Onception: Active Learning with Expert Advice for Real World Machine Translation

Vânia Mendonça*
INESC-ID, Instituto Superior Técnico
vania.mendonca@tecnico.ulisboa.pt

Ricardo Rei
INESC-ID, Instituto Superior Técnico,
Unbabel AI
ricardo.rei@unbabel.com

Luísa Coheur
INESC-ID, Instituto Superior Técnico
luisa.coheur@tecnico.ulisboa.pt

Alberto Sardinha
PUC-Rio
INESC-ID, Instituto Superior Técnico
jose.alberto.sardinha@tecnico.ulisboa.pt

Active learning can play an important role in low-resource settings (i.e., where annotated data is scarce), by selecting which instances may be more worthy to annotate. Most active learning approaches for Machine Translation assume the existence of a pool of sentences in a source language, and rely on human annotators to provide translations or post-edits, which can still be costly. In this article, we apply active learning to a real-world human-in-the-loop scenario in which we assume that: (1) the source sentences may not be readily available, but instead arrive in a stream; (2) the automatic translations receive feedback in the form of a rating, instead of a correct/edited translation, since the human-in-the-loop might be a user looking for a translation, but not be able to provide one. To tackle the challenge of deciding whether each incoming pair source–translations is worthy to query for human feedback, we resort to a number of stream-based active learning query strategies. Moreover, because we do not know in advance which query strategy will be the most adequate for a certain language pair and set of Machine Translation models, we propose to dynamically combine multiple strategies using prediction

* Corresponding author.

Action Editor: Rico Sennrich. Submission received: 12 March 2022; revised version received: 7 November 2022; accepted for publication: 25 November 2022.

https://doi.org/10.1162/coli_a_00473

with expert advice. Our experiments on different language pairs and feedback settings show that using active learning allows us to converge on the best Machine Translation systems with fewer human interactions. Furthermore, combining multiple strategies using prediction with expert advice outperforms several individual active learning strategies with even fewer interactions, particularly in partial feedback settings.

1. Introduction

The state of the art on most NLP tasks has been dominated by supervised neural approaches, and Machine Translation (MT) is no exception (Barrault et al. 2020). The impressive results obtained by neural models became possible due to the growing amount of annotated data available; however, this growth is not observed for most languages, since annotation (or, in the case of MT, translation) is a time-consuming and expensive process. This motivates the use of *low-resource* learning methods (i.e., methods that can make the most of minimal annotation effort). Active learning is one such method, as it allows us to choose which instances should be annotated, based on some criterion that measures their informativeness (Cohn, Atlas, and Ladner 1994; Settles 2010). Active learning has been extensively applied to MT (e.g., Haffari, Roy, and Sarkar 2009; Ambati, Vogel, and Carbonell 2011; González-Rubio and Casacuberta 2014; Zeng et al. 2019), mostly in a pool-based setting (i.e., when a pool of source segments is available and one has to choose which segments should be translated by a human annotator).

Existing active learning approaches for MT assume a scenario where a human annotator either provides the translations for (readily available) segments in the source language or post-edits the translations outputted by an MT model (in the case of Interactive MT). To the best of our knowledge, no active learning work has explored a real world scenario in which the end users of MT systems (e.g., Web MT services) not only provide the source segments to be translated, but may themselves be a source of feedback. In this case, the human would be looking for a translation, rather than providing it, but they might still be able to provide feedback (e.g., in the form of a rating) on whether the automatic translation makes sense in the target language. Such ratings could be useful for fine-tuning an MT model (or an ensemble of models) and require considerably less effort than providing a translation from scratch or post-editing a translation. Given these assumptions, our hypothesis is that we can use active learning to go a step further and reduce the need for human intervention.

Thus, we build on our previous work (Mendonça et al. 2021), in which we leveraged human ratings to learn the weights of an ensemble of multiple arbitrary MT models in an online fashion, in order to *dynamically* select the best MT models more often and continually improve the ensemble's performance, as new source segments arrive, for the language pairs in hand. We extend this proposal by incorporating stream-based active learning strategies that decide, for each pair source–translations, whether it is worthy to query the user for a rating, thus reducing the effort needed to improve the performance of the MT ensemble.

Moreover, because we do not know in advance which strategies will be the most effective for a certain language pair and ensemble of models, we *dynamically* combine multiple active learning strategies using the online framework of prediction with expert advice (drawing inspiration from another previous work of ours that focused on sequence labeling tasks [Mendonça et al. 2020]). Thus, we introduce a second layer of online learning[1] that learns the strategies' weights based on the performance of the online MT ensemble at each user interaction, so that the best performing strategies will be used more often in the next interactions.

We thus address the following research questions:

RQ1 Can an active learning approach converge to the best systems with fewer human interactions?

RQ2 Does it pay off to dynamically combine multiple query strategies using expert advice (rather than using a single strategy)?

Our contribution is four-fold:

1. A stream-based active learning solution that makes the most of an ensemble of pre-trained models by taking advantage of reduced human effort to dynamically adapt the ensemble to any language pair in hand;[2]
2. A set of experiments in a much lower-resource setting than most existing related literature, as we use datasets containing only as many as 1,000 to 2,000 source segments;
3. A set of stream-based query strategies that are model-independent, and do not require any additional pre-training data (unlike recent active learning approaches to MT [Liu, Buntine, and Haffari 2018b; Finkelstein 2020]);
4. A novel application of prediction with expert advice and active learning to MT in which we dynamically combine multiple stream-based strategies (instead of committing to a single strategy a priori).

Our experiments on multiple Conference on Machine Translation (WMT) datasets (Barrault et al. 2019, 2020; Freitag et al. 2021a) show that using active learning indeed allows us to converge to the top rated MT systems for each language pair, without prior knowledge of them, while sparing human interactions. Moreover, since the best query strategy varies across language pairs and online learning settings, combining multiple active learning strategies using prediction with expert advice is generally a safer option than committing to a single strategy (especially in scarce and/or unreliable feedback settings), and often outperforms several individual strategies.

The remainder of this article is structured as follows: in Section 2, we provide some background on the frameworks of active learning and prediction with expert advice; in Section 3, we present related work regarding the use of online and active learning in MT; in Section 4, we provide the starting point of our work and define the problem we are tackling; in Section 5, we propose a set of stream-based active learning strategies to reduce human intervention while improving an MT ensemble; in Section 6, we propose the combination of multiple active learning strategies under the framework of prediction with expert advice; in Section 7, we present the details of our experiments, whose results are reported and discussed in Section 8; finally, in Section 9, we wrap up this article and present future work directions.

[1] Thus *onception*, in the sense of "dream within a dream" used in the movie *Inception*.

[2] The code for our experiments can be found at `https://github.com/vania-mendonca/Onception`.

2. Background

In this section, we provide the necessary background on the learning frameworks applied in this work: active learning (Section 2.1) and prediction with expert advice (Section 2.2).

2.1 Active Learning

Active learning is a learning framework that aims at minimizing the amount of annotated data needed to train a supervised model by choosing which instances should be annotated, based on some criterion that measures their informativeness (**query strategy**) (Cohn, Atlas, and Ladner 1994; Settles 2010).

Active learning is most commonly used in a **pool-based** setting (see Algorithm 1): A model is trained on an initially small labeled set $\mathcal{L}$ and further retrained as $\mathcal{L}$ is iteratively augmented with an instance (or a batch of instances) selected from a pool of unlabeled instances $\mathcal{U}$. Each instance (or batch) u_t is selected according to a query strategy (line 4) and sent to be annotated by a human (line 5). The now annotated instance(s) u_t^* are removed from $\mathcal{U}$ and added to $\mathcal{L}$, and this process repeats itself until a budget of T instances is exhausted.

However, there might be situations in which one does not have access in advance to a pool of unlabeled instances, but rather accesses them progressively (consider, for example, an interactive system). In such situations, a **stream-based** setting can be followed instead (see Algorithm 2). In this setting, for each unlabeled instance (or batch of instances) u_t that occurs in the stream, the query strategy has to decide whether it is worthy to ask the human for its annotation (line 3).

Active learning query strategies can be based on information related to the learning models, on the characteristics of the data, or a mix of both. **Model-based** strategies rely on criteria such as the model's confidence on its prediction (e.g., Uncertainty Sampling [Lewis and Gale 1994]), the disagreement among the predictions of different models (e.g., Query-by-Committee [Seung, Opper, and Sompolinsky 1992]), or the change expected in the model after being retrained with the selected instances (e.g., Fisher Information [Settles and Craven 2008]). **Data-based** strategies, on the other hand, focus on the representativeness of the instances—either how similar an instance is to the unlabeled set $\mathcal{U}$ (Density), or how much it differs from the labeled set $\mathcal{L}$ (Diversity) (Fujii et al. 1998), with some strategies combining both criteria (e.g., Exploration-Guided Active Learning [Hu, Jane Delany, and Mac Namee 2010]). Finally, a popular strategy

Algorithm 1 Pool-based active learning

```
Input: model, labeled set $\mathcal{L}$, unlabeled set $\mathcal{U}$, budget $T$, batch size $B$
1: for $t \leftarrow 1$ to $T$ do
2:     $model.train(\mathcal{L})$
3:     $\hat{Y} \leftarrow model.predict(\mathcal{U})$
4:     $u_t \leftarrow selectInstances(\mathcal{U}, \hat{Y}, \mathcal{L}, B)$
5:     $u_t^* \leftarrow askAnnotation(u_t)$
6:     $\mathcal{L} \leftarrow \mathcal{L} \cup u_t^*$
7:     $\mathcal{U} \leftarrow \mathcal{U} - u_t$
8: end for
```

Algorithm 2 Stream-based active learning

```
Input: model, labeled set $\mathcal{L}$, stream of unlabeled instances $\mathcal{S}$
for each $u_t \in \mathcal{S}$ do
    $\hat{y}_t \leftarrow model.predict(u_t)$
    if $selectInstance?(u_t, \hat{y}_t)$ then
        $u_t^* \leftarrow askAnnotation(u_t)$
        $\mathcal{L} \leftarrow \mathcal{L} \cup u_t^*$
        $model.train(\mathcal{L})$
    end if
end for
```

that combines both the model uncertainty about its predictions and the instances' Density with respect to $\mathcal{U}$ is Information Density (Settles and Craven 2008). In the case of a stream-based setting, in which the decision on whether to select an instance cannot depend on the remaining unlabeled instances, query strategies may rely on a threshold to make its decision instead.

More recently, several studies have proposed to learn the query strategy from data (known as **Learning To Active Learn**), instead of using the previously mentioned heuristics (or in combination with them). These works framed the problem of selecting an instance to be annotated as a regression problem (Konyushkova, Raphael, and Fua 2017), as a policy that can be learned using reinforcement learning (Fang, Li, and Cohn 2017) or imitation learning (Liu, Buntine, and Haffari 2018a,b; Vu et al. 2019), and also as an adversarial problem (Deng et al. 2018).

2.2 Prediction with Expert Advice

Prediction with expert advice is a popular framework for online learning (i.e., a learning setting in which learning takes place continually, as the data arrives—in a sequential order—rather than through offline training (Hoi et al. 2021)). A problem of prediction with expert advice can be seen as an iterative game between a **forecaster** and the **environment**, in which the forecaster consults different sources (**experts**) in order to predict an outcome that will occur in the environment (Cesa-Bianchi and Lugosi 2006). Thus, at each iteration t, the forecaster consults the predictions $\hat{p}_{j,t}, j = 1 \ldots J$ made by a set of J weighted experts, in the decision space $\mathcal{D}$. Considering these predictions, the forecaster makes its own prediction, $\hat{p}_{f,t} \in \mathcal{D}$. At the same time, the environment reveals an outcome y_t in the decision space $\mathcal{Y}$ (which may not necessarily be the same as $\mathcal{D}$). Based on this outcome, a loss can be derived in order to update the experts' weights $\omega_1, \ldots, \omega_J$.

To learn the experts' weights, one can use an online learning algorithm, such as Exponentially Weighted Average Forecaster (EWAF) (Cesa-Bianchi and Lugosi 2006), an algorithm with well-established performance guarantees. In EWAF, at each iteration t, the forecaster randomly selects one of the experts' predictions following the probability distribution based on the experts' weights $\omega_{1,t-1} \ldots \omega_{J,t-1}$, as shown in Equation (1):

$$\hat{p}_{f,t} = \frac{\sum_{j=1}^{J} \omega_{j,t-1} \hat{p}_{j,t}}{\sum_{j=1}^{J} \omega_{j,t-1}} \tag{1}$$

where $\hat{p}_{j,t}$ is the vector containing the current probabilities for each possible decision in $\mathcal{D}$, according to expert j. Then, the forecaster and each of the experts receive a non-negative loss ($\ell_{f,t}$ and $\ell_{j,t}$, respectively), measuring how far their predictions were from guessing the outcome y_t that occurred in the environment. The weight $\omega_{j,t}$ of each expert $j = 1 \ldots J$ is updated according to the loss received by that expert, as follows:

$$\omega_{j,t} = \omega_{j,t-1} e^{-\eta \ell_{j,t}} \tag{2}$$

In the update rule above, if:

$$\eta = \sqrt{\frac{8 \log J}{T}} \tag{3}$$

it can be shown that the forecaster's **regret** for not following the best expert's advice is bounded, as follows:

$$\sum_{t=1}^{T} \ell_{f,t} - \min_{j=1,\ldots,J} \sum_{t=1}^{T} \ell_{j,t} \leq \sqrt{\frac{T}{2} \log J} \tag{4}$$

that is, that the forecaster quickly converges to the performance of the best expert after T iterations (Cesa-Bianchi and Lugosi 2006).

3. Related Work

In this section, we present an overview of existing studies that have applied the frameworks introduced in the previous section to MT, namely, active learning (Section 3.1), online learning (Section 3.2), and combinations of both (Section 3.3).

3.1 Active Learning for Machine Translation

The earliest proposals that attempted to select training data in a clever way in MT, despite not explicitly mentioning active learning, relied on criteria that could be seen as a query strategy (and ended up later inspiring active learning approaches), such as unseen n-gram frequency and sentence TF-IDF (Eck, Vogel, and Waibel 2005), or similarity/n-gram overlap to the test set (Hildebrand et al. 2005; Lü, Huang, and Liu 2007; Ittycheriah and Roukos 2007). Since then, a vast variety of active learning approaches to MT have been proposed, mostly in pool-based and batch-mode settings.[3]

Focusing on Statistical MT, query strategies proposed were mainly based on phrase tables (e.g., phrase frequency on $\mathcal{L}$ or $\mathcal{U}$ [Haffari, Roy, and Sarkar 2009], and phrase translation uncertainty/entropy [Ambati, Vogel, and Carbonell 2011]); language models (e.g., n-gram utility [Haffari, Roy, and Sarkar 2009]; perplexity [Mandal et al. 2008], and KL divergence [Ambati 2012]); alignment of the parallel data (Ambati, Vogel, and Carbonell 2011); the MT model's confidence on the translation (Haffari, Roy, and Sarkar 2009; González-Rubio, Ortiz-Martínez, and Casacuberta 2011); estimations of the translation error (Ananthakrishnan et al. 2010) or of the translation's quality (Logacheva

[3] For a more comprehensive analysis of data selection and active learning approaches in a pool-based setting for MT (up to 2015), see Eetemadi et al. (2015).

and Specia 2014); and the round-trip translation accuracy (i.e., the error between a source sentence and the source obtained by translating the translation) (Haffari, Roy, and Sarkar 2009). Query strategies commonly seen in other tasks have also been used in MT, namely, Query-by-Committee (Mandal et al. 2008), Information Density (González-Rubio and Casacuberta 2014), Diversity, Density and combinations of both (e.g., static sentence sorting [Eck 2008], Density-weighted Diversity [Ambati, Vogel, and Carbonell 2011], and *n*-gram coverage [González-Rubio, Ortiz-Martínez, and Casacuberta 2012; González-Rubio and Casacuberta 2014]). All of this work addressed a pool-based setting, except for that of González-Rubio et al. (González-Rubio, Ortiz-Martínez, and Casacuberta 2011, 2012; González-Rubio and Casacuberta 2014), who addressed a stream-based setting (although in the latter two studies, the authors used pool-based strategies to select the most useful instances from the current batch, rather than using stream-based strategies).

With the Neural MT takeover, strategies based on neural models' details and Diversity/Density strategies based on embedding similarity have been preferred over *n*-gram, alignment, and phrase-based strategies. Peris and Casacuberta (2018) extended the proposals of González-Rubio et al. for Statistical MT (González-Rubio, Ortiz-Martínez, and Casacuberta 2011, 2012; González-Rubio and Casacuberta 2014) by adding strategies based on coverage sampling (i.e., coverage of the attention weights over the source sentence, as a potential indicator of how good the alignment between the source and the translation is) and attention distraction (i.e., whether an MT model's attention is dispersed along the source sentence). Zhang, Xu, and Xiong (2018) used the decoder's probability for the translation (as a form of uncertainty) and proposed a Diversity strategy based on subword-level embeddings, using FASTTEXT (Bojanowski et al. 2017). Zeng et al. (2019) made an extensive comparison of several query strategies found in the MT literature on a Neural MT system based on the Transformer architecture (Vaswani et al. 2017). They also introduced two new strategies: a variant of round-trip translation based on the likelihood of the source sentence according to the reverse translated model, and a Diversity strategy based on the cosine distance applied to the source sentences' subword-level embeddings (following Zhang, Xu, and Xiong 2018), contextual embeddings (BERT [Devlin et al. 2019]), and paraphrastic embeddings (i.e., contextual embeddings fine-tuned in a paraphrase dataset [Wieting and Gimpel 2018]). These two strategies, along with Density-weighted Diversity (following Ambati, Vogel, and Carbonell 2011), outperformed the remaining strategies in use. Finally, Hazra et al. (2021) addressed the problem of redundancy across the batch of instances selected by a given query strategy at each iteration, proposing the removal of redundant sentences using a model-aware similarity approach, on top of either one of three model-based query strategies: Uncertainty Sampling, coverage sampling, and attention distraction (following Peris and Casacuberta [2018] in the latter two strategies).

