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

Crowdsourcing makes it possible to create translations at low cost. We proposed two mechanisms to make this process even cheaper while maintaining a high translation quality. First, we introduce a ranking selection method that allows us to reduce cost by quickly identifying bad translators after they have translated only a few sentences. This allows us to rank translators, so that we only rehire good translators and so that we can select the best translations from among good candidates. Second, we develop a model selection method. Our model evaluates the translation quality on a sentence-by-sentence basis, and fits a threshold between acceptable and unacceptable translations. Unlike past work, which always paid for a fixed number of translations of each source sentence and then choosing the best from among them, we can decide after seeing a single translation whether it is good enough or not. Our model based selection allows us to reduce cost by reducing the number of redundant translations that we solicit.

1 Introduction

Crowdsourcing is a promising new mechanism for collecting large volumes of annotated data at low cost. 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 it possible to hire people to create sufficient volumes of translation in order to train SMT systems (for example, Zbib et al. (2013), Zbib et al. (2012), Post et al. (2012)).

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 in more traditional employment scenarios, where our annotator are pre-vetted and their skills are clear 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 at 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 from 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) To reduce the number of worker we hire, and retain only high quality translators, we quickly identify and filter out workers who produce low quality translations. (2) To reduce the number of translations that we solicit for each source sentence, instead of soliciting a fixed number of translations for each foreign sentence, we stop soliciting trans-

lations after we get an acceptable one. We do so by building models to distinguish between acceptable translations and unacceptable ones. 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 0.84), 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 0.84 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 70% of the total cost using ranking selection method and at 60% of the non-professional translations' cost using model selection method.

2 Previous work

We use the data collected by Zaidan and Callison-Burch (2011) through Amazon's Mechanical Turk. 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] and these workers also filled out a survey about their language skills and their countries of origin. In each HIT, they posted 10 Urdu sentences to be translated. A total of 51 Turkers contributed translations.

[1]LDC Catalog number LDC2010T23

2.1 Professional quality from non-professionals

Zaidan and Callison-Burch (2011) used the features in order to train models to select the best translation from 4 candidate translations, and the translation quality is comparable to professional quality. They 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, 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. Finally, they integrate a worker calibration feature, that computes the averaged aggregation BLEU score for a fraction of Tukers in their translations against the professional translations.

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 their 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%). Although <\$1,000 is certainly a modest amount, if we want to collect data at a very large scale, the cost for non-professional translations will dramatically increase.

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 estimate the cost of professional translation at \$6.03 per sentence, then the cost of the calibration data is \$4,322.30.

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 for the non-professionals cost is $716.80 and the upperbound for total cost is $5,039.10 ($716.80 + $4,322.30 for 10% calibration).

3 Problem Definition

Our problem definition of the cost optimization task is: given a small portion of translation data (non-professionals and professionals), we want to identity bad workers and unacceptable translations to reduce cost by avoiding hiring bad workers continually or purchasing redundant translations after we get acceptable ones. At the same time, we want to maintain in high translation quality.

3.1 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 reference as C_p, and the unit cost for one non-professional translation as C_{np}. Suppose we have N_p source sentences with α matching professional translations for each of them and N_{np} source sentences with β matching non-professional translations for each of them, the total cost C is:

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

where in our case, basically $\alpha = \beta = 4$. 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.

3.2 Quantifying the Goodness of Translations

For this data set we have access to 4 sets of professional translations, which were created by different translation agencies hired by the LDC. While minimizing costs, we want to ensure that the quality of the translations does not suffer. To evaluate translation quality, we compute the quality of selected non-professional translations using the BLEU score (Papineni et al., 2002) against all professional translations. However, to reduce cost of computing the calibration feature and labeling each training sample by its BLEU score, we select **only one** professional translation randomly as reference to calculate the BLEU score.

3.3 Model Definition

To evaluate each translation, we use a linear regression model to predict score ($\hat{y} \in R$) for an input translation t.

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

where $\hat{w}$ is the associated weight vector and $f(t)$ is the feature vector of the translation t.

3.4 Features

Besides features (Zaidan and Callison-Burch, 2011) used, we introduce a new bilingual feature. We use the IBM Model 1 to construct the word alignment with 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.

