aside from being designed as a corpus-wide metric (not to be applied to a single document) and for use with multiple source translations (rather than just one gold translation), much recent research has shown BLEU scores are often misleading and easily influenced by hyperparameter choices (e.g. Post [26]) and, worse, that BLEU scores are not correlated with human judgments for comparing NMT systems that are relatively similar in quality (Mathur et al. [17]). We measure human effort in terms of a variant of the calculated keystroke mouse ratio (KSMR) score, similar to the approach in Kreutzer and Riezler [11] and Peris and Casacuberta [24],

except that it is modified to also include a base effort amount for each sentence that receives a feedback request, as in our system the human translator is not prompted to evaluate all output sentences (unless the threshold for asking for feedback is set to 0) and we want to account for the base effort of being prompted to evaluate a given translation, even if no correction is made.

For each simulated document translation loop (i.e. our simulated translator policy responding to the prompted sentences and further post editing if deemed necessary by the specified behavior policy), we calculate several different metrics:

1. The original BLEU score of the NMT output document.

2. The original chrF score of the NMT output document.

3. The percentage of sentences for which the feedback-requester prompts translator feedback.

4. The BLEU score for the document after translator input has been solicited (but before any final post-editing).

5. The chrF score for the document after translator input has been solicited (but before any final post-editing).

6. The human effort score (KSRM variant) required to address the feedback requests for the document.

7. The BLEU score improvement after translator input has been solicited (but before any final post-editing). This is simply the score calculated in metric #4 minus the score calculated in metric #1.

8. The chrF score improvement after translator input has been solicited (but before any final post-editing). This is simply the score calculated in metric #5 minus the score calculated in metric #2.

After calculating these results for each document, we create graphs to evaluate these metrics for a large number of documents, and to see whether and how they change over time, as the simulated translator uses the system. We provide tools for easily generating

these plots, enabling the user to specify a number of documents to group by (i.e. the metrics are averaged for each sequential specified number of documents), to make the plots easier to read for large numbers of documents. Or if the user wishes to keep each document as a unique data point on the plot, this **per_docs** parameter can simply be set to the value 1.

For the multiple document plots, we chart the eight metrics listed above that are calculated for each document, showing changes over time. We also provide graphs showing how the BLEU score improvement and chrF score improvement (i.e. metrics #7 and #8) correspond with varying levels of human effort (KSMR variant).

4.5 Compared Approaches

We evaluate and compare five different settings for the system in our experiments. First, we run the baseline system with no online learning, testing both of the translator behavior policies. These approaches are referred to as **"Full Policy"** and **"Efficient Policy"** in the plots. As there is no updating of the feedback-requester weights throughout the simulations for these two settings, we would expect the results to stay approximately the same, or vary randomly, over time.

In the rest of the three compared approaches, we also use the efficient policy behavior (though we do not put this in the graph keys to save space), as we believe this behavior is more representative of how a human translator would use the system. The third approach, **"Online Learning"** we test is adding the online learning component to the system, so that the feedback-requester model is updated after each document to incorporate the simulated translator's prompted and post-editing feedback. The fourth approach, **"Entropy-based AL"**, shows the results of the entropy-based active learning strategy, also with online learning after each document. And the fifth approach, **"Learned Sampling AL"**, shows the results of the second active learning strategy with online learning.

4.6 Results

These experiments were performed using 1249 documents of 50 sentences each, and then again using 4679 documents of 32 sentences. However, as the results were largely comparable, we display the results on the sample of 1249 documents, as it is easier to see variance between documents (i.e. requires less grouping to fit all of the data points on the plots). For space considerations, we include only some select graphs here. However, all of the plots for both rounds of experiments (1249 documents and 4679 documents), including the BLEU metric evaluations, can be found in the Github repository. [1]

4.6.1 Request Thresholds and Percentage of Sentences Requested

This first set of plots (Figures 4.1, 4.2, 4.3, and 4.4) shows the percentage of sentences the received requests for feedback by the feedback-requester for the document. The documents are grouped by 10 in sequential order and their metrics averaged, to make the plots legible.

It can be seen that the percentage of sentences requested for **Full Policy** and **Efficient Policy** of the baseline system is always the same, which is as expected, as the only difference here is how the simulated translator responds to the request, and their is no online learning in place. We can also see that the percentage of sentences requested changes very little between the different threshold values. This is because after the initial round of training, the large majority of our feedback-requester model's predictions are very close to either 0 or 1, and as we are not doing any online learning in these configurations, that never changes.

[1] https://github.com/bolducp/human-assisted-nmt/tree/master/hnmt/feedback_requester/experiments/plots

Figure 4.1: Percentage of Sentences Feedback Requested Threshold 0.25

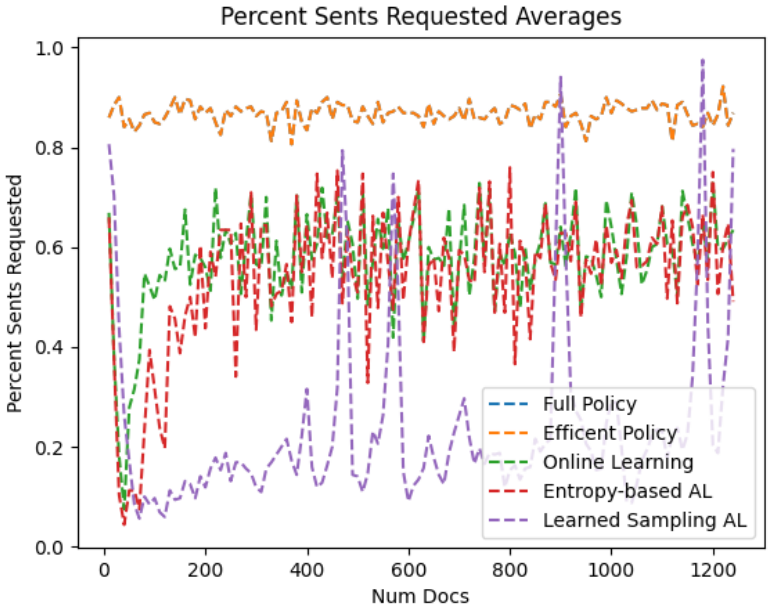


Figure 4.2: Percentage of Sentences Feedback Requested Threshold 0.5

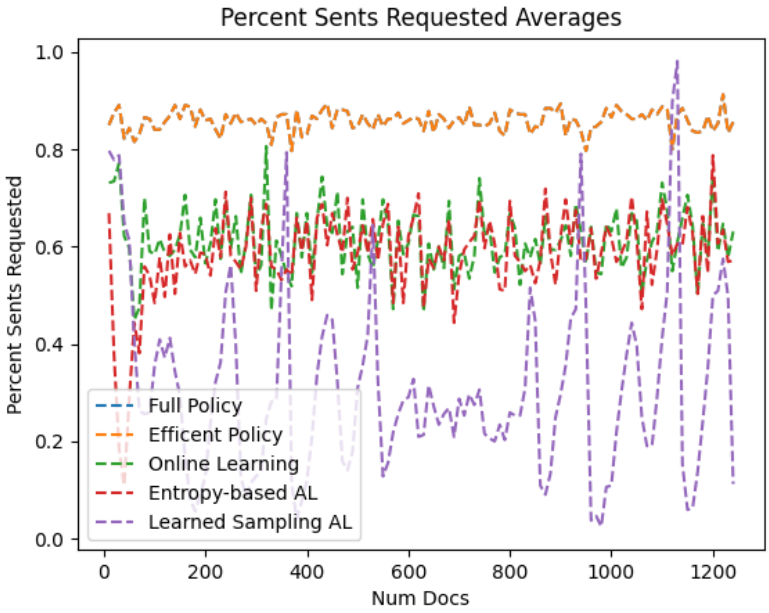


Figure 4.3: Percentage of Sentences Feedback Requested Threshold 0.75

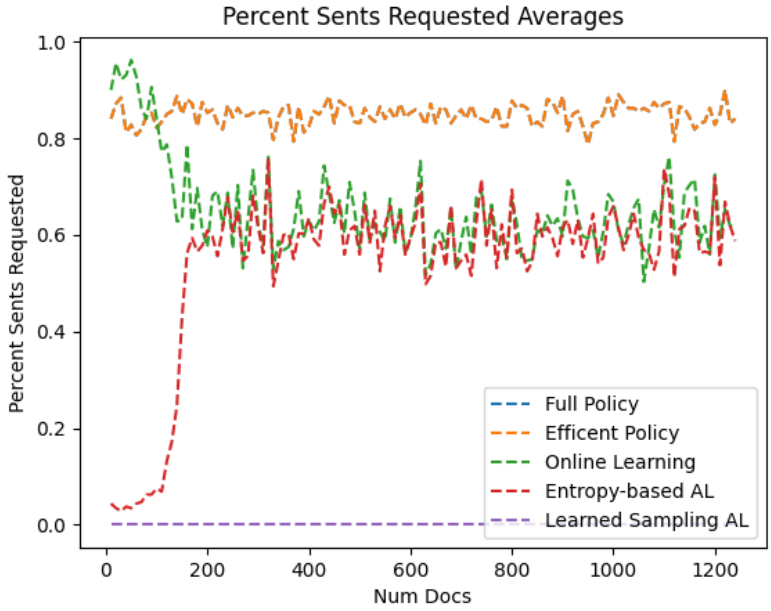
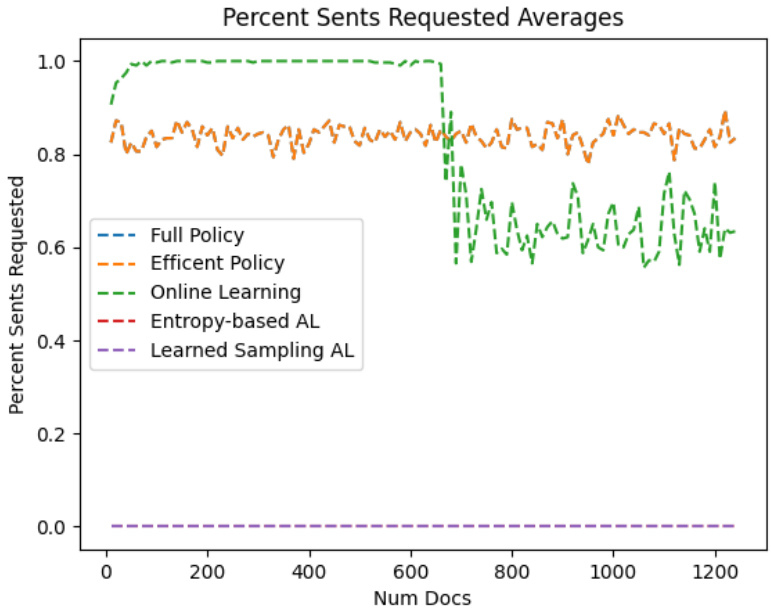


