The Name of My Thesis

Mingkun Gao

A THESIS

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Master of Science in Engineering

2015

____
Chris Callison-Burch
Supervisor of Thesis

____
Graduate Group Chairman

Contents

0.1 ABSTRACT

Crowdsourcing makes it possible to create translations at much lower cost than hiring professional translators. However, it is still expensive to obtain millions of translations needed to train high performance statistical machine translation systems. We proposed two mechanisms to reduce the cost of crowdsourcing while maintaining high translation quality. First, we develop a translation reducing method. We train a linear model to evaluate the translation quality on a sentence-by-sentence basis, and fit 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 chose the best from them, we can stop earlier and pay less when we receive a translation that is good enough. Second, we introduce a translator reducing method by quickly identifying bad translators after they have translated only a few sentences. This also allows us to rank translators, so that we re-hire only good translators to reduce cost.

Chapter 1

Introduction

As the globalizing process continues all over the world, people's demand in automatically translate materials in other languages increases. This promoted the improvement in the quality of machine translation systems. Currently, machine translation is a technique widely used in people's daily life. However, to build a machine translation system, high quality parallel corpus is always necessary. For some popular languages, such as French or Spanish, there are plenty of parallel corpus with English which could be used for training a machine translation system. However, for some languages used only by a small amount of people, how to get high quality bilingual corpus is a big issue. Usually, we can hire professional translators or linguists to translate the source document in the target language to English and build the corpus. This can guarantee a good translating quality of the corpus. But there are two limitations for this approach:

1. It's hard to get a large amount of language resources in this way for the lack of professional translators.
2. The cost for hiring professional translators or linguists is very high, especially a large amount of resources need to be translated.

Crowdsourcing is a mechanism to collect data from a large amount of people at relatively low cost. The popularization of the Internet makes it possible to do crowdsourcing tasks from online communities. In online crowdsourcing communities, anyone can be a worker as long as he or she has the access to the Internet. This good nature of crowdsourcing makes it a promising mechanism in Natural Language Processing (NLP). Many NLP researchers have started to create language resource data through crowdsourcing (for example, Snow et al. (2008), Callison-Burch and Dredze (2010) and others).

Machine translation is a suitable field where crowdsourced data collection can be used in corpus construction since it needs a large volume of bilingual sentence-aligned parallel corpora. This thesis contains two aspects of crowdsoucing machine translation: quality control (Zaidan and Callison-Burch, 2011) and cost optimization for crowdsourcing machine translation.

1.1 Quality Control for Crowdsourcing Machine Translation

Crowdsourcing is a promising new mechanism to collect large volumes of annotated data. 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. Crowdsourcing's low cost has made it possible to hire large number of people online to collect language resource data in order to train machine translation systems (for example, Zbib et al. (2013), Zbib et al. (2012), Post et al. (2012), Ambati and Vogel (2010)). However, there is a price for crowdsourcing's low cost. Crowdsourcing is different from traditional employing mode. There is no pre-test or interview before we hire a crowdsourcing worker online, which means we don't know the proficiency and working ability of the worker on the crowdsourcing platform. In our case for machine translation, there are no professional translators and there are no built in mechanism to test the ability of them. They work completely out of anyone's oversight. Thus, translations produced via crowdsourcing may be in low quality. Previous research work has solved this problem. Zaidan and Callison-Burch (2011) proposed a framework to improve the quality of crowdsourcing machine translation to a professional level. Instead of soliciting only one translation for each Urdu source sentence, they collected multiple translations as candidates for the corresponding source sentence in Urdu. Then, they extracted features and built the feature vector for each candidates. They used professional translations as calibration data to gold-standard label each training and testing sample in BLEU (Papineni et al., 2002). Finally, they trained a MERT (Och, 2003; Zaidan, 2009) model to score each translation and selected the translation with the highest BLEU score. This framework lead to a corpus BLEU score comparable to the BLEU score of the professionally translated corpus.

We extend their crowdsourcing translation framework using other models, such as linear regression model and decision tree model, and get similar result showing that the framework is effective to control the quality of bilingual parallel corpus built via crowdsourcing.

1.2 Cost Optimization for Crowdsourcing Machine Translation

Even though the cost for crowdsourcing is low, if we want to collect a huge corpus of non-professional translations, it still spends lots of money. For example, suppose we have a corpus constitutes of one million sentences, and the estimate cost for translating one source sentence is $0.10, if we plan to solicit one set of non-professional translations, the total cost is $100,000; and if we plan to solicit four sets of non-professional translations, the total cost increases to $ 400,000. We explore the possibility and methods to achieve same high quality while minimizing the associated cost.

In (Zaidan and Callison-Burch, 2011)'s framework, multiple translations are solicited. Based on the intuition that good translation may arrive earlier and we don't have to collect all candidates for the source sentence, we design and implement a mechanism 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 translations after we get an acceptable one. We do so by building models to distinguish between acceptable translations and unacceptable ones.

In addition, we find that workers' performance is consistent over time. Thus, we can quickly identify spam workers only after soliciting their early submissions and filter them out as soon as possible. In this way, we save the cost by avoiding hiring spam workers.

1.3 Main Contribution

The main contribution of this thesis is that:

- We extend Zaidan and Callison-Burch (2011)'s quality control framework to other models.

- Our model can predict whether a given translation is acceptable with high accuracy, substantially reducing the number of redundant translations needed for every source segment.

- Translators can be ranked well after observing only small amounts of data compared with the gold standard ranking (reaching a correlation of 0.94 after seeing the translations of only 20 sentences from each worker). Therefore, bad workers can be filtered out quickly.

- The translator ranking can also be obtained by using a linear regression model with a variety of features at a high correlation of 0.95 against the gold standard.

- We can achieve a similar BLEU score as Zaidan and Callison-Burch (2011) at half the cost using our cost optimizing methods.

Chapter 2

Quality Control for Crowdsourcing Machine Translation

Quality control is an important issue for crowdsourcing annotation and labeling since workers in the crowdsourcing community are not required to provide the qualification to do the job. Sheng et al. (2008)'s work on repeated labeling presents a way of solving the problems of uncertainty in labeling in crowdsourcing. 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 noisy 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.

To improve the quality of crowdsourcing machine translation, (Zaidan and Callison-Burch, 2011) solicited four translations for each source sentences. By selecting the best translation among them, they achieved a professional level of quality compared to gold standard references. We extended their framework to other models.