Moreover, Learning To Active Learn strategies, which learn the query strategy from data, have also been applied to Neural MT. Liu, Buntine, and Haffari (2018b) viewed the query strategy as a policy network learned on a higher-resource language pair using imitation learning: The query strategy corresponds to a policy that learns by observing the inputs and output of an optimal policy; this optimal policy is trained on a higher-resource dataset, and its learning goal is to distinguish which instances are worthy of being annotated, based on how much each instance improves the performance of the task model on a development set, if added to the task model's labeled set. Their approach outperformed three heuristic strategies based on sentence length and uncertainty for most of the language pairs considered. Finkelstein (2020) learned a a stream-based query strategy that should decide, for each incoming sentence, whether its translation should receive human feedback. This query strategy was a neural network pre-trained on a parallel corpus, which learned when to ask for feedback based on the CHRF score (Popović 2015) between each sentence's automatic translation and its gold translation. The query strategy network was later retrained based on the human translator feedback (which can be given to a sentence when the query strategy asks for it, or by post-editing the final document), upon each document completion, based on the CHRF score between automatic and human made/post-edited translations. However, this proposal was not compared to other active learning query strategies.

Our proposal for applying active learning to MT differs from these works in several aspects. First, only a few proposals consider a stream-based setting (González-Rubio, Ortiz-Martínez, and Casacuberta 2012; González-Rubio and Casacuberta 2014; Peris and Casacuberta 2018), and most of them end up applying pool-based strategies to select a subset of instances to be annotated among the current batch, instead of using stream-based strategies. Second, the two exceptions that use stream-based strategies rely solely on strategies that are either model-based (González-Rubio, Ortiz-Martínez, and Casacuberta 2011; Finkelstein 2020) or require pre-training on additional data (Finkelstein 2020). However, because our starting point is an ensemble of arbitrary (and potentially black-box) MT models, and considering that we want to minimize the need for additional data, we constrain our solution to be model-independent and purely heuristic. Finally, our proposal differs from all of the works reviewed above in that we assume that the human-in-the-loop will not provide translations nor post-edits. Instead, we assume that the human will just rate the automatic translations, which is a form of feedback that requires less effort and is more plausible in practical applications such as Web MT services or MT shared tasks.

3.2 Online Learning for Machine Translation

There have been a number of online learning approaches applied to MT in the past, mainly in Interactive MT and/or post-editing MT systems. Most approaches aimed at learning the parameters or feature weights of an MT model (Mathur, Cettolo, and Federico 2013; Denkowski, Dyer, and Lavie 2014; Ortiz-Martínez 2016; Sokolov et al. 2016; Nguyen, Daumé III, and Boyd-Graber 2017; Lam, Kreutzer, and Riezler 2018), fine-tuning a pre-trained model for domain adaptation (Turchi et al. 2017; Karimova, Simianer, and Riezler 2018; Peris and Casacuberta 2019), or incorporating new parallel data from an unbounded stream (Levenberg, Callison-Burch, and Osborne 2010).

Most of these studies used human post-edited translations as a source of feedback, with the exceptions being the systems competing for WMT'17 shared task on online bandit learning for MT (Sokolov et al. 2017), as well as Lam, Kreutzer, and Riezler (2018), who used (simulated) quality judgments.

In contrast to these approaches, few studies used online learning to address the challenge of combining multiple MT models and dynamically finding the most appropriate ones for the language pair and set of models in hand. One such study is that of Naradowsky, Zhang, and Duh (2020), who dynamically selected the best MT system for a given MT task or domain using stochastic multi-armed bandits and contextual bandits. The bandit algorithms learned from feedback simulated using a sentence-level BLEU score (Papineni et al. 2002) between the selected automatic translation and a reference translation. Another one is a previous work of ours (Mendonça et al. 2021), in which we framed the problem of dynamically converging to the performance of the best individual MT system as a problem of prediction with expert advice (when feedback is available to the translations outputted by all the systems in the ensemble) and adversarial multi-armed bandits (Robbins 1952; Lai and Robbins 1985) (when feedback is only available for the final translation chosen by the ensemble). We simulated the human-in-the-loop by using actual human ratings obtained from an MT shared task (Barrault et al. 2019), when available in the data, and proposed different fallback strategies to cope with the lack of human ratings for some of the translations.

We thus build upon our previous work (Mendonça et al. 2021), since it is the only one that takes advantage of human quality ratings rather than translations or post-edits. This setting not only reduces the human effort involved in improving the MT ensemble, but it also should prove to be more suitable to represent real-world MT scenarios, such as Web translation systems or MT shared tasks, in which the human-in-the-loop is not expected to provide a translation.

3.3 Combining Online Learning and Active Learning

To the best of our knowledge, there are no studies that combine multiple query strategies using online learning frameworks for MT. The studies that relate the most to our proposal in MT are those of Haffari, Roy, and Sarkar (2009), who combined multiple pool-based strategies in a weighted ensemble, learned a priori on a development set, and Peris and Casacuberta (2018), who proposed an ensemble of stream-based query strategies, in which all strategies contributed equally (the strategies' votes are combined using a vote-entropy function [Dagan and Engelson 1995]). Both works differ from what we are proposing, since the weights of our ensemble of query strategies are learned dynamically based on human feedback, rather than fixed or learned a priori. Moreover, to the best of our knowledge, there are no online ensembles of stream-based query strategies in the literature either.

However, the combination of multiple pool-based active learning strategies using online learning frameworks has been previously proposed for other tasks, mainly binary and multi-class classification.

Baram, El-Yaniv, and Luz (2004) framed the problem of selecting a query strategy using multi-armed bandits (Auer et al. 1995) and multi-armed bandits with experts (Auer et al. 2002), which, unlike prediction with expert advice (recall Section 2.2), assume that only the expert/arm selected knows its loss. The variant based on experts took advantage of multiple strategies and performed in line with the best individual strategy for binary classification problems.

Osugi, Kun, and Scott (2005) also used multi-armed bandits with experts (Auer et al. 2002) to learn the best strategy for binary classification problems (but with a different loss function than Baram, El-Yaniv, and Luz [2004]), and outperformed the two individual query strategies considered, as well as Baram, El-Yaniv, and Luz (2004).

Hsu and Lin (2015) presented a modification of an algorithm for multi-armed bandits with experts so that it would compute a reward (loss function) for all the query strategies that selected the same instance as the chosen strategy (instead of only to that strategy). This approach performed in line with the best individual query strategies, and outperformed Baram, El-Yaniv, and Luz (2004).

Chu and Lin (2016) built on Hsu and Lin (2015), but applied contextual bandits instead, and also proposed to transfer the ensembles of query strategies learned in one dataset to other datasets. Their approach performed in line with the best two individual query strategies and outperformed Hsu and Lin (2015).

Pang et al. (2018) proposed a modification of multi-armed bandits with experts, to account for non-stationary loss functions (i.e., the best expert might vary over time), in binary and multi-class classification tasks. Their approach outperformed or performed in line with the best individual strategies and outperformed both Baram, El-Yaniv, and Luz (2004) and Hsu and Lin (2015) in non-stationary datasets.

Finally, in a previous study of ours (Mendonça et al. 2020), we combined multiple well-studied pool-based query strategies for two sequence labeling tasks (Part-of-Speech tagging and Named Entity Recognition), using prediction with expert advice. Our approach was able to converge to the performance of the best individual query strategies, and nearly reached the performance of a supervised baseline (which used twice the annotation budget).

Inspired by our previous work, we propose combining multiple query strategies using prediction with expert advice, now in a stream-based setting. Our goal is to converge to the most useful strategy without needing to evaluate them a priori for each language pair and set of MT models, since a change in these factors might have an impact on which query strategy proves more useful, as discussed by Lowell, Lipton, and Wallace (2019).

4. Problem Definition

In this article, we build on our previous work in which we combined online learning and MT (Mendonça et al. 2021), as it is compatible with the online interaction assumptions under consideration.

As mentioned in Section 3.2, in our previous work, we considered two feedback settings: a **full feedback** setting (i.e., feedback is assumed to be available to the translations outputted by all the MT systems in the ensemble), and a **partial feedback** setting (feedback is only available for the final translation chosen by the ensemble). For the first case, we applied prediction with expert advice, using EWAF to learn the systems' weights; for the partial feedback setting, we turned to adversarial multi-armed bandits, learning the systems' weights using Exponential-weighting for Exploration and Exploitation (EXP3) (Auer et al. 1995). The multi-armed bandits framework follows a slot-machine metaphor, so each "expert" is referred to as an **arm**. At each iteration, EXP3 selects one arm in a similar fashion to EWAF (randomly based on the distribution of the arms' weights); the main difference between these two algorithms lies in the loss, which is computed only for the selected arm in the case of EXP3.

Thus, in this scenario, each MT system is an expert (or arm) $j = 1 \dots J$, associated with a weight ω_j (all the systems start with the same weight). An overview of the learning process is illustrated in Figure 1: At each iteration t, a segment src_t is selected from the source language corpus and handed to all the MT systems. Each system outputs a translation $transl_{j,t}$ in the target language, and one of these translations is selected as the forecaster's decision, according to the probability distribution given by the systems' weights. The chosen translation $transl_{f,t}$ (in the partial feedback setting) or the translations outputted by all the systems (in the full feedback setting) receive a human score $score_{j,t}$, from which a loss $\ell_{j,t}$ is derived for the respective MT system. Finally, the weight $\omega_{f,t-1}$ of the chosen system (in the partial feedback setting) or the weights $\omega_{1,t-1} \dots \omega_{J,t-1}$ of all the systems (in the full feedback setting) are updated based on the loss computed.

In this work, our first challenge is to decide whether each incoming segment src_t and respective translations $transl_{1,t}, \dots, transl_{J,t}$ are informative enough so that the translations are worthy of being rated by a human. To address this challenge, we propose using a number of stream-based query strategies, which we detail in Section 5. We note that we make no assumptions about whether we have full or partial feedback; therefore, our contribution in this article builds on both feedback settings.

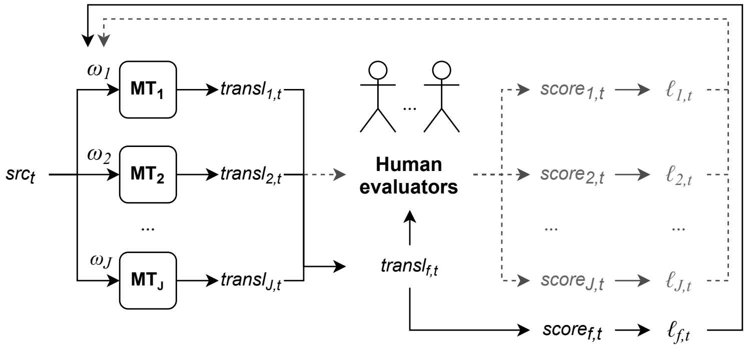

Figure 1
Overview of the online learning process applied to MT, at each iteration t (Mendonça et al. 2021): A source segment is translated by each of the MT systems in the ensemble; out of their translations, one of them is chosen as the most likely according to the online algorithm in use; the chosen translation (in the partial feedback setting), or all the translations (in the full feedback setting) receive a human score, based on which the MT system behind the chosen translation (partial feedback) or all the MT systems (full feedback) receive a loss and see their weights updated. The gray dashed arrows represent flows that only occur in the full feedback setting.

Our second challenge is to make the best use of multiple query strategies available, considering that we do not know a priori which query strategy will be the most adequate for a given language pair and set of models. To tackle this challenge, we propose combining multiple strategies under the framework of prediction with expert advice, updating their weights in an online fashion. We describe this approach in Section 6.

5. Stream-based Active Learning for MT

Given the online nature of our scenario, we operate under a stream-based setting (recall Section 2.1), in which we assume that we only have access to one source segment (and its respective automatic translations) at a time. However, in order to more easily adapt certain query strategies to this setting, we store the segments that have been scored by the human in a human-scored set $\mathcal{L}$ and the segments that have been discarded (i.e., for which the human was not asked a score) in a discarded set $\mathcal{U}$. Because we are dealing with a stream-based setting, the decision of whether to select a certain segment is based on whether the values computed by each query strategy are above or below a given threshold. The stream-based active learning process applied to our MT scenario is illustrated in Figure 2 and detailed in Algorithm 3.

For each source segment src_t, we obtain the translations outputted by each MT model $m_j \in \mathcal{M}$ (lines 4–7). Given these translations, the forecaster chooses a translation $transl_{f,t}$ as the most likely to be the best (line 8), according to the online algorithm under consideration (EWAF for the full feedback setting or EXP3 for the partial feedback setting). Then, we apply a query strategy to decide whether we should ask a human to score that segment's translations (line 9). Depending on the query strategy in use, $\mathcal{L}$ or $\mathcal{U}$ may also be taken into account. If the query strategy decides that the segment's translations should be scored by the human, the weights $\omega_1, \ldots, \omega_J$ associated with the MT models are updated based on the score received by the respective translations $transl_{1,t}, \ldots, transl_{J,t}$ (in the full feedback setting—lines 10–15), or only the weight of the MT model that outputted the forecaster's choice, m_f, is updated (in the partial feedback

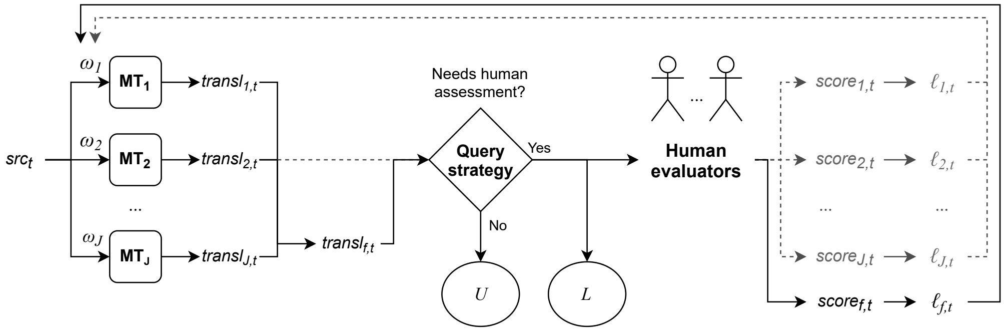

Figure 2
Overview of the stream-based active learning process applied to our online MT ensemble, at each iteration *t*: The translation chosen by the online algorithm (partial feedback) or the translations outputted by all the MT systems (full feedback) only receive a human score if the query strategy decides so. We keep track of the pairs source–translation(s) that were scored in the $\mathcal{L}$ set, and the ones that were not scored are stored in the $\mathcal{U}$ set. The gray dashed arrows represent flows that only occur in the full feedback setting.

Algorithm 3 Stream-based active learning for an online MT scoring

```
Input: MT models $\mathcal{M} = m_1, \ldots, m_J$, stream of source segments $\mathcal{S}$, human-scored set $\mathcal{L}$,
    discarded set $\mathcal{U}$
$\omega_{1,0}, \ldots, \omega_{J,0} \leftarrow 1$
$\mathcal{L}, \mathcal{U} \leftarrow \emptyset$
for each $src_t \in \mathcal{S}$ do
    for each $m_j \in \mathcal{M}$ do
        $transl_{j,t} \leftarrow m_j.translate(src_t)$                     # Obtaining translations
        $transl_t \leftarrow transl_t \cup transl_{j,t}$
    end for
    $transl_{f,t}, m_f \leftarrow forecasterMT(transl_t)$                # Choosing a translation
    if $selectInstance?(src_t, transl_{f,t}, transl_t, \mathcal{L}, \mathcal{U})$ then        # Query strategy
        if full feedback setting then
            for each $m_j \in \mathcal{M}$ do
                $score_{j,t} \leftarrow askHumanScore(src_t, transl_{j,t})$
                $scores_t \leftarrow scores_t \cup score_{j,t}$
                $\omega_{j,t} \leftarrow m_j.updateWeight(score_{j,t})$      # MT systems' weights update
            end for
        else if partial feedback setting then
            $score_{f,t} \leftarrow askHumanScore(src_t, transl_{f,t})$
            $\omega_{f,t} \leftarrow m_f.updateWeight(score_{f,t})$    # Chosen MT system's weight update
        end if
        $\mathcal{L} \leftarrow \mathcal{L} \cup \{src_t, transl_t, score_{f,t}, scores_t\}$          # Tracking scored segments
    else
        $\mathcal{U} \leftarrow \mathcal{U} \cup \{src_t, transl_t\}$          # Tracking segments that were not scored
    end if
end for
```

setting—lines 16–19). The scored segment is then added to $\mathcal{L}$ (line 20). If the current segment was not selected by the query strategy to be scored, it is added to $\mathcal{U}$ instead (line 22). The process repeats itself for as long as the stream of segments goes on.

Our stream-based setting, as well as the model-independence and low-resource requirements in our scenario, severely constrain which query strategies are worthy to use. In other words, we want to *avoid* strategies that: (1) rely on model details, since we assume the MT systems available to be black-box and that we only have access to the model's output (translation); (2) rely on an unlabeled set, since it is not available in a stream-based setting; or (3) imply any kind of additional training (as it is the case, for instance, of Learning To Active Learn or strategies based on language models). We thus propose the use of the following criteria as query strategies.