Figure 4.4: Percentage of Sentences Feedback Requested Threshold 0.9



4.6.2 Translation Quality v. Human Effort

These plots (Figures 4.5–4.8)[2] show the gain in translation quality, measured in net increase in document chrF score from the original NMT output document to the document after the prompted feedback requests (but not including any post-editing), and the corresponding amount of human effort (normalized KSMR variant score) required to obtain the increase. The normalized KSMR variant is computed by starting with a base effort score that each feedback-prompted sentence receives, to represent the minimal effort required to read the source sentence and its NMT translation, and adding this base score to the number of character edits between the NMT output translation and the user-corrected sentence (or reference translation for these simulations), as computed by the Python `difflib`[3] library.

A few observations of interest here include the fact that the baseline system results for both the simulated translator policies (i.e. **Full Policy** and **Efficient Policy**) are very similar at all thresholds, that the active learning strategies (in the settings where they do not uniformly predict no feedback) both widely expand the range of translation quality/effort trade-offs across documents, and that the relationship between improvement in chrF score and required effort seems to be largely linear.

[2]For separate graphs of each approach type, see Appendix .1.

[3]https://docs.python.org/3/library/difflib.html

Figure 4.5: Translation Quality v. Human Effort Threshold 0.25

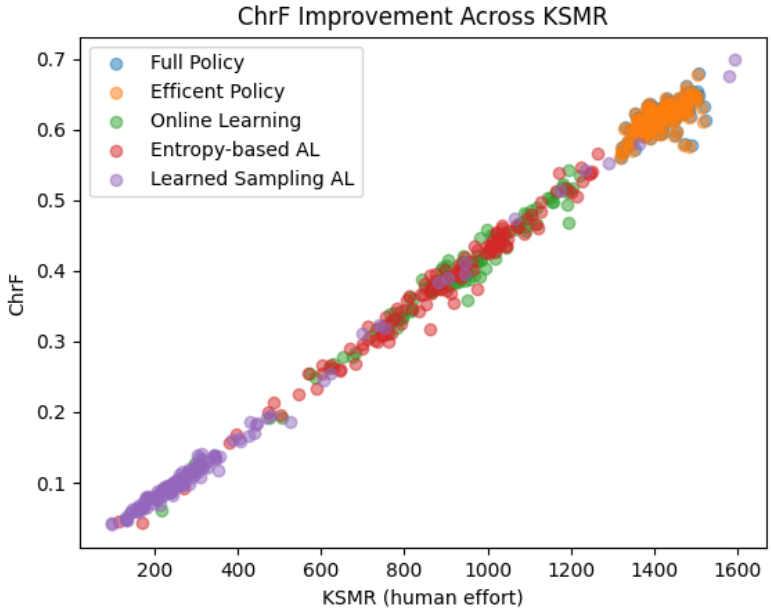


Figure 4.6: Translation Quality v. Human Effort Threshold 0.5

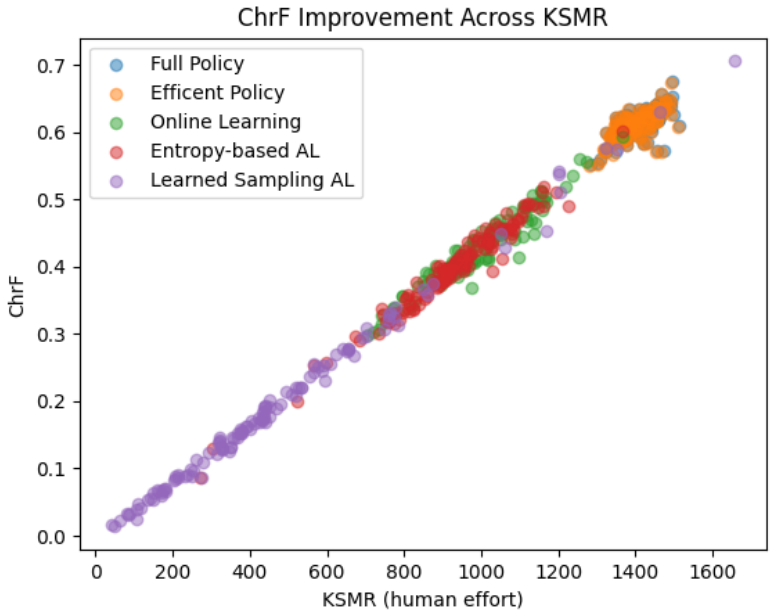


Figure 4.7: Translation Quality v. Human Effort Threshold 0.75

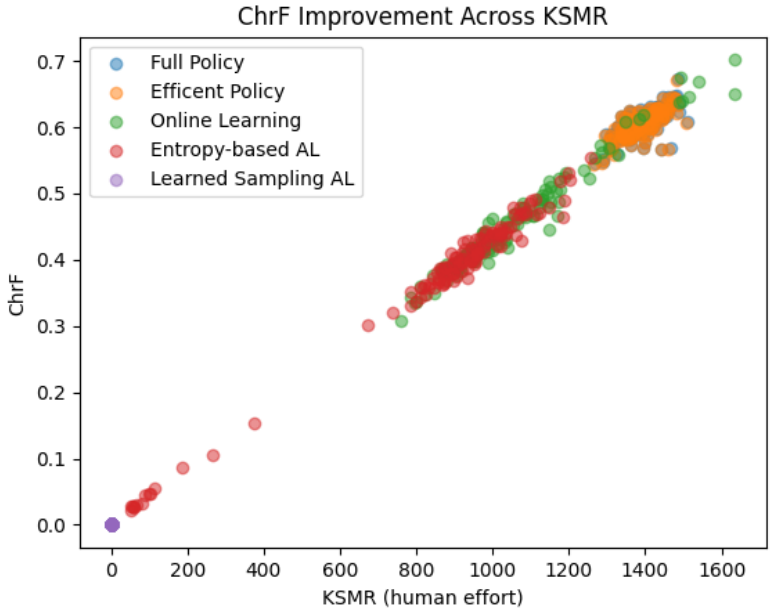
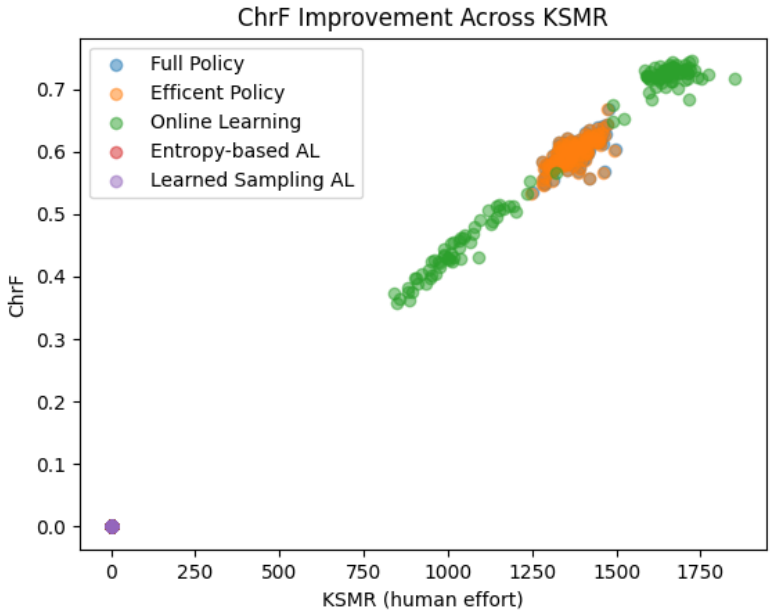


Figure 4.8: Translation Quality v. Human Effort Threshold 0.9



4.7 Examples

It is also informative to consider some specific examples of the sentences receiving feedback requests for a given document in the different approaches. In order to compare, we'll consider a small subtitle document segment, and examine which sentences are requested for feedback by each approach, using the weights that were saved after the experimental run on 1249 documents described above and setting the threshold value at 0.5 for all approaches.