4 Automatically Ranking Translators

We present ranking selection methods to compute ranks of workers from a small portion of work they submitted. We filter out bad workers and select the best translation from translations provided by surviving workers based on their ranks.

4.1 Workers are consistently good (or bad) over time

Figure 1 illustrates the consistency of workers' performance by showing the gold-standard ranking of Turkers created by computing their BLEU scores compared to professionals for all of the translations that they submitted, 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 and bad workers rarely produce good translations.

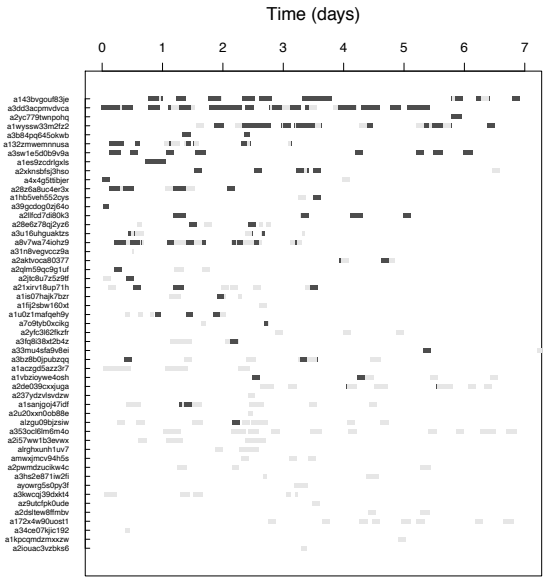

Figure 1: A time-series plot of all of the translations produced by Turkers (identified by their WorkerID serial number). Turkers are sorted based on the gold-standard ranking against professionals, with the best translators on top on y-axis. Each tick represent a single translation HIT, and depicts the HIT's BLEU score (color) and its size/number of sentences (thickness). A HIT contains 10 sentences for translation. We use the average BLEU score for each HIT and show its tick in black if its BLEU score is higher than the median and in light grey other wise.

4.2 Ranking Selection Method

Since workers' performance is consistent, workers' rankings are sufficiently accurate to reflect the quality of translations provided by them and we can select the translation which is provided by the worker with the best rank. We propose two methods to rank workers and select translations. They are both very 'cheap' since we only use a small portion of professional translations and avoid hiring bad workers after we get workers' ranking.

4.2.1 Ranking workers using a model

We use a linear regression model to score each translation and rank workers' by their model predicted performance. The model predicted score for translation t is defined as $M(t)$. The model predicted performance of the worker w is:

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

where T_w is the set of translations completed by w. After we rank workers, we keep top workers in the list and select translation provided by the worker with best rank among top workers.

4.2.2 Ranking workers using their first k translations

Rather than using a model to rank workers, we use the quality of first-K translation sentences provided by each Turker as calibration to rank workers. Then, we filter out bad workers and select translation based on the ranks of remaining workers.

4.3 Experiments

We report ranking's correlation to gold standard ranking and translation quality for both two methods.

4.3.1 Oracles

In ranking workers, we use workers' gold standard ranking as the oracle. We compute the gold standard performance score for each worker using the average BLEU score of all translations provided by this worker and rank workers by their scores. For each translation, the BLEU is computed against 4 professional references. In selecting best translations, we set the gold standard ranking selection method as the oracle method, in which we select translation provided by the worker with best rank in gold standard ranking, and the BLEU score achieved is denoted as B_{gold}.

4.3.2 Baselines

We set two baselines for ranking correlation(ρ) against gold standard ranking for our proposed approaches. For the first baseline, we choose the MERT(Och, 2003) baseline, which achieves a correlation of 0.67 when trained on ranking features. This is the highest correlation that MERT achieves across all feature sets. The second baseline is a simpler baseline that reserves 10% of the data for calibration, and computes a ranking of translators based on their BLEU scores against professionals over this calibration set, the correlation reaches 0.68. Besides, we set the random selection method as the baseline method in translation selection.

