Cost Optimization in Crowdsourcing Translation: Low cost translations made even cheaper

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

Crowdsoucing makes it possible to solicit translation candidates for further selection in machine translation tasks at low cost. We proposed two mechanisms to make this process even cheaper while maintaining a high translation quality: ranking selection method and model selection method. For ranking selection method, we reduce cost by identifying bad translators after we solicit only several translations from each of them, keeping hiring top workers and selecting the best translation based their rankings. In model selection method, we train models to evaluate the quality of translation candidates and fit a threshold between acceptable and unacceptable translations. The total cost is optimized by reducing the redundant translations which can be pointed out by our model and the the threshold.

1 Introduction

Crowdsourcing is a promising new mechanism for collecting large volumes of annotated data at low cost. Platforms like Amazon Mechanical Turk (MTurk) provide researchers with access to large groups of people, who can complete 'human intelligence tasks' that are beyond the scope of current artificial intelligence. Since statistical natural language processing benefits from increased amount of labeled training data, many NLP researchers have focused on creating speech and language data through crowdsourcing (for example, Snow et al. (2008; Callison-Burch and Dredze (2010) and others). One NLP application that has been the focus of crowdsourced data collection is statistical machine translation (SMT) which requires large bilingual sentence-aligned parallel corpora to train translation models. Crowdsourcing's low costs has made i possible to hire people to create sufficient volumes of translation in order to train SMT systems.

However, crowdsourcing is not perfect, and one of its most pressing challenges is how to ensure the quality of the data that is created by it. Unlike more traditional employment mechanism, where our annotator are pre-vetted and their skills are attested for, in crowdsourcing very little is known about the annotators. They are not professional translators, and there are no built-in mechanisms for testing their language skills. They complete tasks without any oversight. Thus, translations produced via crowdousrcing may be low quality . Previous work has addressed this problem, showing that non-professional translators hired on Amazon Mechanical Turk (MTurk) can achieve professional-level quality, by soliciting multiple translations of each source sentence and then choosing the best translation (Zaidan and Callison-Burch, 2011).

In this paper we focus on a different aspect of crowdsourcing than Zaidan and Callison-Burch (2011). We attempt to achieve the same high quality while **minimizing the associated costs**. We reduce costs using two complementary methods: (1) We quickly identify and filter out workers who produce low quality translations. The goal is to reduce the number of worker we hire, and retain only high quality translators. (2) Instead of soliciting a fixed number of translations for each foreign sentence, we stop soliciting translations after we get an acceptable one.

We do so by building models to distinguish between acceptable translations and unacceptable ones. The goal is to reduces the number of independent translations that we solicit for each source sentence. Our work stands in contrast with Zaidan and Callison-Burch (2011) who had no model of annotator quality, and who always solicited and paid for a fixed number of translations of each source segment.

In this paper we demonstrate that

- Workers can be ranked by quality with high correlation against a gold standard ranking (ρ of 0.XXX), using logistic regression and a variety of features, or initially testing them using a small amount of calibration data with known professional translations.
- This ranking can be established after observing very small amounts of data (reaching ρ of 0.XXX after seeing only 10 translations from each worker), so bad workers can be filtered out quickly.
- Our models can predict whether a given translation is acceptable with high accuracy, subtantially reducing the number of redundant translations needed for every source segment.
- We can achieve a similar BLEU score as Zaidan and Callison-Burch (2011) at $\frac{1}{X}$ of the cost.

2 Previous work

We use the data collected by Zaidan and Callison-Burch (2011) through Amazon's Mechanical Turk (MTurk). MTurk is an online marketplace for work where workers (called Turkers) complete microtasks called Human Intelligence Tasks (HITs) in return for micropayments. Zaidan and Callison-Burch (2011) hired Turkers to translate 1792 Urdu sentences from the 2009 NIST Urdu-English Open Machine Translation Evaluation set.[1]. In each HIT, they posted 10 Urdu sentences to be translated. A total of 51 Turkers contributed translations.

Along with the translations, Zaidan and Callison-Burch (2011) also surveyed the Turkers, and collected self-reported language skills (what was their native language, how long they had spoken English and Urdu), and information about what country they lived in.

The Linguistics Data Consortium produced four sets of professional translations for each of the Urdu sentences in this set. This makes it possible to compare the Turkers' translation quality to professionals.

2.1 Professional quality from non-professionals

Zaidan and Callison-Burch (2011) used the features in order to try to select the best translation from among the 4 candidate translations, either by predicting the best translation on a sentence-by-sentence basis, or by trying to rank the Turkers and then taking the translation from the best translator of each sentence.

Zaidan and Callison-Burch (2011) extracted a number of features from the translations and workers' self-reported language skills in order to predict the best translations. These features included 9 sentence-level features:

- Language model features: we assign a log probability and a per-word perplexity score for each sentence, based on 5-gram language model trained on English Gigaword corpus.
- A Web *n*-gram log probability feature using Microsoft Web N-Gram Corpus, up to 5-grams.
- Geometric averages of Web *n*-grams.
- Sentence length features: we use the ratios of the length of the Urdu source sentence to the length of its English translation, and vice versa.
- Edit rate to other Turkers' translations of that sentence.

They also used 15 worker-level features that aggregate over the sentence-level features, plus features based on their language abilities and their location, and a set of 3 features based on a second-pass HIT where English speakers ranked the translations (average rank, % of time ranked best, % of time ranked better than others). Finally, they posit a worker calibration feature, that computes the averaged aggregation BLEU score for a fraction of Tukers in their translations against the professional translations.