Translation Disagreement: We follow the Query-by-Committee strategy (Seung, Opper, and Sompolinsky 1992), by computing the disagreement among the translations outputted by the MT models in the ensemble (see Equation (5)). If the average agreement among all the translations $transl_{1,t}, \ldots, transl_{J,t}$, $AvgAgr$, is below a given threshold, then the segment should be scored by the human. We compute the agreement between two translations using: (1) a lexical similarity measure (Jaccard; Jaccard 1912), considering the segment's words as the basic unit; (2) the cosine similarity between pre-trained contextual segment-level embeddings (BERT; Devlin et al. 2019); (3) a translation evaluation metric (BLEU; Papineni et al. 2002), applied in a segment-level fashion.

$$AvgAgr\left(transl_t\right) = \frac{1}{\frac{J^2-J}{2}} \left(\sum_{j=1}^{J} \sum_{j'=1}^{j-1} agreement\left(transl_{j,t}, transl_{j',t}\right) \right) \tag{5}$$

Translation Difficulty: Inspired by earlier work that used Quality Estimation as a data selection criterion (Logacheva and Specia 2014), we measure the difficulty of translating a segment src_t using a Quality Estimation metric based on the perplexity of the translation, PRISM (Thompson and Post 2020) (see Equation (6)). If the average quality of a segment's translations, *AvgQuality*, is below a given threshold (i.e., if the metric does not expect the translations outputted by the MT models considering the source segment src_t) then the segment should be scored by the human.

$$AvgQuality\left(transl_t\right) = \frac{1}{J} \left(\sum_{j=1}^{J} Prism\left(transl_{j,t}, src_t\right) \right) \tag{6}$$

Diversity with Respect to Scored Segments: We apply the Diversity strategy (Fujii et al. 1998) to a stream-based setting by computing how much the current source segment src_t differs from the human-scored set $\mathcal{L}$. If the average similarity between src_t and each source segment in $\mathcal{L}$ is below a given threshold, then the segment should be scored by the human. We compute the similarity between two segments using: (1) a lexical similarity measure (Jaccard; Jaccard 1912), considering the segment's words as the basic unit, and (2) the cosine similarity between pre-trained contextual segment-level embeddings (BERT; Devlin et al. 2019). We also compute a variant of Diversity based on n-gram coverage (Ambati, Vogel, and Carbonell 2011; Zeng et al. 2019), according to Equation (7) (where $\mathbb{I}$ is the indicator function).

$$Div\left(src_t, \mathcal{L}\right) = \frac{\sum_{s \in ngram(src_t)} \mathbb{I}\left(s \notin ngram(\mathcal{L})\right)}{|ngram\left(src_t\right)|} \tag{7}$$

Density with Respect to Discarded Segments: We introduce a modified version of Density (Fujii et al. 1998), in which we compare the current source segment src_t to the segments in the discarded set $\mathcal{U}$ (i.e., those that were not scored by the human). In other words, if the average similarity between src_t and each source segment in $\mathcal{U}$ is above a given threshold, then the segment should be scored by the human. We compute the similarity between two segments using the same measures as in the Diversity strategy. Once again, we also compute a variant based on n-gram coverage (Ambati, Vogel, and Carbonell 2011; Zeng et al. 2019), according to Equation (8) (where $\#\left(s|\mathcal{X}\right)$ denotes the frequency of the n-gram s in the n-grams of set $\mathcal{X}$, and λ is a decay parameter to penalize n-grams seen in the human-scored set $\mathcal{L}$).

$$Den\left(src_t, \mathcal{U}, \mathcal{L}\right) = \frac{\sum_{s \in ngram(src_t)} \#\left(s|\mathcal{U}\right) e^{-\lambda \#(s|\mathcal{L})}}{|ngram(src_t)||ngram(\mathcal{U})|} \tag{8}$$

The inclusion of this strategy may be counterintuitive, since it chooses segments that are representative of those ignored so far (i.e., of those that were considered not informative enough to be scored by the human); however, at a certain point, it might be the case that the ignored segments are actually more representative of potential future sentences; therefore, it may be useful to ask the human to score them.

6. Combining Query Strategies Using Expert Advice

The introduction of active learning query strategies to decide whether to prompt the human to score the translation(s) outputted by the MT ensemble raises a new challenge: choosing the most appropriate query strategy for a given language pair and MT ensemble, without any prior knowledge nor pre-evaluation of the performance of each strategy, considering that their performance might be inconsistent across settings (Lowell, Lipton, and Wallace 2019). To tackle this challenge, we follow another previous work of ours, in which we combined different query strategies in a pool-based scenario for two sequence labeling tasks, using online learning under the framework of prediction with expert advice (Mendonça et al. 2020).

In our stream-based active learning scenario, each expert corresponds to a query strategy $k = 1, \ldots, K$, to which a weight ϕ_k is assigned (all the query strategies start with the same weight). The learning process is illustrated in Figure 3 and Algorithm 4. At each iteration, each query strategy casts a Boolean value indicating whether the current segment's translation(s) should be scored by the human (lines 10–13). The forecaster randomly chooses one of the query strategies' votes (line 14), based on the distribution of the strategies' weights (recall Equation (1) in Section 2.2). Depending on the forecaster's vote, the current segment's translation(s) are scored by the human (or not), and the weights of the MT systems in the ensemble are updated accordingly, as already seen in Section 5.

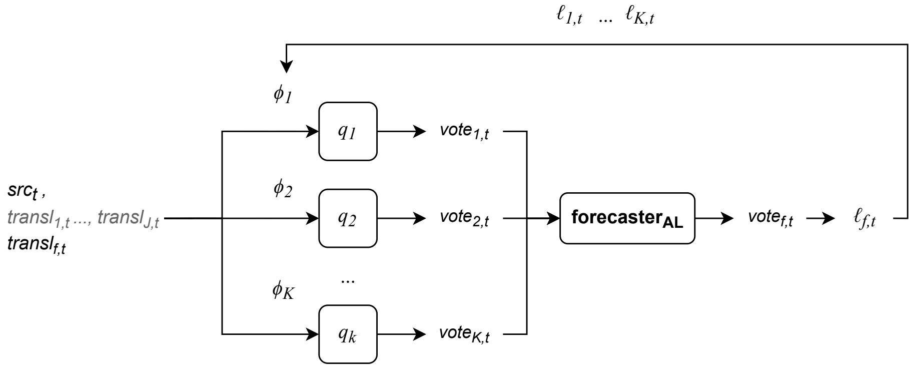

Figure 3
Overview of the stream-based active learning process with expert advice at each iteration t: instead of applying a single query strategy (Figure 2), we have multiple query strategies in parallel. For each pair source–translation(s), each strategy votes on whether the translation(s) should receive a human score. One of the strategies' votes is chosen according to EWAF. If the chosen vote is in favor of scoring, the improvement of the MT ensemble after the scoring is used as a loss function to update the query strategies' weights.

We want to award a greater weight to the strategies that cast the best votes, to make those strategies more likely to be considered in future iterations. This raises a question: How can we measure the success of each query strategy? Because our end goal is to improve the MT ensemble, we measure its improvement at each weight update by considering the **regret** $R_{\mathcal{M}}$ for not choosing the best MT system's translation at each iteration, up to the current iteration T (Equation (9)), which can be seen as a *dynamic* regret (Pang et al. 2018). Note that this regret formulation deviates from the traditional formulation, in that we compare the forecaster to the best sequence of decisions overall (whose cumulative loss is given by $\sum_{t=1}^{T} \min_{j=1,\ldots,J} \ell_{j,t}$), instead of the best expert overall (whose cumulative loss would be given by $\min_{j=1,\ldots,J} \sum_{t=1}^{T} \ell_{j,t}$). This way, we can ensure that our ensemble of active learning strategies learns from the best policy possible.

$$R_{\mathcal{M}} = \sum_{t=1}^{T} \ell_{f,t} - \sum_{t=1}^{T} \min_{j=1,\ldots,J} \ell_{j,t} \tag{9}$$

Thus, at each iteration, we consider the variation of the regret, $\Delta_{R_{\mathcal{M}}} = R_{\mathcal{M},t} - R_{\mathcal{M},t-1}$, to compute the loss function $\ell_{k,t}$ that, in turn, will allow us to update the weights $\phi_1, \ldots, \phi_K$ of all the query strategies (line 29): Those who voted in favor of scoring will receive a loss proportional to the increase in the regret, while the remaining strategies will receive a loss inversely proportional to the regret's increase (Equation (10)).

$$\ell_{k,t} = \begin{cases} \Delta_{R_{\mathcal{M}}} & \text{if } vote_{k,t} = 1 \\ 1 - \Delta_{R_{\mathcal{M}}} & \text{if } vote_{k,t} = 0 \end{cases} \tag{10}$$

Note that, if the forecaster's vote was against human scoring, there is no change in the regret of the MT ensemble; therefore we cannot update the weights of the query strategies.

Algorithm 4 Stream-based active learning with expert advice

```
Input: MT models $\mathcal{M} = m_1, \ldots, m_J$, stream of source segments $\mathcal{S}$, query strategies $\mathcal{Q} = q_1, \ldots, q_K$, human-scored set $\mathcal{L}$, discarded set $\mathcal{U}$
$\omega_{1,0}, \ldots, \omega_{J,0} \leftarrow 1$
$\phi_{1,0}, \ldots, \phi_{K,0} \leftarrow 1$
$\mathcal{L}, \mathcal{U} \leftarrow \emptyset$
for each $src_t \in \mathcal{S}$ do
    for each $m_j \in \mathcal{M}$ do
        $transl_{j,t} \leftarrow m_j.translate(src_t)$                          # Obtaining translations
        $transl_t \leftarrow transl_t \cup transl_{j,t}$
    end for
    $transl_{f,t}, m_f \leftarrow forecasterMT(transl_t)$                     # Choosing a translation
    for each $q_k \in \mathcal{Q}$ do
        $vote_{k,t} \leftarrow q_k.selectInstance?(src_t, transl_{f,t}, transl_t, \mathcal{L}, \mathcal{U})$     # Query strategy
        $votes_t \leftarrow votes_t \cup vote_{k,t}$
    end for
    $vote_{f,t} \leftarrow forecasterAL(votes_t)$                         # Choosing a query strategy
    if $vote_{f,t} == True$ then
        if full feedback setting then
            for each $m_j \in \mathcal{M}$ do
                $score_{j,t} \leftarrow askHumanScore(src_t, transl_{j,t})$
                $scores_t \leftarrow scores_t \cup score_{j,t}$
                $\omega_{j,t} \leftarrow m_j.updateWeight(score_{j,t})$         # MT systems' weights update
            end for
        else if partial feedback setting then
            $score_{f,t} \leftarrow askHumanScore(src_t, transl_{f,t})$
            $\omega_{f,t} \leftarrow m_f.updateWeight(score_{f,t})$        # Chosen MT system's weight update
        end if
        $R_{\mathcal{M},t-1} \leftarrow R_{\mathcal{M},t}$
        $R_{\mathcal{M},t} \leftarrow updateRegret()$                  # Updating the regret of the MT ensemble
        for each $q_k \in \mathcal{Q}$ do
            $\phi_{k,t} \leftarrow q_k.updateWeight(R_{\mathcal{M},t-1}, R_{\mathcal{M},t})$  # Query strategies' weights update
        end for
        $\mathcal{L} \leftarrow \mathcal{L} \cup \{src_t, transl_t, score_{f,t}, scores_t\}$             # Tracking scored segments
    else
        $\mathcal{U} \leftarrow \mathcal{U} \cup \{src_t, transl_t\}$                # Tracking segments that were not scored
    end if
end for
```

We follow a slightly different approach for the scenario when we only have partial feedback for the translations outputted by the MT systems (i.e., when using multi-armed bandits to learn the weights of the MT ensemble). Because only the MT system corresponding to the currently chosen arm receives a loss, we do not know what would have been the optimal choice in hindsight, on a real world scenario. Thus, instead of considering the cumulative loss for the forecaster and what would be the optimal cumulative loss in hindsight, we compute the regret as the difference between the *average* loss for the forecaster and the *average* loss of the arm with the lowest average loss at that iteration (obtained considering the amount of times that arm was chosen). In this case, $\Delta_{R_{\mathcal{M}}}$ may vary between -1 and 1, thus we compute $\ell_{k,t}$ as shown in Equation (11).

$$\ell_{k,t} = \begin{cases} \frac{\Delta_{R_{\mathcal{M}}}+1}{2} & \text{if } vote_{k,t} = 1 \\ 1 - \frac{\Delta_{R_{\mathcal{M}}}+1}{2} & \text{if } vote_{k,t} = 0 \end{cases} \tag{11}$$

For the cases where the lowest loss is zero, not as a merit of the arm's performance, but because such arm has not been chosen yet, we assume its loss to be the average between zero and the highest average loss so far among the remaining arms.

7. Experimental Setup

To validate our proposals, we performed an experiment using data from two MT shared tasks, which include human scores, allowing us to simulate an online interaction setting. In this simulation, the learning goal of the online MT ensemble is to give a greater weight to the top ranked MT systems competing for each language pair, based on the human scores received by their translations, and without knowing in advance which systems are the best.

Our experiment's goal is to converge to the top MT systems with as little human intervention as possible. Particularly, we want to observe how fast this happens: (1) using each individual stream-based query strategy proposed in Section 5; (2) using the online ensemble of query strategies, proposed in Section 6. However, we note that our true goal is to *gradually* and *dynamically* converge to the MT systems that received the best feedback in an online scenario where there is no gold knowledge of which systems would be the best.

In this section, we detail the data used (Section 7.1), the loss functions for the MT ensemble (Section 7.2), the implementation details for the query strategies (Section 7.3), the computing infrastructure on which the experiments were performed (Section 7.4), and the evaluation metrics considered (Section 7.5).

7.1 Data

We performed our experiments on the test datasets made available for the News Translation shared tasks in both WMT'19 (Barrault et al. 2019) and WMT'20 (Barrault et al. 2020). Regarding WMT'19, we performed our experiments on the same selection of language pairs as in our previous work (Mendonça et al. 2021), since it offers a diverse coverage of phenomena in the dataset: English → German (`en-de`), French → German (`fr-de`), German → Czech (`de-cs`), Gujarati → English (`gu-en`), and Lithuanian → English (`lt-en`). The test sets for these language pairs are summarized in Table 1. For each language pair, each source segment is associated with the following information:

- A reference translation in the target language (produced specifically for the task);
- The automatic translation outputted by each system competing in the task for that language pair;
- The average score obtained by each automatic translation, according to human assessments made by one or more crowd-sourced human

Table 1
Overview of the language pairs from the WMT'19 datasets considered in our experiments.

	en-de	fr-de	de-cs	gu-en	lt-en
Test set size (# segments)	1,997	1,701	1,997	1,016	1,000
Competing systems	22	10	11	12	11
Human assessments coverage	86.80%	23.52%	62.94%	75.00%	100.00%

evaluators, in two formats: a raw score in [0;100] and a z-score in $[-\infty; +\infty]$. Not all the automatic translations received a human assessment;

- The number of human evaluators for each automatic translation (if there were any).

Regarding WMT'20, we used the two language pairs that were scored by professional translators after the shared task, using Scalar Quality Metric (SQM) scores (henceforth referred to as pSQM) (Freitag et al. 2021a): English → German (en-de) and Chinese → English (zh-en).[4] These datasets are summarized in Table 2. For each language pair, each source segment is associated with:

- The automatic translation outputted by each MT system selected by Freitag et al. (2021a), out of the ones competing in the shared task;
- A set of human-made translations (which were also scored by the professional translators);
- The pSQM score (a scalar number between 0 and 7) reflecting the quality of each translation (human or automatic) according to a given professional translator;
- The ID of the professional translator responsible for each score.

Due to inconsistencies in the format of the .tsv files provided with the pSQM annotations, some source segments and/or translations were missing/misplaced; because these represented just a few cases, we discarded the entries with empty sources or translations. Then we computed the average of the pSQM scores per translation.

For each language pair, we report our experiment on a single shuffle of the respective test set, so that the original order of the segments would not bias the results. We report a single run instead of an average of multiple runs, since different runs of the active learning approaches could lead to the weights being updated in different iterations for the same segments (depending on whether each active learning strategy/Onception decided to ask for human feedback); thus it would not be viable to average across runs accurately.[5]

[4] The datasets annotated with pSQM scores are available in: https://github.com/google/wmt-mqm-human-evaluation.

[5] However, we ran our experiments in additional shuffles of the gu-en dataset, and we observed a similar pattern of performance across shuffles.

Table 2
Overview of the language pairs from the WMT'20 datasets annotated with pSQM scores.

	`en-de`	`zh-en`
Test set size (# segments)	1,418	2,000
Competing systems	7	8
Human translators	3	2
Human scores coverage	100%	100%

7.2 MT Loss Functions

In order to compute the loss function for the online learning algorithms applied to the ensemble of MT systems (EWAF for the full feedback setting, or EXP3 for the partial feedback setting), we used the human raw scores available for each segment (WMT'19) or the average of the pSQM scores provided (WMT'20). We normalized all the scores to be in the interval $[0, 1]$ and rounded them to two decimal places, to avoid exploding weight values due to the exponential update rule. It should be noted that the use of these specific scores merely aims at simulating a human in a realistic situation, but our approach should be agnostic to the score format or range.