4.3.3 Ranking workers using a model

We use %10 of data to train a linear regression model to rank workers and select best translation by workers' ranking. Table 1 shows that our model achieves the highest BLEU score of 38.37 when trained on all features. If we calculate calibration feature and training sample label against only one reference rather than 4 references, we achieve a BLEU score of 37.52 with a correlation equaling to 0.71 which is higher that two baselines. To further reduce cost, we keep retaining top 25% workers and select the translation with the best rank provided by top workers. We achieve a BLEU score of 36.98 while S_{gold} is 38.51. The cost is:

$$C = \frac{\alpha}{4} \cdot \frac{1}{10} N_p \cdot C_p + \frac{\beta}{4} \cdot \frac{9}{10} N_{np} \cdot C_{np} = 1,241.86(\$)$$

Thus, we can achieve a comparable translation quality by spending almost only 25% of money with a quality lost of 1.53 in BLEU.

Feature Set	ρ	BLEU
(S)entence features	0.69	36.66
(W)orker features	0.65	36.92
(R)anking features	0.79	36.94
Calibration features	0.79	38.27
Calibration features*	**0.68**	**37.56**
S+W+R features	0.78	37.39
S+W+R+Bilingual features	0.80	37.59
All features	**0.84**	**38.37**
All features*	**0.71**	**37.52**

Table 1: Spearman's correlation(ρ) and translation quality of model predicted ranking selection method for linear regression model trained using different feature sets. We don't filter out bad workers when selecting the best translation. * indicates that we choose **only one** professional reference to calculate the BLEU score as calibration feature and the true label of a training sample while in other cases, we use 4 references to calculate BLEU score.

4.3.4 Ranking workers using their first k translations

Without using any model, we rank workers using their first k translations and select best translations based on rankings of the top %25 workers. 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 sentences as T_k and for each source sentence $s \in T_k$:

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

where $C(s)$ is the translating candidates set of the source sentence s, 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). Table 4 shows the results of Spearman's correlations for different value of K. Compared with 4-reference calibration, we achieve very strong correlation when calibrating the workers using one reference based on the translations of their first 40 sentences. Even we only use the first 20 sentences to evaluate and rank workers, the correlation (ρ^*) is close to 0.90. Consequently, we can decide whether to continue to hire

Proportion of Calibration Data		ρ^+	ρ^*
First k sentences	Percentage		
1	0.7%	0.57	0.41
2	1.3%	0.62	0.48
3	2.0%	0.69	0.59
4	2.7%	0.72	0.59
5	3.3%	0.78	0.69
6	4.0%	0.80	0.70
7	4.7%	0.79	0.71
8	5.3%	0.81	0.69
9	6.0%	0.84	0.77
10	6.6%	0.84	0.76
20	13.3%	0.93	0.88
30	19.9%	0.96	0.93
40	26.6%	0.97	0.95
50	33.2%	0.98	0.95
60	39.8%	0.99	0.94

Table 2: Spearman's Correlations for calibration data in different proportion. * indicates that the calibration is computed against **only one** reference while + indicates that the calibration is computed against 4 references.

a worker in a very short time after analyzing the first k sentences ($k \leq 20$) provided by each worker.

Figure 5 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 plotted the BLEU scores for random selection and the BLEU score for selection based on the gold standard ranking(B_{gold}). Figure 6 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 B_{gold}. Surprisingly, we see that when only a small part of sentences (say 20 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.

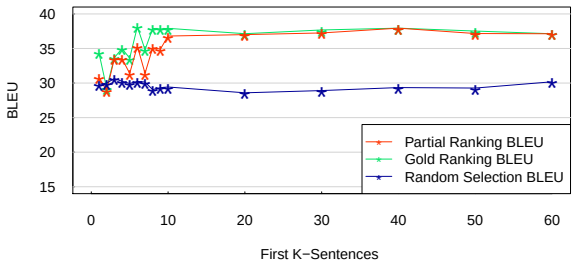

Figure 2: The BLEU score for selecting the best translation by the top 25% Turkers' ranking based on the first k sentences (red line), which is denoted as Partial Ranking BLEU. The green line shows the BLEU score for selecting the best translation by the gold standard ranking, which is denoted as Gold Ranking BLEU. The dark blue line shows the BLEU score for selecting translation randomly. We denote it as Random Selection BLEU.