We introduce a new bilingual feature. We use the IBM Model 1 to construct the word alignment with

[1] LDC Catalog number LDC2010T23

probabilities between Urdu and English. For each foreign sentence, we calculate the word alignment feature by averaging the alignment probabilities of all words in Urdu sentence.

2.2 Cost

Compared to the cost of professional translations, the cost of crowdsourced translations is already low. Zaidan and Callison-Burch (2011) paid $0.10 per sentence. The cost to translate each of the sentences in the Urdu data set once was $179.20, plus a 10% commission to Amazon. The major cost involved with Zaidan and Callison-Burch (2011)'s method is the need to redundantly translate every source sentence. Every sentence in their set was independently translated by 4 workers. So the total cost to create the translations in their data set was $716.80 (+10%).

Another component cost of the Zaidan and Callison-Burch (2011) is the need for some amount of professionally-translated data, used to calbrate the goodness of the non-professional Turker translators. Zaidan and Callison-Burch (2011) vary the amount of calibration data used. The minimum amount is 10% of the data set. If we esimate the cost of professional translation at XXX, then the cost of the calibration data is XXX.

Here we attempt to minimize cost by reducing the number of translations needed for each sentence, and reducing the amount of professionally-translated calibration data. The lower-bound on cost is $179.20, for single translations from Turkers with no calibration data. The upperbound is $XXX ($716.80 + $XXX for YY% calibration).

2.3 Cost Quantification

Throughout this paper we will analyze the cost savings of the various methods that we propose. To make it clear how we compute the savings, we define that the unit cost for one professional references as C_p, and the unit cost for one non-professional translation as C_{np}, and the number of professional and non-professional translations we solicited are N_p and N_{np} respectively. Thus for the total cost C, we have:

$$C = N_p \cdot C_p + N_{np} \cdot C_{np}$$

The costs associated with professional translations result from the calibration data that is used to estimate the goodness of the non-professional translation. This typically will be a fraction of the total data being translated. Conversely, the number of non-professional translations will typically exceed the total number of sentences being translated, because we typically solicit multiple (redundant) translations of the same input sentence from different non-professionals, and then pick the best translation.

2.4 Quantifying the Goodness of Translations

While minimizing costs, we want to ensure that the quality of the translations does not suffer. We compute the quality of the non-professional translators using the BLEU score against professional translators. For this data set we have access to 4 sets of professional translations, which were created by different translation agencies hired by the LDC. We want to set an upper bound by seeing what the expected BLEU score is for professional translation. We therefore report BLEU scores in this paper is using a leave-one-out strategy, where we leave out one set of professional and use other 3 sets of professional translations as the reference set to calculate. Each BLEU score reported in the paper is the average of 4 numbers. This also allows us to compute an average score for the professional translators, compared against the other 3 translation agencies.

3 Data Analysis

We created a gold-standard ranking of Turkers by computing their BLEU scores compared to professionals for all of the translations that they submitted. Figure 1 shows this ranking, along with the number of HITs produced by each worker and their timing information. From this graph we see that that most workers' performance stays consistent as time passes. Good translators tend to produce consistently good translations. Bad translators tend to produce consistently bad translations. This observation may enable us to predict workers' performance based on their early submissions, so that we can come to an early decision about whether to continue to hire them.

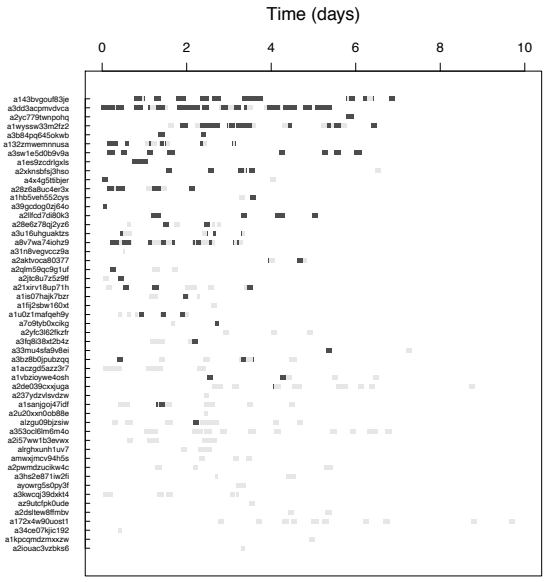

Figure 1: A time-series plot of all of the translations produced by Turkers (identified by their WorkerID serial number). Turkers are sorted based on the gold-standard ranking against professionals, with the best translators on top. Each tick represent a single translation HIT, and depicts the HIT's BLEU score (color) and its size/number of sentences (thickness). We calculated the median of all HITs' BLEU scores. HIT's color is dark if its BLEU score is higher than the median, and light if it is lower.

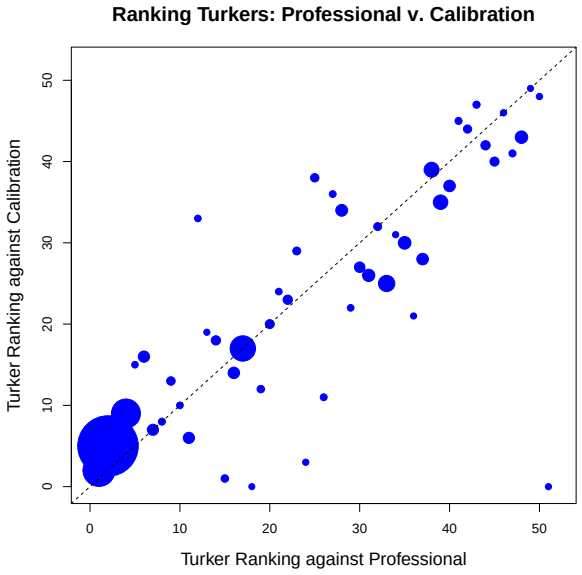

Figure 2: Correlation between gold standard ranking and calibration ranking. We use 10% training data as calibration data to rank workers. The corresponding Spearman's Correlation is 0.79. Each bubble represents a worker with his/her rank in gold standard ranking on x-axis and rank in calibration ranking on y-axis. The radius of each bubble shows the relative volume of translations completed by the worker.