As we can see in Table 1, not all segments received at least one human score in the WMT'19 shared task. Thus, we rely on the fallback strategies proposed in our previous work (Mendonça et al. 2021):

- HUMAN-ZERO: If there is no human score for the current translation, a score of zero is returned;
- HUMAN-AVG: If there is no human score for the current translation, the average of the previous scores received by the system behind that translation is returned as the current score;
- HUMAN-COMET: If there is no human score for the current translation, the COMET score (Rei et al. 2020) between the translation and the pair source–reference available in the corpus is returned as the current score (see Mendonça et al. [2021] for details).

In this experiment, for each combination of language pair and feedback setting, we considered only the fallback strategy that obtained the best results in Mendonça et al. (2021), as shown in Table 3. For WMT'20, all the translations received at least one pSQM score, so we did not need to use any fallback strategies.

Table 3
Loss functions considered for each combination of language pair (WMT'19) and feedback setting.

Language pair	**Full feedback**	**Partial feedback**
`en-de`	HUMAN-AVG	HUMAN-ZERO
`fr-de`	HUMAN-COMET	HUMAN-ZERO
`de-cs`	HUMAN-COMET	HUMAN-COMET
`gu-en`	HUMAN-ZERO	HUMAN-COMET
`lt-en`	HUMAN-ZERO	HUMAN-ZERO

It should be noted that, in our work, the decision on whether to prompt the human for feedback regarding a translation does not take into account whether the human is actually available, interested, or capable of providing such feedback (since, in a realistic scenario, one cannot guess that in advance). We thus keep these fallback strategies to cope with situations in which the human is prompted to provide feedback but chooses not to do it.

7.3 Active Learning Implementation Details

For each combination of language pair and feedback setting, we compared the following approaches:

- A baseline using all the data available to update the weights of the MT models;
- Each individual query strategy proposed in Section 5 and summarized below, plus a **random** strategy:
  - Diversity based on (1) Jaccard (**DivJac**), and (2) the cosine of sentence-level BERT (**DivBERT**);
  - Density based on (1) Jaccard (**DenJac**), and (2) the cosine of sentence-level BERT (**DenBERT**);
  - Translation Disagreement based on (1) Jaccard (**TDisJac**), (2) the cosine of sentence-level BERT (**TDisBERT**), and (3) sentence-level BLEU (**TDisBLEU**);
  - Translation Difficulty based on PRISM (**TDiff**);
  - *N*-gram coverage Diversity (**DivNgram**) and Density (**DenNgram**).
- An active learning approach combining multiple individual query strategies using prediction with expert advice, as described in Section 6:
  - **Onception (all)** combines all the query strategies listed above (including the random baseline);
  - **Onception (no Density)** combines all the query strategies except the Density based ones, since this query strategy is based on segments that were previously discarded for scoring;
  - **Onception (no Density and TDiff)** combines all the query strategies except the Density based ones and Translation Difficulty; we only report this variation for gu-en since PRISM was not trained in Gujarati.

Concerning the implementation of the query strategies, we made the following decisions. For the strategies based on *n*-gram coverage, we extracted all *n*-grams up to trigrams, using NLTK (Bird, Klein, and Loper 2009). For Translation Disagreement, we computed BLEU using SacreBLEU (Post 2018). For the query strategies using contextual embeddings, we extracted pre-trained embeddings using BERT as a Service (Xiao 2018) and the Base Cased models (English for the English segments and Multilingual for the remaining languages).[6] Before extracting the embeddings, as well as when using Jaccard or before extracting *n*-grams, we removed punctuation from the segments. Before extracting *n*-grams or using Jaccard, we also lowercased the segments and tokenized them (we used the JIEBA tokenizer for Chinese[7] and the NLTK tokenizer for the remaining languages).

Regarding the thresholds based on which the query strategies decide whether to query the human for feedback, we obtained them by averaging the similarity/agreement/PRISM/*n*-gram coverage values from the first five iterations, so that the thresholds could be within a viable range of values. This was needed to make sure that the strategies would not end up never selecting any segment to be scored or selecting all of them. We refer to Appendix A for the list of thresholds used.

7.4 Computing Infrastructure

We performed our experiment on a PC with a Windows operating system with an Intel Core CPU i7-9750H @ 2.60GHz CPU, 32GB RAM, and a NVIDIA GeForce RTX 2060 6GB GDDR6, except for the extraction of the BERT embeddings for the WMT'19 datasets, which was performed on a shared machine with 24 Intel Xeon CPU E5-2630 v2 @ 2.60GHz and 257 GB RAM.

7.5 Evaluation Metrics

The main goal of our experiment is to observe how fast each approach converges (i.e., gives a greater weight) to the best MT models for each language pair. As the gold standard, we consider the top ranked models on the WMT'19 official ranking (listed in Table 4), and the top "systems" (most of which being the human translators) according to Multidimensional Quality Metric (MQM) scores given by professional translators for the WMT'20 datasets (listed in Table 5).[8] Thus, for each human interaction (and subsequent update to the MT ensemble's weights), we report the performance of each approach by computing the overlap between the top $n = 3$ systems with greatest weights according to our approaches, $\hat{s_n}$, and the top $n = 3$ systems according to the shared task's official ranking/MQM scores, s_n^*:

$$OverlapTop_n = \frac{|\hat{s_n} \cap s_n^*|}{n}, n = 3 \tag{12}$$

We are focused only on the top MT systems since, in a realistic scenario (e.g., a Web MT service), a user would most likely only care about the main translation returned, or would at most consider one or two alternative translations. Moreover, the success of our approach is not restricted to converging to the best system, since the scores obtained in the WMT'19 shared task are not reliable enough to discriminate between

[6] `https://github.com/google-research/bert/`.

[7] `https://github.com/fxsjy/jieba`.

[8] Given that we could not find the third top system reported by Freitag et al. (2021a)—VolcTrans (Wu et al. 2020)—in the datasets provided, we considered the fourth top system (WeChat_AI) instead.

Table 4
Top 3 performing systems for each language pair in the WMT'19 News Translation shared task (Barrault et al. 2019). The systems named "online-[letter]" correspond to publicly available translation services and were anonymized in the shared task.

	Top 3	z-score	Raw score
en-de	Facebook-FAIR (Ng et al. 2019)	0.347	90.3
	Microsoft-sent-doc (Junczys-Dowmunt 2019)	0.311	93.0
	Microsoft-doc-level (Junczys-Dowmunt 2019)	0.296	92.6
fr-de	MSRA-MADL (Xia et al. 2019)	0.267	82.4
	eTranslation (Oravecz et al. 2019)	0.246	81.5
	LIUM (Bougares et al. 2019)	0.082	78.5
de-cs	online-Y	0.426	63.9
	online-B	0.386	62.7
	NICT (Dabre et al. 2019)	0.367	61.4
gu-en	NEU (Li et al. 2019)	0.210	64.8
	UEDIN (Bawden et al. 2019)	0.126	61.7
	GTCOM-Primary (Bei et al. 2019)	0.100	59.4
lt-en	GTCOM-Primary (Bei et al. 2019)	0.234	77.4
	tilde-nc-nmt (Pinnis, Krišlauks, and Rikters 2019)	0.216	77.5
	NEU (Li et al. 2019)	0.213	77.0

Table 5
Top 3 performing systems for each language pair according to MQM scores (Freitag et al. 2021a). The entries named "Human-[letter]" correspond to human translators. Note that lower MQM scores correspond to higher quality translations.

	Top 3	MQM score	pSQM score
en-de	Human-B	0.75	5.16
	Human-A	0.91	4.90
	Human-P	1.41	4.32
zh-en	Human-A	3.43	4.34
	Human-B	3.62	4.29
	WeChat_AI (Meng et al. 2020)	5.13	4.02

similarly performing systems (due to the lack of a large enough coverage of human assessments as well as the fact that these scores were crowd-sourced, rather than given by professionals). However, for a more detailed account of the performance of each approach, we report additional metrics in the appendix: (1) the evolution of the rank correlation (using Kendall's τ) between the MT systems sorted by their online learning weights and the official ranking/MQM scores (Appendix B); (2) the evolution of the weights of the query strategies when using Onception (Appendix C).

8. Results and Discussion

In this section, we focus on the evolution of the *OverlapTop*$_{n=3}$ throughout human feedback interactions. The overlap value is represented by the heatmap color scale: The darkest blue corresponds to a full overlap of the top 3 systems, while light yellow represents a lack of overlap between the online algorithm's learned top systems and the official top. Each iteration corresponds to a segment that was seen and whose translations received a human score, leading to an update to the MT systems' weights in the online ensemble. In other words, the approaches for which the heatmap bars are shorter consult the human fewer times, thus requiring less human effort, which is what we are aiming for (provided that it does not impact the ability to find the top systems). We note that, in this work, we assume a potentially endless stream of incoming source segments; thus, in this experiment, the maximum number of iterations for each language pair is given by the total amount of source segments available in the respective dataset.

8.1 Full Feedback Setting

Figures 4–8 (WMT'19) and Figures 9–10 (WMT'20) represent the evolution of the *OverlapTop*$_{n=3}$ for the full feedback setting (i.e., when all the translations for a given segment receive a human—or fallback—score). The first thing we can notice is that, for all language pairs, most active learning approaches allow us to reduce the number of segments for which the human is prompted to score the translations (as indicated by the shorter heatmap bars).

Starting with the WMT'19 dataset, for `en-de`, the query strategies that make their decision based on the translations (TDisJac, TDisBERT, TDisBLEU, and TDiff) are the most promising, converging to the total of the top 3 MT systems within much fewer interactions than the baseline without active learning (within 200–400 weight updates), allowing us to spare hundreds of human interactions without harming the MT ensemble performance. Onception (no Density) gets to the full top 3 slightly earlier than the baseline, while the Onception version that uses all the query strategies follows a very similar pattern as the baseline (which means it did not spare many human feedback interactions).

As for `fr-de`, both the baseline and most query strategies converge to the full top 3 early (within the first dozens of interactions), which indicates that, in this situation, any active learning strategy would be useful to avoid redundant human interactions.

A similar landscape can be found for `de-cs`, with several query strategies settling on the top 3 systems slightly earlier than the baseline (namely, DenJac, DenBERT, all the

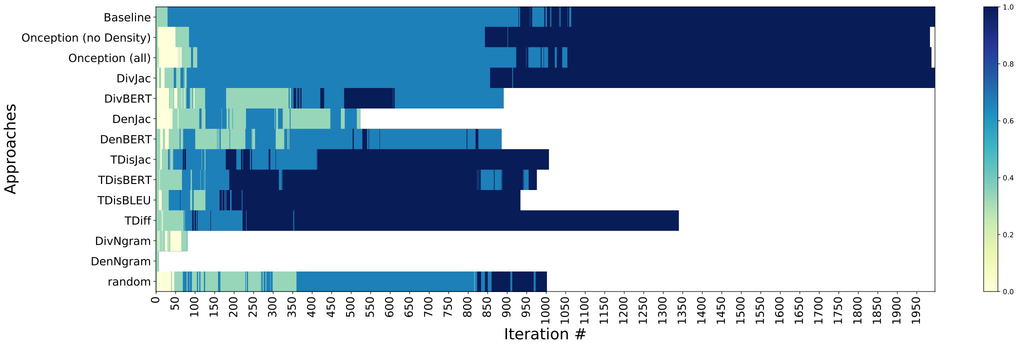

Figure 4
Overlap of the top 3 MT systems for `en-de` (WMT'19) in the full feedback setting (figure best seen in color).

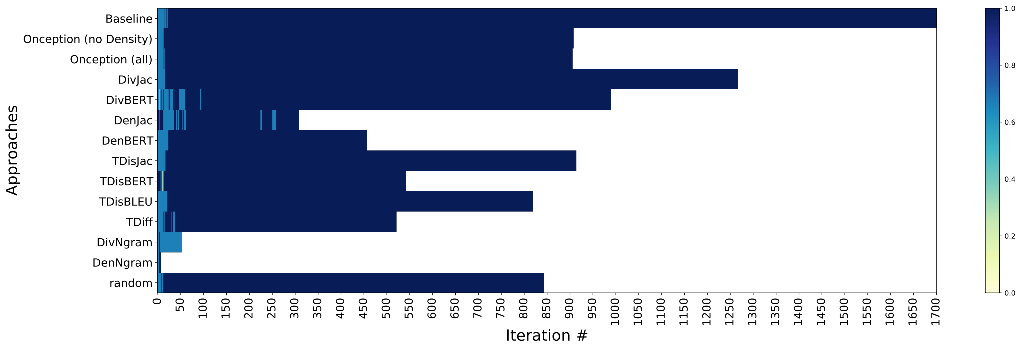

Figure 5
Overlap of the top 3 MT systems for `fr-de` (WMT'19) in the full feedback setting (figure best seen in color).

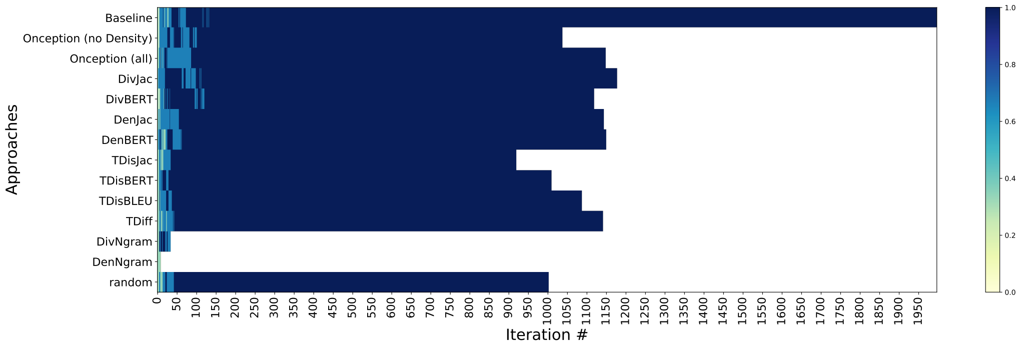

Figure 6
Overlap of the top 3 MT systems for `de-cs` (WMT'19) in the full feedback setting (figure best seen in color).

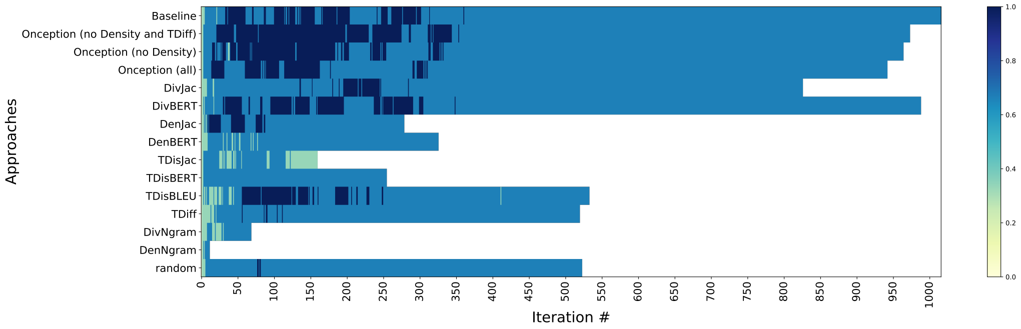

Figure 7
Overlap of the top 3 MT systems for `gu-en` (WMT'19) in the full feedback setting (figure best seen in color).

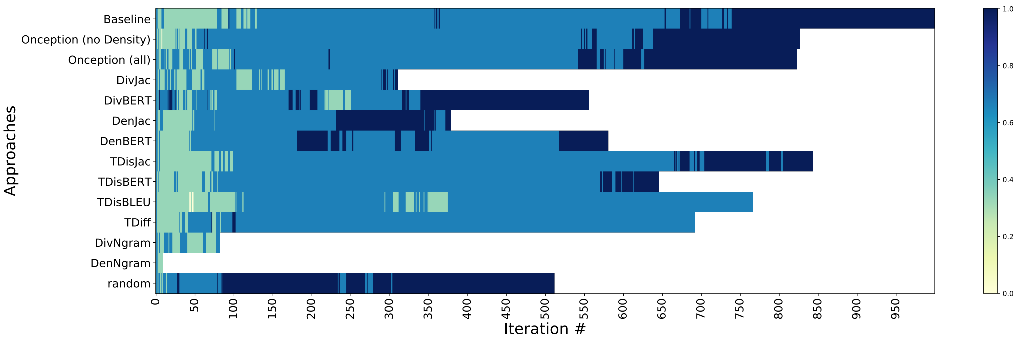

Figure 8
Overlap of the top 3 MT systems for `lt-en` (WMT'19) in the full feedback setting (figure best seen in color).

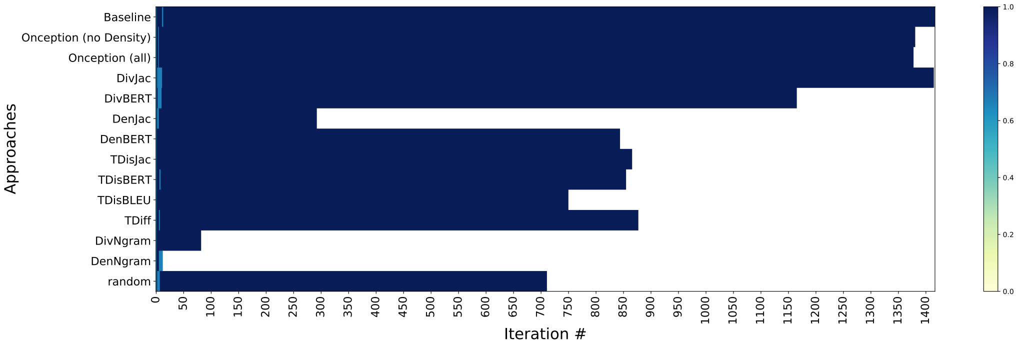

Figure 9
Overlap of the top 3 MT systems for `en-de` (WMT'20) in the full feedback setting (figure best seen in color).