We'll use a randomly selected 15-sentence document segment, with the following source Japanese text:

ミサキって知ってるか
三年三組のミサキ
三組にそんな人いたっけ
二十六年前の話な
そいつさ一年の頃から人気者でさ
頭も顔も性格もよかったらしいんだ
だから生徒にも先生にも好かれててさ
いるよねえ、学年に一人はそういう人
ところが三年に上がってすぐそのミサキが死んじゃったのさ
どうして
事故って聞いた
それでな
みんなものすごいショックを受けてさ
それから三組じゃその後もずっとミサキは生きてるっていうふりをし続けることに
したのさ
ちょっと不気味ね
結局そのふりは卒業まで続いてさ
卒業式の時は校長の計らいでミサキの席が用意されたらしい

The corresponding gold translation for this document is:

> Have you heard of Misaki? From the 9th grade, class 3.
> Was there anyone named that in class 3?
> It happened 26 years ago.
> She'd been popular ever since 7th grade.
> She was smart, pretty, and had a great personality.
> So she was loved by students and teachers alike.
> Yeah, there's at least one person like that in every grade.
> But shortly after she started 9th grade, Misaki died.
> How?
> I heard it was an accident.
> So everyone was really shocked.
> But from that day on, class 3 continued to behave like Misaki was still alive.
> That's kind of creepy.
> They kept the act up right to graduation.
> The principal even arranged to have Misaki's seat included in their graduation ceremony.

And the original NMT output of this document (sentences numbered for later reference) is:

> 1. Do you know about Misaki or Misaki for three years?
> 2. Such a person in the three groups
> 3. The story is twenty-six years ago.
> 4. It has been popular since the first year.
> 5. It seems that the head, face and personality were good.
> 6. So it's good for both students and teachers.
> 7. That's the kind of person who is in school.
> 8. But as soon as I got up for three years, that Misaki died.

```
9. Why?
10. I heard the accident.
11. That's why everyone is shocked.
12. After that, the three groups decided to continue pretending that
Misaki would be alive even after that.
13. It's a bit eerie.
14. After all, the pretend continues until graduation.
15. At the graduation ceremony, it seems that the head of the school
prepared a Misaki seat.
```

4.7.1 Approach Requests Comparison

Baseline Feedback-requester (i.e. Full Policy and Efficient Policy)

The baseline feedback-requester model solicited feedback on 12 of the 15 sentences, specifically, all sentences **except** for #5 ("It seems that the head, face and personality were good.") and #9 ("Why?").

Baseline Feedback-requester after Online Learning

The baseline feedback-requester model after online learning updates for the 1249 documents, solicited feedback on 3 of the 15 sentences, specifically, sentences #1 ("Do you know about Misaki or Misaki for three years?"), #9.("Why?"), and #13 ("It's a bit eerie.").

Entropy-based AL

The entropy-based AL approach model (after online learning updates for the 1249 documents), did not solicit feedback on any of the sentences.

Learned Sampling AL

The learned sampling AL approach model (after online learning updates for the 1249 documents), did not solicit feedback on any of the sentences.

4.7.2 Active Learning Approach Selected Sentence Comparison

To observe sentences selected for feedback by the active learning strategies, we examine 100 random sentences[4] and compare the sentences receiving feedback requests.

Entropy-based AL

The entropy-based AL approach model (after online learning updates for the 1249 documents), solicited feedback on six of the 100 random sentences, listed following:

1.救援を待つか 俺一人で やつを...
2. 行け!
3. 誰の仕業か 知ってるわ
4. 俺 そのものだ
5. 落ち付いてくだ...
6. そう そうよね

The corresponding gold translations for these sentences are:

1. wait for the reinforcement. ill take that bastard on
2. go now!
3. i know who did it.
4. you complete me.
5. eh, relax.
6. oh, no!

And the original NMT output sentences are:

1. `Wait for the relief? I'm alone.`
2. `Let's go!`
3. `Who's the Work Know`
4. `I'm the one.`

[4]See the full sentence list and their reference translations in Appendix .2

5. I'm not sure what to do.
6. That's right.

Learned Sampling AL

The learned sampling AL approach model (after online learning updates for the 1249 documents), solicited feedback on two of the 100 random sentences. Those two sentences were the following:

1. 誰の仕業か 知ってるわ
2. 落ち付いてくだ...

The corresponding gold translations for these sentences are:

1. i know who did it.
2. eh, relax.

And the original NMT output for the sentences:

1. Who's the Work Know
2. I'm not sure what to do.

We can see that the two sentences requested for feedback by the learned sampling AL approach were also among the six sentences requested by the entropy-based approach. Among the other four sentences, it is not apparent from the provided gold reference translations, but the original NMT translations for the 4th requested sentence ("I'm the one.") and the 6th requested sentence ("That's right.") are actually perfect valid translations in most contexts.

Chapter 5

DISCUSSION

From merely reading the randomly-selected subtitles example in the previous section, the online learning and active learning approaches for the feedback requester model seem to be performing quite poorly by failing to solicit feedback on many badly translated sentences. This is observed by reading samples from the NMT output that do not receive feedback requests, as well as reflected in the BLEU scores of the original NMT translated documents, which range from 6 to 12 (indicating low quality).[1] The baseline model ends up with a much more satisfactory output, but only by asking about the majority of the NMT output sentences. Looking at the output of the original NMT output for the first example of a document section reveals that none of the sentences are actually translated in a manner requiring no feedback to be natural for their context in the document. While some of the NMT output sentences are actually perfectly valid sentence translations out of context (i.e. "Why?" as the translation of "どうして" for line #9 in the first example), they are clearly incorrect in the context of the document. With original NMT output that is as poor as this example, the feedback-requester likely does not provide any user-experience advantage where even a less careful translator will likely need to address almost every sentence. A more well-suited example for the intended use case of this human-assisted NMT system would need to have at least some sentences that are well enough translated that the user gains time by skipping some sentences. In other words, starting with a better pre-trained NMT model likely would yield more promising results for the system. Additionally, having more precisely trained weighting hyper-parameters for the loss functions may also result in more promising translation quality for human effort trade offs. However, the current experiments still do

[1] https://github.com/bolducp/human-assisted-nmt/blob/master/hnmt/feedback_requester/experiments/plots

show the potential of the system to identify highly problematic translations, and present some intriguing patterns.

5.1 Results Analysis

5.1.1 Online Learning and Active Learning Strategies

One observation that emerges from the graphs showing the percentage of sentences that received feedback requests (Figures 4.1–4.4) is how adding the online learning element expands the breadth of range of percentage of sentences receiving feedback requests. This might be particularly expected in the case of adding in the active learning strategies, which widens the model's optimization to include finding examples that not only need feedback but will also teach the model itself. The fact that we can see that the range of percentage of sentences receiving feedback requests increases not only for the active learning approaches, but also for the online learning experiments, suggests that this may have something to do with the loss function we are using for updating the model in response to each simulated full-document interaction with the feedback-requester. This additional training, in particular for the learned sampling active learning model, was quite volatile, and it was difficult to choose hyper-parameters that didn't result in the model making nearly identical all near 1 positive judgments or all near 0 negative judgments. We suspect that further exploration of hyper-parameter lambda values could yield experimental runs for the 0.75 and 0.9 threshold with varying predictions. However, as we wanted to keep the hyper-parameters constant across the different threshold experiments for this current analysis, we were unable to find hyper-parameters lambda values that worked for the learned sampling AL strategy at all thresholds.

5.1.2 Impact of Threshold on Online Learning Updates

Somewhat surprisingly, adjusting the threshold values after the initial training for the baseline feedback-requester model changed the outcome of feedback requests very little, because

the majority of the model outputs were very close to either 0 or 1. Playing around with even higher thresholds, in the range of 0.98 or 0.99, may have resulted in more variance. For online learning and the active learning strategies, which updated according to a new loss function, these values did seem to cause a difference both in the sentence feedback requests and in how the model adapted. Interesting, in all of the cases where the number of requests did not start and stay at 0, the percentage of sentences requested eventually regulated to the same range (between 50% and 75% of sentences for a given document). This suggests perhaps that something in the loss function resulted in eventually optimizing for that range. If, by chance, the ideal percentage of sentences to be solicited feedback for falls within this range, then we suspect these versions of the feedback-requester may do well on identifying the worst sentences. However, as the ideal usage would be for the feedback-requester to adapt to the quality of the specific document, these results are a bit of a disappointment.

5.2 Future Work

5.2.1 Learned Sampling Active Learning Adjustments

Aside from redoing some of the experiments using a more accurate pre-trained NMT model and/or documents from a corpus in a similar domain to those used to initially train the feedback-requester model, we have ideas about another potential change to make for obtaining better active learning outcomes for the learned sampling approach. In this work, we use the magnitude of the gradients of the feedback-model's weights with respect to the user objective as a measure for how much the model has "learned" from a given example, as described in Section 3.3.2. However these gradients may not always reliably measure how much the model is changing. In particular, it is possible that after an update step in which the gradients are large, the model will make very similar predictions as it did before, due to the fact that neural networks are over-parameterized. In future work, we might be interested in targeting the amount that the feedback-requester "learns", as characterized by how much its predictions change, more directly. To do so, we would create a large and varied unlabeled

data set V (specifically, a set of source sentences and their resulting NMT output translations that we would feed into the feedback-requester model) that we hold constant, and we also have a sentence with human feedback D. We denote the feedback-requester model's predictions for the user objective on this set as M(V). After the feedback-requester is updated by being trained against D, we denote the model as M_D. Then we represent the amount that the feedback-requester model learned about the user objective from the training data D as $|M(V) - M_D(V)|$, where |x| is some vector norm of x. Intuitively, if the model's predictions change dramatically, the training data was informative, and if they do not change at all, the model had nothing new to learn from the training data.

Under this approach, the feedback-requester must be trained to predict the system objective, represented as $|M(V) - M_D(V)|$, from only the unlabeled portion of D, i.e. the portion that is visible before human feedback is obtained, which we denote D_U.

