# Empirical Evaluation of Active Learning Techniques for Neural MT

**Xiangkai Zeng**[1*] **Sarthak Garg**[2] **Rajen Chatterjee**[2] **Udhyakumar Nallasamy**[2] **Matthias Paulik**[2]
[1]Carnegie Mellon University
[2]Apple Inc.
xiangkaiz@cs.cmu.edu,{sarthak_garg, rajen_c, udhay, mpaulik}@apple.com

## Abstract

Active learning (AL) for machine translation (MT) has been well-studied for the phrase-based MT paradigm. Several AL algorithms for data sampling have been proposed over the years. However, given the rapid advancement in neural methods, these algorithms have not been thoroughly investigated in the context of neural MT (NMT). In this work, we address this missing aspect by conducting a systematic comparison of different AL methods in a simulated AL framework. Our experimental setup to compare different AL methods uses: i) State-of-the-art NMT architecture to achieve realistic results; and ii) the same dataset (WMT'13 English-Spanish) to have fair comparison across different methods. We then demonstrate how recent advancements in unsupervised pre-training and paraphrastic embedding can be used to improve existing AL methods. Finally, we propose a neural extension for an AL sampling method used in the context of phrase-based MT - Round Trip Translation Likelihood (RTTL). RTTL uses a bidirectional translation model to estimate the loss of information during translation and outperforms previous methods.

## 1 Introduction

Active learning (AL) is an iterative supervised learning procedure where the learner is able to query an oracle for labeling new data points. Since the learner chooses the data points for annotation, the amount of labeling needed to learn a concept can be much lower than annotating the whole unlabeled dataset (Balcan et al., 2009). This approach is useful in low-resource scenarios where unlabeled data is abundant but manual labeling is expensive. AL has been successfully applied to many areas of NLP like classification, sequence labeling, spoken language understanding (Cohn et al., 1994; Guo and Greiner, 2007; Dagan and Engelson, 1995; Settles and Craven, 2008; Tur et al., 2005) as well as machine translation (MT) (Ambati, 2011; Haffari et al., 2009; Eck, 2008; Peris and Casacuberta, 2018; Zhang et al., 2018). In MT, most of the AL methods have been investigated under the phrase-based paradigm. Although neural MT has dominated the field (Barrault et al., 2019), there has only been limited effort to investigate and compare existing AL algorithms in this newer paradigm. The few recently published papers in this direction (Peris and Casacuberta, 2018; Zhang et al., 2018) use LSTM-based MT systems, whereas, the latest state-of-the-art systems are based on the Transformer architecture (Vaswani et al., 2017). Moreover, these papers either investigate different algorithms of the same class or compare only a handful of methods from different classes. Thus a global picture showing the effect of different AL methods on the same dataset for the state-of-the-art (SotA) MT system has been missing.

In this work, we fill this missing gap by performing a comprehensive evaluation of different AL algorithms on a publicly available dataset (WMT'13) using the SotA NMT architecture. To make our analysis thorough, we take into account different evaluation metrics to avoid any bias arising because of similarity between the evaluation metric and some components of the AL algorithm. Finally, we propose two extensions of existing AL algorithms. One leverages recent advances in paraphrastic embeddings (Wieting and Gimpel, 2018) and other is based on round-trip translation - a neural variant of the approach proposed in phrase-based MT (Haffari et al., 2009). Both of these approaches outperform existing methods with the latter showing the best results.

[*] Work done during internship at Apple Inc.



*Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo)*, pages 84–93
Hong Kong, China, November 3, 2019. 
https://doi.org/10.18653/v1/P17

## 2 Active Learning Framework

We simulate AL in a batch setup because it is more practical to send batches of data for manual translation rather than a single sentence at disjoint intervals of time. Algorithm 1 summarizes the procedure. It expects: i) a labeled parallel corpus ($\mathcal{L}$), which is used to train the NMT system ($\mathcal{M}$) ; ii) an unlabeled monolingual corpus ($\mathcal{U}$), which is used to sample new data points for manual translation; iii) a scoring function ($\psi$), which is used to estimate the importance of data points in ($\mathcal{U}$); and iv) batch size ($B$), which indicates the number of data points to sample in each iteration.[1] In practice, the AL algorithm will iterate until we exhaust the budget for annotation (step 2). However, in our simulation we already have reference translations for all the unlabeled data points (see Footnote 1), therefore, we iterate until we exhaust all the data points in $\mathcal{U}$. In each iteration, we first train an NMT system from scratch using $\mathcal{L}$ (step 3). We then score all the sentences in $\mathcal{U}$ with $\psi$ that takes in to account $\mathcal{L}, \mathcal{U}$, and $\mathcal{M}$ (step 4-6). The $\psi$ function is a key component in all AL algorithms, which is discussed in detail along with its variants in the next section. We then select the highest scoring $\mathcal{B}$ sentences for manual translation (step 7-8). These sentences are removed from $\mathcal{U}$ (step 9), and added to $\mathcal{L}$ along with their reference translations (step 10). The algorithm then proceeds to step 2 for the next round.