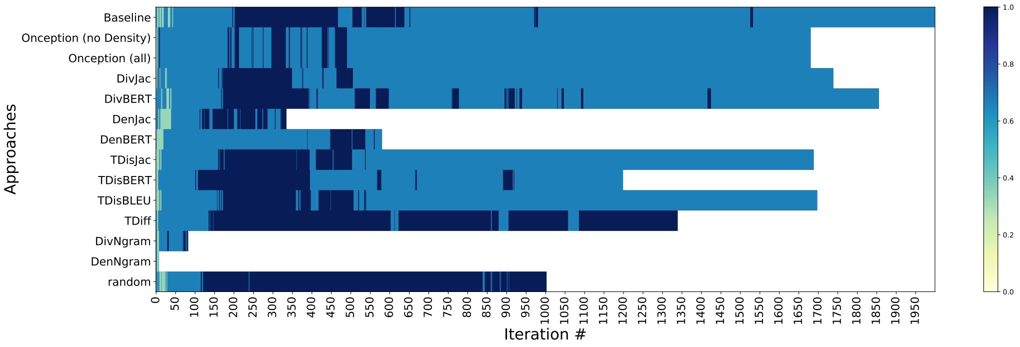

Figure 10
Overlap of the top 3 MT systems for `zh-en` (WMT'20) in the full feedback setting (figure best seen in color).

TDis strategies, and TDiff). Once again, using active learning could allow us to avoid redundant human effort.

As for `gu-en`, both Onception approaches seem to pay off, as they give a greater weight to the top 3 systems more consistently than the baseline and the individual query strategies. Even so, all the approaches (starting with the baseline) seem to "forget" at least one of the top systems. Among the Onception approaches, the one that does not include TDiff performs slightly better overall, which is expected, considering that the metric used by TDiff (PRISM) had not been trained on Gujarati.

For `lt-en`, both Onception approaches and some query strategies settle on the full top 3 earlier than the baseline, with the best strategy being random sampling. This suggests that using an active learning strategy that does not rely on a threshold (as is the case of random sampling) might be useful.

Moving to the WMT'20 dataset, for `en-de`, both the baseline and all the active learning approaches (either the individual strategies or Onception) reach the top 3 systems in about 20 interactions, which suggests that most interactions in this situation would be redundant. As such, using any active learning strategy would allow us to skip those redundant interactions. Recall that, for this dataset, the translation ratings are more reliable and consistent than the crowd-sourced scores available for the WMT'19 datasets, which likely explains the quick convergence toward the top 3.

Finally, for `zh-en`, the strategies that allow us to settle on the top 3 systems for the longest are TDiff and random sampling. Both the baseline and the remaining strategies end up only allowing us to find 2 of the top 3 systems, similarly to `gu-en` (perhaps indicating that the most valuable feedback would be at later interactions).

8.2 Partial Feedback Setting

Figures 11–15 (WMT'19) and Figures 16–17 (WMT'20) represent the evolution of the *OverlapTop*$_{n=3}$ for the partial feedback setting (i.e., when only the forecaster's selected translation receives a human—or fallback—score). At first glance, we can notice that the overall performance of both the baseline and all the active learning approaches is generally more inconsistent than in the full feedback setting, which can be explained by

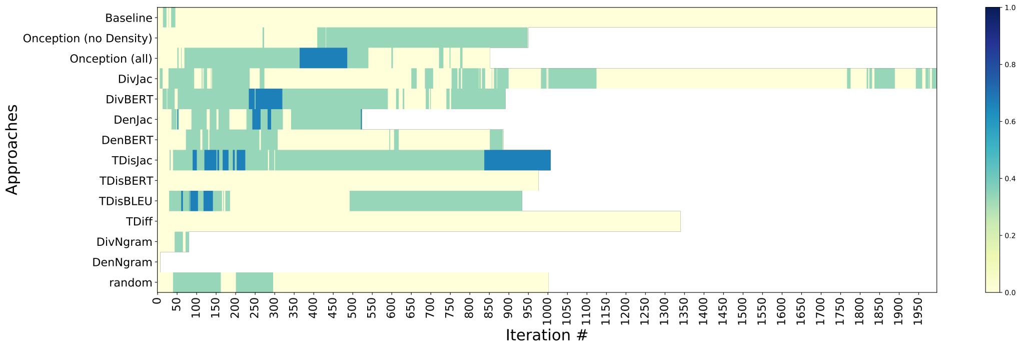

Figure 11
Overlap of the top 3 MT systems for `en-de` (WMT'19) in the partial feedback setting (figure best seen in color).

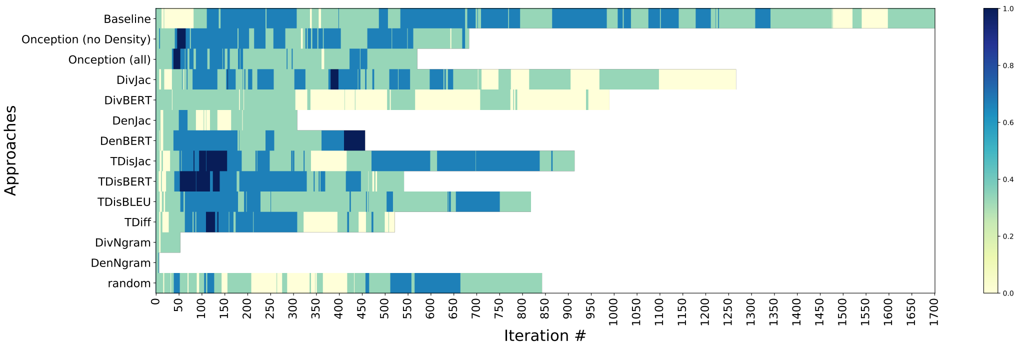

Figure 12
Overlap of the top 3 MT systems for `fr-de` (WMT'19) in the partial feedback setting (figure best seen in color).

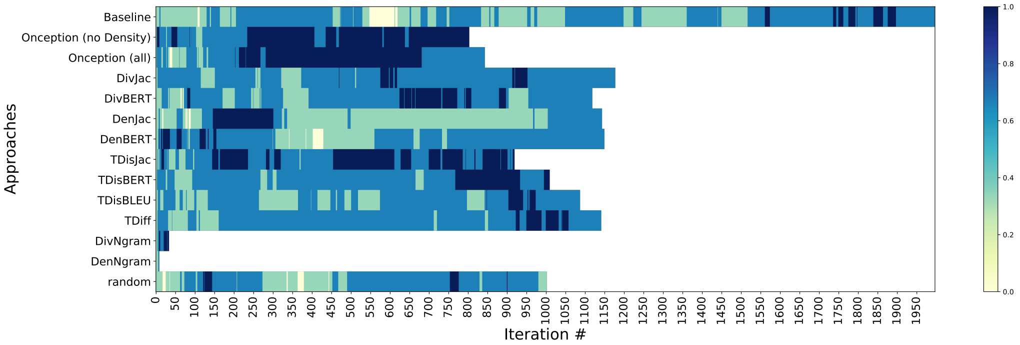

Figure 13
Overlap of the top 3 MT systems for `de-cs` (WMT'19) in the partial feedback setting (figure best seen in color).

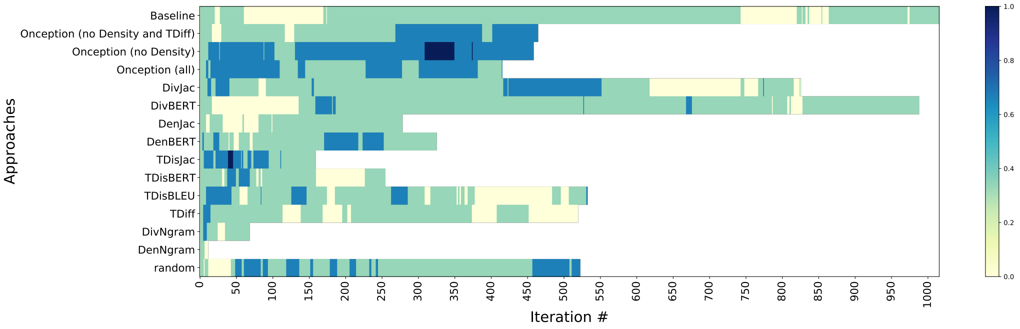

Figure 14
Overlap of the top 3 MT systems for `gu-en` (WMT'19) in the partial feedback setting (figure best seen in color).

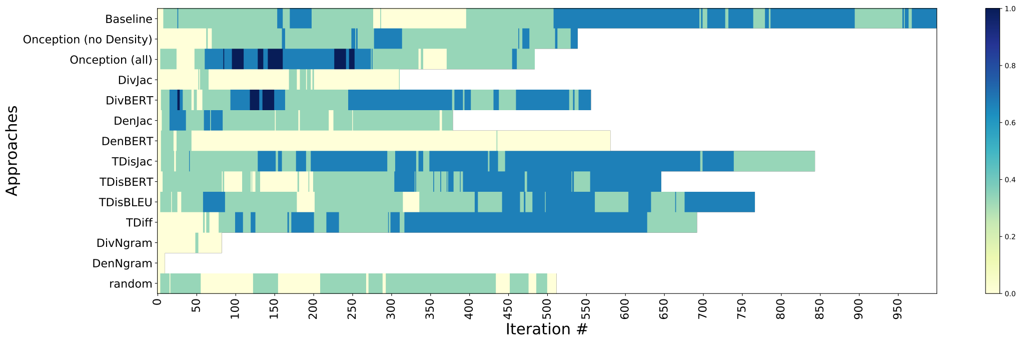

Figure 15
Overlap of the top 3 MT systems for `lt-en` (WMT'19) in the partial feedback setting (figure best seen in color).

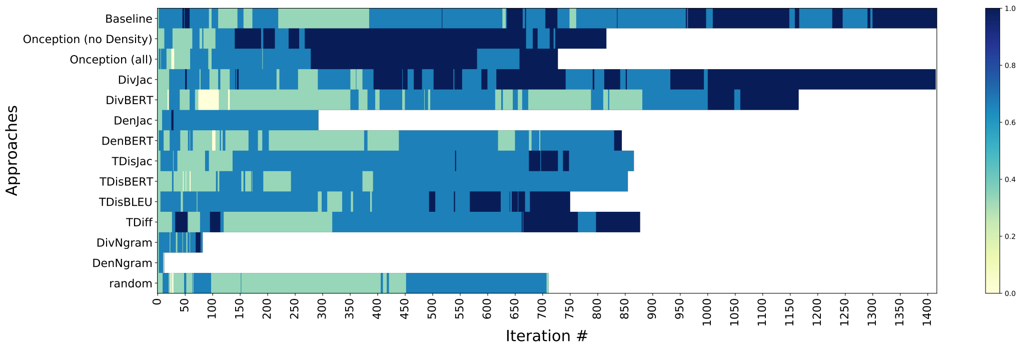

Figure 16
Overlap of the top 3 MT systems for `en-de` (WMT'20) in the partial feedback setting (figure best seen in color).

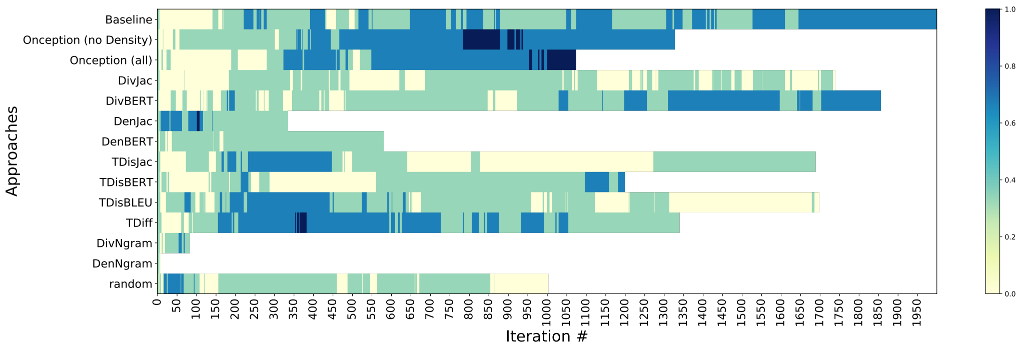

Figure 17
Overlap of the top 3 MT systems for `zh-en` (WMT'20) in the partial feedback setting (figure best seen in color).

the fact that there is fewer feedback to learn from (since only the MT system selected at each iteration receives a score).

Starting with the WMT'19 dataset, for `en-de`, most active learning strategies and Onception not only allowed us to spare human interactions, but it allowed us to find 1 or 2 top systems (most notably, TDisJac), while the baseline was not able to do so for most of the interactions. This suggests that using active learning might be useful, preferably Onception, to avoid strategies that do not perform so well.

As for `fr-de`, once again the baseline never finds the top 3 systems, while Onception and some individual strategies do (albeit briefly). While the overlap values are mostly inconsistent throughout the human interactions for all the approaches, it might still pay off to use active learning.

For `de-cs`, both Onception and several individual strategies (most notably TDisJac) find and stick with the top 3 systems earlier/for longer than the baseline, with Onception doing so more consistently. This suggests that, not only using active learning might pay off, but using Onception might be a safer bet than committing to one strategy.

As for `gu-en`, the baseline never finds more than one top system, while some query strategies do, with Onception (No Density) and TDisJac briefly finding the full top 3. Surprisingly, the version of Onception that does not include TDiff does not perform as well as Onception (No Density), despite the fact that PRISM (the metric used by TDiff) had not been trained on Gujarati.

For `lt-en`, once again, only two approaches found the full top 3 (albeit briefly): Onception and DivBERT. Despite the overall "forgetfulness" of some top systems by most approaches, using active learning still seems to pay off, as most strategies are able to find 2 top systems for longer/earlier than the baseline.

Moving to the WMT'20 dataset, for `en-de`, both Onception versions allow us to find the full top 3 more consistently and much earlier than the baseline and the individual query strategies, suggesting its usefulness in this setting.

Finally, for `zh-en`, only Onception (both versions) and TDiff are able to find the full top 3 (albeit not for long) and overall outperform the baseline in the earliest interactions. Once again, despite the inconsistent performances found across the partial feedback setting, Onception appears to be more useful than sticking to an arbitrary individual strategy or not using active learning at all.

8.3 Discussion

Full vs. Partial Feedback:. From the results above, we can see that for most language pairs in the full feedback setting, active learning can be helpful to skip redundant human effort in providing feedback (i.e., without harming the MT ensemble performance), with most strategies keeping up with the baseline's performance. This suggests that, in some cases, even just a random sampling strategy might be enough to make the most of as little feedback as possible. Using Onception might not be as critical in this setting, as opposed to the partial feedback setting, in which the individual strategies' performance is far more inconsistent. In the potential impossibility of pre-evaluating strategies prior to deployment, Onception is a safer bet for partial feedback settings, as it allows us to, at least, dodge the worst strategies.

WMT'19 vs. WMT'20 Datasets:. The pSQM ratings available for the WMT'20 datasets were provided by a pool of only 6 raters (which means that many segments' translations were scored by the same raters), all of whom were professional translators. Moreover, there was a full score coverage for these datasets. This contrasts with the datasets for

WMT'19, whose scores were crowd-sourced and often lacking for certain translations. This might explain why the overall performance for the WMT'19 datasets is more inconsistent across iterations (as well as the inconsistency across language pairs for WMT'19). However, the overall performance for the WMT'20 was also more inconsistent than initially expected—this can be explained by the fact that all the "competing systems" were very balanced (some were actually human translators and the remaining systems selected for pSQM/MQM annotations by Freitag et al.a [2021a] were mostly the top systems according to the WMT'20 ranking).

Performance across Different Strategies:. Although we could not find any pattern in terms of which strategies were the best (as expected from Lowell, Lipton, and Wallace's [2019] findings), it was anticipated that Diversity and Density strategies would be more prone to be inconsistent, as they more critically depend on an appropriate choice of threshold values—if the threshold is too strict, these strategies will not have any previous segments in $\mathcal{L}$ or $\mathcal{U}$ to compare the current segment with. We observed this sensitivity at its peak with the n-gram coverage strategies, which barely asked for any human feedback in our experiments, while often failing to capture 2 or more top systems.

RQ1: Can an active learning approach converge to the best systems with fewer human interactions?

As observed in the results reported above, for all language pairs and feedback settings considered, there is at least one query strategy that accelerates the process of finding the top 3 systems (or a subset of them). However, *which* query strategy should be used in each situation is not a trivial question to answer, since the performance of the query strategies under consideration varies greatly across language pairs and feedback settings. This was a predictable result considering the previous findings of Lowell, Lipton, and Wallace (2019) (observed in a pool-based active learning setting), which we now corroborate for a stream-based active learning setting.

RQ2: Does it pay off to dynamically combine multiple query strategies using expert advice (rather than using a single strategy)?

Considering the result discussed in the previous research question (i.e., that the best performing query strategy varies across settings), it generally pays off combining multiple strategies using online learning (i.e., using Onception), since we do not know in advance which strategy will perform better for a given setting, and could end up committing to one of the worst ones. We should note that we are not concerned regarding which specific query strategies are chosen more often, but whether Onception is able to perform in line with or better than the best strategies and dodge the worst ones, in terms of the MT ensemble performance.