If we use the first 20 sentences to rank workers, the correlation is 0.88 and the BLEU score achieved is 37.01 while and B_{gold} is 37.14. The difference between these two scores is 0.13 and the cost is:

$$C = \frac{\alpha}{4} \cdot (0.133 \cdot N_p) \cdot C_p + \frac{\beta}{4} \cdot (0.867 \cdot N_{np}) \cdot C_{np}$$
$$= 1,592.53(\$)$$

We can achieve almost the same translation as the oracle method with only spending 32% of the total cost.

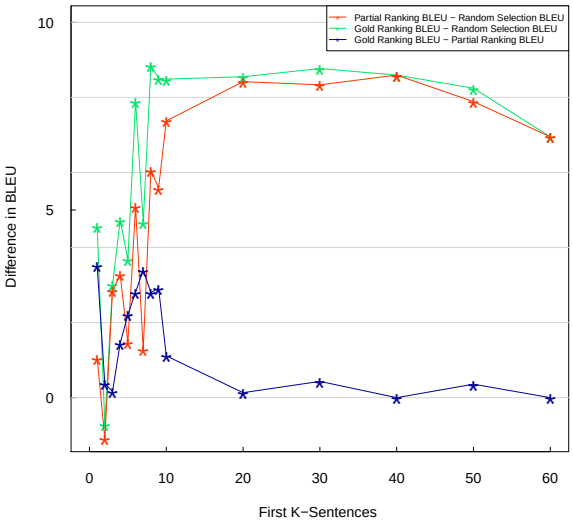

Figure 3: The difference between BLEU scores reported from three different methods in Figure 5.

5 Get Another Translation?

We present the model selection method 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 (δ) of the upper bound of BLEU score. The upper bound of BLEU score is computed on best translations selected from the full translation set of training data. Algorithm 1 details the process. After we get the θ (threshold between acceptable and unacceptable translations) for a specific value of δ (say 95%), for given a a new translation, we score it by our model and if the its score is higher than θ, we stop soliciting another translation. Otherwise, we continually solicit translations. Table 6 illustrate this mechanism.

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 3: 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.

Algorithm 1 Model selection algorithm

Input: δ, the proportion of the upperbound of BLEU score; a training set $S = \{(x^s_{i,j}, y^s_{i,j})^4_{j=1}\}^n_{i=1}$ and a validation set $V = \{(x^v_{i,j}, y^v_{i,j})^4_{j=1}\}^n_{i=1}$ where $x_{i,j}$ is the feature vector for the jth translation of the source sentence s_i and $y_{i,j}$ is the label for $x_{i,j}$.

Output: θ, the threshold between acceptable and unacceptable translations; m(x), a linear regression model.

initialize $\theta \leftarrow 0$,$m(x) \leftarrow \emptyset$
$m'(x) \leftarrow$ train a linear regression model on T
$maxbleu \leftarrow$ use $m'(x)$ select best translations on T and calculate the translation quality
while $\theta \neq maxbleu$ **do**
  for $i \leftarrow 1$ to n **do**
    for $j \leftarrow 1$ to 4 **do**
      if $m'(x^v_{i,j}) > \theta \wedge j < 4$ **then** select $x^v_{i,j}$ for s_i and **break**
      if $j == 4$ **then** select $x^v_{i,j}$ for s_i
  $q \leftarrow$ calculate translation quality for V
  if $q > \delta \cdot maxbleu$ **then break**
  else $\theta = \theta + stepsize$
$m \leftarrow$ train a linear regression model on $S \cup V$
Return: θ and model $m(x)$

5.1 Comparability of Translation Quality between Different Translation Selection Strategies

To show the comparability of translation quality between ranking selection strategy and model selection strategy, we train MERT, Regression Trees, and Linear Regression models to rank translators, calculate rankings' correlations against the gold standard ranking and compare the translation qualities between ranking selection and model selection. Besides features (Zaidan, 2009) used, we introduce a new bilingual feature. We use the IBM Model 1 to construct the word alignment with 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. 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. We use the training data to create a model, which we denote as M. For each source sentence s_i, we have a translation set $T = \{t_{i,1}, t_{i,2}, t_{i,3}, t_{i,4}\}$. The model predicted score for translation t is defined as $M(t)$. 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.