**Algorithm 1** Batch Active Learning for NMT

1: **Given:** Parallel data $\mathcal{L}$, Monolingual source language data $\mathcal{U}$, Sampling strategy $\psi(\cdot)$, Sampling batch size $\mathcal{B}$.
2: **while** Budget $\neq$ EMPTY **do**
3: $\quad \mathcal{M} = TrainNMTsystem(\mathcal{L})$ ;
4: $\quad$ **for** $x \in \mathcal{U}$ **do**
5: $\quad\quad f(x) = \psi(x, \mathcal{U}, \mathcal{L}, \mathcal{M})$;
6: $\quad$ **end for**
7: $\quad X_B = TopScoringSamples(f(x), \mathcal{B})$;
8: $\quad Y_B = HumanTranslation(X_B)$ ;
9: $\quad \mathcal{U} = \mathcal{U} - X_B$;
10: $\quad \mathcal{L} = \mathcal{L} \cup \{X_B, Y_B\}$ ;
11: **end while**
12: **return** $\mathcal{L}$

[1] In our simulation, $\mathcal{U}$ is basically the source side of a parallel corpus $\mathcal{L}'$ ($\mathcal{L}' \neq \mathcal{L}$), and to label new data points from $\mathcal{U}$ we simply extract the corresponding references from $\mathcal{L}'$ rather than asking a human annotator.

## 3 Methodology

In this section we outline the AL methods - i.e. the scoring functions ($\psi$), which have been proposed to work best for NMT, SMT and various sequence labeling tasks (Peris and Casacuberta, 2018; Zhang et al., 2018; Ambati, 2011; Haffari et al., 2009; Settles and Craven, 2008). These approaches can be broadly categorized into two classes: model-driven and data-driven.

Model-driven approaches sample instances based on the model, the labeled dataset and the unlabeled dataset, i.e. $\psi(x, \dots) = \psi(x, \mathcal{M}, \mathcal{U}, \mathcal{L})$. These methods receive direct signal from the model, which can potentially help in sampling more sentences from regions of the input space, where the model is weak. We first describe several model-driven approaches from the above works, all of which sample instances where the model $\mathcal{M}$ is least certain about the prediction. We then propose Round Trip Translation Likelihood, a neural extension of an existing method, which outperforms other model-driven methods substantially.

Data-driven approaches on the other hand only rely on $\mathcal{U}$ and $\mathcal{L}$ to sample sentences, i.e. $\psi(x, \dots) = \psi(x, \mathcal{U}, \mathcal{L})$. Since these methods are model independent, model training in step 3 of Algorithm 1 can be ignored, making these methods computationally faster. We summarize various existing data-driven approaches from MT literature and demonstrate how these approaches can benefit considerably from sentence embeddings specifically trained for capturing semantic similarity.

### 3.1 Model-Driven

In this class of methods, we explore uncertainty sampling (Lewis and Catlett, 1994) strategies that have been widely used in MT. In this strategy an unlabeled example $x$ is scored with some measure of uncertainty in the probability distribution over the label classes assigned by the model $p_{\mathcal{M}}(y|x)$. In the case of classification tasks, using the entropy $H(p_{\mathcal{M}}(y|x))$ is the most obvious choice, but in the case of structure prediction tasks, the space of all possible labels is usually exponential, making entropy calculation intractable. Settles and Craven (2008) found two heuristics: Least Confidence and N-best Sequence Entropy, which seemed to be the most effective estimators of model uncertainty across for two sequence labeling tasks. In addition to these, we also investigate Coverage Sampling (Peris and Casacuberta, 2018)

proposed for interactive NMT, and our version of Round Trip Translation Likelihood inspired from the work in phrase-based MT (Haffari et al., 2009).

#### 3.1.1 Least Confidence (LC)

This method estimates the model uncertainty of a source sentence $x$ by averaging token-level log probability of the corresponding decoded translation $\hat{y}$. In our formulation, we further add length normalization to avoid any bias towards the length of the translations.

$$\psi_{\text{LC}}(x, \mathcal{M}) = -\frac{1}{L} \log p_{\mathcal{M}}(\hat{y}|x), \quad (1)$$

where $L$ denotes the length of $\hat{y}$.

