CROWDSOURCING MACHINE TRANSLATION

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

Lyle Ungar
Graduate Group Chairperson

Acknowledgments

I would like to thank my thesis advisor, Prof. Chris Callison-Burch, for his patient guidance on my thesis work. He is very inspiring and uplifting. I would also like to thank Dr. Wei Xu (a Postdoctoral Researcher in the CIS department of the University of Pennsylvania) for her support in my research work. Finally, I would like to thank Mike Felker and Betty Gentner for making administrative matters run smoothly.

Abstract

Crowdsourcing makes it possible to create translations at much lower cost than hiring professional translators. We achieve similar translation quality compared with professional translations using different machine learning models to select the best translation among several candidate translations provided by non-professional translators. However, it is still expensive to obtain the millions of translations that are needed to train high performance statistical machine translation systems. We propose 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 selected 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 that quickly identifies 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. These two mechanisms work well on a previously studied set of Urdu translations and save the cost associated with data collection. In addition, we extend the translation reducing method to the Tamil translation data and achieve similar cost reduction effect.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As the globalization process continues across the world, the demand for translation increases. This facilitates the improvement in the quality of machine translation (MT) systems. Through online systems such as Google Translate, machine translation has become widely used in people's daily life. However, to build a machine translation system, large amounts of bilingual training data, called bilingual parallel corpora, is necessary. For some 'high resource' languages, such as French or Spanish, there is an abundance of parallel data that can be used to train machine translation systems. However, for other so-called 'low resource' languages, obtaining a sufficiently large bilingual corpus is a big issue. Although bilingual training data are normally created as a byproduct of some other human industries (for instance, some of them are created by the European Union which translates its official documents into all of the languages of its memberships), we might consider hiring profes-

sional translators or linguists to translate documents from foreign languages into English in order to build a training corpus for our machine translation system. This would ensure good translation quality of the corpus. But there are two limitations for this approach:

1. The cost of hiring professional translators is prohibitively high, especially for the large amounts of data need to train an MT system.

2. For very low resource languages, it is sometimes hard to find professional translators.

Crowdsourcing is a mechanism to collect data from a large population at relatively low cost. It parallelizes the creation of the data across a large number of people (the crowd). The popularization of the Internet makes it possible to do crowdsourcing tasks from many places across the globe. Anyone can be a crowd worker as long as he or she has access to the Internet. This makes crowdsourcing a promising mechanism for 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 good fit for crowdsourced data construction, since it needs a large volume of bilingual training data. This thesis examines two aspects of crowdsourcing for machine translation: quality control (Zaidan and Callison-Burch, 2011) and cost optimization.

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 led 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 results. We validate that such models can be used to perform effective quality control for crowdsourced translation.

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, the cost is still potentially very high. For example, supposed that we have a corpus containing one million sentences, the estimated cost for translating one source sentence is $0.10. If we plan to solicit one set of non-professional translations for each source sentence, 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. In this thesis, we explore methods to minimizing the associated cost while maintaining the same level of translating quality.

In Zaidan and Callison-Burch (2011)'s framework, a fixed number of redundant translations are solicited for each source sentence. One of our cost reduction methods is based on the intuition that we may receive a good translation early. If we can identify when we have received a high quality translation, then we don't have to collect additional redundant

translations of the source sentence. Our mechanism reduces 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.

Another cost reduction method makes a prediction about who are good translators and who are bad translators. Our analysis shows that workers' performance is consistent over time. Thus, if we can quickly identify bad workers after soliciting only a few translations from them, then we can filter them out as soon as possible. In this way, we save the cost by avoiding re-hiring bad workers.

1.3 Extension for Cost Optimization

We extend the translation reducing framework to the Tamil data. For each Tamil source sentence, four non-professional translations are collected and non-professional workers are hired to select the best translation among them. Since we don't have gold standard references to calibrate, we propose a method to label each non-professional translation based on a second-pass ranking by native English speakers. The experiment shows that the translation reducing method works well on Tamil data and reduces the cost to collect data, even in absence of professional translations to use as gold standard calibration data.

1.4 Main Contribution

The main contributions of this thesis are:

- 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.
- In addition to Urdu, our model works well on the Tamil translation data.

Chapter 2

Literature Review

Quality control is an important issue for crowdsourcing since anonymous workers whose skills are unknown and small financial incentives encourage workers to participate even if they do not have appropriate skills.

Snow et al. (2008) were the first to research the efficacy of crowdsourcing for natural language processing. They explored ways of making the quality of non-expert annotation achieve the quality that we would expect from professional annotators. In particular, they showed that redundant non-expert annotation and majority voting led to an expert-like annotation in several annotation tasks, such as word sense disambiguation and word similarity annotation. Moreover, after using a small amount of gold standard data for calibration, they reduced the amount of redundancy required and achieved expert-level annotation with fewer non-expert annotators.