For each test sample, we select the translation provided by the worker with the best rank, and evaluate the translation quality by calculating the BLEU score against references. As a comparison, for the same test sample, we also select the translation with the highest model predicted score and evaluate translation quality. Table **??** presents ranking lists' correlation scores corresponding to different models trained using different features. Table **??** shows the translation quality comparisons for the two selecting strategies. From Table **??** and Table **??**, we know that if the predicted ranking list is highly correlated to the gold standard ranking, the translation quality of ranking selection method is comparable to that of model selection method. Besides, as a comparison, if we directly compute the gold standard ranking of all translators using all of the data and select translation based on the gold ranking, then the BLEU score (S_{rgold}) is 38.99, which is quite close to 39.80 (S_{mgold}), the highest BLEU score we get by model selection method. Thus, it's reasonable to use S_{rgold} as the upper-bound of the translation quality instead of S_{mgold} since it's difficult to compare translation qualities between a ranking selection method and a model selection method on dynamic testing sets. Table **??** 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.

5.2 Baselines

We set two baselines for ranking correlation for our proposed approaches. For the first baseline, we choose the MERT baseline, which achieves a correlation of 0.67 when trained on ranking features. This is the highest correlation that MERT achieves across all feature sets. The second baseline is 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.

5.3 Save Cost by Filtering out Bad Workers

5.3.1 Ranking workers using a model

Table **??** shows that the Logistic Regression Model achieves the highest BLEU score trained on all features among all ranking selection methods. Thus we train a Logistic Regression Model to rank workers and keep retaining top 25% workers. In testing, we only select the translation with the best rank provided by top workers. We achieve a high ranking correlation of 0.84 and a BLEU score of 37.94 while S_{rgold} is 38.99. The difference between these two BLEU scores is 1.05. The cost is:

$$\begin{aligned} C &= 10\% \cdot N_p \cdot C_p + (1 - 10\%) \cdot 25\% \cdot N_{np} \cdot C_{np} \\ &= 0.1 \cdot N_p \cdot C_p + 0.225 \cdot N_{np} \cdot C_{np} \end{aligned}$$

5.3.2 Decide whether hire workers after their first k translations

Rather than using a model to rank workers, we use the quality of first-K translation sentences provided by each Turker as calibration to rank workers. The translation quality is computed against references. Table 4 shows the results of Spearman's Correlation for different value of K. We get a surprisingly strong correlation with the gold standard ranking of workers, without using a model at all. 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. 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. Consequently, we can decide whether to continue to hire a worker in a very short time after analyzing the first k sentences (k may be equal to or less than 10) provided by each worker. 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*

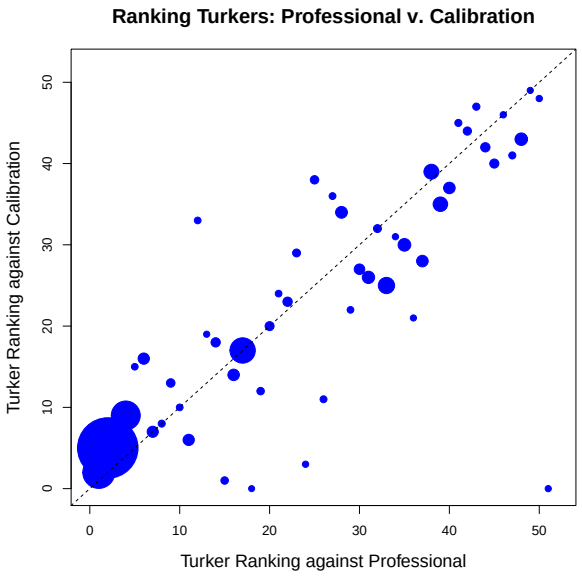

Figure 4: Correlation between gold standard ranking and calibration ranking. We use 10% training data as calibration data to rank workers. The corresponding Spearman's Correlation is 0.79. Each bubble represents a worker with his/her rank in gold standard ranking on x-axis and rank in calibration ranking on y-axis. The radius of each bubble shows the relative volume of translations completed by the worker.