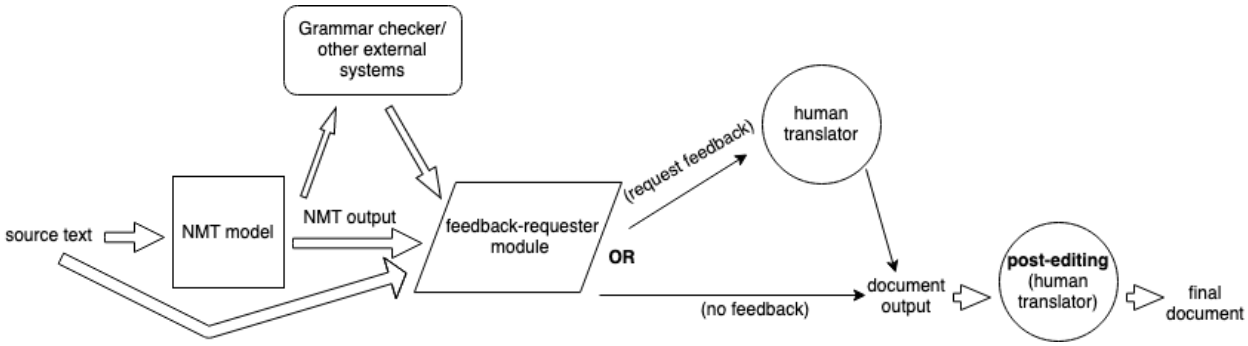
Then letting $S(D_U)$ represent the prediction of $|M(V) - M_D(V)|$ (i.e. the model's prediction of how much it will "learn" from observing feedback on D), we train against the squared error loss $(S(D_U) - |M(V) - M_D(V)|)^2$. During the initial training period, we can evaluate this loss for every sentence, as we will have "human feedback" in the form of gold-standard translations for every sentence. During the "online" training following deployment, we can only evaluate this loss when human feedback is requested. We note that it might not be of use to begin training for the system objective until the user objective predictions are relatively stable, because we are trying to predict a complex function of the user objective model. Depending on the computational complexity of evaluating $|M(V) - M_D(V)|$, it may also be necessary to subsample, rather than computing the loss for every sentence.

At evaluation time, we use the prediction of $|M(V) - M_D(V)|$ (i.e. our proxy for how much the user objective will benefit from the training data that results from a feedback request) as δ_2 in Equation (3.3).

5.2.2 Additional Feedback-Requester Model Inputs

We're interested in adapting the current system to provide additional inputs. For example, we suspect the feedback-requester model would be able to make more informed decisions if it had access to the output of a grammar checker run on the NMT translation. There could be other additional external systems that may provide useful inputs for the feedback-requester model, such as a machine-translation quality-estimation system, as well. Such additions could be added to the system architecture as demonstrated in Figure 5.1.

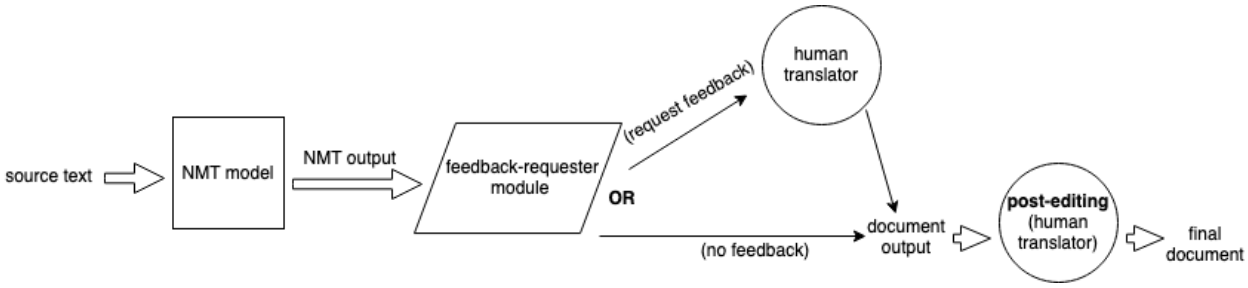
Figure 5.1: System Architecture With External System Inputs



5.2.3 Analysis of the Impact of Including Source Text

We are also interested in exploring whether including the source text as an input into the feedback-requester makes a difference in the model's performance. To gauge this, we could conduct experiments where we compare training and evaluating two versions of the system in the same manner, except that one will receive the source text as an input for the feedback-requester model (per the normal configuration) and the other will not. This adjustment to the system architecture is demonstrated in Figure 5.2.

Figure 5.2: System Architecture Without Source Text Inputs



5.2.4 Transfer Between Source Languages

As an additional inquiry for the variant of our system described in 5.2 that does not take the source text as one of the inputs into the feedback-requester model, we would be interested to see how the feedback-requester trained on one language pair performs when evaluated with a different source language. To examine this, we could perform experiments using our trained feedback-requester model on a different source language and examine how the output of the (pre post-editing) document produced by the human-assisted NMT system compares with the plain NMT output at various threshold levels for requesting human translator impact.

5.3 Conclusion

Preliminary experimentation of the interactive command line tool on texts from a similar domain as the training corpus (the Kyoto Free Translation Task) were quite promising in prompting feedback on the more problematic sentences in the NMT output translation. For our experiments using a corpus from the very different domain of conversational dialogue in movie and television subtitles, our pre-trained NMT system produced much worse initial output (between 6 and 10 document-level BLEU scores) [2] and our various online learning and

[2] All of the original NMT output document BLEU scores and chrF scores can be found in the experiment plots at: https://github.com/bolducp/human-assisted-nmt/tree/master/hnmt/feedback_requester/experiments/plots

active learning adaptation attempts presented somewhat disappointing results. However, we believe that the initial baseline system as well as the results showing the trade-off between translation quality and human effort still represent a contribution to working towards efficiently using human input to improve NMT system output. Furthermore, we suspect that additional experiments using a more accurate pre-trained NMT model and/or testing on documents in a similar domain to those used to initially train the feedback-requester model may result in more promising outcomes for the online learning and active learning adaptations.

BIBLIOGRAPHY

[1] Philip Bachman, Alessandro Sordoni, and Adam Trischler. Learning algorithms for active learning. *CoRR*, abs/1708.00088, 2017.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[3] Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. Statistical approaches to computer-assisted translation. *Comput. Linguist.*, 35(1):3–28, March 2009.

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

[5] Jesús González-Rubio, Daniel Ortiz-Martínez, and Francisco Casacuberta. Active learning for interactive machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, page 245–254, USA, 2012. Association for Computational Linguistics.

[6] Spence Green, Sida Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christoper Manning. Human effort and machine learnability in computer aided translation. pages 1225–1236, 10 2014.

[7] Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy, July 2019. Association for Computational Linguistics.

[8] R. Knowles and Philipp Koehn. Neural interactive translation prediction. 2016.

[9] Rebecca Knowles, Marina Sanchez-Torron, and Philipp Koehn. A user study of neural interactive translation prediction. *Machine Translation*, May 2019.

[10] Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. Can neural machine translation be improved with user feedback? *CoRR*, abs/1804.05958, 2018.

[11] Julia Kreutzer and Stefan Riezler. Self-regulated interactive sequence-to-sequence learning. *CoRR*, abs/1907.05190, 2019.

[12] Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[14] Tsz Kin Lam, Julia Kreutzer, and Stefan Riezler. A reinforcement learning approach to interactive-predictive neural machine translation. *CoRR*, abs/1805.01553, 2018.

[15] Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. Interactive-predictive neural machine translation through reinforcement and imitation. *CoRR*, abs/1907.02326, 2019.

[16] Ming Liu, Wray Buntine, and Gholamreza Haffari. Learning to actively learn neural machine translation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 334–344, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[17] Nitika Mathur, Timothy Baldwin, and Trevor Cohn. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online, July 2020. Association for Computational Linguistics.

[18] Makoto Morishita, Jun Suzuki, and Masaaki Nagata. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3603–3609, Marseille, France, May 2020. European Language Resources Association.

[19] Graham Neubig. The Kyoto free translation task. http://www.phontron.com/kftt, 2011.

[20] Franz Josef Och, Richard Zens, and Hermann Ney. Efficient search for interactive statistical machine translation. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, April 2003. Association for Computational Linguistics.

[21] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 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, June 2019. Association for Computational Linguistics.

[22] Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, and Josef van Genabith. A neural network based approach to automatic post-editing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 281–286, Berlin, Germany, August 2016. Association for Computational Linguistics.

[23] Álvaro Peris and Francisco Casacuberta. Active learning for interactive neural machine translation of data streams. *CoRR*, abs/1807.11243, 2018.

[24] Álvaro Peris and Francisco Casacuberta. Active learning for interactive neural machine translation of data streams. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 151–160, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[25] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[26] Matt Post. A call for clarity in reporting BLEU scores. *CoRR*, abs/1804.08771, 2018.

[27] Reid Pryzant, Yongjoo Chung, Dan Jurafsky, and Denny Britz. Jesc: Japanese-english subtitle corpus, 2018.

[28] Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, page 203–206, USA, 2007. Association for Computational Linguistics.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[30] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation, 2016.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[32] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[33] Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. A survey of deep learning techniques for neural machine translation, 2020.

[34] Xiangkai Zeng, Sarthak Garg, Rajen Chatterjee, Udhyakumar Nallasamy, and Matthias Paulik. Empirical evaluation of active learning techniques for neural MT. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 84–93, Hong Kong, China, November 2019. Association for Computational Linguistics.