Sheng et al. (2008) were interested in training a model using crowdsourced labels. Their work on repeated labeling presented 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, Sheng et al. (2008) proposed a framework for repeated-labeling that resolves the uncertainty in labeling via majority voting to keep the model trained on noisy labeled data having a high accuracy in predicting, . The experimental results showed that a model's 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. They applied it to the problem of word sense annotation. Passonneau and Carpenter (2013) also proposed an approach to detect and to avoid spam workers. They measured the performance of workers by comparing workers' labels to the current majority labels. Workers with bad performance can be identified and blocked.

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 showed that if the goal was to train a classifier on the labels, that the properties of the classifier would determine whether it was better to re-label data or to 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.

Chapter 3

Quality Control for Crowdsourcing Machine Translation

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 extend their framework to other models.

3.1 Data Collection

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[1]. 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 Linguistic Data Consortium (LDC). The cost of non-professional translation was $0.10 per sentence and we estimate the cost of professional translation to be approximately $0.30 per word (or $6 per sentence, with 20 words on average).

3.2 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 features to build feature vectors used in tuning model and choose the best translations from the candidates. We replicate Zaidan and Callison-Burch (2011)'s feature sets (sentence-level features, worker-level features, ranking features and calibration features) and extend to include additional bilingual features, which are not part of that original work.

3.2.1 Sentence-Level Features (9 Features)

This feature set contains language-based features that indicate the quality of an English sentence without making very much direct use of the original source sentence. This set of features tells good English sentences apart from bad ones. The reason why we use this

[1]LDC Catalog number LDC2010T23

set of features is that a good English sentence is the prerequisite of being a good English translation.

- Language model features: we assign a log probability and a per-word perplexity score to each sentence. We use SRILM toolkit to calculate a 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 source sentence to the length of the translation sentence as a 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 translations and the other is for unreasonably long translations).

- 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) in the Microsoft Web N-Gram Corpus. For short sentences whose length are 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 use 3 features corresponding to max n-gram order of 3,4 and 5. Specifically, P_i denotes the maximum log probability of i-gram for a translation 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 the most similar to other translations. Taking this into consideration, we add the edit rate feature to implement the similarity among all candidate translations.

3.2.2 Worker-Level Features (15 Features)

We take the quality of workers into consideration and add worker level features based on the intuition that good workers are more likely to generate 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: Zaidan and Callison-Burch (2011) asked each worker questions about their language abilities. They asked whether the worker was a native Urdu speaker or a native English speaker, and how long they had spoken English or Urdu. We add four features corresponding to the four questions.

- Worker Location: we add two binary features to indicate whether a worker is located in Pakistan or India.

3.2.3 Ranking Features (3 Features)

Zaidan and Callison-Burch (2011) collected 5 ranking labels for each translation and refined 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 candidate translations.
- Is-Better percentage: how often a translation is ranked as a better translation based on the pairwise comparisons.

3.2.4 Calibration Features (1 Feature)

- Calibration features: 1 feature shows the average BLEU score of a worker's translations when they are compared with professional references.

3.2.5 Bilingual Features (1 Feature)

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 3.1, we show how the bilingual feature allows us to distinguish between a valid translation (top) and an invalid/spammy translation (bottom).

3.3 Supervised Learning in Machine Translation

Koehn (2009) summarized statistical machine translation and showed that supervised learning methods can be used to discriminate good translations and bad translations, and to train models to estimate the quality of translations. Zaidan and Callison-Burch (2011) proposed

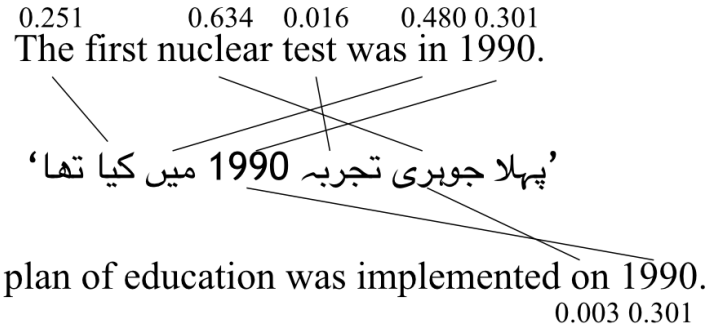

Figure 3.1: Example bilingual features for two crowdsourced translations of an Urdu sentence. The numbers are alignment probabilities for each aligned word. The bilingual feature is the average of these probabilities, thus 0.240 for the good translation and 0.043 for the bad translation. Some words are not aligned if potential word pairs don’t exist in bilingual training corpus.

a framework to select the best translation among all candidates and achieved professional translating quality. They used a parameter tuning method for machine translation, called MERT, to select the best translation. We extend their framework by using a decision tree model and a linear regression model.