2.1 Data Collection

We use the data collected by Zaidan and Callison-Burch (2011) through Amazon's Mechanical Turk(MTurk). MTurk is an online platform provided to people for completing Human Intelligence Tasks(HIT) with a relatively low cost. We use their Urdu-to-English 2009 NIST Evaluation Set as our corpus. Zaidan and Callison-Burch (2011) translated the Urdu side to English through MTurk. They collected the translations in the unit of Human Intelligence Tasks(or HITs). In every HIT, they posted 10 Urdu sentences to be translated. Every sentence is translated by 4 workers, and subsequently post-edited by 10 additional workers.[1] This data set also has four corresponding professional translations for each of the Urdu sentences, collected by LDC. This makes it possible to compare the Turkers' translation quality to professionals.

2.2 Feature Extraction

Following Zaidan and Callison-Burch (2011), we extract a number of features from the translations and workers' self-reported language skills. We use these to build feature vectors used in tuning model and choosing the best translations from the candidates. POTENTIALLY: We extend Zaidan and Callison-Burch (2011)'s feature set to include additional bilingual features, which were not part of that original work.

2.2.1 Sentence-Level Features (9 Features)

This feature set contains language based features to solely implicate the quality of an English sentence without any suggestion on the bond of the meaning between the source sentence and the translation . This set of features tells good English sentences apart bad ones. The reason we use this set of features is that a good English sentence is the prerequisite of being a good English translation.

[1] Zaidan and Callison-Burch (2011) collected their translations in two batches. The first batch contained 1 translation, each with 1 post-edited version. The second contained an additional 3 translations, each of which was post-edited by 3 workers.

- Language model features: we assign a log probability and a per-word perplexity score for each sentence. We use SRILM toolkit to calculate perplexity score for each sentence based on 5-gram language model trained on English Gigaword corpus.

- Sentence length features: we use the ratio of the length of the Urdu source sentence to the length of the translation sentence as feature since a good translation is expected to be comparable in length with source sentence. We add two such ratio features(one is designed for unreasonably short translation and the other is for unreasonably long translation).

- Web n-gram log probability feature: we add the Web n-gram log probability feature to reflect the probability of the n-grams(up to length 5) exist in the Microsoft Web N-Gram Corpus. For short sentences whose length is less than 5, we use the sentence length as the order of the n-gram in calculation.

- Web n-gram geometric average features: we calculate the geometric average n-gram to evaluate the average matching over different n-grams. We add 3 features correspondent to max n-gram length of 3,4 and 5. Specifically, P_i denotes the log probability of i-gram and these 3 features are represented in $\sqrt[3]{P_1P_2P_3}$,$\sqrt[4]{P_1P_2P_3P_4}$ and $\sqrt[5]{P_1P_2P_3P_4P_5}$.

- Edit rate to other translations: In posterior methods, to minimize Bayes risk, we choose the translation that is most similar to other translations. Taking this into consideration, we add the edit rate feature to implement the similarity among all candidates translations.

2.2.2 Worker-Level Features

We take the quality of workers into consideration and add worker level features since the intuition that good workers can always high quality translations.

- Aggregate features: for each sentence level feature, we use the average values over all translations provided by the same worker as that worker's aggregate feature values.
- Language abilities: we collect worker's language ability information about whether the worker is a native Urdu speaker or native English speaker and how long they have spoken English or Urdu and add four features correspondent to the four aspects above.
- Worker Location: we add two features to indicate whether a worker is located in Pakistan or India.

2.2.3 Ranking Features

Zaidan and Callison-Burch (2011) collected 5 ranking labels for each translation and refine 3 features from these labels.

- Average Ranking: the average of the 5 ranking labels for this translation.
- Is-Best percentage: this feature shows how often a translation is ranked as the best translation among all candidates translation.
- Is-Better percentage: how often a translation is ranked as a better translation based on the pairwise comparisons.

2.2.4 Calibration Features

- Calibration features: 1 feature based on the average BLEU score of a worker's translations provided is computed against professional references.

We additionally introduce a new bilingual feature based on IBM Model 1. We align words between each candidate translation and its corresponding source sentence. The bilingual feature for a translation is the average of its alignment probabilities. In Figure 2.4, we show how the bilingual feature allows us to distinguish between a valid translation (top) and an invalid/spammy translation (bottom).

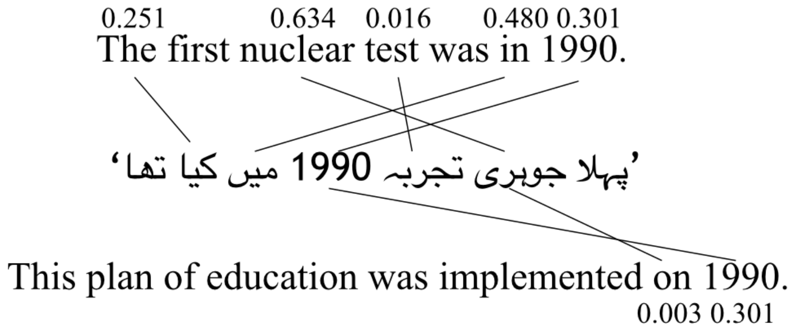

0.251 0.634 0.016 0.480 0.301
The first nuclear test was in 1990.
‘پہلا جوہری تجربہ 1990 میں کیا تھا’
This plan of education was implemented on 1990.
0.003 0.301

Figure 2.1: Example bilingual features for two crowdsourced translations of an Urdu sentence. The numbers are alignment probabilities for each aligned word. The bilingual feature is the average of these probabilities, thus 0.240 for the good translation and 0.043 for the bad translation. Some words are not aligned if potential word pairs don't exist in bilingual training corpus.

2.3 Supervised Learning in Machine Translation

Supervised learning methods can be used to discriminate good translations and bad translations, and train models to estimate the quality of translations. (Zaidan and Callison-Burch, 2011) proposed a framework to select the best translation among all candidates and achieved professional translating quality. They used MERT as the parameter tuning model. We extended their framework by using decision tree model and linear regression model.