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

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 sentences 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).

Figure 5 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 plotted the BLEU scores for random selection and the BLEU score for selection based on the gold standard ranking. Figure 6 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

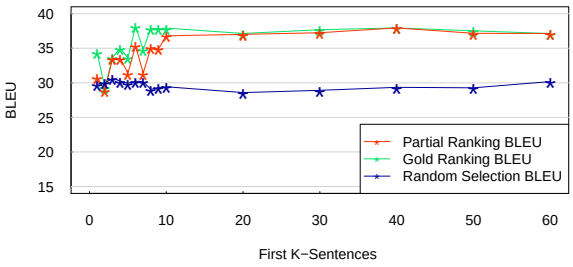

Figure 5: The BLEU score for selecting the best translation by the top 25% Turkers' ranking based on the first k sentences (red line), which is denoted as Partial Ranking BLEU. The green line shows the BLEU score for selecting the best translation by the gold standard ranking, which is denoted as Gold Ranking BLEU. The dark blue line shows the BLEU score for selecting translation randomly. We denote it as Random Selection BLEU.

rank Turkers, the BLEU score we get from the ranking approaches the BLEU score (B_g) we get by se-

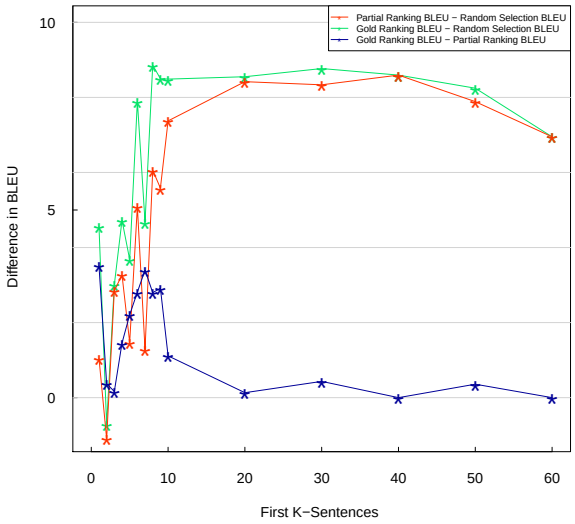

Figure 6: The difference between BLEU scores reported from three different methods in Figure 5.

lecting 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 and B_g is 37.57. The difference between B_g and B_{10} is 1.8 while the cost is:

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

which is only a small part of the whole cost. 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 decreases to 0.13 and the cost is:

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

6 Get Another Translation?

We present the model selection method 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's 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 up-

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*).

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.

per 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 positive correlations between *delta* and *# of Trans*. 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.

6.1 Experimental Setup

In 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.

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

6.2 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*

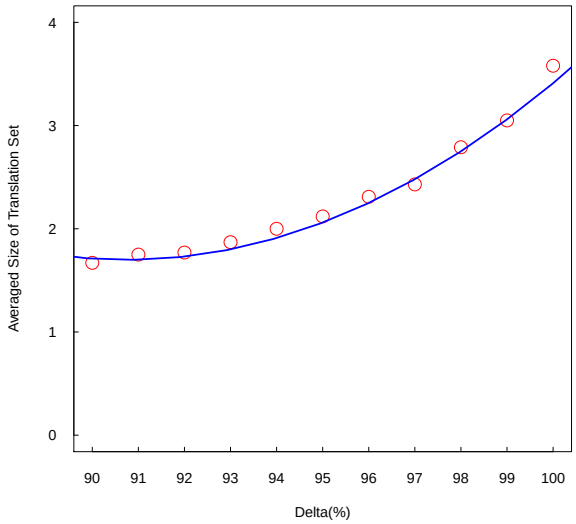

Figure 7: Relationship between *delta* and *# of Trans*. Each red circle shows the *delta* on x-axis and the average size of translation set on y-axis. The blue curve represents the model we fit to describe the relationship.

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 7 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:

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

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 re-translated. The cost savings here comes from reducing the number of redundant 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 8 plots the cost versus the BLEU scores for the different configurations that we experimented with.

In Figure 8-(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. The quality peaks at 37.9 BLEU after $11,600. 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.

