Predicting Human-Targeted Translation Edit Rate via Untrained Human Annotators

Omar F. Zaidan and **Chris Callison-Burch**
Dept. of Computer Science, Johns Hopkins University
Baltimore, MD 21218, USA
{ozaidan,ccb}@cs.jhu.edu

Abstract

In the field of machine translation, automatic metrics have proven quite valuable in system development for tracking progress and measuring the impact of incremental changes. However, human judgment still plays a large role in the context of *evaluating* MT systems. For example, the GALE project uses human-targeted translation edit rate (HTER), wherein the MT output is scored against a post-edited version of itself (as opposed to being scored against an existing human reference). This poses a problem for MT researchers, since HTER is not an easy metric to calculate, and would require hiring and training human annotators to perform the editing task. In this work, we explore soliciting those edits from *untrained* human annotators, via the online service Amazon Mechanical Turk. We show that the collected data allows us to predict HTER-ranking of documents at a significantly higher level than the ranking obtained using automatic metrics.

1 Introduction

In the early days of machine translation (MT), it was typical to evaluate MT output by soliciting judgments from human subjects, such as evaluating the fluency and adequacy of MT output (LDC, 2005). While this approach was appropriate (indeed desired) for evaluating a system, it was not a practical means of tracking the progress of a system during its development, since collecting human judgments is both costly and time-consuming. The introduction of automatic metrics like BLEU contributed greatly to MT research, for instance allowing researchers to measure and evaluate the impact of small modifications to an MT system.

However, manual evaluation remains a core component of system evaluation. Teams on the GALE project, a DARPA-sponsored MT research program, are evaluated using the HTER metric, which is a version of TER whereby the output is scored against a post-edited version of itself, instead of a preexisting reference. Moreover, emphasis is placed on performing well across all documents and across all genres. Therefore, it is important for a research team to be able to evaluate their system using HTER, or at least determine the ranking of the documents according to HTER, for purposes of error analysis. Instead of hiring a human translator and training them, we propose moving the task to the virtual world of Amazon's Mechanical Turk (AMT), hiring workers to edit the MT output and predict HTER from those edits. We show that edits collected this way are better at predicting document ranking than automatic metrics, and furthermore that it can be done at a low cost, both in terms of time and money.

The paper is organized as follows. We first discuss options available to predict HTER, such as automatic metrics. We then discuss the possibility of relying on human annotators, and the inherent difficulty in training them, before discussing the concept of soliciting edits over AMT. We detail the task given to the workers and summarize the data that we collected, then show how we can combine their data to obtain significanly better rank predictions of documents.

2 Human-Targeted TER

Translation edit rate (TER) measures the number of edits required to transform a hypothesis into an appropriate sentence in terms of grammaticality and meaning (Snover et al., 2006). While TER usually scores a hypothesis against an existing reference sentence, *human-targeted* TER scores a hypothesis against a post-edited version of itself.

While HTER has been shown to correlate quite well with human judgment of MT quality, it is quite challenging to obtain HTER scores for MT output, since this would require hiring and training human subjects to perform the editing task. Therefore, other metrics such as BLEU or TER are used as proxies for HTER.

2.1 Amazon's Mechanical Turk

The high cost associated with hiring and training a human editor makes it difficult to imagine an alternative to automatic metrics. However, we propose soliciting edits from workers on Amazon's Mechanical Turk (AMT). AMT is a virtual marketplace where "requesters" can post tasks to be completed by "workers" (aka *Turkers*) around the world. Two main advantages of AMT are the pre-existing infrastructure, and the low cost of completing tasks, both in terms of time and money. Data collected over AMT has already been used in several papers such as Snow et al. (2008) and Callison-Burch (2009).

When a requester creates a task to be completed over AMT, it is typical to have completed by more than one worker. The reason is that the use of AMT for data collection has an inherent problem with data quality. A requester has fewer tools at their disposal to ensure workers are doing the task properly (via training, feedback, etc) when compared to hiring annotators in the 'real' world. Those redundant annotations are therefore collected to increase the likelihood of at least one submission from a faithful (and competent) worker.

2.2 AMT for HTER

The main idea it to mimic the real-world HTER setup by supplying workers with the original MT output that needs to be edited. The worker is also given a human reference, produced independently from the MT output. The instructions ask the worker to modify the MT output, using as few edits as possible, to match the human reference in meaning and grammaticality.

The submitted edited hypothesis can then be used as the reference for calculating HTER. The idea is that, with this setup, a competent worker would be able to closely match the editing behavior of the professionally trained editor.