3.3.1 MERT

Och (2003) proposed the Minimum Error Rate Training (MERT) framework for statistical machine translation. This framework is used to train models to score each translation and discriminate between 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, 1964) in the parameter tuning process where every time they only change the value of one parameter and accordingly detect the performance based on that value. We make a detailed instruction of MERT below based on the summary provided by Koehn (2009).

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

$$H(x) = \{h_1(x), h_2(x), .., h_n(x)\} \tag{3.3.1}$$

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

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x) \tag{3.3.2}$$

where λ_i is the parameter for the i_{th} feature. In Powell Search (Powell, 1964), 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:

$$p(x) = \exp(\lambda_c h_c(x) + u(x)) \tag{3.3.3}$$

$$u(x) = \sum_{i \neq c} \lambda_i h_i(x) \tag{3.3.4}$$

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 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 3.2 is the framework (Koehn, 2009) for MERT to tune the parameter.

3.3.2 Decision Tree

A Decision Tree (Breiman et al., 1984) is a classical machine learning model that is used for classification. We use a variant known as the regression tree, which is very similar to the classification tree. The basic framework to train a regression tree is partitioning. We want to divide the data based on some attributes so that data in the same sub-division has similar property (label). The framework (SPSS, 2011) to grow a decision tree is shown

```
Input: sentences with n-best list of translations, initial parameter values
 1:  repeat
 2:    for all parameter do
 3:      set of threshold points T = {}
 4:    for all sentence do
 5:      for all translation do
 6:        compute line l: parameter value → score
 7:      end for
 8:      find line l with steepest descent
 9:      while find line l_2 that intersects with l first do
10:       add parameter value at intersection to set of threshold points T
11:       l = l_2
12:     end while
13:   end for
14:   sort set of threshold points T by parameter value
15:   compute score for value before first threshold point
16:   for all threshold point t ∈ T do
17:     compute score for value after threshold point t
18:     if highest do record max score and threshold point t
19:    end for
20:   if max score is higher than current do update parameter value
21:  end for
22:  until no changes to parameter values applied
```

Figure 3.2: The framework for the parameter tuning process using Powell Search.

below:

1. Start with an empty tree.

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

3. Perform recursion on each leaf.

Variance reduction (SPSS, 2011) 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. $\Delta i(s,t)$ (SPSS, 2011) is defined as:

$$\Delta i(s,t) = i(t) - P_L i(t_L) - P_R i(t_R) \tag{3.3.5}$$

$$i(t) = \frac{\sum_{n \in h(t)} w_n f_n (y_n - \bar{y}(t))^2}{\sum_{n \in h(t)} w_n f_n} \tag{3.3.6}$$

$$P_L = \frac{N_w(t_L)}{N_w(t)} \tag{3.3.7}$$

$$P_R = \frac{N_w(t_R)}{N_w(t)} \tag{3.3.8}$$

$$N_w(t) = \sum_{n \in h(t)} w_n f_n \tag{3.3.9}$$

$$\bar{y}(t) = \frac{\sum_{n \in h(t)} w_n f_n y_n}{N_w(t)} \tag{3.3.10}$$

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.

3.3.3 Linear Regression

Linear Regression (Hastie et al., 2001) is a linear model whose goal is to reduce the residual squared error. It is 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 $w = (X^T X)^{-1} X^T \vec{y}$.

3.4 Experiments

We extend Zaidan and Callison-Burch (2011)'s framework using different models trained on different feature sets. We use 10% of the data set as the training set and use the rest

as the test set. Each source sentence has four non-professional translations from workers on Mechanical Turk. We evaluate the translation quality in BLEU by comparing the non-professional translations that our model selects against a set of four references translations created by the LDC. We report results based on five-fold cross validation.

3.4.1 Baseline

Random selection is used as the baseline method. If we randomly select a translation among all four translations then the BLEU score is 29.56. We also perform an Oracle experiment to calculate the average BLEU of one professional translation against the other three professional translations. Oracle experiment achieves a BLEU score of 42.89.

3.4.2 MERT

We replicate Zaidan and Callison-Burch (2011)'s framework on MERT with the new added bilingual feature. Table 3.1 shows the translation quality.

Feature Set	BLEU Score
(S)entence features	38.51
(W)orker features	37.89
(R)anking features	36.74
Calibration feature	38.27
S+W+R features	38.44
S+W+R+Bilingual features	38.80

All features	39.47

Table 3.1: The translation quality for MERT.

3.4.3 Decision Tree

We use the Decision Tree model to substitute the MERT model in the original framework. Table 3.2 shows the translation quality.

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

Table 3.2: The translation quality for Decision Tree.

We visualize the decision tree that we train by using all features. Figure 3.3 shows the visualization. In the visualization graph, label names are shorten form of the feature names. Table 3.3 shows label names and their corresponding feature names.