2.3.1 MERT

Och (2003) proposed the Minimum Error Rate Training (MERT) framework in statistical machine translation. This framework is used to train models to score each translation and discriminate good translations and bad translations. Since each translation candidate is represented in feature vector format, the model is just a set of parameters corresponding to each feature. Given the n-best list translations of each source sentence and their corresponding professional references, instead of searching the huge space for all parameters, they used Powell algorithm (PoWell,) in the parameter tuning process where every time they only change the value of one parameter and detect the performance based on that value.

Suppose the feature vector used to represent the translation candidate x is defined as:

$$H(x) = \{h_1(x), h_2(x), .., h_n(x)\} \tag{2.3.1}$$

and in the log-linear model, the overall translation probability (quality) is predicted as:

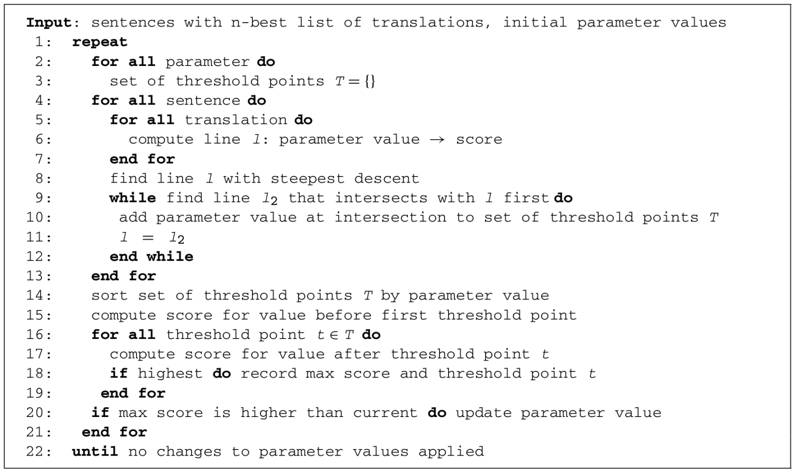
$$p(x) = \exp\sum_{i=1}^{n} \lambda_i h_i(x) \tag{2.3.2}$$

where λ_i is the parameter for the i_{th} feature. In Powell Search, if we want to search for the best value of feature $h_c(x)$ in some iteration, then the probability of that translation could be represent as:

$$p(x) = \exp(\lambda_c h_c(x) + u(x)) \tag{2.3.3}$$

$$u(x) = \sum_{i \neq c} \lambda_i h_i(x) \tag{2.3.4}$$

Each translation is a line with a slope of $h_c(x)$ and an offset of $u(x)$ in a 2-dimensional space. Thus, for the n-best translation candidates, we have n lines in the space and the top line means the corresponding translation has the highest model predicted probability. However, as the value of λ_c changes, the top line may also change since there might be intersects among these lines. Thus, there exists several intervals for the value of λ_c and for each interval, there is a particular top line which means when the value of λ_c belongs to that interval, the corresponding translation has the highest model predicted score. These intersects are called threshold points. For every value v that could be assigned to λ_c, we could rank the n-best translations for each source sentence in the training set based on the metric of $p(x) = \exp(v \cdot h_c(x) + u(x))$, select the top translation for each source sentence and calculate the quality score for these translations against professional translations in some evaluation metric, such as BLEU. Our goal is to find the best value for λ_c that results the highest quality score for those top translations we select for each source sentence. Even though we only search for the best value searching for the best value for a single parameter,it still costs lots of time, especially when the parameter could be in real numbers. However, we know that for each source sentence, the top line only changes at threshold points, which means we only have to search for the best value of λ_c in a finite state set. Figure 2.2 is the framework (Koehn, 2009) for MERT to tune the parameter.



```
Input: sentences with n-best list of translations, initial parameter values
repeat
  for all parameter do
    set of threshold points $T = \{\}$
  for all sentence do
    for all translation do
      compute line $l$: parameter value → score
    end for
    find line $l$ with steepest descent
    while find line $l_2$ that intersects with $l$ first do
     add parameter value at intersection to set of threshold points $T$
      $l = l_2$
    end while
  end for
  sort set of threshold points $T$ by parameter value
  compute score for value before first threshold point
  for all threshold point $t \in T$ do
    compute score for value after threshold point $t$
    if highest do record max score and threshold point $t$
   end for
  if max score is higher than current do update parameter value
 end for
until no changes to parameter values applied
```

Figure 2.2: The Framework for the parameter tuning process using Powell search.

2.3.2 Decision Tree

Decision Tree is a classical machine learning model in classification and regression. In our case, we only introduce the regression tree, which is very similar to classification tree. The basic framework to train a regression tree is partition since we want to divide the data based on some attributes so that data in the same sub-division has similar property (label). The framework to grow a decision tree is shown below:

1. Start with an empty tree.

2. If the stopping rule is not satisfied, make partition on next best feature selected by variance reduction.

3. Recursion on each leaf.

Variance reduction is a splitting criterion to evaluate the effectiveness of the best feature and the splitting threshold for that feature. At node t, we want to maximize the variance reduction $\Delta i(s,t)$ by choosing the best split s. $\Delta i(s,t)$ is defined as:

$$\Delta i(s,t) = i(t) - P_L i(t_L) - P_R i(t_R) \tag{2.3.5}$$

$$\Delta i(t) = \frac{\sum_{n \in h(t)} w_n f_n (y_n - \bar{y}(t))^2}{\sum_{n \in h(t)} w_n f_n} \tag{2.3.6}$$

$$P_L = \frac{N_w(t_L)}{N_w(t)} \tag{2.3.7}$$

$$P_R = \frac{N_w(t_R)}{N_w(t)} \tag{2.3.8}$$

$$N_w(t) = \sum_{n \in h(t)} w_n f_n \tag{2.3.9}$$

$$\bar{y}(t) = \frac{\sum_{n \in h(t)} w_n f_n y_n}{N_w(t)} \tag{2.3.10}$$

where $h(t)$ is the learning samples at node t,w_n is the weight associated with sample n, f_n is the frequency weight associated with sample n. The splitting process stops when a node becomes pure, all samples have the same set of input attributes, the variance reduction is less than some user set threshold and so on.

2.3.3 Linear Regression

Linear Regression is a linear model based on the goal to reduce the residual squared error. It's an approach to model the relationship between a scalar variable y and the corresponding feature vector x. From a matrix perspective, given a set of feature matrix X and its corresponding label vector $\vec{y}$, the model is $w = (X^T X)^{-1} X^T \vec{y}$.