#### 3.1.2 N-best Sequence Entropy (NSE)

Another tractable approximator of model uncertainty is computing the entropy of the $n$-best hypothesis. Corresponding to a source sentence $x$, let $\mathcal{N} = \{\hat{y}_1, \hat{y}_2 \dots \hat{y}_n\}$ denote the set of $n$-best translations. The normalized probability $\hat{P}$ of each hypothesis can be computed as:

$$\forall \hat{y} \in \mathcal{N}, \quad \hat{P}(\hat{y}) = \frac{p_{\mathcal{M}}(\hat{y}|x)}{\sum_{\hat{y} \in \mathcal{N}} p_{\mathcal{M}}(\hat{y}|x)}. \quad (2)$$

Each source sentence is scored with the entropy of the probability distribution $\hat{P}$:

$$\psi_{\text{NSE}}(x, \mathcal{M}) = -\sum_{\hat{y} \in \mathcal{N}} \hat{P}(\hat{y}) \log \hat{P}(\hat{y}). \quad (3)$$

#### 3.1.3 Coverage Sampling (CS)

Under-translation is a well known problem in NMT (Tu et al., 2016), wherein not all source tokens are translated during decoding. The attention mechanism in LSTM based encoder-decoder architecture (Bahdanau et al., 2015) can model word alignment between translation and source to some degree. The extent of coverage of the attention weights over the source sentence can be an indicator of the quality of the translation. Peris and Casacuberta (2018) proposed Coverage Sampling (CS), which uses this coverage to estimate uncertainty. Formally:

$$\psi_{\text{CS}}(x, \mathcal{M}) = -\frac{\sum_{j=1}^{|x|} \log(\min(\sum_{i=1}^{|\hat{y}|} \alpha_{i,j}, 1))}{|x|} \quad (4)$$

where $x$ and $\hat{y}$ are the source sentence and the decoded translation respectively, $|\cdot|$ denotes the number of tokens and $\alpha_{i,j}$ denotes the attention probability on the $j^{th}$ word of $x$ while predicting the $i^{th}$ word of the $\hat{y}$.

#### 3.1.4 Round Trip Translation Likelihood (RTTL)

Ott et al. (2018) showed that even a well trained NMT model does not necessarily assign higher probabilities to better translations. This behavior can be detrimental for methods like **LC** in which sentences with highly probable translations are not selected for human translations. In this scenario we assume that a low quality translation will lose some source-side information and it will become difficult to reconstruct the original source from this translation. To this end, we train models $\mathcal{M}$ and $\mathcal{M}_{rev}$ to translate from source language to target language and the reverse direction respectively. $\mathcal{M}_{rev}$ is identical to $\mathcal{M}$ except that it is trained on data obtained by flipping source and target sentences in $\mathcal{L}$. Formally, for any source sentence $x$ of length $L$, we first translate it to a target sentence $\hat{y}$ using $\mathcal{M}$. Then we translate $\hat{y}$ back using $\mathcal{M}_{rev}$, but instead of decoding, we compute the probability of the original source sentence $x$ and use it as a measure of uncertainty.

$$\hat{y} \approx \underset{y}{\text{argmax}}\, p_{\mathcal{M}}(y|x). \quad \text{(beam search)} \quad (5)$$

$$\psi_{\text{RTTL}}(x, \mathcal{M}, \mathcal{M}_{rev}) = -\frac{1}{L} \log p_{\mathcal{M}_{rev}}(x|\hat{y}). \quad (6)$$

RTTL is inspired by one of the methods proposed by Haffari et al. (2009), but differs from it in terms of modeling uncertainty. In their formulation, $x$ is first translated to $\hat{y}$ like us but instead of scoring the likelihood of $x$ given $\hat{y}$, under $\mathcal{M}_{rev}$, they use $\mathcal{M}_{rev}$ to translate $\hat{y}$ to a new source sentence $\hat{x}$ and measure uncertainty using sentence-level BLEU between $x$ and $\hat{x}$. They showed that their approach did not perform better than a random baseline, however, in our experiments, RTTL outperforms the random baseline as well as all other model-driven methods. We suspect that this might be due to model log probability being a much finer grained metric than sentence-level BLEU.

### 3.2 Data-Driven

The data-driven approaches usually score sentences based on optimizing either one or a trade-

off between the following two metrics:

- Density: This metric scores sentences based on how similar they are with respect to the entire data in $\mathcal{U}$. In other words, sentences with higher likelihood under the data distribution of $\mathcal{U}$ are scored higher. This strategy assumes that the test set has the same distribution as $\mathcal{U}$, which makes achieving good translations on the *dense regions* of $\mathcal{U}$ more important.
- Diversity: This metric compliments the above and encourages sampling sentences which are less similar to the data in $\mathcal{L}$. This eventually leads to $\mathcal{L}$ containing a *diverse* set of sentences, leading to better generalization performance of model $\mathcal{M}$.

A key component in the above two metrics is how the similarity between two sentences is measured. We select the two common practices in literature are using n-gram overlap and cosine similarity between sentence embeddings. In the sections below, we describe the formulation of various data-driven methods based on how sentence similarity is measured.