Sentence-Level Features	
LOGPROB	Sentence Log Probability
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
Worker-Level Features	
AGLOGPROB	Workers' Aggregate Feature of Sentence Log Probability
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
Ranking Features	
AvgRank	Average Ranking Features
IsBetterP	How Often a Translation Is Ranked as A Better Translation
IsBestP	How Often a Translation Is Ranked as The Best Translation
Calibration and Bilingual Features	
Cali	Calibration Feature
Bilin	Bilingual Feature

Table 3.3: Labels for features.

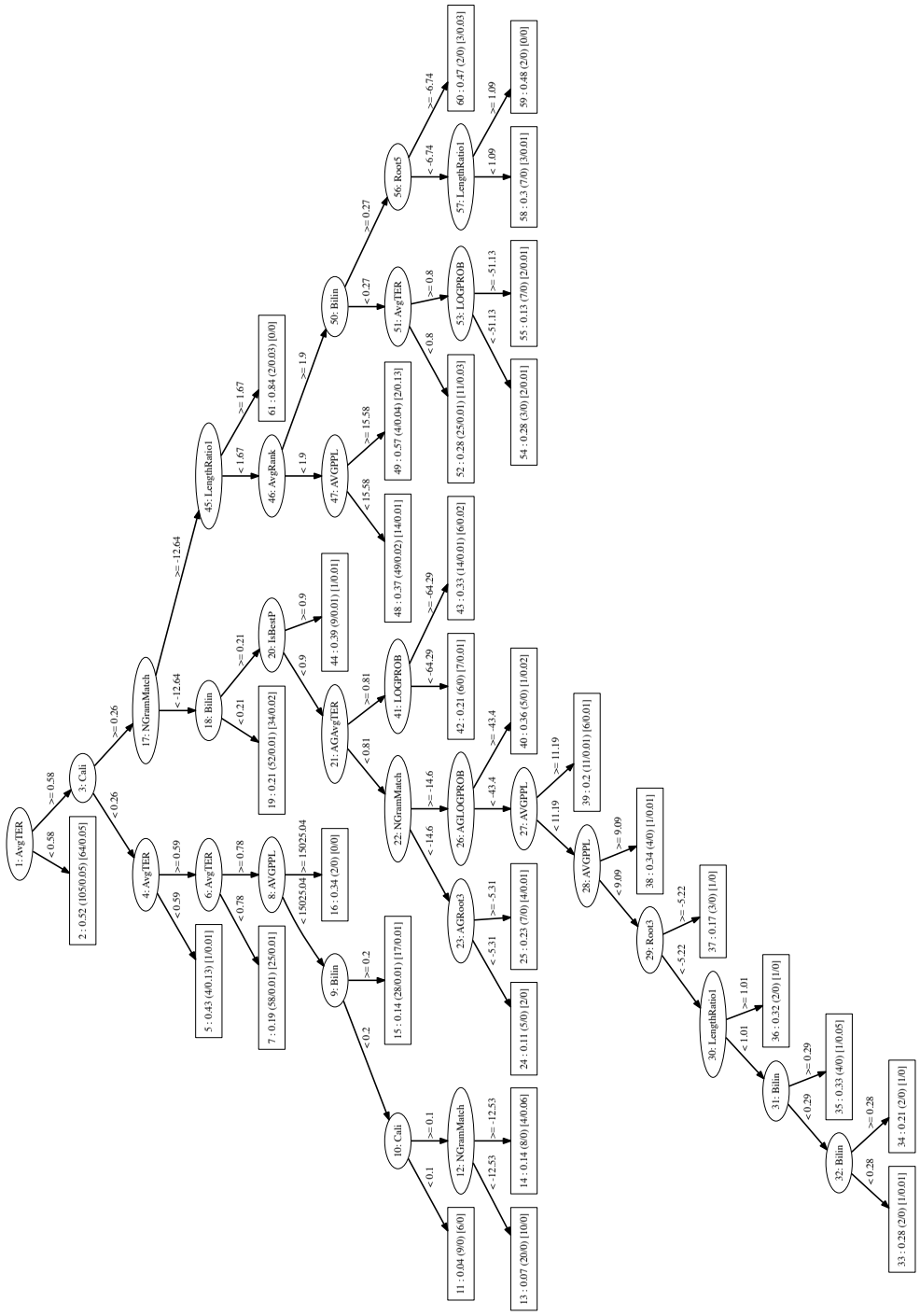

Figure 3.3: The visualization for the Decision Tree Model.

3.4.4 Linear Regression

Table 3.4 shows the translation quality using Linear Regression model. The Linear Regression model achieves the highest translation quality compared with other models and the highest BLEU score is 39.80 when all features are used in model tuning process.

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

Table 3.4: The translation quality of the best non-professional selected according to the Linear Regression model.

3.5 Quality Control Analysis

Compared to the baseline method, the MERT model, the Decision Tree Model and the Linear Regression Model all achieve much better performances, which means supervised learning framework works well on several popular machine learning models and shows

its effectiveness in quality control. Among all these three models, the Linear Regression Model achieves the highest BLEU score of 39.80 when all features are used in the training process. Compared to the professional translation which achieves a BLEU score of 42.89, this machine learning based quality control mechanism achieves a similar translation quality with a loss of 3.09 in BLEU.