2.4 Experiments

We extended (Zaidan and Callison-Burch, 2011)'s framework using different models trained on different feature sets. We use 10% of the whole data set as the training set and use the rest as the test set. Each source sentence has four translations in total.

We evaluate the translation quality in BLEU. We report results based on five-fold cross validation.

2.4.1 Baseline

Random selection is set as the baseline method which select a translation among all four translations randomly and achieves a BLEU score of 29.56. We also perform an Oracle experiment to select the best professional translation among all four references by comparing one translation against other three translations and selecting the one with highest similarity to others. Oracle experiment achieves a BLEU score of 42.38.

2.4.2 MERT

We replicate (Zaidan and Callison-Burch, 2011)'s framework on MERT with the new added bilingual feature. Table 2.1 shows the translation quality.

Feature Set	BLEU Score
(S)entence features	38.51
(W)orker features	37.89
(R)anking features	36.74
(C)alibration feature	38.27
S+W+R features	38.44
S+W+R+B features	38.80
All features	39.47

Table 2.1: The translation quality for MERT

2.4.3 Decision Tree

We use the Decision Tree model to substitute the MERT model in the original framework. Table 2.2 shows translation quality. We visualize the decision tree we trained using all features. Figure 2.3 shows the visualization. In the visualization graph, label names is the shorten form of the feature names. Table 2.4.3, 2.4.3,2.4.3,2.4.3

Feature Set	BLEU Score
(S)entence features	35.32
(W)orker features	37.59
(R)anking features	36.17
(C)alibration feature	38.27
S+W+R features	37.04
S+W+R+B features	37.00
All features	37.19

Table 2.2: The translation quality for Decision Tree

show label names and the corresponding feature names.

Sentence-Level Features	
LOGPROB	Sentence Log Probabilty
AVGPPL	Per-Word Perplexity Score
LengthRatio1	Length Ratio Feature 1
LengthRatio2	Length Ratio Feature 2
NGramMatch	Web N-Gram Log Probability Feature
Root3	Web 3-Gram Geometric Average Feature
Root4	Web 4-Gram Geometric Average Feature
Root5	Web 5-Gram Geometric Average Feature
AvgTER	Edit Rate Feature

Table 2.3: Labels for sentence-level features

2.4.4 Linear Regression

2.5 Quality Control Analysis

Compared to the baseline method, the MERT model, Decision Tree Model and Linear Regression Model all achieve much better performance, which means.

Worker-Level Features	
AGLOGPROB	Workers' Aggregate Feature of Sentence Log Probabilty
AGAVGPPL	Workers' Aggregate Feature of Per-Word Perplexity Score
AGLengthRatio1	Workers' Aggregate Feature of Length Ratio 1
AGLengthRatio2	Workers' Aggregate Feature of Length Ratio 2
AGNGramMatch	Workers' Aggregate Feature of Web N-Gram Log Probability
AGRoot3	Workers' Aggregate Feature of Web 3-Gram Geometric Average
AGRoot4	Workers' Aggregate Feature of Web 4-Gram Geometric Average
AGRoot5	Workers' Aggregate Feature of Web 5-Gram Geometric Average
AGAvgTER	Workers' Aggregate Feature of Edit Rate
EngNative	Is an English Native Speaker
UrduNative	Is an Urdu Native Speaker
LocationIndia	Is the Worker in India
LocationPakistan	Is the Worker in Pakistan
YearEng	How Long the Worker Speaking English
YearUrdu	How Long the Worker Speaking Urdu

Table 2.4: Labels for worker-level features

Ranking Features	
AvgRank	Average Ranking Features
IsBetterP	How Often a Translation Is Ranked as The Best Translation
IsBestP	How Often a Translation Is Ranked as A Better Translation

Table 2.5: Labels for ranking features

Calibration Features & Bilingual Features	
Cali	Calibration Feature
Bilin	Bilingual Feature

Table 2.6: Labels for calibration and bilingual features

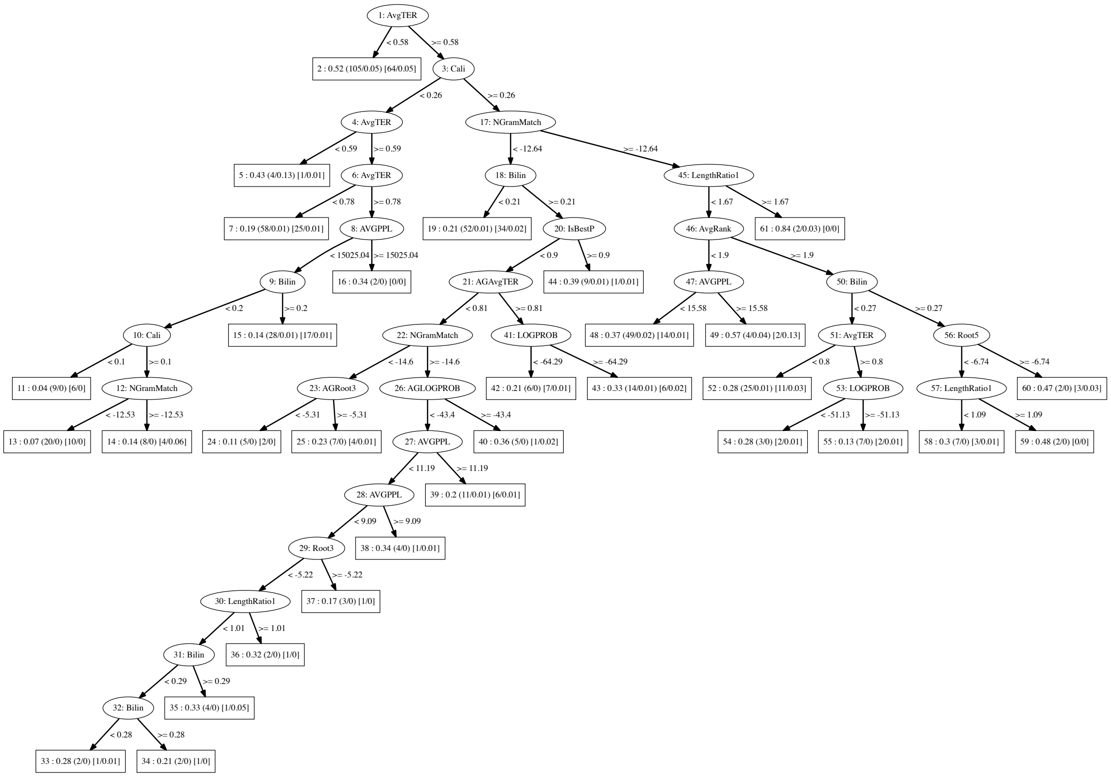


Figure 2.3: The visualization for the Decision Tree Model.