Keeping that in mind, at a first look at the results reported, one could think that Onception would not always pay off, since this approach is not always perfectly aligned with the performance of the best query strategies (i.e., it does not verify the performance guarantees of EWAF). However, this behavior is foreseeable since Onception is learning from an *imperfect* source of feedback—the regret of the MT ensemble, which, during the first updates, may not reflect accurately what would be the best options overall, since it is being learned simultaneously. Despite this apparent limitation (caused by the lack of a reliable metric to learn the best strategies), our Onception approaches still pay off: They often perform better than the baseline and most individual strategies, finding the full

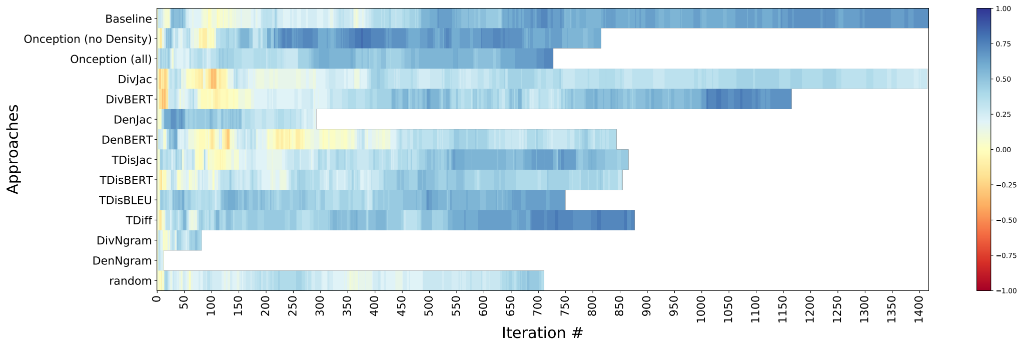

Figure 18
Evolution of the rank correlation for `en-de` (WMT'20) and the partial feedback setting (figure best seen in color).

top 3 systems earlier (i.e., requiring less human effort), and, perhaps more importantly, they tend to ignore the worst strategies, proving to be especially useful in the partial feedback setting.

Moreover, Onception often correlates well with the official ranking overall, as it is the case for `en-de` and the WMT'20 dataset (Figure 18). This result is particularly relevant for the WMT'20 datasets, as the competing systems are more balanced, meaning that even if Onception does not get the full top 3 correctly, it is probably relying on the next best systems.

9. Conclusions and Future Work

In this work, we contributed with an active learning solution that reduces the human effort needed to learn an online ensemble of MT models deployed simultaneously. We adapted multiple well-known query strategies to a stream-based setting, all of them being model-independent and compliant with a low-resource setting (as they do not require any additional pre-training data). Because we do not know in advance which query strategies will be more useful on a given setting, we dynamically combined multiple strategies using prediction with expert advice. In our experiments, this revealed itself to be a safer option than to commit to a single query strategy for most settings, often outperforming several individual strategies and sparing the need for human interaction even further.

One aspect to take into account is the the trade-off between the computational effort of running multiple strategies simultaneously and the resulting benefit. In practice, this trade-off will depend on how much effort is behind each strategy; thus, although our Onception approach is agnostic to the query strategies in the ensemble, this does not necessarily exclude a step of pre-filtering the strategies to use based on the effort to use each strategy. Moreover, the decision to use Onception *versus* using a single query strategy (or even just random sampling) should also be weighted against the characteristics of setting in hand: Settings in which feedback might be less reliable or might not be available for multiple MT systems seem more likely to benefit from the effort involved in running a combination of multiple strategies (as opposed to settings in which feedback might be more reliable or abundant, for which random sampling might suffice).

For future work, some extensions to our solution could be considered, namely: (1) using dynamic thresholds for the query strategies, rather than fixed ones (since some strategies may need to be more or less strict when choosing segments at different points of the learning process, particularly in the case of the Diversity and Density strategies); (2) using stochastic and/or contextual bandits, as well as non-stationary experts, to combine the query strategies (since the best strategy tends to vary as new segments arrive, and a greater emphasis on exploration might be beneficial); and (3) extending these experiments on more recent datasets, such as the ones from the WMT'21 shared tasks (Akhbardeh et al. 2021; Freitag et al. 2021b).

Appendix A. Query Strategies' Thresholds

These values represent the thresholds of similarity/agreement/quality estimation based on which each strategy decides whether it is worthy to ask for human feedback on a pair source-translation(s).

Table A1
Threshold values used for each query strategy and dataset.

Query strategy	**WMT'19**					**WMT'20**	
	`en-de`	`fr-de`	`de-cs`	`gu-en`	`lt-en`	`en-de`	`zh-en`
DivJac	0.09	0.06	0.03	0.02	0.003	0.09	0.07
DivBERT	0.91	0.90	0.90	0.95	0.91	0.94	0.90
DenJac	0.05	0.07	0.03	0.02	0.01	0.06	0.07
DenBERT	0.92	0.91	0.90	0.94	0.91	0.93	0.89
TDisJac	0.63	0.59	0.41	0.25	0.65	0.59	0.75
TDisBERT	0.975	0.969	0.962	0.948	0.983	0.983	0.993
TDisBLEU	0.48	0.40	0.28	0.14	0.41	0.45	0.62
TDiff	−0.70	−0.89	−1.33	−4.74	−1.16	−0.76	−1.06
DivNgram	0.90	0.88	0.95	0.96	0.99	0.88	0.91
DenNgram	0.00035	0.00052	0.00049	0.00025	0.00020	0.00015	0.00025

Appendix B. Rank Correlation Evolution

The following figures depict the evolution of the rank correlation (Kendall's τ) between the shared task's official ranking (WMT'19) or the ranking given by the MQM ratings (WMT'20) and the ranking given by the weights learned using online learning.

Full Feedback

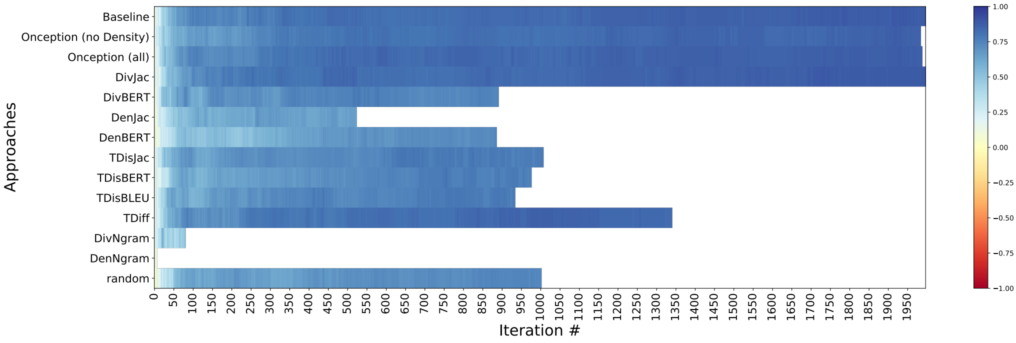

Figure B.1
Evolution of the rank correlation for `en-de` (WMT'19) (figure best seen in color).

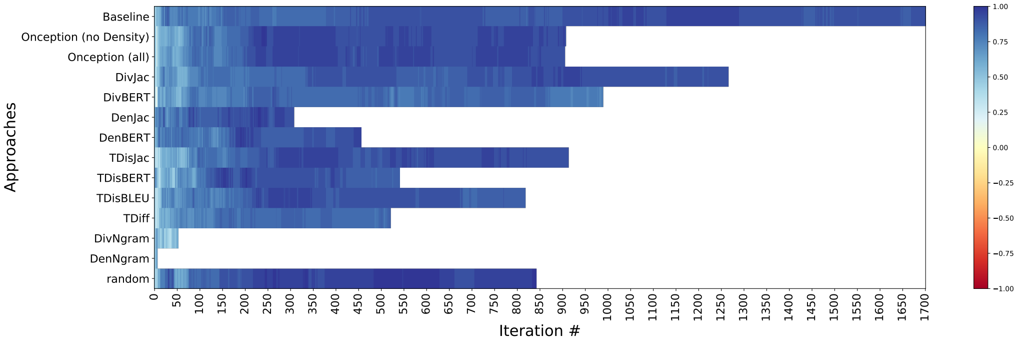

Figure B.2
Evolution of the rank correlation for `fr-de` (WMT'19) (figure best seen in color).

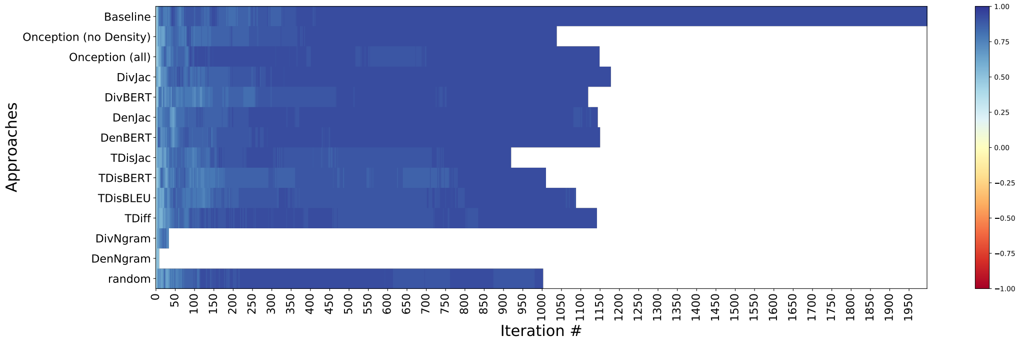

Figure B.3
Evolution of the rank correlation for `de-cs` (WMT'19) (figure best seen in color).

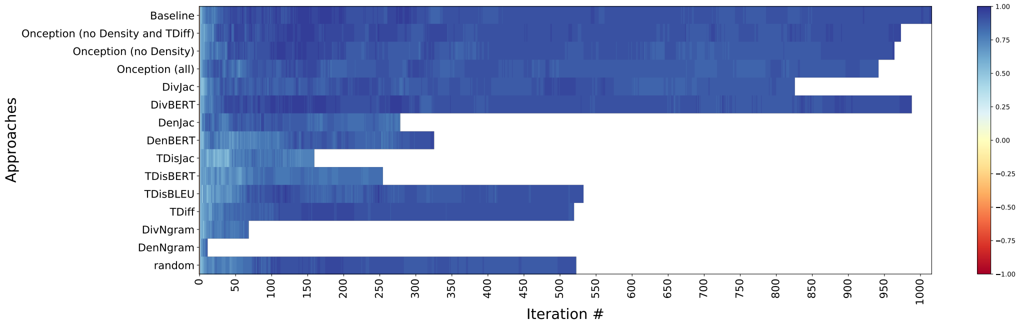

Figure B.4
Evolution of the rank correlation for `gu-en` (WMT'19) (figure best seen in color).

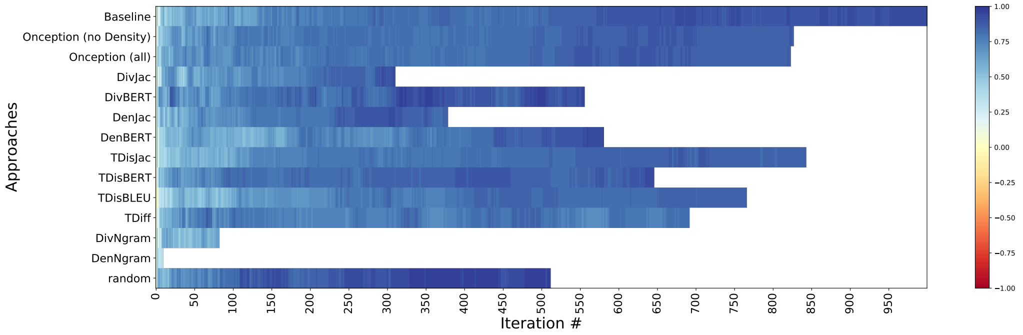

Figure B.5
Evolution of the rank correlation for `lt-en` (WMT'19) (figure best seen in color).

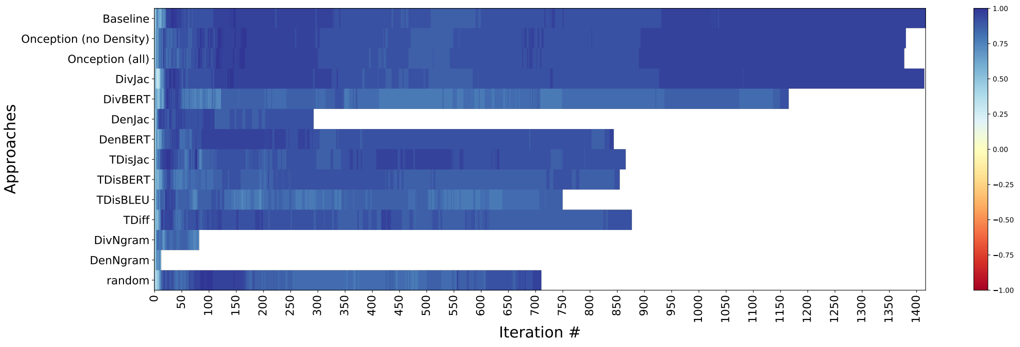

Figure B.6
Evolution of the rank correlation for `en-de` (WMT'20) (figure best seen in color).

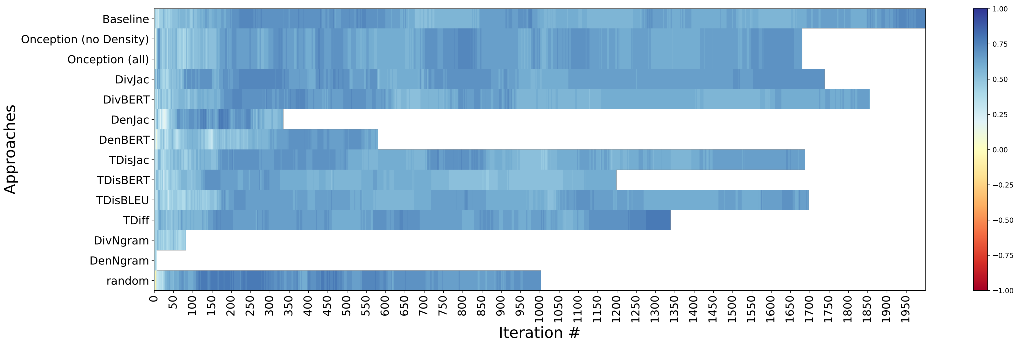

Figure B.7
Evolution of the rank correlation for `zh-en` (WMT'20) (figure best seen in color).

Partial Feedback

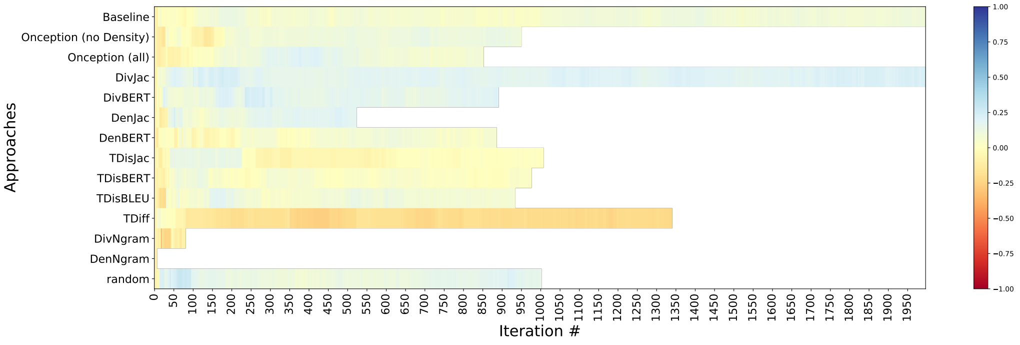

Figure B.8
Evolution of the rank correlation for `en-de` (WMT'19) (figure best seen in color).

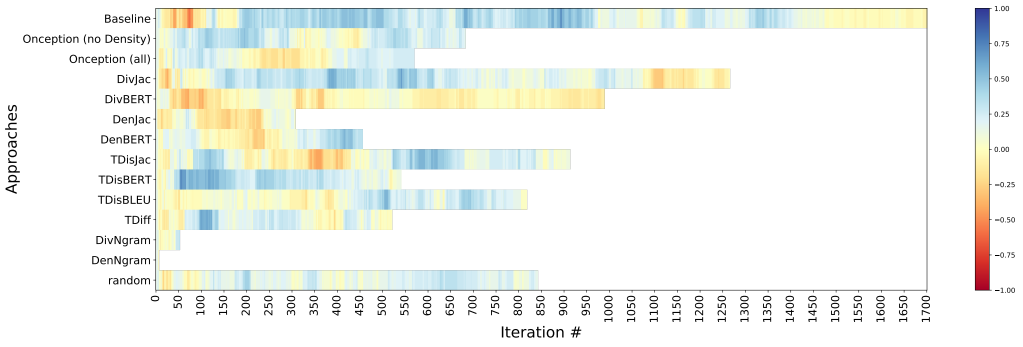

Figure B.9
Evolution of the rank correlation for `fr-de` (WMT’19) (figure best seen in color).