Chapter 4

Cost Optimization for Crowdsourcing Machine Translation

In this chapter [1], we focus on a different aspect of crowdsourcing than Zaidan and Callison-Burch (2011). We attempt to achieve the same high quality while **minimizing the associated costs**.

We propose two complementary methods: (1) We 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. (2) We reduce the number of workers we hire, and retain only high quality translators

[1] Chapters 4 extend the exposition and analysis presented in Gao et al. (2015). The experimental results are the same as in the published work.

by quickly identifying and filtering out workers who produce low quality translations. Our work stands in contrast with Zaidan and Callison-Burch (2011) who always solicited and paid for a fixed number of translations for each source sentence, and who had no model of annotator quality.

4.1 Problem Setup

We start with a corpus of source sentences to be translated, and we may solicit one or more translations for every sentence in the corpus. Our targeted task is to assemble a single high quality translation for each source sentence while minimizing the associated cost. This process can be repeated to obtain multiple high quality translations.

We study the data set which is created by Zaidan and Callison-Burch (2011) and mentioned in Chapter 3. Following Zaidan and Callison-Burch (2011), we use BLEU to gauge the quality of human translations. We can compute the expected quality of professional translation by comparing each of the professional translations against the other 3 and selecting the best translation among them. This results in an average BLEU score of 42.38. In comparison, the average Turker translations score is only 28.13 without quality control. Zaidan and Callison-Burch trained a MERT (Och, 2003; Zaidan, 2009) model to select one non-professional translation out of the four and pushed the quality of crowdsourcing translation to 38.99, closer to the expected quality of professional translation. They used a small amount of professional translations (10%) as calibration data to estimate the good-

ness 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. Cost reduction plays an important role if we want to assemble a large enough parallel corpus to train a statistical machine translation system which typically require millions of translated sentences. Here, we introduce several methods for reducing the number of non-professional translations while still maintaining high quality.

4.2 Estimating Translation Quality

Since the linear regression model achieves the highest translation quality, we use the linear regression model to estimate translation quality. We replicate the feature set used in Chapter 3 to train models.

4.3 Reducing the Number of Translations

The first way that we optimize cost is to solicit fewer redundant translations. The strategy 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

Algorithm 1 How good is good enough

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 $S = \{\vec{f}^{s}_{i,j}, y^{s}_{i,j})^{j=1..m}_{i=1..n}\}$ and a validation set $V = \{(\vec{f}^{v}_{i,j}, y^{v}_{i,j})^{j=1..m}_{i=1..n}\}$ 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 m **do**

            if $\vec{w'} \cdot \vec{f}^{v}_{i,j} > \theta \wedge j < m$ **then** select $t^{v}_{i,j}$ for s_i and **break**

            if $j == m$ **then** select $t^{v}_{i,m}$ 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: θ and model parameter $\vec{w}$

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 the allowable degradation in translation quality. We train a model to search for a threshold θ between acceptable and unacceptable translations for a specific value of δ. For instance, we may fix δ at 95%, meaning that the resulting BLEU score should not drop below 95% of the α after reducing the number of translations.

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.

4.3.1 Experiments

We divide data into a training set (10%), a validation set (10%) and a test set (80%). Each source sentence has four translations in total. We use the validation set to search for θ. The Oracle upper bound on BLEU is set to be 40.13 empirically. 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.

$\delta(\%)$	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 4.1: The relationship between 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 average number of translation candidates set for each source sentence (*# Trans*).

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 to select the best translation for each source

segment from full sets of candidates. It reaches a BLEU score of 40.13.

Translation reducing method

Table 4.1 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 translations. If we set $\delta < 0.95$ 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.

Examples	System Selection	Candidates	BLEU Score
Example 1		Abstain from decrease eating in order to escape from flue.	3.7
		In order to be safer from flu quit dieting.	17.1
	✓	Avoiding dieting to prevent from flu.	18.4
		abstention from dieting in order to avoid Flu.	5.5
Example 2		This research of American scientists was shown after many experiments on mouses.	22.5

		According to the American Scientist this research has come out after much experimentations on rats.	16.6
	✓	This research of American scientists came in front after experimenting on mice.	27.8
		This research from the American Scientists have come up after the experiments on rats.	13.5
Example 3		The research proved this old talk that decrease eating is useful in fever.	6.3
		This Research has proved the very old saying wrong that it is good to starve while in fever.	15.3
	✓	Research disproved the old axiom that " It is better to fast during fever"	18.1
		research has proven this old myth wrong that its better to fast during fever.	12.0

Example 4		The Police said that they were killed in the Frontier Corps.	13.7
	✓	It is being said that the dead belonged to the frontier core and the police.	22.2
		The information about the killed are being said through the Front-air core and police.	12.4
		The deceased were told to be related to Frontier core and the police.	20.0

Table 4.2: Examples of translation reducing method where model selections agree with the gold standard calibration.