4 Automatically Ranking Translators

Here, we try to compute the ranks of the Turkers, with the goal of trying to filter out bad workers. Instead of indirectly evaluating our rankings by the translation quality, we instead evaluate our predicted rankings directly, calculating their correlation with the gold standard ranking given in Table 1.

We train MERT, Regression Trees, and Linear Regression models to rank translators. MERT, the baseline method, achieves a correlation of 0.67 when trained on ranking features. This is the highest correlation that MERT achieves across all feature sets. MERT is poorly suited for ranking translators. If we contrast it with a simpler baseline that reserves 10% of the data for calibration, and computes a ranking of translators based on their BLEU scores against the professionals over this calibration set, the correlation reaches 0.79. We target 0.67 and 0.79 as the baseline correlation values to beat for our more sophisticated models.

Table 1 shows an unexpected result. The MERT trained on the complete set of features produces a correlation that is weaker than one trained only use ranking features. The reason for this is that the model is trained using the MERT algorithm (Och, 2003), which is typically used to set the parameters of a statistical machine translation system such that the 1-best translation is ranked the highest among an n-best list containing thousands of translations. Setting the feature weights using MERT does a poor job at producing a total ordering on the translators.

We get a surprisingly strong correlation with the gold standard ranking of workers, without using a model at all. Instead, we can use a small amount of calibration data (where gold standard translations are known). If we rank the worker based solely on

[2]Combination of (S)entence, (W)orker and (R)anking features

[3]Combination of (S)entence, (W)orker , (R)anking and (B)ilingual features

[4]Linear Regression

[5]Regression Tree

Feature Set	Spearman Correlation		
	MERT	Linear Regression	Regression Tree
Sentence features	0.36	0.69	0.71
Worker features	0.44	0.65	0.59
Ranking features	0.67	0.79	0.76
Calibration feature	0.79	0.79	0.79
S+W+R features [2]	0.42	0.78	0.74
S+W+R+B features [3]	0.47	0.80	0.72
All features	0.56	0.84	0.81

Table 1: Spearman's correlation for different models trained using different feature sets

Feature Set	Bleu Score (selected by ranking)			Bleu Score (selected by model)		
	MERT	LR [4]	RT [5]	MERT	LR [4]	RT [5]
Sentence features	30.04	36.66	36.97	38.51	37.84	35.32
Worker features	37.89	36.92	37.96	37.89	36.92	37.59
Ranking features	37.25	36.94	37.04	36.74	35.69	36.17
Calibration feature	38.27	38.27	38.27	38.27	38.27	38.27
S+W+R features[2]	33.04	37.39	37.60	38.44	38.69	37.04
S+W+R+B features [3]	34.30	37.59	37.27	38.80	39.23	37.00
All features	35.58	38.37	37.80	39.74	39.80	37.19

Table 2: Bleu score for different models trained using different feature sets

their first HITs' BLEU score, comparing their translations of the 10 sentences in that HIT against the reference translations, then we do well at predicting their BLEU score for all of their translations. The Spearman Correlation is 0.84 when comparing this first-HIT (10 sentences) ranking with gold standard ranking.

If we rank workers using their first k sentences (where $k \geq 1$ and ≤ 10 with step size 1 and $k \geq 20$ and $\leq$ 100 with step size 10), we can calculate the Spearman Correlation against the gold standard ranking list as k increases. The correlation converges to 1 after k is larger than 60, in part because the average number translation each worker submitted is 150.6 and the median number of translation is XXX.

4.1 Experimental Setup

4.1.1 Ranking workers using a model

In the first approach ranking workers by model prediction, we evaluate models through five-fold cross validation. We divided the data into 2 parts: 20% data for feature values calculations and remaining 80% data for testing. We use the 20% portion to calculate the worker aggregation feature, and half of the data in the 20% portion (10% of the full data set) to calculate the worker calibration feature against their references. This 10% portion of data is used as training set where the label of each sample is the BLEU score against 4 references. Since we want to show that we can save money by reducing professionals translation as calibration, the proportion of the training set is smaller than that in general machine learning problem. We use the training data to create a model, which we denote as M. For each source sentence s_i, we have a translation candidates set $T = \{t_{i,1}, t_{i,2}, t_{i,3}, t_{i,4}\}$. We select the translation with highest model predicted score $M(t)$. For MERT and Linear Regression, M is a vector of weights corresponding to dimensions in the feature space. $M(t)$ is defined as:

$$M(t) = \hat{w} \cdot f(t)$$

where $\hat{w}$ is weight vector and $f(t)$ is the feature vector of the translation t. For Regression Tree, M is a tree where each internal node has a threshold to split