Feature Set	BLEU Score
(S)entence features	37.84
(W)orker features	36.92
(R)anking features	35.69
(C)alibration feature	38.27
S+W+R features	38.69
S+W+R+B features	39.23
All features	39.80

Table 2.7: The translation quality for Linear Regression

2.6 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 noisy 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).

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.

2.7 Problem Setup

We start with a corpus of source sentences to be translated, and we may solicit one or more translation for every sentence in the corpus. Our goal is to assemble a single high quality translation for each source sentence while minimizing the associated cost.

We study the data collected by Zaidan and Callison-Burch (2011) through Amazon's Mechanical Turk. They hired Turkers to translate 1792 Urdu sentences from the 2009 NIST Urdu-English Open Machine Translation Evaluation set[2]. A total of 52 Turkers contributed translations. Turkers also filled out a survey about their language skills and their countries of origin. Each Urdu sentence was translated by 4 non-professional translators (the Turkers) and 4 professional translators hired by the LDC. The cost of non-professional translation is $0.10 per sentence and the cost of professional translation is approximately $0.30 per word (or $6 per sentence, since they are 20 words long on average).

Following Zaidan and Callison-Burch (2011), we use BLEU (Papineni et al., 2002) to gauge the quality of human translations. We can compute the expected quality of professional translation by comparing each of the professional translators against the other 3. This results in an average BLEU score of 42.38. By comparison, the Turker translations score only 28.13 on average. Zaidan and Callison-Burch trained a MERT model to select one non-professional translation out of the four and pushed the quality of crowdsourcing translation to 39.06, closer to the expected quality of professional translation. They used a small amount of professional translations (10%) as calibration data to estimate the goodness of the non-professional translation. The component costs of their approach are the 4 non-professional translations for each source sentence, and the professional translations for the calibration data.

Although Zaidan and Callison-Burch demonstrated that non-professional translation was significantly cheaper than professionals, we are interested in further reducing the costs. This plays a role if we would like to assemble a large enough parallel cor-

[2] LDC Catalog number LDC2010T23

pus (on the order of millions of sentence translations) to train a statistical machine translation system. Here, we introduce several methods for reduce the number of non-professional translations while still maintaining high quality.

2.8 Estimating Translation Quality

We use a linear regression model[3] to predict a quality score ($score(t) \in R$) for an input translation t.

$$score(t) = \vec{w} \cdot \vec{f}(t)$$

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

We replicate the feature set used by Zaidan and Callison-Burch (2011) in their MERT model:

- Sentence-level features: 9 features based on language model, sentence length, edit distance to other translations.

- Worker-level features: 15 features based on worker's language ability, location and average sentence-level scores.

- Ranking features: 3 features based on the judgments of monolingual English speakers ranking the translations from best to worst.

- Calibration features: 1 feature based on the average BLEU score of translations provided by the same worker, which is computed against professional references.

We additionally introduce a new bilingual feature based on IBM Model 1. We align words between each candidate translation and its corresponding source sentence. The bilingual feature for a translation is the average of its alignment probabilities. In Figure 2.4, we show how the bilingual feature allows us to distinguish between a valid translation (top) and an invalid/spammy translation (bottom).

[3] We used WEKA package: `http://www.cs.waikato.ac.nz/ml/weka/`

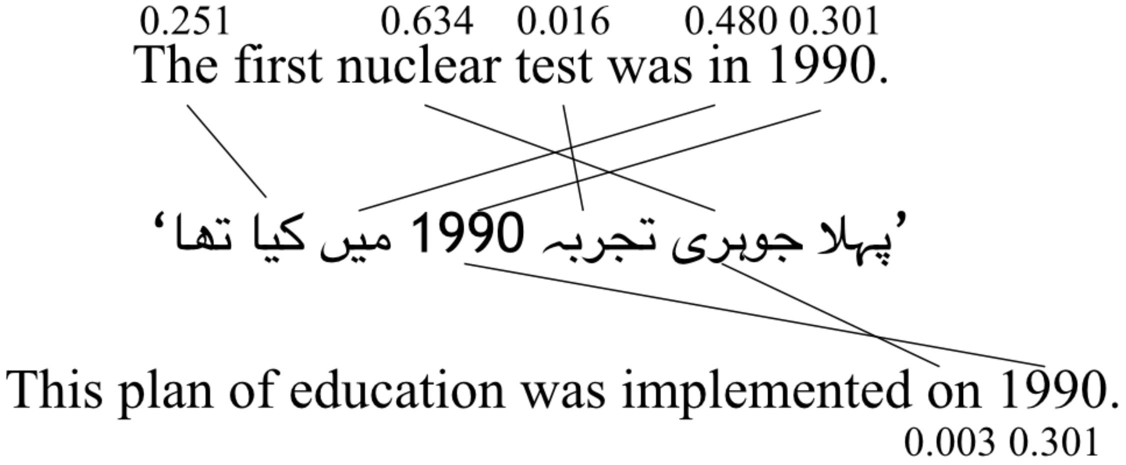

0.251
0.634
0.016
0.480
0.301
The first nuclear test was in 1990.
‘پہلا جوہری تجربہ 1990 میں کیا تھا’
This plan of education was implemented on 1990.
0.003
0.301

Figure 2.4: Example bilingual feature for two crowdsourced translations for a sentence in Urdu. The numbers are alignment probabilities for each aligned word. The bilingual feature is the average of these probabilities, thus 0.240 for the good translation and 0.043 for the bad translation. Some words are not aligned if potential word pairs don’t exist in corpus.

Algorithm 1

Input: δ, the allowable deviation from the expected upper bound on BLEU score (using all redundant translations); α, the upper bound BLEU score; a training set $S = \{\vec{f}^s_{i,j}, y^s_{i,j})^{j=1..4}_{i=1..n}\}$ and a validation set $V = \{(\vec{f}^v_{i,j}, y^v_{i,j})^{j=1..4}_{i=1..n}\}$ where $\vec{f}_{i,j}$ is the feature vector for $t_{i,j}$ which is the jth translation of the source sentence s_i and $y_{i,j}$ is the label for $\vec{f}_{i,j}$.