#### 3.2.1 N-gram Overlap

Ambati (2011) and Eck (2008) investigated density and diversity metrics using n-gram overlap for phrase-based MT and concluded that the best approach is to combine both of them together in the scoring function. Therefore, we select Density Weighted Diversity method from the former and Static Sentence Sorting from the latter in our study. Both methods use the following notations:

- $\mathcal{I}$: denotes the indicator function,
- n-gram($x$): denotes the multiset of n-grams in a sentence (or a set of sentence) $x$,
- #($a|\mathcal{X}$): denotes the frequency of an n-gram $a$ in n-gram($\mathcal{X}$).

**Density Weighted Diversity (DWDS)** combines the density and diversity metrics using a harmonic mean. Equation 7 and 8 respectively define the density ($\alpha$) and diversity ($\beta$) metrics, which are combined together in Equation 9 to obtain the DWDS scoring function.

$$\alpha(x,\mathcal{U},\mathcal{L}) = \frac{\sum_{s\in\text{n-gram(x)}} \#(s|\mathcal{U})e^{-\lambda\#(s|\mathcal{L})}}{|\text{n-gram}(x)||\text{n-gram}(\mathcal{U})|} \quad (7)$$

Here, $\lambda$ is used as a decay parameter to give discount the n-grams which have already been seen in the bilingual data.

$$\beta(x,\mathcal{U},\mathcal{L}) = \frac{\sum_{x\in\text{n-gram}(x)} \mathcal{I}(s \notin \text{n-gram}(\mathcal{L}))}{|\text{n-gram}(x)|} \quad (8)$$

$$\psi_{\text{DWDS}}(x,\mathcal{U},\mathcal{L}) = \frac{\alpha(x,\mathcal{U},\mathcal{L})\beta(x,\mathcal{U},\mathcal{L})}{k\alpha(x,\mathcal{U},\mathcal{L}) + \beta(x,\mathcal{U},\mathcal{L})} \quad (9)$$

Here, $k$ controls the relative weighting of $\alpha$ and $\beta$.

**Static Sentence Sorting (SSS)** is a much simpler formulation which samples sentences from dense regions of $\mathcal{U}$, containing n-grams which are absent in $\mathcal{L}$.

$$\psi_{\text{SSS}}(x,\mathcal{U},\mathcal{L}) = \frac{\sum_{s\in\text{n-gram}(x)} \mathcal{I}(s \notin \mathcal{L})\#(s|\mathcal{U})}{|x|} \quad (10)$$

#### 3.2.2 Cosine Similarity

Zhang et al. (2018) proposed **S-Score (SS)** to use cosine similarity between sentence embeddings rather than n-gram overlap as a measure of sentence similarity. S-Score mainly relies on the diversity metric for selection. It samples sentences from $\mathcal{U}$ which are furthest from their nearest neighbors in $\mathcal{L}$. Essentially sentences which are semantically different from all the sentences in $\mathcal{L}$ would be selected. Let $\mathbf{e}(x)$ denote the embedding vector of the sentence $x$ and $cos(\cdot,\cdot)$ denote the cosine similarity, then S-Score is defined as:

$$\psi_{SS}(x,\mathcal{L}) = \min_{y\in\mathcal{L}} cos(\mathbf{e}(x),\mathbf{e}(y)) \quad (11)$$

Zhang et al. (2018) used learnt sentence embeddings starting from `fasttext` (Bojanowski et al., 2017) and fine-tuned using Paragraph Vector (Le and Mikolov, 2014).

To better understand how recent advances in unsupervised pre-training can benefit active learning, we perform an ablation study of the S-Score method with varying the source of embeddings. We experiment with the following three increasingly expressive sentence representations:

**Bag of words (SS-BoW):** This is the simplest method in which the sentence embeddings are computed by taking the average of all the word embeddings. The word embeddings are obtained from the `fasttext` tool.

**Contextual embedding (SS-CE):** In this method, we leverage unsupervised pre-training techniques like BERT which have significantly advanced the SotA in NLP (Devlin et al., 2019). Specifically, we train the Transformer Encoder using the Masked Language Modeling (MLM) objective proposed in BERT (Devlin et al., 2019). We then compute the sentence embedding by averaging outputs from the trained encoder corresponding to all the sentence tokens.

**Paraphrastic embedding (SS-PE):** The sentence embedding methods listed above and those used by Zhang et al. (2018) are all trained with the objective of predicting tokens based on their context. While this allows the embeddings to be somewhat correlated with the semantics, explicitly fine-tuning the embeddings on semantic similarity can be helpful for our use case. Therefore, we fine-tune the contextual embedding model discussed above on the paraphrase task as proposed in Wieting and Gimpel (2018).

Wieting and Gimpel (2018) created a dataset[2]containing pairs of English paraphrases by back-translating the Czech side of an English-Czech corpus. We fine-tune the embeddings of the paraphrase pairs to be close to each other using a contrastive loss. We specifically choose this task because it does not utilize any supervised human annotated corpus for semantic similarity while achieving competitive performance on SemEval semantic textual similarity (STS) benchmarks.