Appendices

.1 Separate Approach Graphs for ChrF Improvement Across KSMR Scores

.1.1 Threshold 0.25

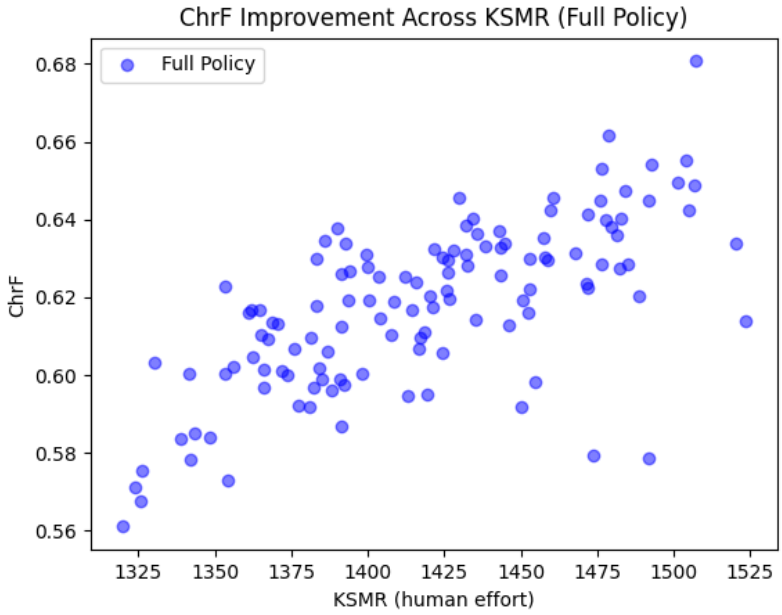


Figure 3: Threshold 0.25, Full Policy

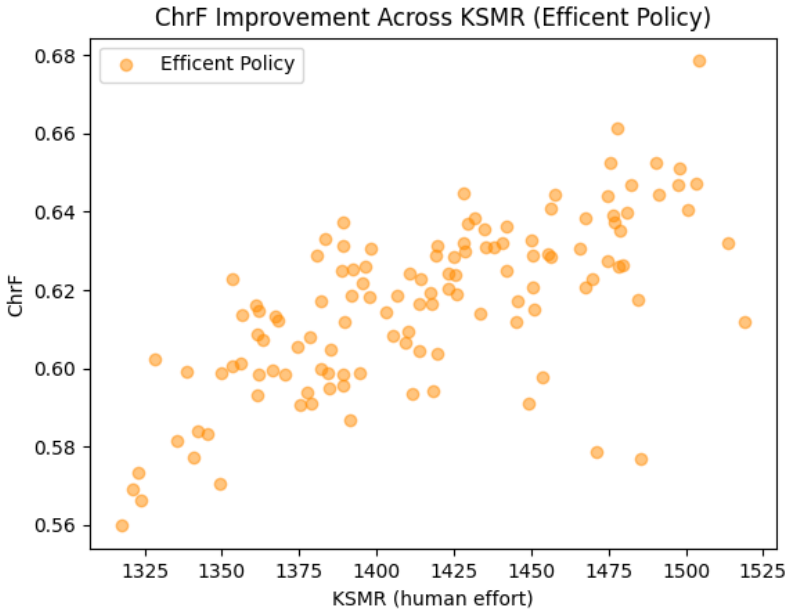


Figure 4: Threshold 0.25, Efficient Policy

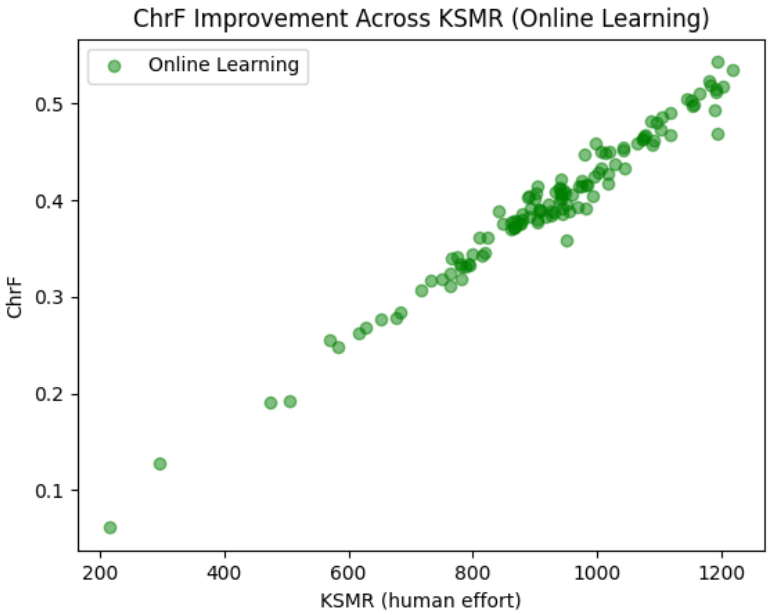


Figure 5: Threshold 0.25, Online Learning

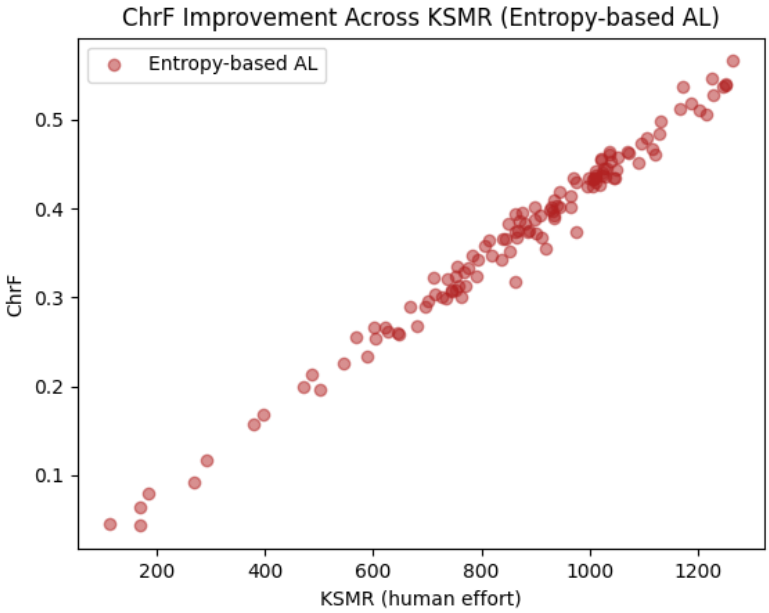


Figure 6: Threshold 0.25, Entropy-based AL

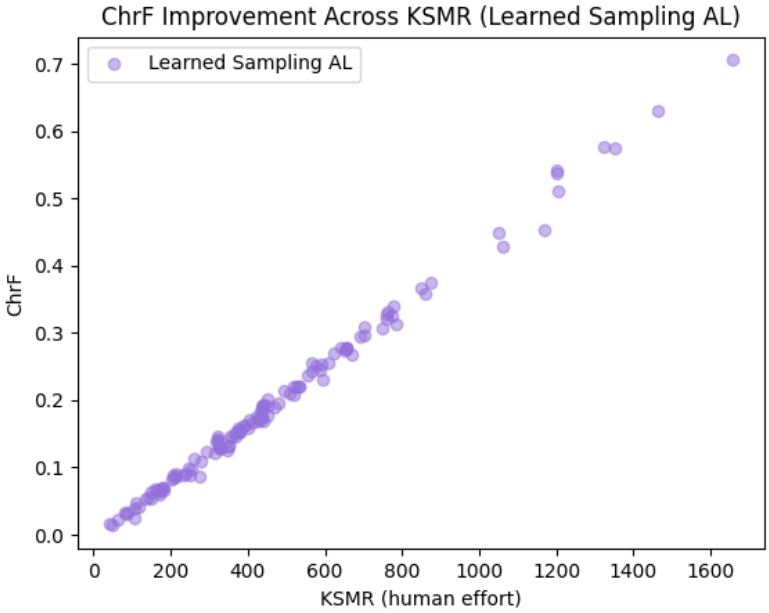


Figure 7: Threshold 0.5, Learned Sampling AL

.1.2 *Threshold 0.5*

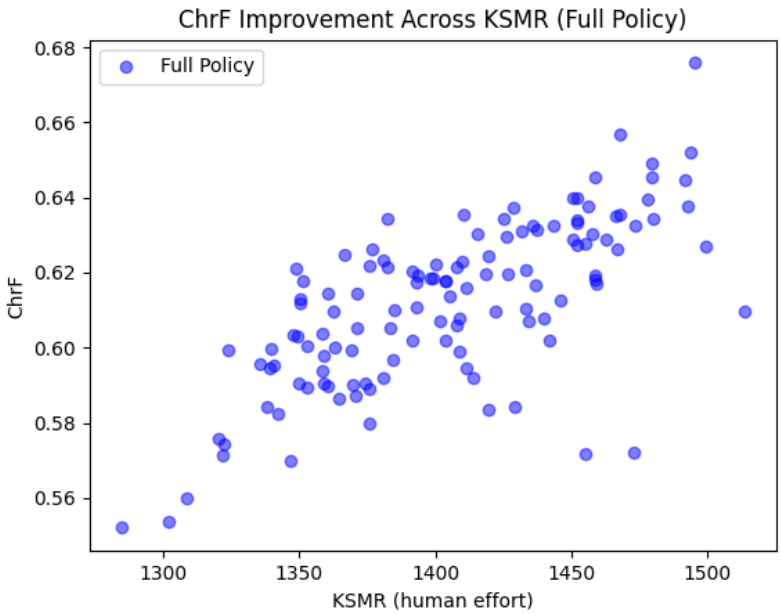


Figure 8: Threshold 0.5, Full Policy

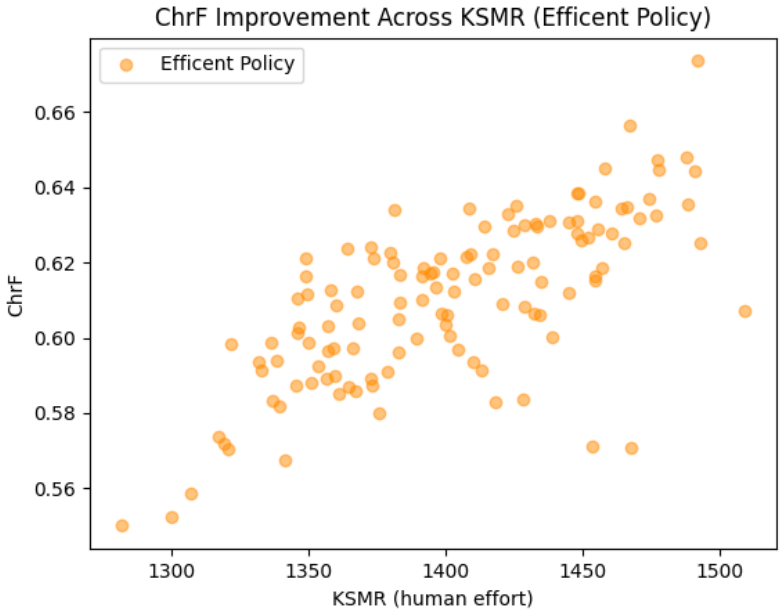


Figure 9: Threshold 0.5, Efficient Policy

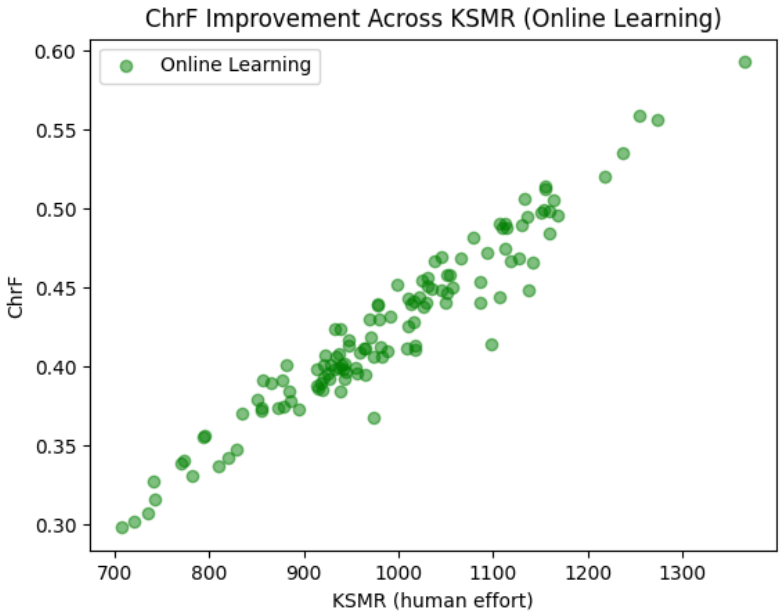


Figure 10: Threshold 0.5, Online Learning

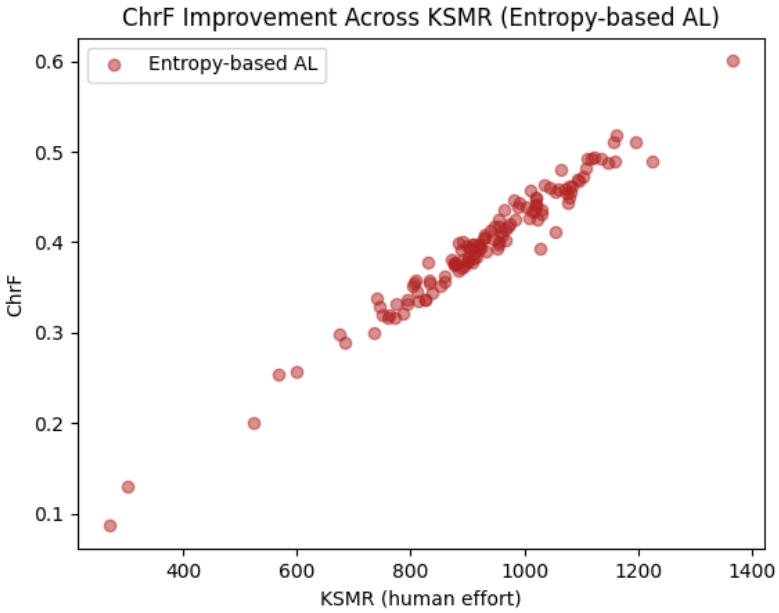


Figure 11: Threshold 0.5, Entropy-based AL

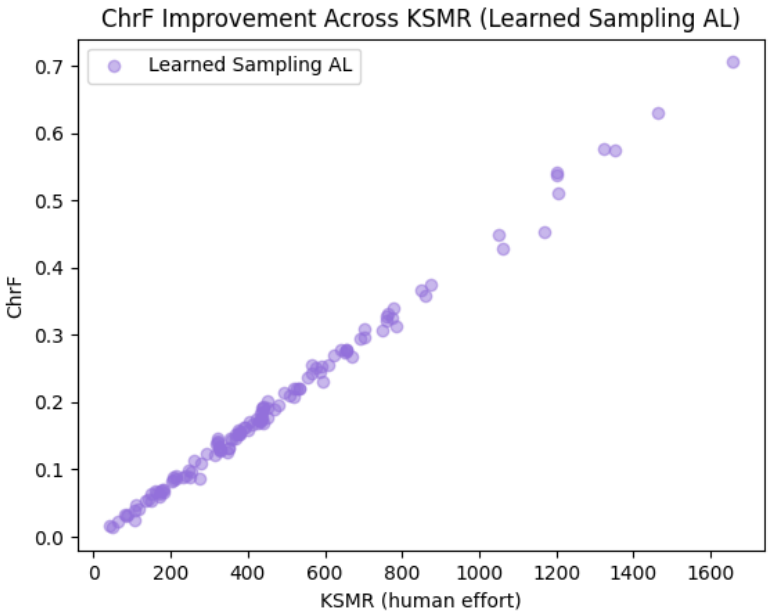


Figure 12: Threshold 0.5, Learned Sampling AL

.1.3 *Threshold 0.75*

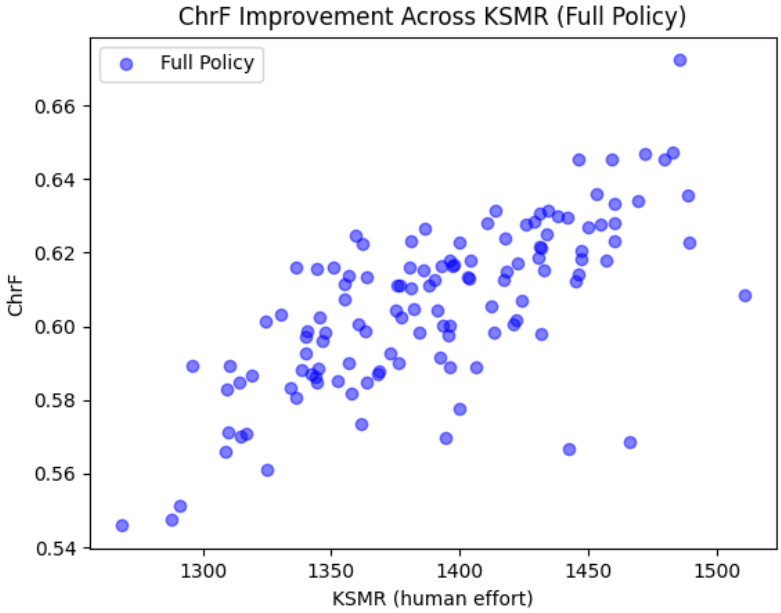


Figure 13: Threshold 0.75, Full Policy

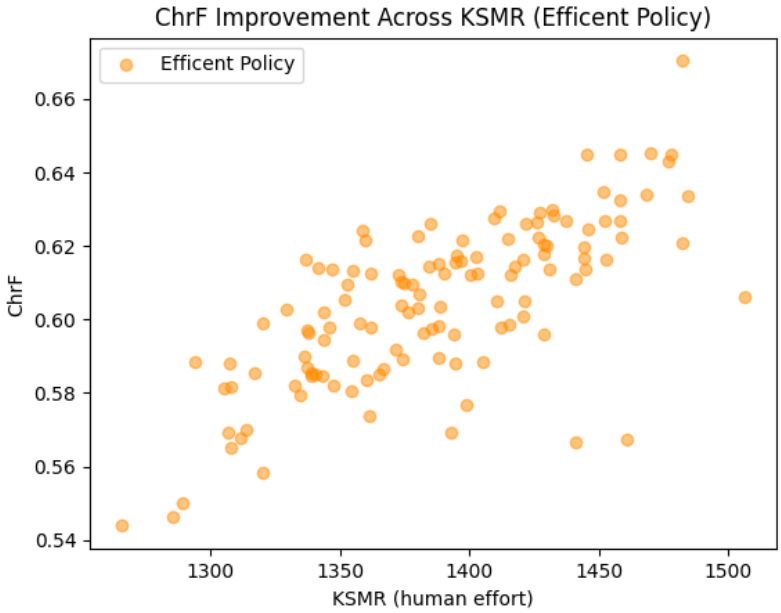


Figure 14: Threshold 0.75, Efficient Policy

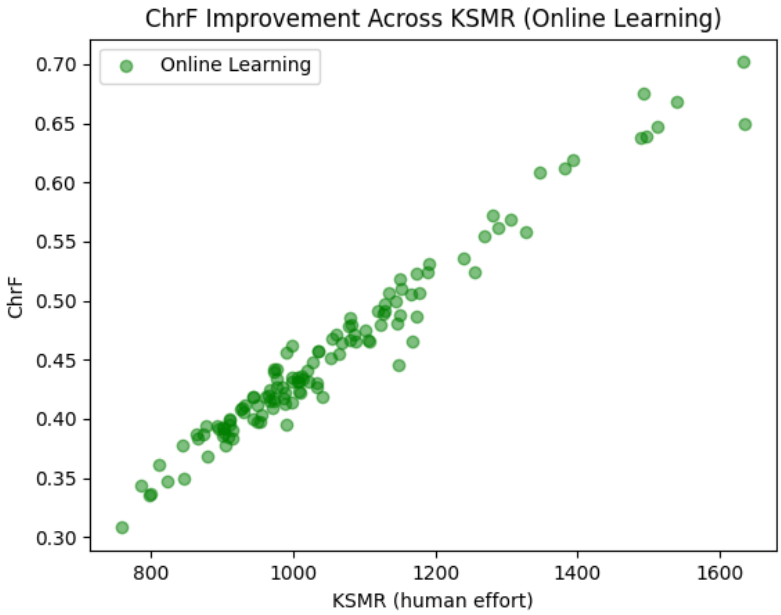


Figure 15: Threshold 0.75, Online Learning

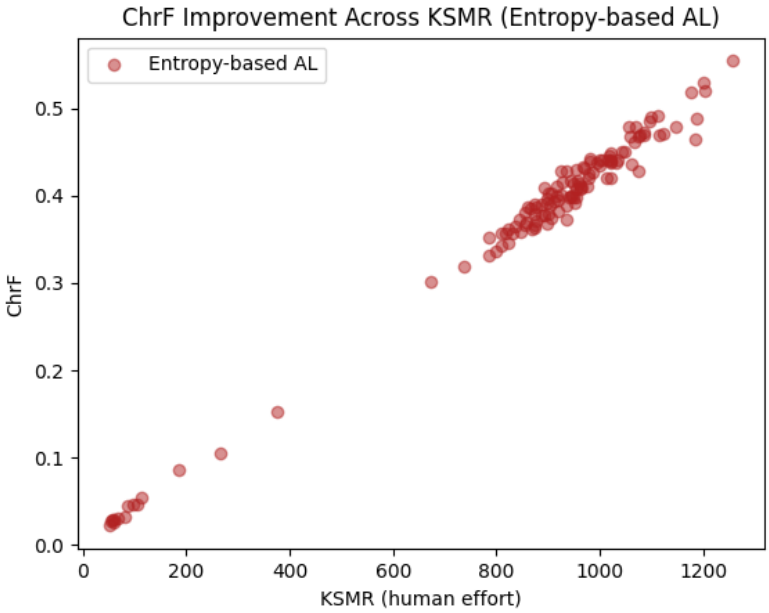


Figure 16: Threshold 0.75, Entropy-based AL

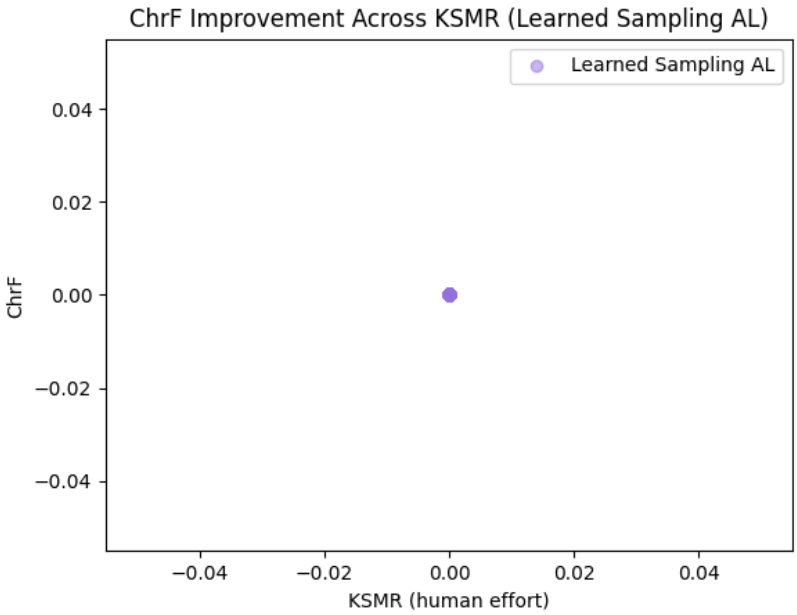


Figure 17: Threshold 0.75, Learned Sampling AL

.1.4 *Threshold 0.9*

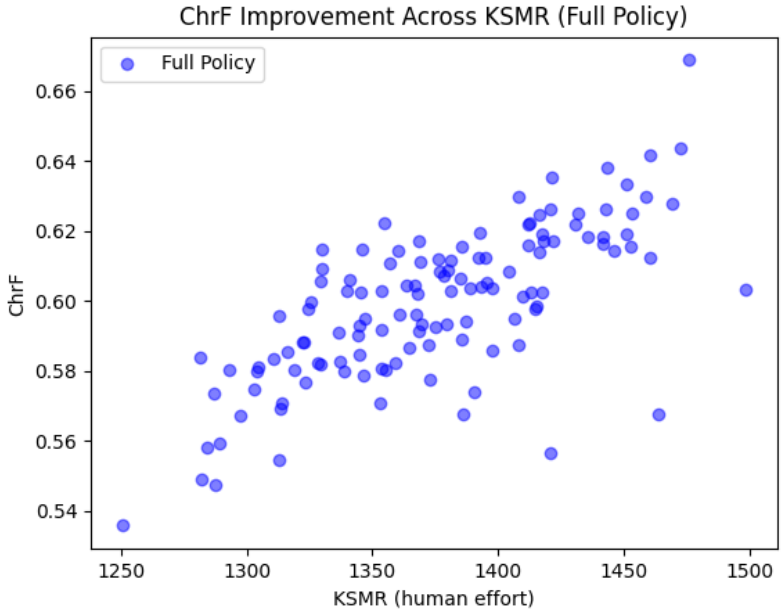


Figure 18: Threshold 0.9, Full Policy

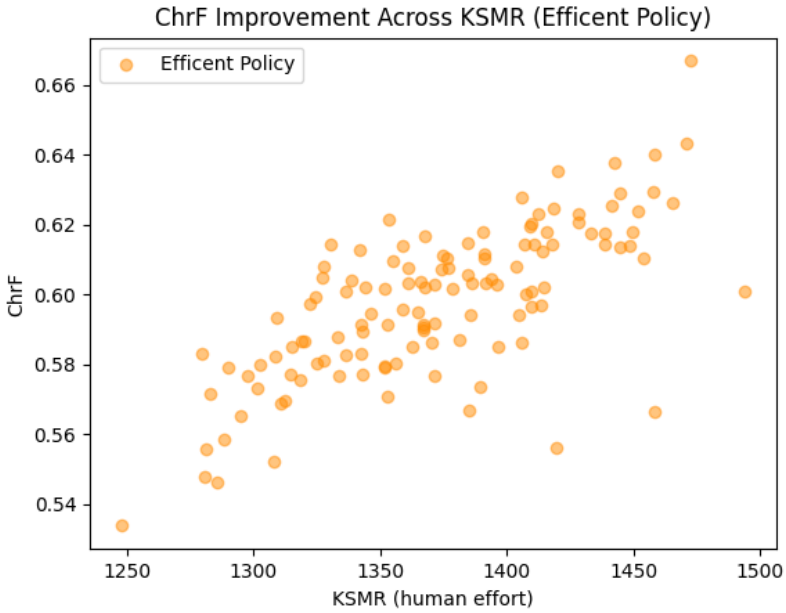


Figure 19: Threshold 0.9, Efficient Policy

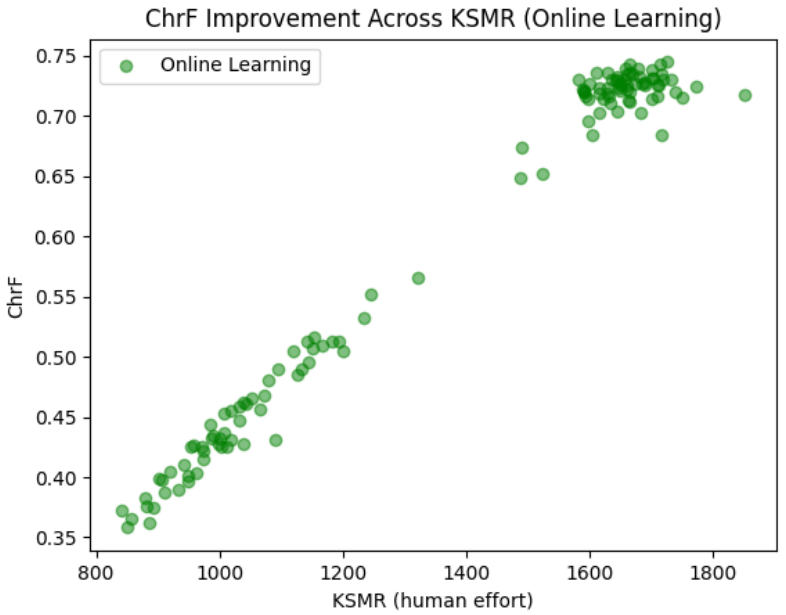


Figure 20: Threshold 0.9, Online Learning

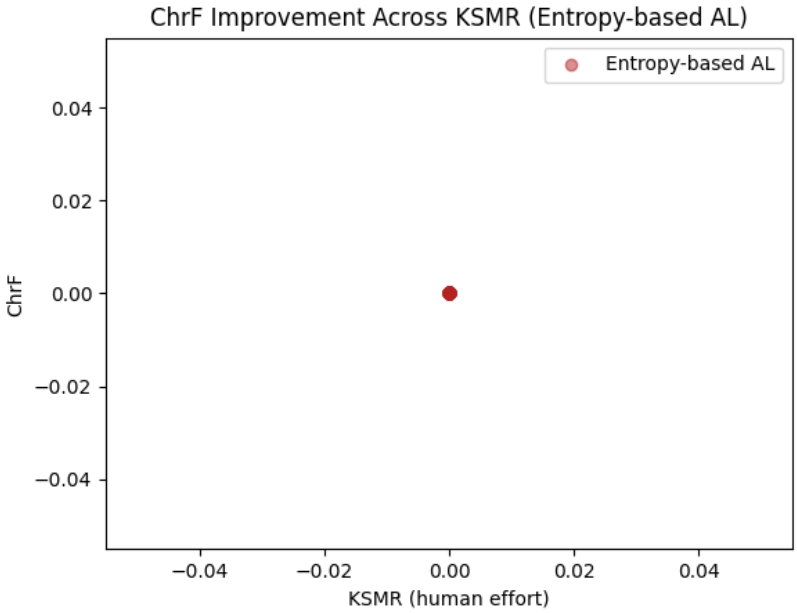