Output: θ, the threshold between acceptable and unacceptable translations; $\vec{w}$, a linear regression model parameter.

```
initialize $\theta \leftarrow 0$,$\vec{w} \leftarrow \emptyset$
$\vec{w'} \leftarrow$ train a linear regression model on $S$
$maxbleu \leftarrow$ select best translations for each $s_i \in S$ based on the model parameter
    $\vec{w'}$ and record the highest model predicted BLEU score
while $\theta \neq maxbleu$ do
    for $i \leftarrow 1$ to $n$ do
        for $j \leftarrow 1$ to 4 do
            if $\vec{w'} \cdot \vec{f}^v_{i,j} > \theta \wedge j < 4$ then select $t^v_{i,j}$ for $s_i$ and break
            if $j == 4$ then select $t^v_{i,j}$ for $s_i$
    $q \leftarrow$ calculate translation quality for V
    if $q > \delta \cdot \alpha$ then break
    else $\theta = \theta + stepsize$
$\vec{w} \leftarrow$ train a linear regression model on $S \cup V$
Return: $\theta$ and model parameter $\vec{w}$
```

2.9 Reducing the Number of Translations

The first mechanism that we introduce to optimize cost is one that reduces the number of translations. Our goal is to recognize when we have got a good translation of a source sentence and to immediately stop purchasing additional translations of that sentence. The crux of this method is to decide whether a translation is 'good enough,' in which case we do not gain any benefit from paying for another redundant translation.

Our translation reduction method allows us to set an empirical definition of 'good enough'. We define an oracle upper bound α to be the estimated BLEU score using the full set of non-professional translations. We introduce a parameter δ to set how much degradation in translation quality is allowable. For instance, we may fix δ at 95%, meaning that the resulted BLEU score should not drop below 95% of the α after reducing the number of translations. We train a model to search for a threshold θ between acceptable and unacceptable translations for a specific value of δ.

For a new translation, our model scores it, and if its score is higher than θ, then we do not solicit another translation. Otherwise, we continue to solicit translations. Algorithm 1 details the process of model training and searching for θ.

2.9.1 Experiment

We divide data into a training set (10%), a validation set (10%) and a test set (80%). We use the validation set to search for θ. The upper bound BLEU is set to be 40.13 empirically as the oracle upper bound. We then vary the value of δ from 90% to 100%, and sweep values of θ by incrementing it in step sizes of 0.01. We report results based on a five-fold cross validation, rotating the training, validation and test sets.

Baseline and upper bound

The baseline selection method of randomly picking one translation for each source sentence achieves a BLEU score of 29.56. To establish an upper bound on translation quality, we perform an oracle experiment selects best translation for each source segment. It reaches a BLEU score of 40.13.

Translation reducing method

Table 2.8 shows the results for translation reducing method. The δ variable correctly predicts the deviation in BLEU score when compared to using the full set of

δ(%)	BLEU Score	# Trans.
90	36.26	1.63
91	36.66	1.69
92	36.93	1.78
93	37.23	1.85
94	37.48	1.93
95	38.05	2.21
96	38.16	2.30
97	38.48	2.47
98	38.67	2.59
99	38.95	2.78
100	39.54	3.18

Table 2.8: The relation among the δ (the allowable deviation from the expected upper bound on BLEU score), the BLEU score for translations selected by models from partial sets and the averaged size of translation candidates set for each source sentence (# *Trans*).

translations. If we set δ <0.95 then we lose 2 BLEU points, but we cut the cost of translations in half, since we pay for only two translations of each source segment on average.