First-k Sentences	BLEU Score (by selection)			Score Comparison		
	Partial Data Ranking	Gold Ranking	Random	P- R[6]	G - R [7]	G-P [8]
1	29.48	30.26	30.07	-0.59	0.19	0.78
2	32.31	33.72	29.93	2.38	3.79	1.41
3	32.42	33.91	29.80	2.62	4.11	1.49
4	32.97	34.34	29.37	3.6	4.97	1.37
5	35.27	37.63	28.54	6.73	9.09	2.36
6	35.65	37.56	28.65	7.00	8.91	1.91
7	35.10	37.57	29.30	5.80	8.27	2.47
8	34.15	34.22	29.96	4.19	4.26	0.07
9	35.59	37.57	29.81	5.78	7.76	1.98
10	35.77	37.57	29.29	6.48	8.28	1.8
20	36.97	37.10	29.21	7.76	7.89	0.13
30	37.23	37.76	28.44	8.79	9.32	0.53
40	37.93	37.94	30.12	7.81	7.82	0.01
50	37.52	37.52	29.22	8.30	8.30	0.00
60	37.13	37.13	28.66	8.47	8.47	0.00

Table 4: Spearman's Correlation for calibration data in different proportion

on the corresponding dimension and each leaf node has the BLEU score value.

Using the remaining 80% of the data as our test set, first we rank workers by their performance predicted by models and evaluate the ranking list by calculating the Spearman's correlation against the gold standard ranking. The performance of the worker w according to a model is computed as:

$$Performance(w) = \frac{\sum_{t \in T_w} M(t)}{|T_w|}$$

where T_w is the set of translations completed by w. We then assign a rank to each worker and maintain a ranked list R of the workers, according to their performance score under the model.

We select the best translation according to each model using workers' rankings in R. We select the translation from the worker with the best rank, and evaluate the translation quality by calculating the BLEU score against references. This allows us to test whether a model is suitable for ranking workers, and also to assess how much impact the assigning correct ranks to translators has on the resultant translation quality when we select the best translation using the ranking.

We intentionally avoid using the same test set to compute BLEU score and Spearnman's correlation, because (SAY WHY). We therefore divide the test set into two equal parts, one for assessing worker rankings and the other for evaluation translation quality.

4.1.2 Decide whether hire workers after their first k translations

In the second approach, we compared workers' first k sentences with references and ranked workers by their performance on these sentences. At this point we make a decision on whether to continue to hire them to do more translations. We kept the 'best' translators, defining the best as the top 25% workers in our ranked list. The idea is to retain only the top-performing workers, and to make that decision quickly (after seeing only k of their translations).

To evaluate this method, we created several test sets. Each test set excluded any item that was used to rank the workers, or which did not have any translations from the top 25% of workers according to our predicted rankings. We therefore have *different test sets* for each value of k. This makes the results slightly more difficult to analyze than in normal experiments, although the trends are still clear.

Formally, we define the test set for first k sen-

Proportion of Calibration Data		Spearman's Correlation
First k Sentence	Percentage	
1	0.7%	0.57
2	1.3%	0.62
3	2.0%	0.69
4	2.7%	0.72
5	3.3%	0.78
6	4.0%	0.80
7	4.7%	0.79
8	5.3%	0.81
9	6.0%	0.84
10	6.6%	0.84
20	13.3%	0.93
30	19.9%	0.96
40	26.6%	0.97
50	33.2%	0.98
60	39.8%	0.99
70	46.5%	0.99
80	53.1%	0.99
90	60.0%	0.99
100	66.4%	0.99

Table 3: Spearman's Correlation for calibration data in different proportion

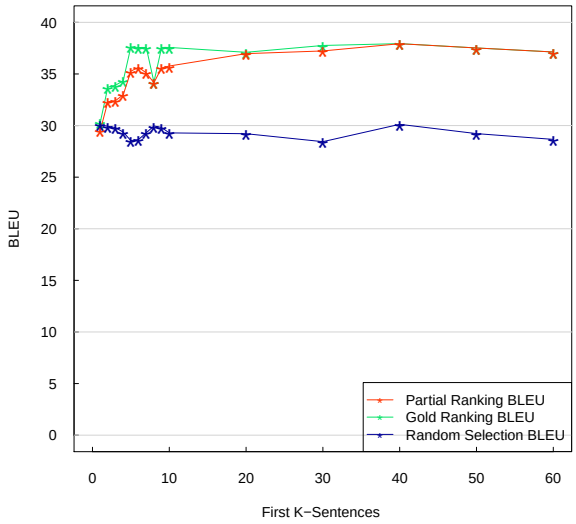

Figure 3: The BLEU score for selecting the best translation by the top 25% Turkers' ranking based on the first k sentences (red line), which is denoted as Partial Ranking BLEU. The green line shows the BLEU score for selecting the best translation by the gold standard ranking, which is denoted as Gold Ranking BLEU. The dark blue line shows the BLEU score for selecting translation randomly. We denote it as Random Selection BLEU.

tences as T_k and for each source sentence $t \in T$:

$$\{t \mid (C(t) \cap S_k = \emptyset) \wedge (C(t) \cap S_w \neq \emptyset)\}$$

where $C(t)$ is the translating candidates set of the source sentence t, S_k is the translation set consists of each worker's first k translations and S_w is the translation set consists of translations provided by selected workers (some top ranking workers).

4.2 Cost Savings

In the first approach, we quickly evaluate workers and rank them for filtering out workers with low rankings. We train linear regression models using a variety of features to score each translation and evaluate workers by averaging the scores of his/her translations.