Figure 21: Threshold 0.9, Entropy-based AL

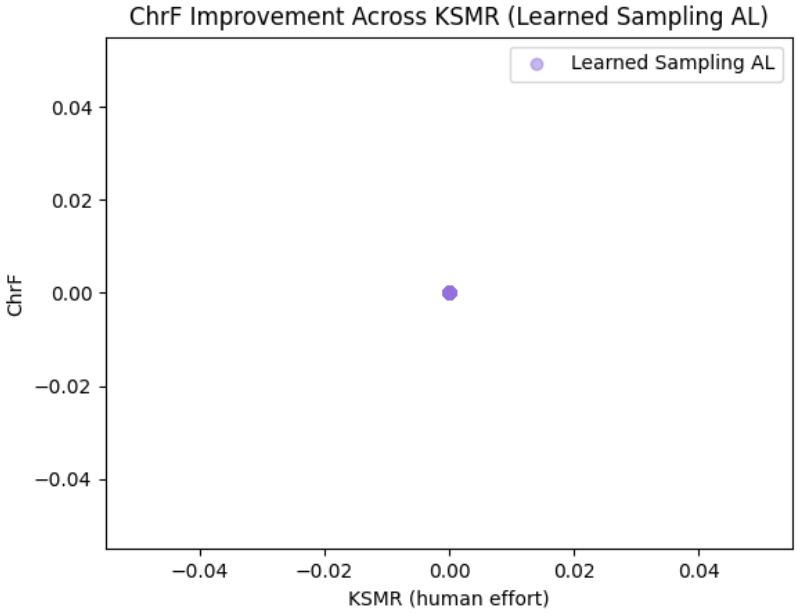


Figure 22: Threshold 0.9, Learned Sampling AL

.2 100 Japanese-English Subtitle Corpus Parallel Sentences

i can help. 助けてくれ。

why would you fix me? なぜ私を治療するの?

so despite your opposition to the v's, you don't condone this recent rash of violence against them? 「Vに反対する立場でありながら」 「Vに危害を加えようとする この所の動きは許せないと」

was the transistor in the internet. トランジスタとインターネットでした

you complete me. 俺 そのものだ

maybe she chose not to. 彼女は敢えてそうしたのかも

ok, just walk it in a little. オーケー 少し近づいて

two years ago, i was the highestrated lecturer 2年前 私はMITの起業家マスタープログラムで

all right, i will. もちろん

was that your fingers would get the screen dirty 指は画面を汚してしまうからだめだというものでした

it's nick's. ニックのよ

is he in trouble? あの子が問題を起こしたのかい?

i know who did it. 誰の仕業か 知ってるわ

so here they are. こちらが その楽器です

that is not the point. そこは重要ではない

she's much more of a winter. hmm. see here... wireless connector. 彼女は もっと白いよ これだ

when those two kiss? あの子たちがキスしたら

my, another turn of events. 予想外の展開ね

who the hell are you, man? おまえ 誰なんだよ?

occasional altruistic tasks throughout the day 人助けの仕事をやり遂げると

that's when i want to go to him. 彼のとこへ行きたい。

but this shows you that the government of uganda と言う事は、ウガンダ政府は

absolutely wonderful things とても素晴らしいことも
how you holding up? どうだった?
shall we eat meat for dinner today? 今夜は肉にしようか?
that... あの...
she got me started in this massage parlor. 私を 今のマッサージ・パーラーで 働かせ始めたのも そいつよ
i'll say. そうです
go now! 行け!
to graduate in mathematics in the central african republic. これはリディアです中央アフリカ共和国で 数学科を卒業した最初の女性です
i used to play this game in my head. いつも頭の中でこう考えるの
i have always been with you. いつも一緒よ。
there is no such episode. これは そういう仕様なんじゃ ボケ!