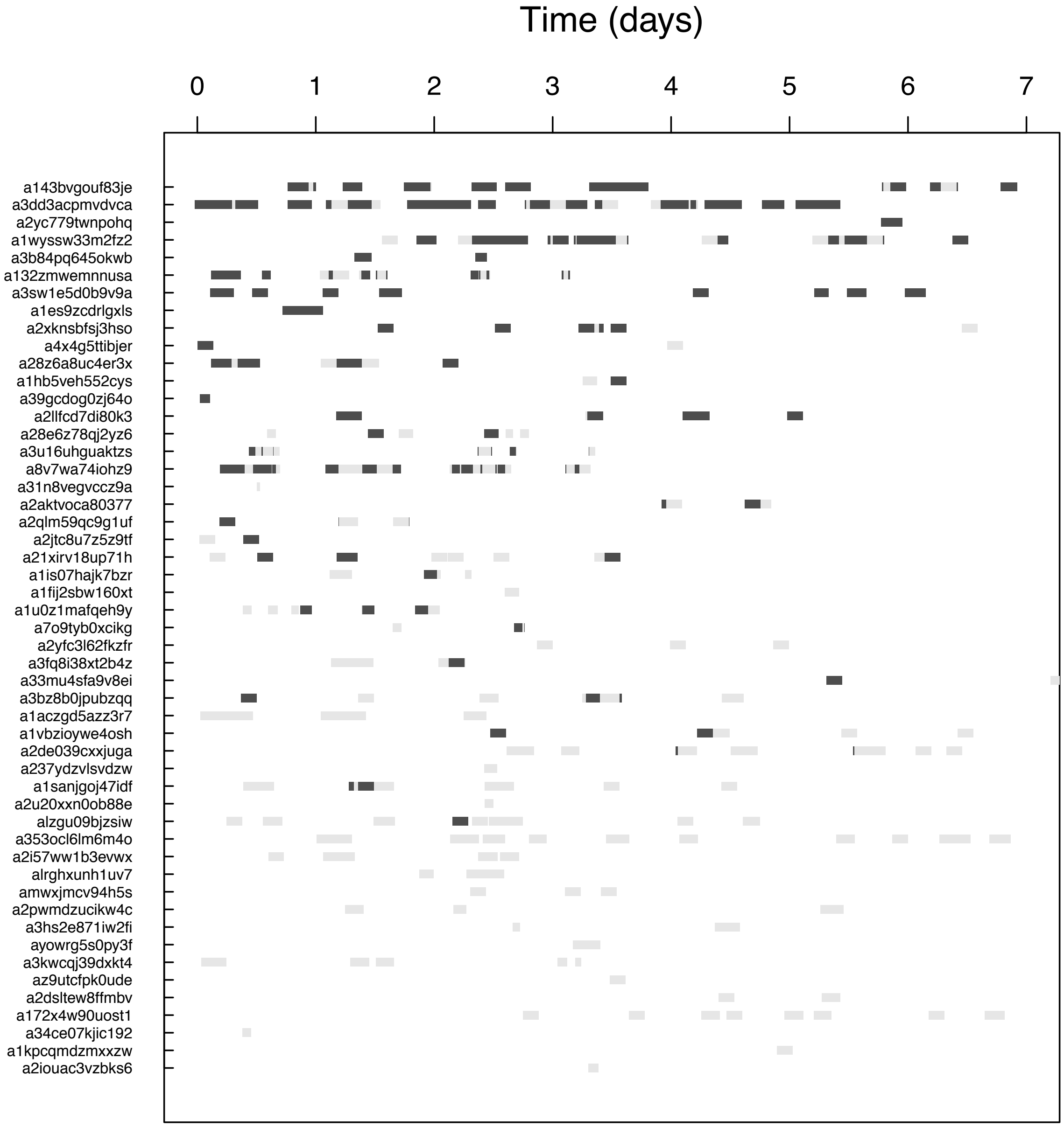


Figure 2.5: A time-series plot of all of the translations produced by Turkers (identified by their WorkerID serial number). Turkers are sorted with the best translator at the top of the y-axis. Each tick represent a single translation and dark color means better than average quality.

2.10 Choosing Better Translators

The second mechanism that we use to optimize cost is to reduce the number of non-professional translators that we hire. Our goal is to quickly identify whether Turkers are good or bad translators, so that we can continue to hire only the good translators and stop hiring the bad translators after they are identified as such. Before presenting our method, we first demonstrate that Turkers produce consistent quality translations over time.

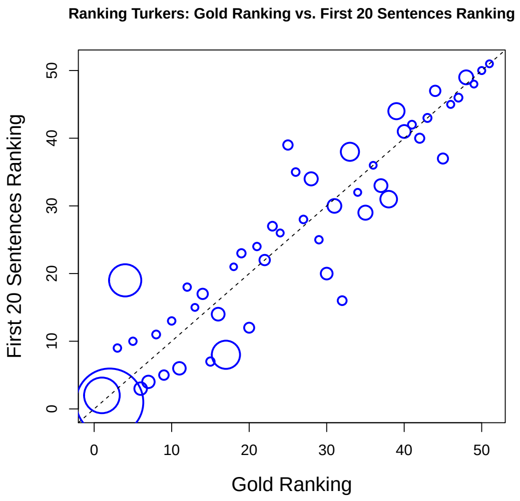
2.10.1 Turkers' behavior in translating sentences

Do Turkers produce good (or bad) translations consistently or not? Are some Turkers consistent and others not? We used the professional translations as a gold-standard to analyze the individual Turkers, and we found that most Turkers' performance stayed surprisingly consistent over time.

Figure 2.5 illustrates the consistency of workers' quality by plotting quality of their individual translations on a timeline. The translation quality is computed based on the BLEU against professional translations. Each tick represent a single translation and depicts the BLEU score using two colors. The tick is black if its BLEU score is higher than the median and it is light grey otherwise. Good translators tend to produce consistently good translations and bad workers rarely produce good translations.

2.10.2 Evaluating Rankings

We use weighted Pearson correlation (Pozzi et al., 2012) to evaluate our ranking of workers against gold standard ranking. Since workers translated different number of sentences, it is more important to rank the workers who translated more sentences correctly. Taking the importance of workers into consideration, we set a weight to each worker using the number of translations he or she submitted when calculating the correlation. Given two lists of worker scores x and y and the weight vector w, the



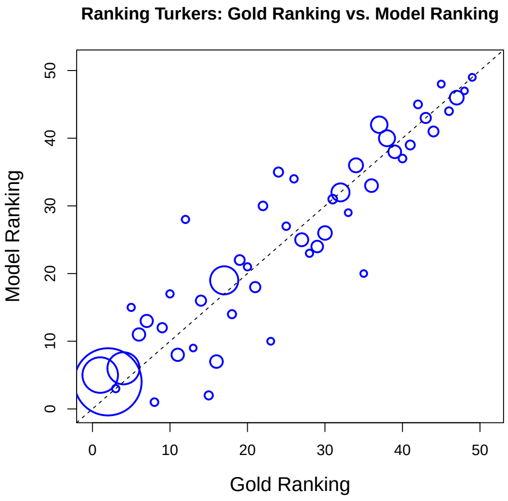


Figure 2.6: Correlation between gold standard ranking and ranking computed using the first 20 sentences as calibration. Each bubble represents a worker. The radius of each bubble shows the relative volume of translations completed by the worker. The weighted correlation is 0.94.

Figure 2.7: Correlation between gold standard ranking and our model's ranking. The corresponding weighted correlation is 0.95.

weighted Pearson correlation ρ can be calculated as:

$$\rho(x, y; w) = \frac{cov(x, y; w)}{\sqrt{cov(x, x; w) cov(y, y; w)}} \tag{2.10.1}$$

where *cov* is weighted covariance:

$$cov(x, y; w) = \frac{\sum_i w_i (x_i - m(x; w))(y_i - m(y; w))}{\sum_i w_i} \tag{2.10.2}$$

and m is weighted mean:

$$m(x; w) = \frac{\sum_i w_i x_i}{\sum_i w_i} \tag{2.10.3}$$

2.10.3 Automatically Ranking Translators

We introduce two approaches to rank workers using a small portion of work they submitted. Our goal is to filter out bad workers, and to select the best translation from translations provided by the remaining workers.

Ranking workers using their first k translations We rank the Turkers using the first few translations that they provide by comparing their translations to the professional translations of those sentences. Ranking workers on gold standard data would allow us to discard bad workers. This is similar to the idea of a qualification test in MTurk.

Ranking workers using a model In addition to ranking workers by comparing them against a gold standard, we also predict their ranks with a model. We use the 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 $score(t)$. The model predicted performance of the worker w is:

$$performance(w) = \frac{\sum_{t \in T_w} score(t)}{|T_w|} \tag{2.10.4}$$

where T_w is the set of translations completed by the worker w.