We show that using contextual sentence embeddings does not give any noticeable gains over simply using bag of words embeddings, however fine-tuning the embeddings on semantic similarity tasks improves the performance of S-Score substantially, enabling it to outperform other data-driven approaches.

## 4 Experiments

### 4.1 Dataset

Our setup is based on the WMT'13 English-Spanish news translation task. We use the Europarl and News Commentary Corpus consisting of $\sim$ 2M sentence pairs. We randomly sample 10% of the whole bilingual data to create the base parallel dataset $\mathcal{L}$ ($\sim$ 200K) which is used to train an initial NMT model. We then randomly sample 50% from the remaining data to the unlabeled dataset $\mathcal{U}$ ($\sim$ 1M) used for simulating the AL experiments. Note that we do the random sampling just once and fix $\mathcal{L}$ and $\mathcal{U}$ for all the experiments for fair comparison. Since we experiment in a simulated AL framework, the target sentences in $\mathcal{U}$ are hidden while scoring source sentences with different AL strategies. Once the AL algorithm samples a batch $B$ containing 100k source sentences from $\mathcal{U}$, the sampled sentences along with their corresponding "hidden" translations are added to $\mathcal{L}$. We use `newstest-2012` as the validation set and `newstest-2013` as the test set, each consisting of about 3000 sentence pairs. For training the contextual embeddings, we use the English News Crawl corpus from years 2007-17, consisting of $\sim$ 200M sentences. For preprocessing, we apply the Moses tokenizer (Koehn et al., 2007) without aggressive hyphen splitting and with punctuation normalization. We learn a joint source and target Byte-Pair-Encoding (BPE, Sennrich et al. (2016)) on the whole bilingual data with 32k merge operations.

### 4.2 Training and Model Hyperparameters

For the NMT models in all the experiments, we use the `base` Transformer configuration with 6 encoder and decoder layers, 8 attention heads, embedding size of 512, shared input and output embeddings, `relu` activation function and sinusoidal positional embeddings. We train with a batch size of 2000 tokens on 8 Volta GPUs using half-precision for 30 epochs. Furthermore we use Adam optimizer (Kingma and Ba, 2015) with a learning rate of $0.001, \beta_1 = 0.9, \beta_2 = 0.98$, learning rate warmup over the first 4000 steps and inverse square root learning rate scheduling. We also apply dropout and label smoothing of 0.1 each. We average the weights of 5 checkpoints with the best validation loss and run inference with a beam size of 5.

While training the Transformer Encoder using the Masked Language Modeling (MLM) objective, we switch to the `big` configuration with an embedding size of 1024 and 16 attention heads. The masking probabilities are the same as described in Devlin et al. (2019), however instead of pair of sentences, we use text streams spanning across multiple sentences (truncated at 256 tokens) (Lample and Conneau, 2019). This model

[2] https://drive.google.com/file/d/19NQ87gEFYu3zOIp_VNYQZgmnwRuSIyJd/view?usp=sharing

is trained using 64 Volta GPUs, each processing a batch of 128 segments. We use a learning rate of 0.0003, with other hyperparameters similar to the NMT model. The above model is fine-tuned on the task of Paraphrastic Sentence Embeddings. Specifically we use a margin of 0.4, learning rate 0.0005, batch size 3500 tokens and megabatches consisting of 32 batches. We train for 5 epochs using 8 Volta GPUs. Lastly for DWDS, we set $k = 1, \lambda = 1$. For both the n-gram based methods, we consider up to tri-grams. In case of NSE, we restrict the $n$-best list to be of size 5. Our baseline is a system that randomly selects sentences from the unlabeled data.

## 5 Results and Discussion

In this section, we compare the performance of different AL algorithms of each class: model-driven and data-driven. For a comprehensive comparison, we evaluate the best approaches of both classes on:

- **Various MT evaluation metrics**. N-gram overlap based metrics like BLEU (Papineni et al., 2002) might be biased towards AL methods based on n-gram similarity (DWDS, SSS). For a fair comparison, we evaluate the AL approaches on BLEU, TER (Snover et al., 2006), which is based on edit distance, and BEER (Stanojević and Sima'an, 2015), which uses a linear model trained on human evaluations dataset.

- **Out-of-domain evaluation sets**. Since AL algorithms are sensitive to the labeled ($\mathcal{L}$) and unlabeled data ($\mathcal{U}$) distributions, it is possible that some AL algorithms perform worse, when evaluated on out-of-domain test sets. To compare the robustness of different AL approaches, we evaluate them on a test set sourced from biological domain, which is very different from the training data distribution (parliament speech and news).

- **Evaluation sets without any translationese source sentences**. Translationese refers to the distinctive style of sentences which are translated by human translators. These sentences tend to be a simplistic, standardized, explicit and lack complicated constructs like idioms etc. These properties might make it easy to reconstruct source sentences from the translationese domain, hence discouraging them to be sampled by RTTL. Presence of translationese source sentences in the test sets might unfairly penalize RTTL.