Table 1 shows that the higest correlation is achieved using the Linear Regression model trained on all features, including the sentence, worker, ranking, bilingual and calibration features. It achieves a Spearman's Correlation of 0.84. Since we use 10% data to calculate calibration feature, and since professional translators are used to created the calibration data, its cost is approximately $ XXX (XXX * YYY sentences or words).

The Linear Regression model with all features acheives a BLEU score of 38.37, its ranked list of translators is used ot select the best translation for each source sentence. As a comparison, if we directly compute the gold standard ranking of all translators using all of the data, then the BLEU score is 38.99. The difference between these two BLEU scores is 0.62. We save 90% cost for LDC data with the penalty of losing 0.62 BLEU score in selecting accuracy.

Since the calibration data represents a substantial portion of the costs involved with collecting our translations via crowdsourcing, we can attempt to reduce it. The Linear Regression model reaches a correlation of 0.80 if we omit the calibration data entirely. Rather than using 10% of each HIT as calibration data, we experimented with using the first-K

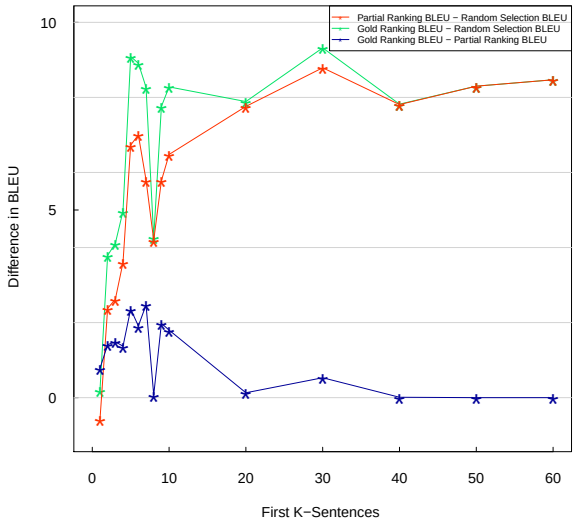

Figure 4: The difference between BLEU scores reported from three different methods. . The green line shows the difference between Gold Ranking BLEU and Random Selection BLEU. The red lines shows the difference between Partial Ranking BLEU and Random Selection BLEU and the dark bleu line shows the difference between methods Gold Ranking BLEU and Partial Ranking BLEU.

translation sentences provided by each Turker. Table 3 shows the results of Spearman's Correlation. We achieve very strong correlation when calibrating the workers based on the translations of their first 40 sentences. Even we only use the first 10 sentences to evaluate and rank workers, the correlation (ρ) is higher than 0.80. Consequently, we can decide whether to continue to hire a worker in a very short time after analyzing the first 10 sentences (or less) provided by each worker. If we use the first 20 sentences, which is still only a small part of data compared with the whole data set, ρ is higher than 0.90, nearly a perfect match with the gold standard ranking.

Figure 3 shows the BLEU score when we select the top 25% workers from the ranking list based on the performance of first k sentences. As a comparison, we also listed the BLEU scores for random selection and the BLEU score for selection based on the gold standard ranking. Figure 4 shows the difference between BLEU scores we get in three different mechanisms in order to make the comparisons clear. As we increase the number of sentences we use to rank Turkers, the BLEU score we get from the ranking approaches the BLEU score we get by selecting translations based on the gold standard ranking. Surprisingly, we see that when only a small part of sentences (say 10 sentences) for each worker are used in ranking, the ranking list is quite similar to the gold standard ranking list and the BLEU score is very close to the BLEU score get by gold standard ranking.

If we use the first 10 sentences, the correlation is 0.84 and B_{10} is 35.77. The difference between B_g and B_{10} is 1.8 while the cost is:

$$\begin{aligned} C &= 6.6\% \cdot N_p \cdot C_p + (1 - 6.6\%) \cdot 25\% \cdot N_{np} \cdot C_{np} \\ &= 0.066 \cdot N_p \cdot C_p + 0.2335 \cdot N_{np} \cdot C_{np} \end{aligned}$$

which is only a small part of the whole cost. The lost of BLEU is 4.03 (39.80 - 35.77) by comparison with model selection methods. If we increase the number of sentences we use for ranking to 20, the correlation increase to 0.93 and B_{20} is 36.97. The difference is 0.13 and the cost is:

$$\begin{aligned} C &= 13.3\% \cdot N_p \cdot C_p + (1 - 13.3\%) \cdot 25\% \cdot N_{np} \cdot C_{np} \\ &= 0.133 \cdot N_p \cdot C_p + 0.21675 \cdot N_{np} \cdot C_{np} \end{aligned}$$

5 Get Another Translation?

delta (%)	S_{upper}	BLEU Score	*# of Trans*
90	40.13	36.46	1.67
91	40.13	36.72	1.75
92	40.13	36.84	1.77
93	40.13	37.11	1.87
94	40.13	37.61	2.00
95	40.13	37.90	2.12
96	40.13	38.32	2.31
97	40.13	38.52	2.43
98	40.13	39.12	2.79
99	40.13	39.45	3.05
100	40.13	39.90	3.58

Table 5: The relation among the *delta* (the proportion of the BLEU score's upper bound S_{upper}),the BLEU score for translations selected by models and the averaged size of translation candidates set for each source sentence (*# of Trans*).