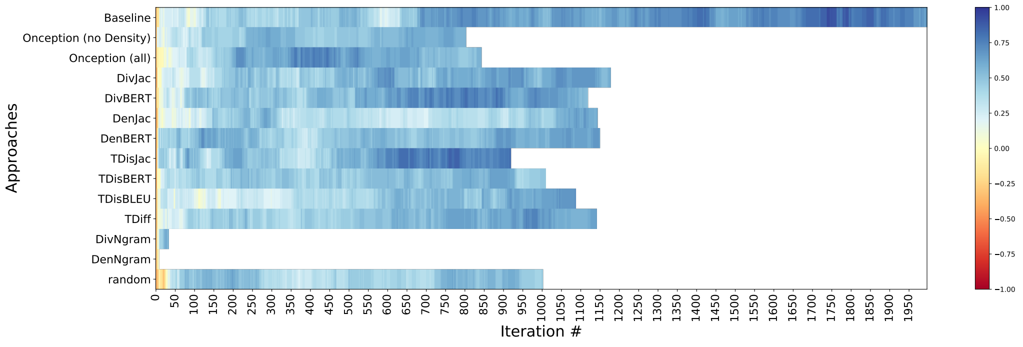

Figure B.10
Evolution of the rank correlation for `de-cs` (WMT’19) (figure best seen in color).

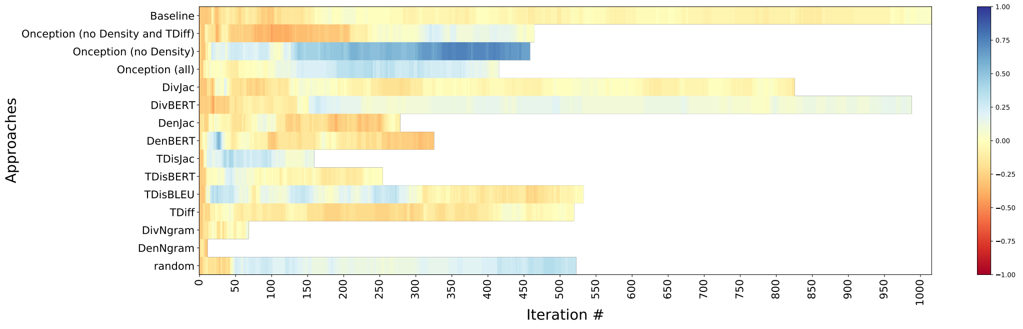

Figure B.11
Evolution of the rank correlation for `gu-en` (WMT’19) (figure best seen in color).

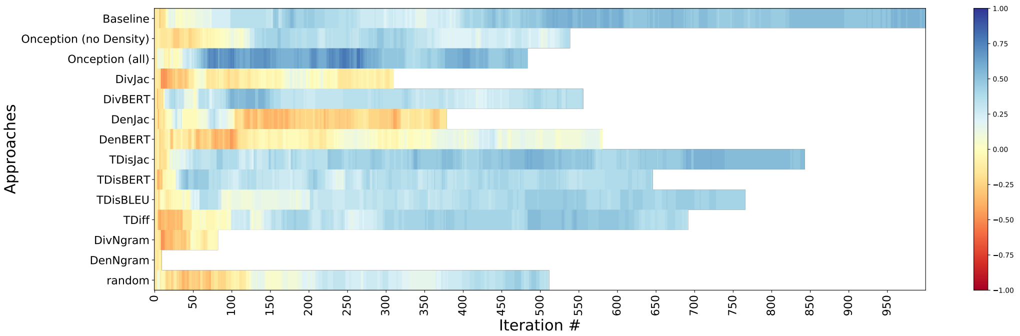

Figure B.12
Evolution of the rank correlation for `lt-en` (WMT’19) (figure best seen in color).

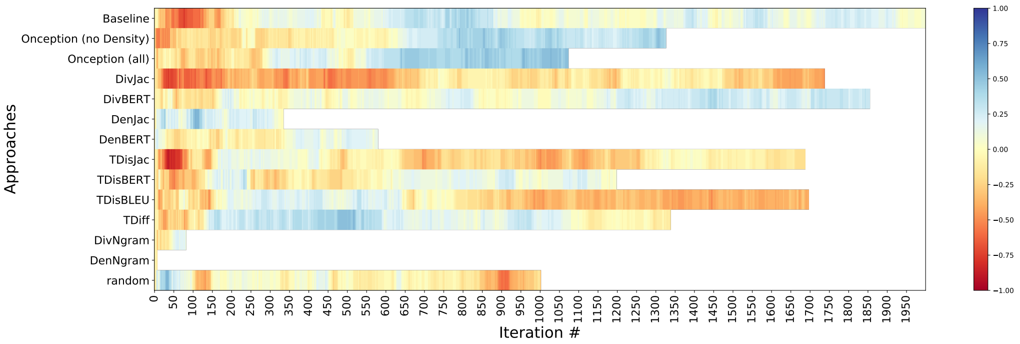

Figure B.13
Evolution of the rank correlation for `zh-en` (WMT’20) (figure best seen in color).

Appendix C. Query Strategies' Weights Evolution

The following figures depict the evolution of the weights of the query strategies when using Onception. We note that the x-axis is shorter than in the previous plots, and its length might vary, since it represents only the iterations for which a request for a score was made.

Full Feedback

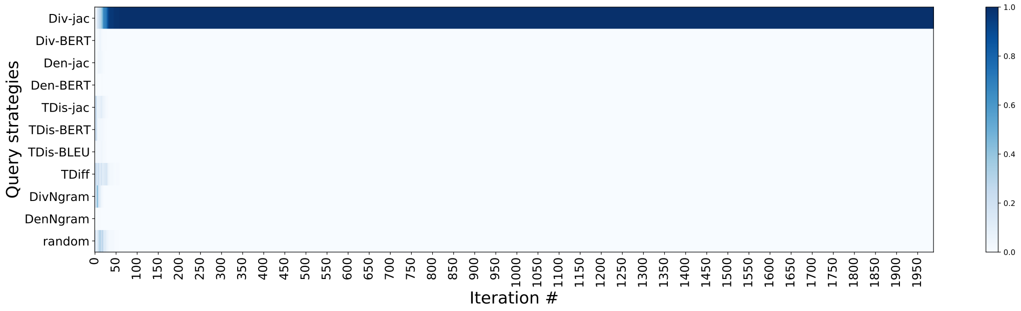

Figure C.1
Evolution of the query strategies' weights for `en-de` and EWAF (figure best seen in color).

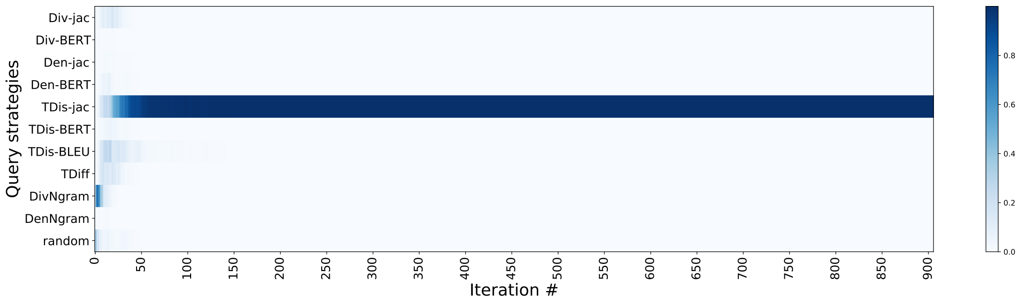

Figure C.2
Evolution of the query strategies' weights for `fr-de` (WMT'19) (figure best seen in color).

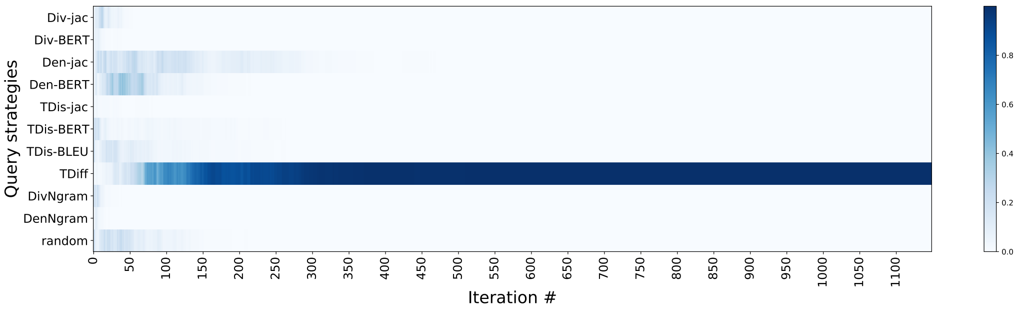

Figure C.3
Evolution of the query strategies' weights for `de-cs` (WMT'19) (figure best seen in color).

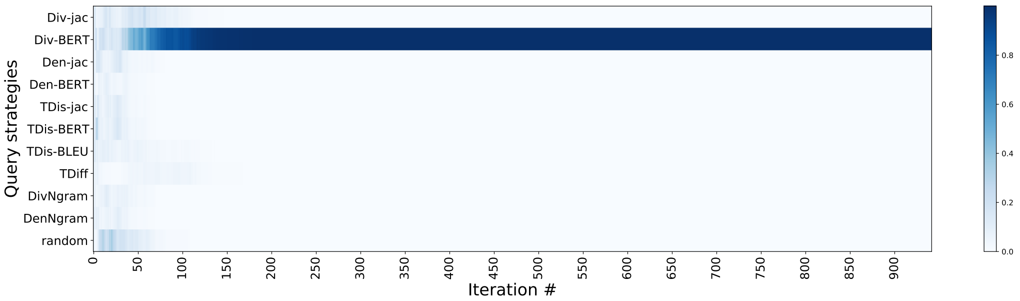

Figure C.4
Evolution of the query strategies' weights for `gu-en` (WMT'19) (figure best seen in color).

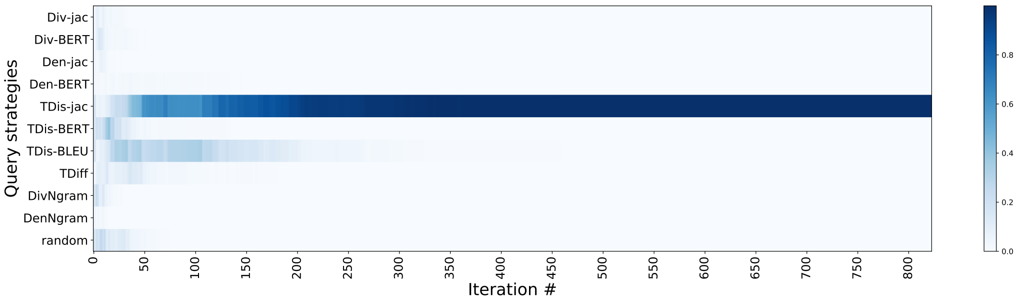

Figure C.5
Evolution of the query strategies' weights for `lt-en` (WMT'19) (figure best seen in color).

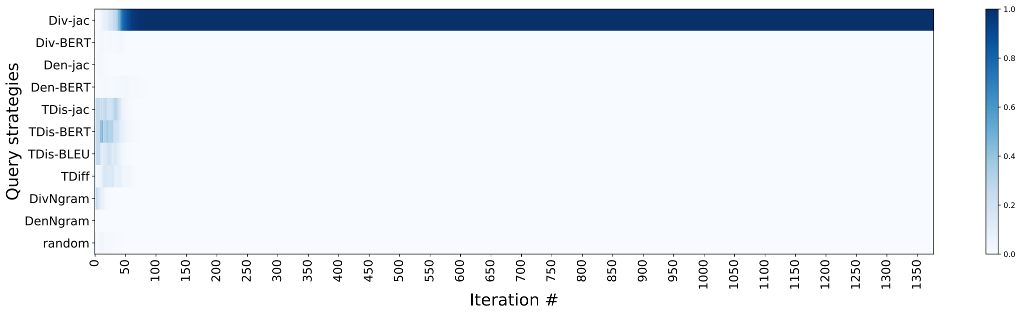

Figure C.6
Evolution of the query strategies' weights for `en-de` (WMT'20) (figure best seen in color).

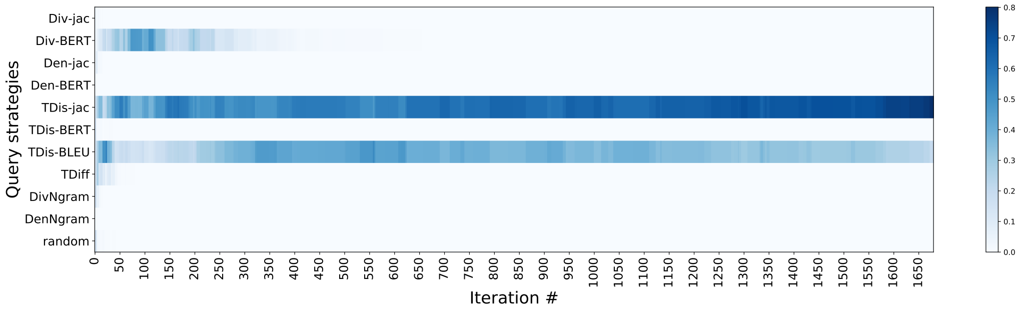

Figure C.7
Evolution of the query strategies' weights for `zh-en` (WMT'20) (figure best seen in color).

Partial Feedback

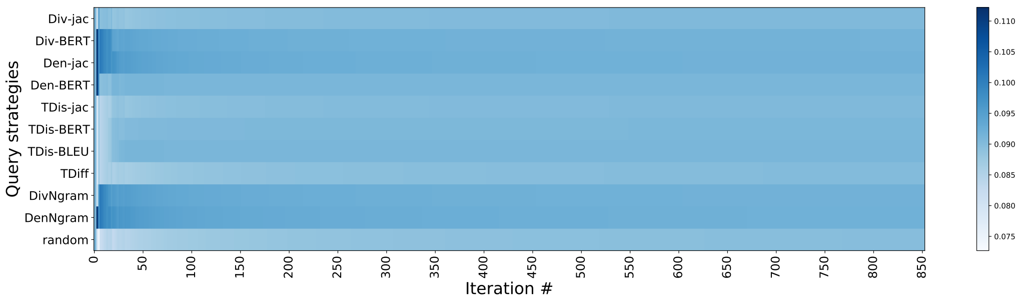

Figure C.8
Evolution of the query strategies' weights for `en-de` (WMT'19) (figure best seen in color).

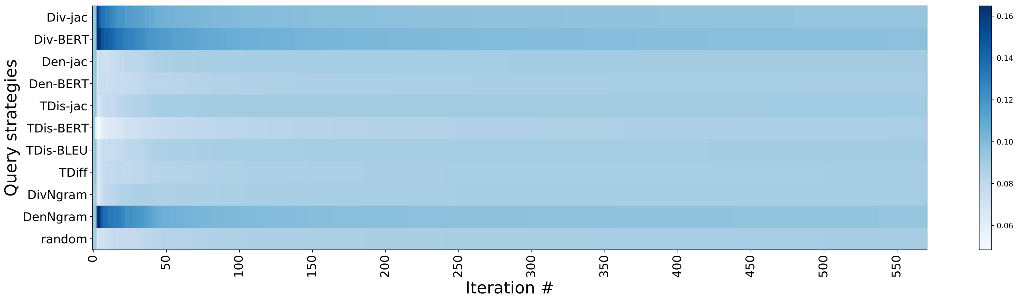

Figure C.9
Evolution of the query strategies' weights for `fr-de` (WMT'19) (figure best seen in color).

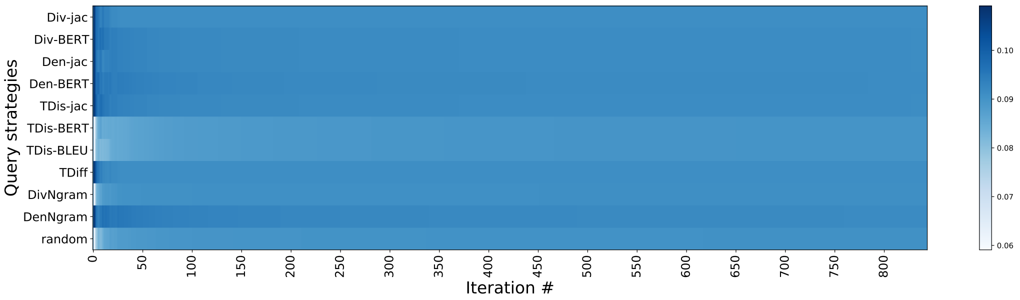

Figure C.10
Evolution of the query strategies' weights for `de-cs` (WMT'19) (figure best seen in color).

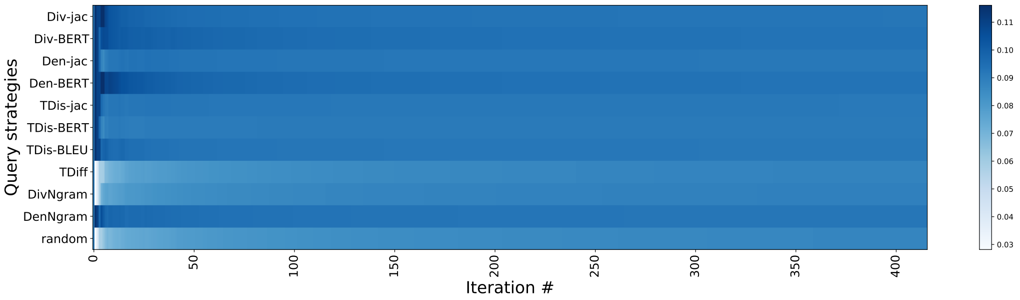

Figure C.11
Evolution of the query strategies' weights for `gu-en` (WMT'19) (figure best seen in color).