### 5.1 Model-Driven Approaches

Figure 1 and Table 1 compares the results of model-driven approaches and random sampling baseline. We observe that CS performs worse than the random baseline. This is in contrast to the results reported in Peris and Casacuberta (2018) where it is amongst the best performing methods. The performance of CS is dependent upon the assumption that attention probabilities are good at modeling word alignments. While this assumption is valid in the case Peris and Casacuberta (2018), which uses attentional sequence-sequence models with LSTMs/GRUs, it breaks down in the presence of multi-layered, multi-headed attention mechanism of the Transformer architecture. Upon closer inspection, we found that this method sampled very long sentences with rare words and many BPE splits, resulting in sub-optimal performance. LC has slightly better performance than NSE, while RTTL outperforms all the other methods consistently by a non-trivial margin. This demonstrates that our proposed extension is an effective approximation of model uncertainty.

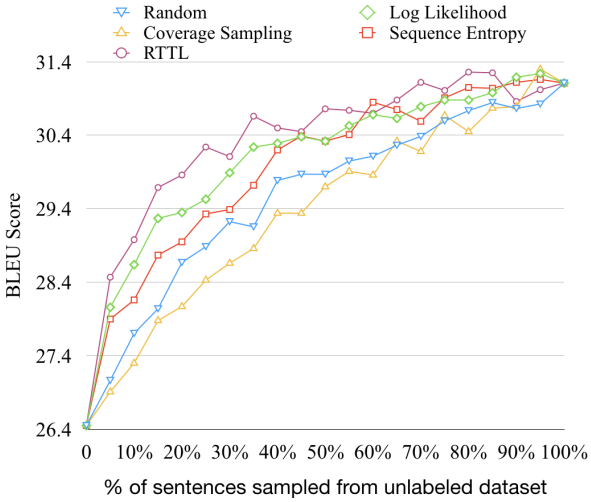


Figure 1: Results of model-driven approaches.

### 5.2 Data-Driven Approaches

Figure 2 shows the results of n-gram based data-driven approaches. As illustrated in Figure 2, these computationally inexpensive methods can consistently outperform the random baseline by a

Table 1: Results for Model-Driven and Data-Driven approaches. We report the average BLEU scores across 20 active learning iterations. Best methods within each category are boldened.

| Random | Model-Driven | | | | Data-Driven (N-Gram) | | Data-Driven (Embedding) | | |
|---|---|---|---|---|---|---|---|---|---|
| | LC | CS | NSE | RTTL | DWDS | SSS | SS-BoW | SS-CE | SS-PE |
| 29.54 | 30.06 | 29.35 | 29.93 | **30.29** | 30.22 | 30.21 | 29.84 | 30 | **30.17** |

large margin. In spite of modeling density and diversity in very different ways, both the methods achieve similar performance.

Figure 3 shows the results of embedding based data-driven approaches (SS) corresponding to different sources of embeddings. It is noteworthy that using bag-of-words (SS-BoW) and contextual embeddings (SS-CE) results in roughly the same performance, barely beating the random baseline. However, fine-tuning the contextual embeddings on the paraphrase task (SS-PE), brings about a large performance gain, emphasizing the effectiveness of fine-tuning on semantic similarity tasks for AL.

The above trends are inline with the results reported in Table 1 as well.

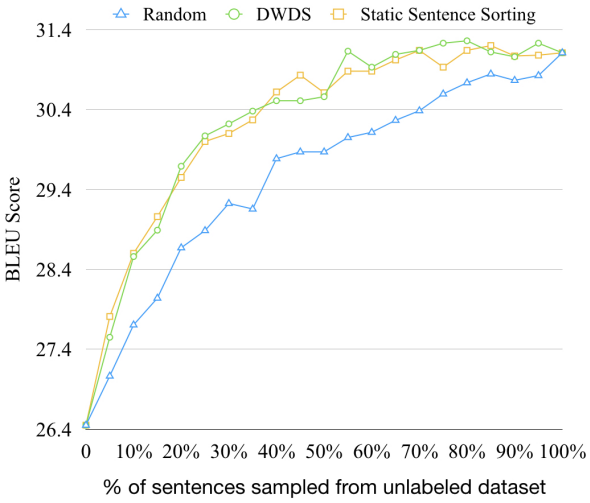


Figure 2: Results of data-driven approaches based on n-gram similarity.