2.10.4 Experiments

After we rank workers, we keep top-ranked workers and select the best translation only from their translations. For both ranking approaches, we vary the number of good workers that we retain.

We report ranking's correlation to gold standard ranking and translation quality. Since the top worker threshold is varied and we change the value of k in first k sentence ranking, we have a different test set in different settings. Each test set exclude any item that was used to rank the workers, or which did not have any translations from the top workers according to our rankings.

In addition to evaluating the correlation of our different ways of ranking translators, we also compute the translation quality when selecting the translation provided by the worker with best rank.

Gold standard and Baseline

We evaluate ranking quality using the weighted Pearson correlation (ρ) compared with the gold standard ranking of workers. To establish the gold standard ranking, we score each Turker based on the average BLEU score of all his or her translations against professional references.

We use the ranking by the MERT model developed by Zaidan and Callison-Burch (2011) as baseline. It achieves a correlation of 0.73 against the gold standard ranking.

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 top workers. To evaluate this method, we calculate the weighted correlation for our rankings against gold ranking.

Table 2.9 shows the results of Pearson correlations for different value of k. As k increases, our rankings fit to the gold ranking better. Consequently, we can decide whether to continue to hire a worker in a very short time after analyzing the first k sentences ($k \leq 20$) provided by each worker. Figure 2.6 shows the correlation of gold ranking and first 20 sentences ranking.

Ranking workers using a model

We train a linear regression model on 10% of the data to rank workers. We use the model to select the best translation in one of two ways:

- By using the model's prediction of workers' rank, and selecting the translation from the best worker.

- By using the model's score for each translation and selecting the highest scoring translation of each source sentence.

Table 2.10 shows that the model trained on all features achieves a very high correlation with the gold standard ranking (Pearson's $\rho = 0.95$), and a BLEU score of 39.80.

Proportion of Calibration Data		ρ
First k sentences	Percentage	
1	0.7%	0.21
2	1.3%	0.38
3	2.0%	0.41
4	2.7%	0.56
5	3.3%	0.70
10	6.6%	0.81
20	13.3%	0.94
30	19.9%	0.96
40	26.6%	0.98
50	33.2%	0.98
60	39.8%	0.98

Table 2.9: Pearson Correlations for calibration data in different proportion.

Figure 2.7 presents a visualization of the gold ranking and model ranking. The workers who produce the largest number of translations (large bubbles in the figure) are ranked extremely well.

2.10.5 Filtering out bad workers

Ranking translators would allow us to reduce costs, by only re-hiring top workers. Table 2.11 shows what happens when we vary the top percentage of workers we retain. In general, the model does a good job of picking the best translations from the remaining good translators. Comparing against knowing the gold ranking, the model loses only 0.55 BLEU when we filter out 75% of the workers. In this case we only need to solicit two translations for each source sentence on average.

		BLEU	
Feature Set	ρ	rank	score
(S)entence features	0.80	36.66	37.84
(W)orker features	0.78	36.92	36.92
(R)anking features	0.81	36.94	35.69
Calibration features	0.93	38.27	38.27
S+W+R features	0.86	37.39	38.69
S+W+R+Bilingual features	0.88	37.59	39.23
All features	**0.95**	**38.37**	**39.80**

Table 2.10: Correlation (ρ) and translation quality for the various features used by our model. Translation quality is computed by selecting best translations based on model-predicted workers' ranking (rank) and model-predicted translations' scores (score). Here we do not filter out bad workers when selecting the best translation.

Top	BLEU				
(%)	random	model	gold	Δ	# Trans
25	29.85	38.53	39.08	0.55	1.95
50	29.80	38.40	39.00	0.60	2.73
75	29.76	38.37	38.98	0.61	3.48
100	29.83	38.37	38.99	0.62	4.00

Table 2.11: A comparison of the translation quality when we retain the top translators under different rankings. The rankings show are random, the model's ranking (using all features from Table 2.10) and the gold ranking. Δ is the different between the BLEU scores for the gold ranking and the model ranking. # Trans is the average number of translations needed for each source sentence.

2.11 Cost Analysis

We have introduced several ways of significantly lowering the costs associated with crowdsourcing translations when a large amount of data are solicited (on the order of millions of samples):

- We show that after we have collected one translation of a source sentence, we can 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 for non-professionals here comes from reducing the number of redundant translations. We can save half of the cost associated with non-professional translations to get 95% of the translation quality using the full set of redundant translations.

- We show that we can quickly identify bad translators, either by having them first translate a small number of sentences to be tested against professional translations, or by estimating their performance using a feature-based linear regression model. The cost savings for non-professionals here comes from not hiring bad workers. Similarly, we reduce the non-professional translation cost to the half of the original cost.

- In both cases we need some amount of professionally translated materials to use as a gold standard for calibration. Although the unit cost for each reference is much higher than the unit cost for each non-professional translation, the cost associated with non-professional translations can dominate the total cost since the large amount of data need to be collected. Thus, we focus on reducing cost associated with non-professional translations.

2.12 Conclusion

In this paper, we propose two mechanisms to optimize cost: the translation reducing method and the translator reducing method. They have different applicable scenarios for large corpus construction. The translation reducing method works if there exists a specific requirement that the quality control must reach a certain threshold. This model is most effective when reasonable amounts of pre-existing professional translations are available for setting the models threshold. The translator reducing method is very simple and easy to implement. This approach is inspired by the intuition that workers' performance is consistent. The translator reducing method is suitable for crowdsourcing tasks which do not have specific requirements about the quality of the translations, or when only very limited amounts of gold standard data are available.

References

Vamshi Ambati and Stephan Vogel. 2010. Can crowds build parallel corpora for machine translation systems? In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 62–65. Association for Computational Linguistics.

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.

Cart algorithm.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Christopher H Lin, Mausam, and Daniel S Weld. 2014. To re (label), or not to re (label). In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing*. Association for the Advancement of Artificial Intelligence (AAAI).

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.

MJD PoWell. An efficient method for finding the minimum of a function of several variables without calculating derivatives.

F Pozzi, T Di Matteo, and T Aste. 2012. Exponential smoothing weighted correlations. *The European Physical Journal B-Condensed Matter and Complex Systems*, 85(6):1–21.

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.