3 The Datasets

We solicited edits of the output from one of GALE's teams on the Arabic-to-English task. This MT output was submitted by this team and HTER-scored by LDC-hired human translators. Therefore, we already had the edits produced by a professional translator. These edits were used as the "gold-standard" to evaluate the edits solicited from AMT and to evaluate our methods of combining Turkers' submissions.

The MT output is a translation of more than 2,153 Arabic segments spread across 195 documents in 4 different genres: broadcast conversations (BC), broadcast news (BN), newswire (NW), and blogs (WB). Table 1 gives a summary of each genre's dataset.

Genre	# docs	Segs/doc	Words/seg
BC	40	15.8	28.3
BN	48	9.6	36.1
NW	54	8.7	39.5
WB	53	11.1	31.6

Table 1: The 4 genres of the dataset.

For each of the 2,153 MT output segments, we collected edits from 5 distinct workers on AMT, for a total of 10,765 post-edited segments by a total of about 500 distinct workers.[1] The segments were presented in 1,210 groups of up to 15 segments each, with a reward of $0.25 per group. Hence the total rewards to workers was around $300, at a rate of 36 post-edited segments per dollar (or 2.8 pennies per segment).

4 What are we measuring?

We are interested in predicting the ranking the documents according to HTER, not necessarily predicting the HTER itself (though of course attempting to predict the latter accurately is the cornerstone of our approach to predict the former). To measure the quality of a predicted ranking, we use Spearman's rank correlation coefficient, ρ, where we first convert the raw scores into ranks and then use the following formula to measure correlation:

$$\rho(X,Y) = 1 - \frac{6\sum_{i=1}^{n}(rank(x_i) - rank(y_i))^2}{n(n^2-1)}$$

[1] Data available at `http://cs.jhu.edu/~ozaidan/hter`.

where n is the number of documents, and each of X and Y is a vector of n HTER scores.

Notice that values for ρ range from –1 to 1, with +1 indicating perfect rank correlation, –1 perfect inverse correlation, and 0 no correlation. That is, for a fixed X, the best-correlated Y is that for which $\rho(X,Y)$ is highest.

5 Combining Tukers' Edits

Once we have collected edits from the human workers, how should we attempt to predict HTER from them? If we could assume that all Turkers are doing the task faithfully (and doing it adequately), we should use the annotations of the worker performing the least amount of editing, since that would mirror the real-life scenario.

However, data collected from AMT should be treated with caution, since a non-trivial portion of the collected data is of poor quality. Note that this does not necessarily indicate a 'cheating' worker, for even if a worker is acting in good faith, they might not be able to perform the task adequately, due to misunderstanding the task, or neglecting to attempt to use a small number of edits.

And so we need to combine the redundant edits in an intelligent manner. Recall that, given a segment, we collected edits from multiple workers. Some baseline methods include taking the minimum over the edits, taking the median, and taking the average.

Once we start thinking of averages, we should consider taking a *weighted* average of the edits for a segment. The weight associated with a worker should reflect our confidence in the quality of that worker's edits. But how can we evaluate a worker in the first place?

5.1 Self Verification of Turkers

We have available "gold-standard" editing behavior for the segments, and we treat a small portion of the segments edited by a Turker as a verification dataset. On that portion, we evaluate how closely the Turker matches the LDC editor, and weight them accordingly when predicting the number of edits of the rest of that group's segments. Specifically, the Turker's weight is the absolute difference between the Turker's edit count and the professional editor's edit count.

Notice that we are not simply interested in a worker whose edited submission closely matches the edited submission of the professional translator. Rather, we are interested in mirroring the professional translator's edit *rate*. That is, the closer a Turker's edit *rate* is to the LDC editor's, the more we should prefer the worker. This is a subtle point, but it is indeed possible for a Turker to have similar edit rate as the LDC editor but still require a large number of edits to get the LDC editor's submission itself.

6 Experiments

We examine the effectiveness of any of the above methods by comparing the resulting document ranking versus the desired ranking by HTER. In addition to the above methods, we use a baseline a ranking predicted by TER to a human reference. (For clarity, we omit discussion with other metrics such as BLEU and (TER–BLEU)/2, since those baselines are not as strong as the TER baseline.

6.1 Experimental Setup

We examine each genre individually, since genres vary quite a bit in difficulty, and, more importantly, we care about the internal ranking within each genre, to mirror the GALE evaluation procedure.

We examine the effect of varying the amount of data by which we judge a Turker's data quality. The amount of this "verification" data is varied as a percentage of the total available segments. Those segments are chosen at random, and we perform 100 trials for each point.