is made possible by attachment to an ipod 通勤には iPodが欠かせません
do you belong to any groups that wish to harm the united states? 「アメリカに害をもたらす組織に属していたか?」
you two are perfect for each other! もう 2時 回ってんだぞ...。
dragon palace. ドラゴンパレス!
and so it's not going to break very easily ですからそんなに簡単には壊れません
what are you saying? 君は?
you didn't tell us you were in a play! さ、忙しくて、言うのは、忘れちゃったかも
to give me breaks in the middle to kind of recuperate from the pain. 休憩を入れてもよかったようです
bless him with steel. 鋼の祝福を
niki, is that other people or is that you? ニキ 他の人じゃない キミのことだろ?
your silver wedding after 25 years. だって結婚記念日じゃないですか。 25年目の銀婚式ですよ。
the fire and wind masks are merging! 《火遁と 風遁の面が...》
wait for the reinforcement. ill take that bastard on 救援を待つか 俺一人で やつを...
by entering the snake nest. ああ ! パンがあった!

i wonder what could have caused it. 是因為有簽約的事嗎

it's all right. you two go on ahead now. この人間の始末は私がする.

what kind of trouble? トラブルって?

in reality, take a guess. 実際にはどうでしょう

so sorry! すいません わざわざ

i was kind of expecting a restaurant. レストランかと思ってた

i had encom. i couldn't be in here all the time. 会社にはまだ言えなかった

you're gonna need it. だが明日は

here are three questions こちらは僕の本から

we'll just take what he wrote down about you and leave. 釈放したって書類を 書けばいい

i no longer need it. 就再也看不到未来

he wasn't coming home when his wife was killed. 妻が殺された時 彼は家に帰ってない

you catch a prion disease by ingesting infected tissue. 感染した組織を摂取して 感染する

wyck is dead. he's dead. ウィックが死んだ 死んじゃった

yes, well, what else is there to do when life turns on you and you've retreated into some small room? そうだ 人生に破れ・・ 部屋で一人きりになって 他にすることがあるかね?

or a shifter. Mと入れてな

you know, if you're my partner, you should at least think about my condition a little bit. 相方のコンディションも少しは考えてくれ

because you just didn't have it in you. お前に力なんかない!