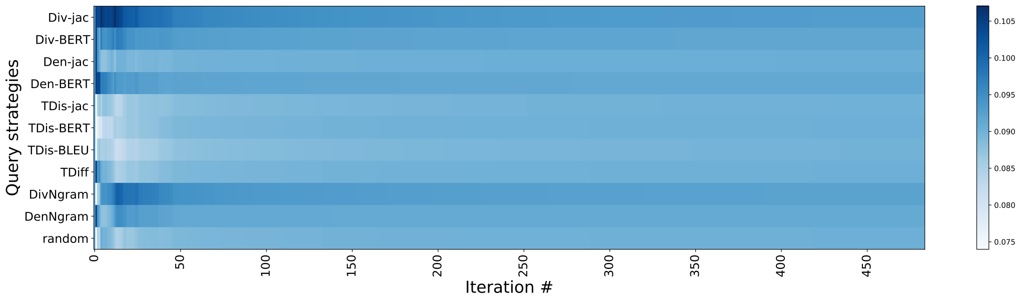

Figure C.12
Evolution of the query strategies' weights for `lt-en` (WMT'19) (figure best seen in color).

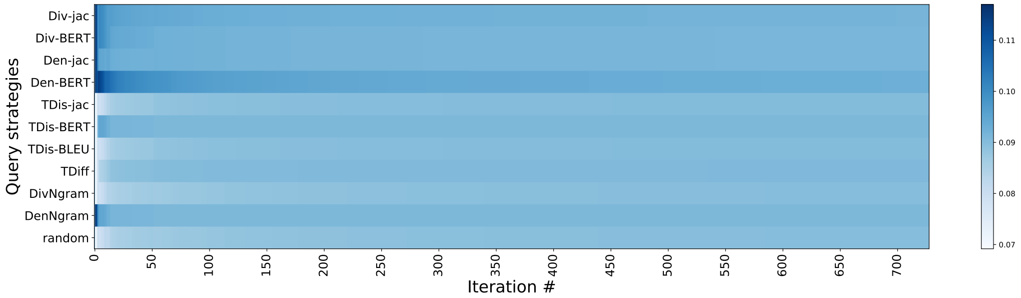

Figure C.13
Evolution of the query strategies' weights for `en-de` (WMT'20) (figure best seen in color).

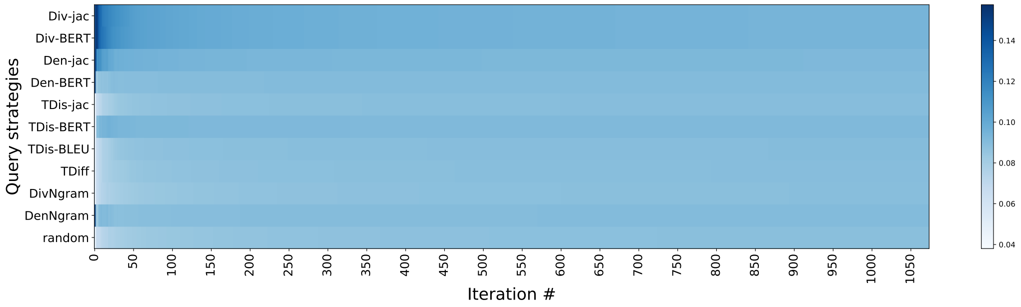

Figure C.14
Evolution of the query strategies' weights for `zh-en` (WMT'20) (figure best seen in color).

Acknowledgments

This work was supported by: Fundação para a Ciência e a Tecnologia under reference UIDB/50021/2020 (INESC-ID multi-annual funding), as well as under the HOTSPOT project with reference PTDC/CCI-COM/7203/2020; Air Force Office of Scientific Research, under award number FA9550-22-1-0475; P2020 program, supervised by Agência Nacional de Inovação (ANI), under the project CMU-PT Ref. 045909 (MAIA). Vânia Mendonça was funded by Fundação para a Ciência e a Tecnologia with a PhD grant, ref. SFRH/BD/121443/2016. The authors would like to thank to Ana Lúcia Santos for proofreading the document, and Soraia M. Alarcão for helping during brainstorming.

References

Akhbardeh, Farhad, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 Conference on Machine Translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88.

Ambati, Vamshi. 2012. *Active Learning and Crowd-Sourcing for Machine Translation in Low Resource Scenarios*. Ph.D. thesis, Carnegie Mellon University.

Ambati, Vamshi, Stephan Vogel, and Jaime Carbonell. 2011. Multi-strategy approaches to active learning for statistical machine translation. In *MT Summit XIII: 13th Machine Translation Summit*, pages 122–129.

Ananthakrishnan, Sankaranarayanan, Rohit Prasad, David Stallard, and Prem Natarajan. 2010. A semi-supervised batch-mode active learning strategy for improved statistical machine translation. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 126–134.

Auer, Peter, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 1995. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Annual Symposium on Foundations of Computer Science - Proceedings*, pages 322–331.

Auer, Peter, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77. `https://doi.org/10.1137/S0097539701398375`

Baram, Yoram, Ran El-Yaniv, and Kobi Luz. 2004. Online choice of active learning algorithms. *Journal of Machine Learning Research*, 5:255–291.

Barrault, Loïc, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, Marcos Zampieri. 2020. Findings of the 2020 Conference on Machine Translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55.

Barrault, Loïc, Ondrej 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. 2019. 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. `https://doi.org/10.18653/v1/W19-5301`

Bawden, Rachel, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The University of Edinburgh's submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 103–115. `https://doi.org/10.18653/v1/W19-5304`

Bei, Chao, Hao Zong, Conghu Yuan, Qingming Liu, and Baoyong Fan. 2019. GTCOM neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 116–121. https://doi.org/10.18653/v1/W19-5305

Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bougares, Fethi, Jane Wottawa, Anne Baillot, Loïc Barrault, and Adrien Bardet. 2019. LIUM's contributions to the WMT2019 news translation task: Data and systems for German-French language pairs. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 129–133. https://doi.org/10.18653/v1/W19-5307

Cesa-Bianchi, N. and G. Lugosi. 2006. *Prediction, Learning and Games*. Cambridge University Press.

Chu, Hong Min and Hsuan-Tien Lin. 2016. Can active learning experience be transferred? In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 841–846, IEEE. https://doi.org/10.1109/ICDM.2016.0100

Cohn, David, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine Learning*, 15:201–221.

Dabre, Raj, Kehai Chen, Benjamin Marie, Rui Wang, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. 2019. NICT's supervised neural machine translation systems for the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 168–174. https://doi.org/10.18653/v1/W19-5313

Dagan, Ido and Sean P. Engelson. 1995, Committee-based sampling for training probabilistic classifiers, *Machine Learning Proceedings 1995*. Elsevier, pages 150–157. https://doi.org/10.1016/B978-1-55860-377-6.50027-X

Deng, Yue, Kawai Chen, Yilin Shen, and Hongxia Jin. 2018. Adversarial active learning for sequence labeling and generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pages 4012–4018.

Denkowski, Michael, Chris Dyer, and Alon Lavie. 2014. Learning from post-editing: Online model adaptation for statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics 2014*, pages 395–404. https://doi.org/10.3115/v1/E14-1042

Devlin, Jacob, Ming-wei Chang, Lee Kenton, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. https://doi.org/10.18653/v1/N19-1423

Eck, Matthias. 2008. *Developing Deployable Spoken Language Translation Systems Given Limited Resources*. Ph.D. thesis, Fakultät für Informatik der Universität Fridericiana zu Karlsruhe.

Eck, Matthias, Stephan Vogel, and Alex Waibel. 2005. Low cost portability for statistical machine translation based on *n*-gram frequency and TF-IDF. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2005).*

Eetemadi, Sauleh, William Lewis, Kristina Toutanova, and Hayder Radha. 2015. Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29:189–223.

Fang, Meng, Yuan Li, and Trevor Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 595–605. https://doi.org/10.18653/v1/D17-1063

Finkelstein, Paige. 2020. Human-assisted neural machine translation: Harnessing human feedback for machine translation. University of Washington.

Freitag, Markus, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474. https://doi.org/10.1162/tacl_a_00437

Freitag, Markus, Ricardo Rei, Nitika Mathur, Chi Kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondrej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation (WMT)*, 1, pages 733–774.

Fujii, Atsushi, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. 1998.

Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4):1–25.

González-Rubio, Jesús and Francisco Casacuberta. 2014. Cost-sensitive active learning for computer-assisted translation. *Pattern Recognition Letters*, 37:124–134. https://doi.org/10.1016/j.patrec.2013.06.007

González-Rubio, Jesús, Daniel Ortiz-Martínez, and Francisco Casacuberta. 2011. An active learning scenario for interactive machine translation. In *ICMI'11 - Proceedings of the 2011 ACM International Conference on Multimodal Interaction*, pages 197–200. https://doi.org/10.1145/2070481.2070514

González-Rubio, Jesús, Daniel Ortiz-Martínez, and Francisco Casacuberta. 2012. Active learning for interactive machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 245–254.

Haffari, Gholamreza, Maxim Roy, and Anoop Sarkar. 2009. Active learning for statistical phrase-based machine translation. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, pages 415–423. https://doi.org/10.3115/1620754.1620815

Hazra, Rishi, Parag Dutta, Shubham Gupta, Mohammed Abdul Qaathir, and Ambedkar Dukkipati. 2021. Active^2 learning: Actively reducing redundancies in active learning methods for sequence tagging and machine translation. In *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pages 1982–1995.

Hildebrand, Almut Silja, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Language model adaptation for statistical machine translation based on information retrieval. In *EAMT 2005 Conference Proceedings*, pages 133–142.

Hoi, Steven C. H., Doyen Sahoo, Jing Lu, and Peilin Zhao. 2021. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289. https://doi.org/10.1016/j.neucom.2021.04.112

Hsu, Wei Ning and Hsuan-Tien Lin. 2015. Active learning by learning. In *Proceedings of the National Conference on Artificial Intelligence*, volume 4, pages 2659–2665.

Hu, Rong, Sarah Jane Delany, and Brian Mac Namee. 2010. EGAL: Exploration guided active learning for TCBR. In *Proceedings of ICCBR*, pages 156–170. https://doi.org/10.1007/978-3-642-14274-1_13

Ittycheriah, Abraham and Salim Roukos. 2007. Direct translation model 2. In *Proceedings of the NAACL HLT 2007*, pages 57–64.

Jaccard, Paul. 1912. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50. https://doi.org/10.1111/j.1469-8137.1912.tb05611.x

Junczys-Dowmunt, Marcin. 2019. Microsoft Translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233. https://doi.org/10.18653/v1/W19-5321

Karimova, Sariya, Patrick Simianer, and Stefan Riezler. 2018. A user-study on online adaptation of neural machine translation to human post-edits. *Machine Translation*, 32(4):309–324. https://doi.org/10.1007/s10590-018-9224-8

Konyushkova, Ksenia, Sznitman Raphael, and Pascal Fua. 2017. Learning active learning from data. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 4226–4236.

Lai, T. L. and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22. https://doi.org/10.1016/0196-8858(85)90002-8

Lam, Tsz Kin, Julia Kreutzer, and Stefan Riezler. 2018. A reinforcement learning approach to interactive-predictive neural machine translation. In *21st Annual Conference of the European Association for Machine Translation*, pages 169–178.

Levenberg, Abby, Chris Callison-Burch, and Miles Osborne. 2010. Stream-based translation models for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Main Conference*, pages 394–402.

Lewis, David D. and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, pages 3–12.

Li, Bei, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The NiuTrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266. https://doi.org/10.18653/v1/W19-5325

Liu, Ming, Wray Buntine, and Gholamreza Haffari. 2018a. Learning how to actively learn: A deep imitation learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1874–1883. https://doi.org/10.18653/v1/P18-1174

Liu, Ming, Wray Buntine, and Gholamreza Haffari. 2018b. Learning to actively learn neural machine translation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018)*, pages 334–344. https://doi.org/10.18653/v1/K18-1033

Logacheva, Varvara and Lucia Specia. 2014. A quality-based active sample selection strategy for statistical machine translation. In *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, pages 2690–2695.

Lowell, David, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 21–30. https://doi.org/10.18653/v1/D19-1003

Lü, Yajuan, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 343–350.

Mandal, A., D. Vergyri, W. Wang, J. Zheng, A. Stolcke, G. Tur, D. Hakkani-Tür, and N. F. Ayan. 2008. Efficient data selection for machine translation. In *2008 IEEE Spoken Language Technology Workshop*, pages 261–264. https://doi.org/10.1109/SLT.2008.4777890

Mathur, Prashant, Mauro Cettolo, and Marcello Federico. 2013. Online learning approaches in computer assisted translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 301–308.

Mendonça, Vânia, Luisa Coheur, Alberto Sardinha, and Ana Lúcia Santos. 2020. Query strategies, assemble! Active learning with expert advice for low-resource natural language processing. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. https://doi.org/10.1109/FUZZ48607.2020.9177707

Mendonça, Vânia, Ricardo Rei, Luisa Coheur, Alberto Sardinha, and Ana Lúcia Santos. 2021. Online learning meets machine translation evaluation: Finding the best systems with the least human effort. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3105–3117. https://doi.org/10.18653/v1/2021.acl-long.242

Meng, Fandong, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. WeChat neural machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 239–247.

Naradowsky, Jason, Xuan Zhang, and Kevin Duh. 2020. Machine translation system selection from bandit feedback. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas*, pages 50–63.

Ng, Nathan, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319. https://doi.org/10.18653/v1/W19-5333

Nguyen, Khanh, Hal Daumé III, and Jordan Boyd-Graber. 2017. Reinforcement learning for bandit neural machine translation with simulated human feedback. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1464–1474.

Oravecz, Csaba, Katina Bontcheva, Adrien Lardilleux, László Tihanyi, and Andreas Eisele. 2019. eTranslation's submissions to the WMT 2019 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 320–326. https://doi.org/10.18653/v1/W19-5334

Ortiz-Martínez, Daniel. 2016. Online learning for statistical machine translation. *Computational Linguistics*, 42(1):121–161. https://doi.org/10.1162/COLI_a_00244

Osugi, Thomas, Deng Kun, and Stephen Scott. 2005. Balancing exploration and exploitation: A new algorithm for active machine learning. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 330–337.

Pang, Kunkun, Mingzhi Dong, Yang Wu, and Timothy M. Hospedales. 2018. Dynamic ensemble active learning: A non-stationary bandit with expert advice. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2269–2276. https://doi.org/10.1109/ICPR.2018.8545422

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. https://doi.org/10.3115/1073083.1073135

Peris, Álvaro and Francisco Casacuberta. 2018. Active learning for interactive neural machine translation of data streams. In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018)*, pages 151–160.

Peris, Álvaro and Francisco Casacuberta. 2019. Online learning for effort reduction in interactive neural machine translation. *Computer Speech and Language*, 58:98–126. https://doi.org/10.1016/j.csl.2019.04.001

Pinnis, Marcis, Rihards Krišlauks, and Matiss Rikters. 2019. Tilde's machine translation systems for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 327–334. https://doi.org/10.18653/v1/W19-5335

Popović, Maja. 2015. chrF: Character *n*-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395. https://doi.org/10.18653/v1/W15-3049

Post, Matt. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. https://doi.org/10.18653/v1/W18-6319

Rei, Ricardo, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702. https://doi.org/10.18653/v1/2020.emnlp-main.213

Robbins, Herbert. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535. https://doi.org/10.1090/S0002-9904-1952-09620-8

Settles, Burr. 2010. Active learning literature survey. Technical report. University of Wisconsin-Madison Department of Computer Sciences.

Settles, Burr and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079. https://doi.org/10.3115/1613715.1613855

Seung, H. Sebastian, Manfred Opper, and Haim Sompolinsky. 1992. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 287–294. https://doi.org/10.1145/130385.130417

Sokolov, Artem, Julia Kreutzer, Christopher Lo, and Stefan Riezler. 2016. Learning structured predictors from bandit feedback for interactive NLP. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1610–1620. https://doi.org/10.18653/v1/P16-1152

Sokolov, Artem, Julia Kreutzer, Kellen Sunderland, Pavel Danchenko, Witold Szymaniak, Hagen Fürstenau, and Stefan Riezler. 2017. A shared task on bandit learning for machine translation. In *Proceedings of the Conference on Machine Translation (WMT)*, volume 2, pages 514–524. https://doi.org/10.18653/v1/W17-4756

Thompson, Brian and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121.

Turchi, Marco, Matteo Negri, M. Amin Farajian, and Marcello Federico. 2017. Continuous learning from human post-edits for neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1):233–244. https://doi.org/10.1515/pralin-2017-0023

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 5999–6009.

Vu, Thuy Trang, Ming Liu, Dinh Phung, and Gholamreza Haffari. 2019. Learning how to active learn by dreaming. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4091–4101.

Wieting, John and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462. https://doi.org/10.18653/v1/P18-1042

Wu, Liwei, Xiao Pan, Zehui Lin, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2020. The volctrans machine translation system for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 305–312.

Xia, Yingce, Xu Tan, Fei Tian, Fei Gao, Weicong Chen, Yang Fan, Linyuan Gong, Yichong Leng, Renqian Luo, Yiren Wang, Lijun Wu, Jinhua Zhu, Tao Qin, and Tie-Yan Liu. 2019. Microsoft Research Asia's systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 424–433. https://doi.org/10.18653/v1/W19-5348

Xiao, Han. 2018. bert-as-service. https://github.com/hanxiao/bert-as-service.

Zeng, Xiangkai, Sarthak Garg, Rajen Chatterjee, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Empirical evaluation of active learning techniques for neural MT. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo)*, pages 84–93. https://doi.org/10.18653/v1/D19-6110

Zhang, Pei, Xueying Xu, and Deyi Xiong. 2018. Active learning for neural machine translation. In *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, pages 153–158. https://doi.org/10.1109/IALP.2018.8629116