Figure8-(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 8-(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 $348.36,60.7% of total non-professional translations' cost, for a BLEU of 38.5 (96% of the maximum).

8 Related Work

(Sheng et al., 2008)'s work on repeated labeling presents a way of solving the problems of uncertainty in labeling. Since we cannot always get high-quality labeled data samples with relatively low costs in reality, 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.

Passonneau and Carpenter (2013) created a Bayesian model of annotation and they applied to the problem of word sense annotation. 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.

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.

9 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 selection method is suitable for crowdsourcing tasks which do not have specific requirements about the quality of the translations, or when the data collection is being performed by a requester who does not have sufficient background in machine learning in order to train a model, or when only very limited amounts of gold standard data are available. The model selection method works if there exists a specific requirement that the quality control must reach a certain threshold, or when more data needs to be collected. This model is most effective when reasonable amounts of pre-existing professional translations are available for setting the models threshold. Its major cost reduction comes from dramatically reducing the amount of non-professional data to maintain the same quality.

Acknowledgments

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

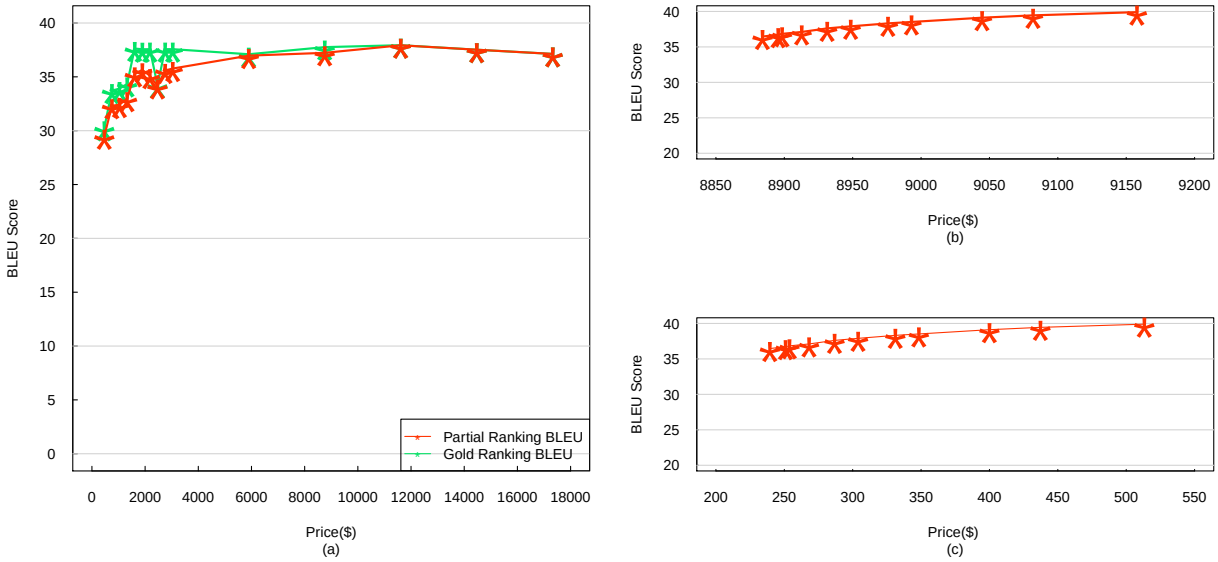

Figure 8: The Relationship between BLEU score and costs. In Figure (a), the red line shows the relationship between BLEU score and the total costs (professional and non-professional) for the ranking based approach. The green line shows the corresponding translation quality for gold standard ranking selection 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.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. 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.

Matt Post, Chris Callison-Burch, and Miles Osborne. 2012. Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 401–409. Association for Computational Linguistics.

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.

Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F Zaidan, and Chris Callison-Burch. 2012. Machine translation of arabic dialects. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 49–59. Association for Computational Linguistics.

Rabih Zbib, Gretchen Markiewicz, Spyros Matsoukas, Richard M Schwartz, and John Makhoul. 2013. Systematic comparison of professional and crowdsourced reference translations for machine translation. In *HLT-NAACL*, pages 612–616.