In the second approach, we train a model to decide whether a translation is 'good enough,' in which case we don't need to pay for another redundant translation of the source sentence. Besides, we quantify the quality control issue: make it possible to control the BLEU score of the translation selected from a partial translation set to a proportion (*delta*) of the upper bound of BLEU score. The upper bound of BLEU score is computed on best translations selected from the full translation set. For a specific delta value, we train the Linear Regression model to score each translation we've got already, and use this score comparing with the threshold between acceptable and unacceptable translations to evaluate whether to get another translation. If the translation we solicit currently satisfies our quality control demand which means the model predicted BLEU score is higher than the threshold, we stop soliciting other translations continually for the source sentence. Table 6 illustrates the idea of this approach. On one hand, since we only collected part of the full translation candidates set, we save money by avoiding paying for redundant translations. On the other hand, in the training and validating process, we reduce the size of the training set and validation set, say only 10% of the full data set for each , which means we only need 10% reference data to calculate the gold standard BLEU score and calibration feature value for each translation candidate in training set and validation set respectively.

To evaluate the performance of the model running with different thresholds, we first compute an upper bound by selecting the best translation among all 4 candidates for each foreign sentence of the validation set according to our model. We call this S_{upper}. S_{upper} is the highest BLEU score we can get by choosing translation using the model, since it has access to all of the available translations. Originally, for each source sentence, the size of the translation set is 4. Since we stop soliciting translations after getting the acceptable one, the averaged size of translation set among all source sentences becomes less than 4. We define the averaged size of translation sets as *# of trans*.

Table 5 shows the following positive correlations between *delta* and *# of Trans*:

$$\#\ of\ Trans \propto delta$$

Thus, it's reasonable to deduce the negative correlations between the cost and *delta*. From Table 5, we see that as the translating accuracy (*delta* and BLEU score) increasing, the averaged size of translation set increases.

5.1 Experimental Setup

To perform this experiment, we divide the data into 3 parts: 10% of the data as a training set, 10% of the data as a validation set and the remaining 80% of the data as a test set.

After training a Linear Regression model, the challenge we are facing with is how to set the threshold to separate acceptable translations and unacceptable ones.

In our design, we set the threshold empirically using the validation set after we have trained the model on the disjoint training set. More specifically, during the training process, we get the upper bound of scores for translations in the training set. Then we search for the threshold through traversing from zero to the upper bound by a small step size.

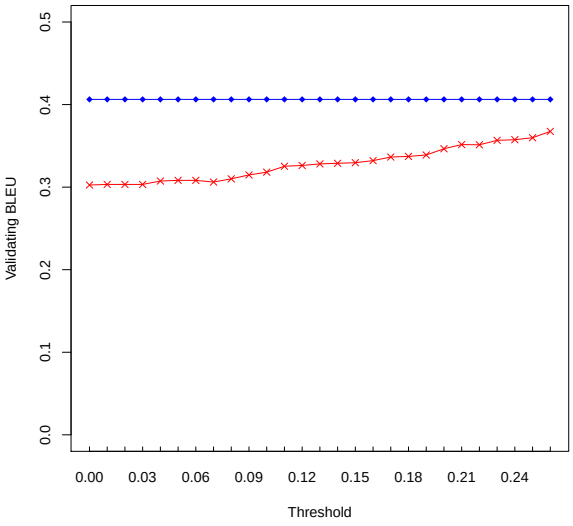

Figure 5: The Process to Sweep the Threshold

We use each value in the process as the potential threshold. We score translations of the foreign

[6] kfds
[7] The
[8] The

Source	References	Translations	Quality
ł ł	Support of France's Recommendation	France has supported the proposal.	0.342
	Support for the Proposal of France	Supporting the French proposal	0.630
	French Proposal Endorsed	France suggestion was appriciatable.	-0.014
	French Proposal Supported	defending the thinking of France.	0.269

Table 6: An example showing how to reduce redundant translations using the model and the threshold. For each source sentence, we solicit 4 references and 4 non-professional translations. The value of 'Quality' is the **model predicted score** for each translation which is different from the BLEU score. In this example, *delta* is %95, and the threshold telling apart acceptable and unacceptable translations is 0.35. Translations listed from top to the bottom are in the chronological order from the earliest submitted one to the latest submitted one. Since the first translation's quality value is lower that the threshold, we need to solicit another one. Knowing that the second translation's quality value is higher than the threshold, we stop soliciting other translations for this source sentence so that we avoid collect redundant translations and reduce cost.

sentences in the validation set. Since this approach assumes a temporal ordering of the translations, we compute the scores for each translation of a source sentence using the time-ordering of when Turkers submitted them. There are 2 conditions on the halt of this process for each foreign sentence: 1) the predicted BLEU score of some translation (submitted earlier than the last translation) is higher than the threshold or 2) we have scored all 4 translations.

After we have used the validation set to sweep various threshold values, we can pick a suitable value for the threshold by picking the lowest value that is within some delta of S_{upper}, say 90%.

Finally, we retrain our model using the union set of the training set and validation set, use the resulting model on the test set. We evaluate the model's performance by counting the average number of candidate translations that it solicits per source sentence, and by computing the loss in overall BLEU score compared to when it had access to all 4 translations. This evaluation shows how much money our model would save by eliminating unnecessary redundancy in the translation process, and how close it is to the upper bound on translation quality when using all of the translations from the original set.

5.2 Cost Savings

From Table 5, we see that the averaged size of translation sets is positive correlated to *delta*. To analyze the cost saving more clearly, we fit a model to describe the relationship between *delta* and *# of Trans*.Thus we can estimate the cost as a function of *delta* which is the goal of quality control, and bridge the gap between quality control and cost optimization.