Table 4.2 shows examples of the selections made by our translation reducing method on the Urdu translation data. In each example, translations are shown in temporal order where the translation at the top of each block 'comes first' and the translation in the bottom of each block 'comes last'. In all of these examples, our algorithm selects a translation before we have observed all translations. If it were run live, then this would have the effect of ceasing

to solicit additional translations for that source segment, and saving the costs associated with further redundant translations. The table also shows the corresponding BELU score for each candidate and our selection mechanism selects the right candidates and stops. In this case our algorithm's selected translations correspond with the quality judgments based on professional reference translations. This is not always true; sometimes our algorithm selects a translation that is not calibrated as the best (Table 4.3).

Examples	System Selection	Candidates	BLEU Score
Example 1	✓	The first nuclear test was in 1990.	24.0
		First nuclear test was done in 1990	34.1
		'first nuclear experiment was done in 1990'	34.9
		This plan of education was implemented on 1990.	4.8
Example 2		Madonna has broken her own record in 2006	59.6
		Madonna has broken her own record of the year 2006.	53.1

✓	Madonna broke her own record of 2006.	34.7
	the male student walks along the long road	3.8

Table 4.3: Examples of translation reducing method where model selections don't agree with the gold standard calibration.

4.4 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 good translators only and stop hiring bad translators after they are identified as such. Before presenting our method, we first demonstrate that Turkers produce consistent quality translations over time.

4.4.1 Turkers' behavior in translating sentences

Do Turkers produce good (or bad) translations consistently or not? Are some Turkers consistent and others not? We use the professional translations as a gold-standard to analyze the individual Turkers, and we find that most Turkers' performance stay surprisingly con-

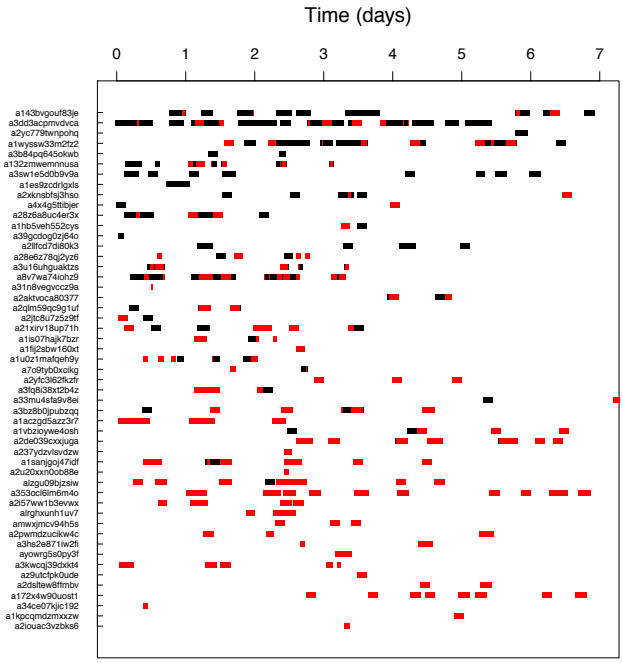

Figure 4.1: A time-series plot of all of the translations produced by Turkers (identified by their WorkerID serial number). Turkers are sorted with the best translator at the top of the y-axis. Each tick represents a single translation and black means better than average quality.

sistent over time.

Figure 4.1 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 represents 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 red otherwise. Good translators tend to produce consistently good translations and bad translators rarely produce good translations.

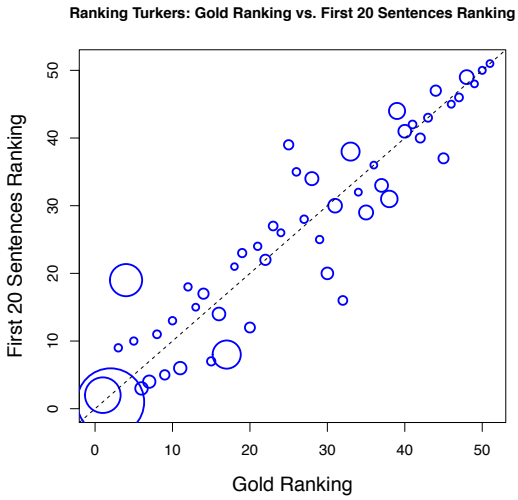

Figure 4.2: Correlation between gold standard ranking and ranking computed using the first 20 sentences as calibration. Each bubble represents a worker. The radius of each bubble shows the relative volume of translations completed by the worker. The weighted correlation is 0.94.