### 5.3 Performance on Different Evaluation Metrics

Figure 4 compares the top three AL methods (RTTL, DWDS, SS-PE) using BLEU. All three methods are quite competitive, with RTTL and SS-PE performing slightly better than DWDS in the beginning. Figures 4 and 5 show consistent performance trends using all the three metrics. It is worth noting from figure 4, that all the methods

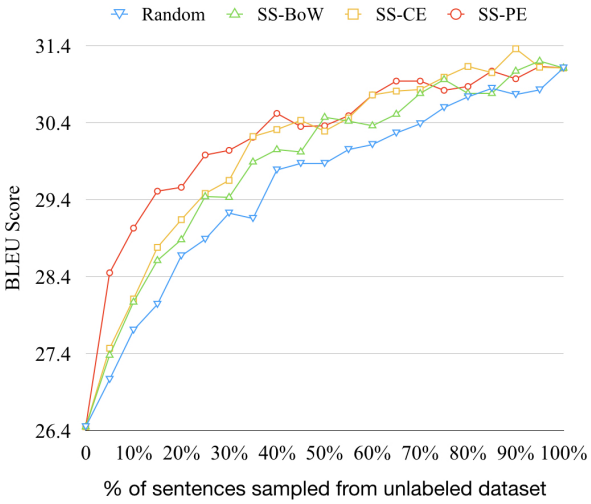


Figure 3: Results of the SS method with various sources of embeddings.

are able to achieve the same BLEU (as with using the whole bitext) with using only 70% of the bitext. This outlines that AL can be quite effective for NMT.

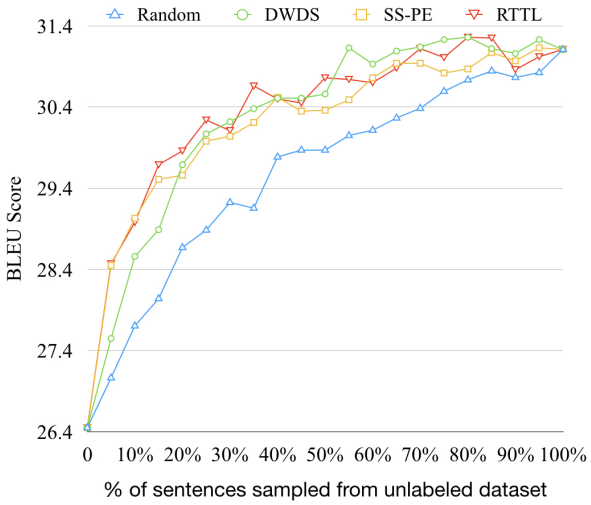


Figure 4: Results of best performing approaches with BLEU.

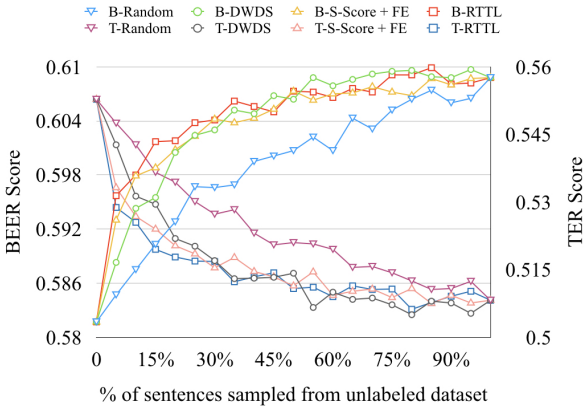


Figure 5: Results of best performing approaches with BEER (B-X) and TER (T-X) evaluation metrics.

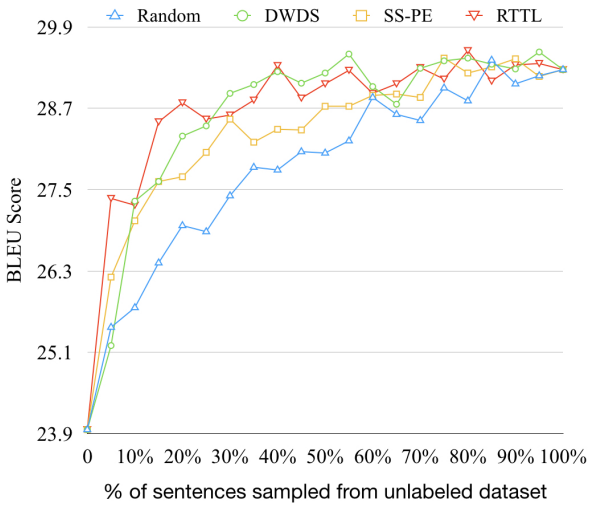


Figure 6: Results of the best performing approaches on Biomedical Translation test set

### 5.4 Evaluation on Test Sets Without Translationese

Zhang and Toral (2019) brought to light the effect of having translationese in the source side of the evaluation sets used in WMT over the years. The situation is even worse for `newstest2013` which contains translations of sentences from 5 different languages, in the source side. We create a new test set, by collecting $\sim$ 2000 sentences from `newstest2009-13` except `newstest2012` (since we use `newstest2012` as validation set) which are originally from English. The BLEU scores of all the methods are much higher on this test set corrected for translationese ($\sim$38 BLEU), as compared to `newstest2013` ($\sim$31 BLEU), however the relative performance trends remain the same.