what? grab your place to belong. 最後はお前が跳べ お前の居場所 手に入れろ

and now to just run away? 逃げるために?

here, we've got a protocell to extract carbon dioxide こちらは、大気から二酸化炭素を取り出し

the misaka network? 御坂御坂着息自己的无力道

no, sir! good. you are to work as a team.. よろしい お前らは チームとして行動せねばならない

shaw's cover identity was burned eight days after one of your isa operatives, guy named grice, saw her in the field. ショーの偽装の身元が焼かれた 8日後 あんたのISAのスパイの1人 グライスという名の男 現場で彼女に会ってる

etta how much i loved you ... エッタ どれほど愛していたか...

if you connect them together 英単語も 何かと関連付け 鎖のように連結させると

here. trust me. ここは正しいな

what is wrong with you? あなた どうかしたの?

fine, fine. いや なんでもない

jessica arndt. ジェシカ・アーント

how long has it been since i've seen you? もう何年会ってない?

i feel like it's a year! 1年に感じるよ!

but now you're back. and it's all yours. ちょっと運転してみるかい?

what happened? 何があったの?

but if during one of those days しかし もし皆さんの体温がその内6時間の間

what did you say? oguri! いつだ 何を言ったんだ 小栗 小栗。

of the light that we see. 微かなパターンを残しました

tony wouldn't have much in common with mr. caspere. トニーはカスパーと 共通点がないと思う

that's why... um... だから あれは挨拶みたいなもんなの

and i like this one, snow has returned to kilimanjaro. これいいよね「キリマンジャロに冠雪戻る」

i only heard about such a thing existing from the children. そんなもの あるなんて聞いたのも子供たちからなのよ。

well, william's last wish was that your father take over this company, my personal feelings are not at issue. そうね、ウイリアムの最後の願いが あなたのお父さんに会社を譲ることだったならば そこに私の気持ちは関係ないわ

it was all white. 真っ白だったわ

what is so important about this book? どうしてあの本がそんなに大事なの?

ancient fairy dragon! 降誕せよ! エンシェント・フェアリー・ドラゴン!

everyone around you dies. 君の周りのみんなが死ぬ

branded it myself. 私の名だよ

oh, no! そう そうよね

perhaps i can find a more sensitive one. もっとデリケートな ものがあるわ。

don't you like surprises? 驚くのは嫌いか?

when did you wake up? 起きた時に 起こすべきじゃ?