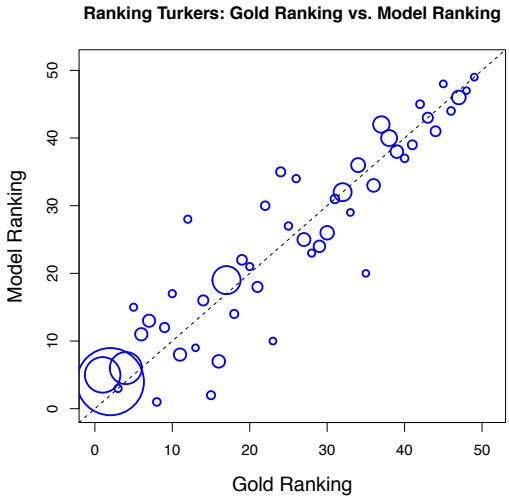

Figure 4.3: Correlation between gold standard ranking and our model's ranking. The corresponding weighted correlation is 0.95.

4.4.2 Evaluating Rankings

We use weighted Pearson correlation (Pozzi et al., 2012) to evaluate our ranking of workers against gold standard ranking. Since workers translate different numbers of sentences, it is more important to rank the workers who translate 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 weighted Pearson correlation ρ can be calculated as:

$$\rho(x,y;w) = \frac{cov(x,y;w)}{\sqrt{cov(x,x;w)cov(y,y;w)}} \tag{4.4.1}$$

where cov is weighted covariance:

$$cov(x,y;w) = \frac{\sum_i w_i(x_i - m(x;w))(y_i - m(y;w))}{\sum_i w_i} \tag{4.4.2}$$

and m is weighted mean:

$$m(x;w) = \frac{\sum_i w_i x_i}{\sum_i w_i} \tag{4.4.3}$$

4.4.3 Automatically Ranking Translators

We introduce two approaches to rank workers using a small portion of the work that they submitted. The strategy is to filter out bad workers, and to select the best translation from translations provided by the remaining workers. We propose two different ranking methods:

Ranking workers using their first k translations We rank the Turkers using their first few translations by comparing their translations against 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 the gold standard, we also attempt to automatically 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 performance of the worker w is:

$$performance(w) = \frac{\sum_{t \in T_w} score(t)}{|T_w|} \tag{4.4.4}$$

where T_w is the set of translations completed by the worker w and $score(t)$ is the model predicted score for translation t.

4.4.4 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 both rankings' correlation with the gold standard ranking. Since the top worker threshold is varied and since we change the value of k in first k sentence ranking, we have a different test set in different settings. Each test set excludes any items which are used to rank the workers, or which do not have any translations from the top workers according to our rankings.

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 BLEU score comparing all of his or her translations to the corresponding 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. We select the best translation of each source sentence from the top ranked worker who translated that sentence.

Table 4.4 shows the results of Pearson correlations for different value of k. As k increases, our rankings fit 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 4.2 shows the correlation of the gold ranking and the ranking based on workers' first 20 sentences.

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:

- Using the model's prediction of workers' rank, and selecting the translation from the best worker.
- Using the model's score for each translation and selecting the highest scoring translation of each source sentence.

Table 4.5 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.

Figure 4.3 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.

4.4.5 Filtering out bad workers

Ranking translators would allow us to reduce costs by only re-hiring top workers. Table 4.6 shows what happens when we vary the percentage of top ranked workers we retain. In general, the model does a good job of picking the best translations from the remaining good translators. Compared to actually 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.

4.5 Cost Analysis

We have introduced several ways of significantly lowering the costs associated with crowd-sourcing translations when a large amount of data is 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

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 4.4: Pearson Correlations for calibration data in different proportion. The percentage column shows what proportion of the whole data set is used for calibration.

here comes from reducing the number of redundant translations. We can save almost half of the cost associated with non-professional translations to get 95% of the translation quality using the full set of redundant translations.

Feature Set	ρ	BLEU rank	BLEU 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**
Baseline (MERT)	0.73	-	38.99

Table 4.5: 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 ranking for workers (rank) and model-predicted scores for translations (score). Here we do not filter out bad workers when selecting the best translation.

- 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.

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 4.6: A comparison of the translation quality when we retain the top translators under different rankings. The rankings shown are random, the model's ranking (using all features from Table 4.5) and the gold ranking. Δ is the difference between the BLEU scores for the gold ranking and the model ranking. # Trans is the average number of translations needed for each source sentence.

- 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.

Chapter 5

Extending the Cost Optimization Framework to a New Language

In last chapter, we demonstrate that we could reduce the cost of crowdsourcing Urdu to English translation. In this chapter, we extend our cost reduction framework to another language, Tamil. Since our Tamil corpus does not have any professionally translated reference translations, we can only apply one of the two cost reduction techniques that we introduced in the previous chapter. We apply the translation reducing method to Tamil-English translation. We achieve similar cost improvements result compared with our Urdu experiments.