### 5.5 Evaluation on Out-of-Domain Test Sets

Figure 6 shows the results on out-of-domain test set from the WMT16 Shared Task on Biomedical Translation. It can be observed from figure 6 that, while RTTL and DWDS are quite robust to the test domain, and strongly outperform the random baseline, there is some degradation in the performance of SS-PE.

## 6 Related Work

Early work on AL for MT includes Ambati (2011); Eck (2008); Haffari et al. (2009) among others. These papers investigated the AL approaches for phrase-based MT systems. Given that the current SotA MT systems are neural-based, in this work, we investigate the effectiveness of their proposed methods in the neural paradigm. Couple of works that did investigate the AL methods for NMT are Peris and Casacuberta (2018) and Zhang et al. (2018). Both of these used RNN/LSTM based NMT architecture, whereas, we use the latest SotA Transformer in our investigation. Peris and Casacuberta (2018) used an interactive NMT setup and mostly focused on model-driven approaches disregarding data-driven methods. Zhang et al. (2018) did compare methods from both the classes but considered only a handful of methods. Our work is closer to Zhang et al. (2018), but we cover much a wider spectrum of methods in AL. We also go one step further and show that the cosine similarity based methods proposed in Zhang et al. (2018) are more effective when the embeddings are optimized for the paraphrase task. As far as we know, most of the prior work concluded that data-driven methods outperform model-driven methods, however, our model-driven RTTL formulation obtains slight gain over the best data-driven method.

## 7 Conclusion

In this work, we performed an empirical evaluation of different AL methods for the state-of-the-art neural MT architecture, a missing aspect in prior work. We explored two classes of approaches: data-driven and model-driven, and observed that all the methods outperform a random baseline, except coverage sampling which relies on the attention mechanism. Coverage sampling was shown to be amongst the best approaches in prior work that used LSTM-based NMT model. Given Transformer's more complex attention ar-

chitecture (multi-headed and multi-layered), it appears that the attention scores are not reliable enough to be used with the AL methods.

From our ablation study on using different sources of embeddings, we discovered that optimizing the embeddings towards a semantic similarity task can give significant performance improvements in data-driven AL methods. Also, for the first time, we observed that a model-driven approach can outperform data-driven methods. The improvement was more evident in the out-of-domain evaluation results. This was possible with our proposed neural extension - RTTL, which computes the likelihood score of re-constructing the original source from its translation using a reverse translation model. Overall, we observed that the performance trends of different AL methods were consistent with all the three evaluation metrics (BLEU, BEER, and TER) and on different evaluation sets (in-domain and out-of-domain).

## Acknowledgments

We would like to thank Tom Gunter, Andrew Finch, Russ Webb and the anonymous reviewers for their helpful comments. Many thanks to the Siri Machine Translation Team for helpful discussions and support.

## References

Vamshi Ambati. 2011. *Active learning and crowdsourcing for machine translation in low resource scenarios*. Ph.D. thesis, Carnegie Mellon University.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, San Diego, CA, USA.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. 2009. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89.

Loïc Barrault, 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*, pages 128–188. Association for Computational Linguistics.

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

David Cohn, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine Learning - Special issue on structured connectionist systems*, 15(2):201–221.

Ido Dagan and Sean P Engelson. 1995. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier, Tahoe City, California, USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, 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, Minneapolis, Minnesota.

Matthias Eck. 2008. *Developing deployable spoken language translation systems given limited resources*. Ph.D. thesis, Verlag nicht ermittelbar.

Yuhong Guo and Russell Greiner. 2007. Optimistic active-learning using mutual information. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, volume 7, pages 823–829, Hyderabad, India.

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

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, San Deigo, CA, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. pages 177–180, Prague, Czech Republic.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, Beijing, China.

David D Lewis and Jason Catlett. 1994. Heterogenous uncertainty sampling for supervised learning.

In *Machine Learning Proceedings 1994: Proceedings of the Eighth International Conference*, pages 148–156, New Brunswick, NJ, USA. Morgan Kaufmann Publishers Inc.

Myle Ott, Michael Auli, David Grangier, et al. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3953–3962, Stockholm, Sweden.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. pages 311–318, Philadelphia, PA, USA.

Álvaro Peris 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*, pages 151–160, Brussels, Belgium.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. pages 1715–1725, Berlin, Germany.

Burr Settles 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, Waikiki, Honolulu, Hawaii.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts.

Miloš Stanojević and Khalil Sima'an. 2015. Beer 1.1: Illc uva submission to metrics and tuning task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 396–401, Lisbon, Portugal.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 76–85.

Gokhan Tur, Dilek Hakkani-Tür, and Robert E Schapire. 2005. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. pages 1–11, Long Beach, CA, USA.

John Wieting 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, Melbourne, Australia.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. *arXiv preprint arXiv:1906.08069*.

Pei Zhang, Xueying Xu, and Deyi Xiong. 2018. Active learning for neural machine translation. In *2018 International Conference on Asian Language Processing (IALP)*, pages 153–158, Bandung, Indonesia. IEEE.