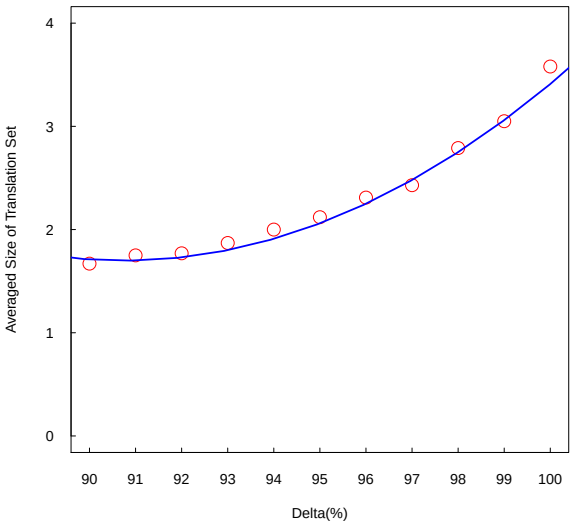

Figure 6: Relationship between *delta* and *# of Trans*. Each red circle shows the average size of translation set on x-axis and the *delta* on y-axis. The blue curve represents the model we fit to describe the relationship.

Figure 6 shows the relationship between *delta* and *# of Trans*. The model we fit can be described as a function $f(x)$:

$$f(x) = 0.02x^2 - 3.63x + 166.41$$

where x is the value of *delta*. The model fits the data pretty good and the average square error rate is 0.0054. Thus for a given value of *delta* x, the cost

is:

$$C = 20\% \cdot N_p \cdot C_p + \frac{f(x)}{4} \cdot 80\% \cdot N_{np} \cdot C_{np}$$

6 Related Work

For one of our approaches for lowering the costs of crowdsourcing, we train models to distinguish between acceptable and unacceptable translation candidates. To do so, we sweep a threshold of BLEU values. The threshold between acceptable and unacceptable translations is fuzzy so there exists some uncertainty in labeling each data sample. This is related to (Sheng et al., 2008)'s work on repeated labeling, which presents a way of solving the problems of uncertainty in labeling and selection of a threshold. In their work, they showed that for single-labeling examples, the labeling quality (the annotator's probability of producing a correct labeling) is critical to the model quality. The model prediction accuracy rises as the labeling quality increases. However, in reality, we cannot always get high-quality labeled data samples with relatively low costs. To keep the model trained on noisy labeled data having a high accuracy in predicting, Sheng et al. (2008) proposed a framework for repeated-labeling that resolves the uncertainty in labeling via majority voting. The experimental results show that a model's predicting accuracy is improved even if labels in its training data are nosity and of imperfect quality. As long as the integrated quality (the probability of the integrated labeling being correct) is higher than 0.5, repeated labeling benefits model training. More closer the quality to 0.5, the more benefits obtained in model prediction.

A very important issue in natural language processing is data annotation. Hiring professional annotators is very expensive. As an alternative, collecting several annotation for each single data sample and pick the best label is more economical. In our work, we collected several translations for each source sentence and pick the best translation. Our work shares many goals in common with Passonneau and Carpenter (2013), who created a Bayesian model of annotation, which they applied to the problem of word sense annotation. Rather than hiring professional annotators, which is very expensive, they hire non-expert annotators on Mechanical Turk. They collected 20 to 25 word sense labels for each word. To decide which label to select for each word, and to compute the quality of the annotation, they proposed the probabilistic model using Bayes's rule. They calculated the product of the prior probability (the initial probability of being the observed label) and the conditional probability (the probability of being the observed label given the true label) and pick one label with the highest score. This sort of a probability estimate provides much more information about the corpus quality than previous methods, such as calculating inter-annotation agreement through Coehn's kappa score. Kappa measures the agreement coefficient among annotators in a chance-adjusted fashion. However, the method only reports how often annotators agree, but does not provide information about the quality of the corpus and the individual data sample.

Although Passonneau and Carpenter (2013) collect word sense labels, which are a small, enumerable set, and we collect translation (which could be thought of as a kind of label, albeit a very complex one), there is a strong commonality in the goals of their word and the goals of our work. Specifically, how can we use all the labels collected in order to select of the best label. And how can we rank the annotators themselves. For selecting the best label for word senses, majority voting is a direct and easy way to solve the problem, but the task is more complex for translation.

Passonneau and Carpenter (2013) also proposed an approach to detect and avoid spam workers. They measured the performance of worker by comparing worker's labels to the current majority labels and worker with bad performance would be blocked. However, this approach suffered from 2 shortcomings: (1) Sometimes majority labels may not reflect the ground truth label. (2) They didn't figure out how much data(HITs) is needed to evaluate a worker's performance. Although they could find the spam after the fact, it was a post-hoc analysis, so they had already paid for that worker and wasted the money. We attempt to identify poor workers as quickly as possible, in order to limit the amount of work that we solicit from them.

Lin et al. (2014) examined the relationship between worker accuracy and budget in the context

of using crowdsourcing to train a machine learning classifier. They show that if the goal is to train a classifier on the labels, that the properties of the classifier will determine whether it is better to re-label data (resulting in higher quality labels) or get more single labeled items (of lower quality). They showed that classifiers with weak inductive bias benefit more from relabeling, and that relabeling is more important when worker accuracy is low (barely higher than 0.5). Counter-intuitively, an infinite budget does not make relabeling work any better.

Novotney and Callison-Burch (2010) showed a similar result for training an automatic speech recognition (ASR) system. When creating training data for an ASR system, given a fixed budget. Their system's accuracy was higher when it is trained on more low quality transcription data compared to when it was trained on fewer high quality transcriptions.