6.2 Experimental Results

Figure 1 shows the rank correlations for various methods across different sizes of verification subsets. Notice that some methods, such as the TER baseline, have horizontal lines, since these do not rate a Turker based on a verification subset.

It is worth noting that the oracle performs very well. This is an indication that predicting HTER accurately is mostly a matter of identifying the best worker. While oracle scenarios usually represent unachievable upper bounds, keep in mind that there are only a very small number of editors per segment (five, as opposed to oracle scenarios dealing with 100-best lists, etc).

Other than that, in general, it is possible to achieve very high rank correlation using Turkers' data, significantly outperforming the TER ranking, even with a small verification subset. The genres do vary quite a bit in difficulty for Turkers, with BC and especially NW being quite difficult, though in the case of NW for instance, this is due to the human reference doing quite well to begin with, rather than Turkers performing poorly.

7 Conclusions and Future Work

We proposed soliciting edits of MT output via Amazon's Mechanical Turk and showed we can predict ranking significantly better than an automatic metric. The next step is to explicitly identify undesired worker behavior, such as not editing the MT output at all, or using the human reference as is instead of editing the MT output. This can be detected by not limiting our verification to comparing behavior to the professional editor's, but also by comparing submitted edits to the MT output itself and to the human reference. In other words, a worker's submission could be characterized in terms of its distance to the MT output and to the human reference, thus building a complete 'profile' of the worker, and adding another component to guard against poor data quality and to reward the desired behavior.

Acknowledgments

This work was supported by the EuroMatrixPlus Project (funded by the European Commission), and by the DARPA GALE program under Contract No. HR0011-06-2-0001. The views and findings are the authors' alone.

References

Chris Callison-Burch. 2009. *Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazon's Mechanical Turk*. In Proceedings of EMNLP.

LDC. 2005. *Linguistic data annotation specification: Assessment of fluency and adequacy in translations*. Revision 1.5.

Matthew Snover, Bonnie J. Dorr, Richard Schwartz. 2006. *A Study of Translation Edit Rate with Targeted Human Annotation*. Proceedings of AMTA.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. *Cheap and fast – but is it good? Evaluating non-expert annotations for natural language tasks*. In Proceedings of EMNLP.

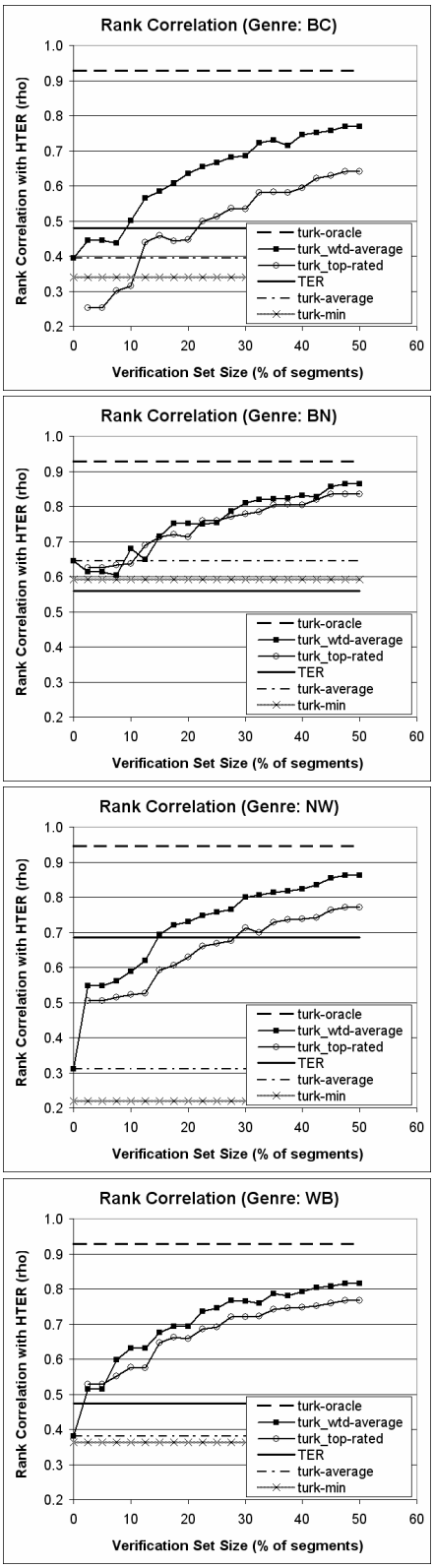

Figure 1: Rank correlation between predicted ranking and HTER ranking for different prediction schemes, across the four genres, and across various sizes of the worker verification set.