7 Discussion

We have introduced several ways of lowering the costs associated with crowdsourcing translations:

- We show that we can quickly identify bad translators, either with a model designed to rank them, or by ranking them by having them first translate a small number of sentences with gold standard translations. The cost savings here comes from not hiring bad workers.

- After we have collected one translation of a source sentence, we consult a model that predicts whether its quality is sufficiently high or whether we should pay to have the sentence retranslated. The cost savings here comes from reducing the number of redant translations.

- In both cases we need a some amount of professionally translated materials to use as a gold standard for calibration. The cost of these professional translations can dominate the cost of our models, so we experiment with how little we can get away with.

In all cases, there is a trade-off between lowering our costs and producing high quality translations. Figure 7 plots the cost versus the BLEU scores for the different configurations that we experimented with.

In Figure 7-(a) the increasing costs are a function of how many sentences we use to rank the translators. Here we use no model, and simply rank the translators by their BLEU score against a small amount of gold standard data. Although the quality peaks at 37.9 BLEU after $11,600, the return on investment is low after spending the first $2,000 to get a BLEU of 35.6. We are able to rank the translators with high accuracy and achieve a relative high BLEU score by paying for a comparatively small number of professional translations to use as calibration. From our experiments, 10-20 professionally translated sentences seems like a reasonable number.

Figure7-(b) uses a model to determine whether to purchase another translation. Here the starting cost is high (nearly $9,000) because the model requires a significant amount of professional translations in order to train the model and to determine the optimal threshold values for whether to solicit another translation. This model allows us to significantly improve the overall translation quality to a BLEU score of nearly 40, for a final cost of $9,200.

To emphasize the effectiveness of model selection approach, Figure 7-(c plots the relationship between BLEU and non-professional component of the overall cost. Past approaches to crowdsourcing translation always solicited 4 non-professional translations of every source sentence. The cost for translating our 1433 test sentences under this approach is $573.44. This produces the maximum BLEU score of 40.1. Using our model to reduce the number of redundant translations, we can reduce the costs with mild degredation in translation quality. We can cut the number of non-translations in half, and pay only $286.72, while achieving a BLEU score of 37.6 (94% of the maximum), or pay $348.36, or %60.7 of total non-professional translations, for a BLEU of 38.5 (%96 of the maximum).

8 Conclusion

In this paper, we propose two mechanisms to optimize cost: the ranking selection method and the model selection method. They have different applicable scenarios. The ranking selection method is a very simple method without any model training. This approach is inspired by the intuition that workers' performance is consistent. The ranking se-

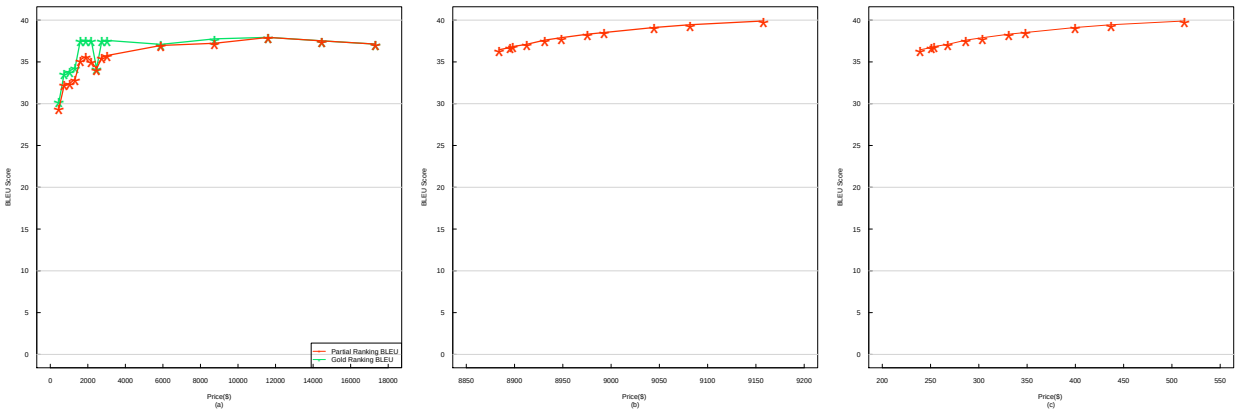

Figure 7: The Relationship between BLEU score and costs. In Figure (a), the red line shows the relationship between BLEU score and the total costs (professional and non-professional) for the ranking based approach. The green line shows the corresponding translation quality for gold standard ranking selection measured in BLEU score. Figure (b) shows the relationship between BLEU score and the total costs for model-based approach. Figure (c) illustrates the relationship between BLEU score and non-professional costs for model based approach.

lection method is suitable for crowdsourcing tasks with vague requirements on the quality control issue and crowdsourcing task requester who has little background in machine learning or data mining and don't know how to train a model. The model selection method works if there exists a specific requirement in quality control and the more data collected, the more benefits will be obtained since this approach reduced the amount of non-professional data dramatically.

Acknowledgments

Do not number the acknowledgment section. Do not include this section when submitting your paper for review.

References

Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12. Association for Computational Linguistics.

Christopher H Lin, Mausam, and Daniel S Weld. 2014. To re (label), or not to re (label).

Scott Novotney and Chris Callison-Burch. 2010. Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 207–215. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Rebecca J Passonneau and Bob Carpenter. 2013. The benefits of a model of annotation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 187–195. Citeseer.

Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM.

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 the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics.

Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1220–1229.