5.1 Data

We study the Tamil data collected by Post et al. (2012), which was created through hiring crowd workers on Amazon Mechanical Turk. About 12,500 Tamil sentences were translated into English and 294 Turkers worked on this project. For each Tamil source sentence, four non-professional translations were solicited. Turkers also filled out a survey about their language skill information and their country of origin. In addition, Post et al. (2012) collected the translation ranking information through a second-pass annotation by English native Turkers on MTurk. They hired five Turkers (ranker) to select the best non-professional translation among four candidates. The ranking HIT included a portion of the data with gold-standard rankings, which was used to test that the rankers were doing the task reliably and not randomly clicking. Testing results are recorded to evaluate rankers' quality.

5.2 Label

Since we don't have the professional references for the Tamil non-professional translations, we propose to label each translation based on the second-pass rankings. We evaluate rankers based on their performance on the control questions (which were constructed by having one human translation paired with 3 machine translation outputs). We define the confidence score for each ranker. For ranker r, the score is:

$$Conf(r) = N_c(r)/N_t(r) \tag{5.2.1}$$

where $N_c(r)$ is the number of correct answered test cases submitted by ranker r and $N_t(r)$ is the total number of test cases submitted by ranker r (e.g. the fraction of time that they rated the human translation higher than the machine translation in the control questions).

A translator's quality is derived from the rankers' scores of his/her translations. Translator quality of Turker T is defined as:

$$Translator(T) = \sum_{t \in Trans(T)} (\sum_{i=1}^{5} \sigma(r_i,t) * Conf(r_i)/5)/N(T) \tag{5.2.2}$$

where $Trans(T)$ is the set of translations submitted by T, $N(T)$ is the number of total translations submitted by T, and $\sigma(r,t) = 1$ iff translation t is picked as the best by ranker r, otherwise $\sigma(r,t) = 0$. When a ranker picks a translation as the best we describe it as an 'endorsement'. The endorsement information for translation t is defined as:

$$Endorsement(t) = \sum_{i=1}^{5} \sigma(r_i,t) * Conf(r_i)/5 \tag{5.2.3}$$

and the predicted score for translation t which is translated by Turker T is defined as:

$$label(t) = 50\% * Endorsement(t) + 50\% * Translator(T) \tag{5.2.4}$$

5.3 Experiments

We divide the data into training set (10%), validation set (10%) and test set (80%), and perform the translation reducing method proposed in Algorithm 1 on the labeled Tamil translation data. We set the upper bound score of 0.304 in our proposed labeling evaluation metric empirically by selecting the non-professional translation with the highest model

predicted score from the full sets of translations. We then vary the value of θ from 90% to 100%, and sweep values of θ by incrementing it in step sizes of 0.01. We evaluate the translation quality in our labeling metric. We report results based on five-fold cross validation.

5.3.1 Baseline

The baseline method is to randomly select a translation for each source sentence. Random selection achieves a score of 0.18. To set the Oracle method, we select the best translation with the highest score. Oracle method achieves a score of 0.45.

5.3.2 Results

Table 5.1 shows the results for translation reducing method on Tamil data. If we want to keep 90% accuracy achieved on the full translation set, we can stop collecting translations after we have got 2.19 translations in average and we can save almost half of the cost to collect data.

Table 5.2 shows examples of the selections made by our translation reducing method. In each example, translations are shown in temporal order where the translation at the top of each block 'comes first' and the translation in the bottom of each block 'comes last'. In all of these examples, our algorithm selects a translation before we have observed all the translations. If it were run live, then this would have the effect of ceasing to solicit additional translations for that source segment, and saving the costs associated with further

$\delta(\%)$	Score	# Trans.
90	0.278	2.19
91	0.280	2.25
92	0.283	2.37
93	0.285	2.42
94	0.290	2.68
95	0.293	2.85
96	0.294	2.92
97	0.298	3.15
98	0.299	3.35
99	0.302	3.65
100	0.303	3.75

Table 5.1: The relationship between δ (the allowable deviation from the expected upper bound on score), the score for translations selected by models from partial sets and the average number of translation candidates set for each source sentence (*# Trans*).

redundant translations. The table also shows the corresponding endorsement score for each candidate and our selection mechanism selects the right candidates and stops. In this case our algorithm's selected translations correspond with the quality judgments collected in the second-pass HIT. This is not always true; sometimes our algorithm selects a translation that is not voted as the best (Table 5.3).

Examples	System Selection	Candidates	Endorsement Score
Example 1	✓	Military troop reduction.	0.78
		less troops	0.0
		Team less	0.0
		reduction in army force	0.0
Example 2	✓	Born as Austrian is the reason to lost some position.	0.59
		(Born ausrian) so lost some positions	0.0
		(Born Austrian) They losed all posts.	0.0
		(Austrians)So some posts were withdrawn	0.17
Example 3		coming Canada welfare letters are clearly mentioned	0.0
	✓	These details are in the letter from Canadian Institutes of Health Research	0.60

		Canada welfare department mentioned in the letter are following.	0.15
		Canada health department explained in letter about coming back	0.0
Example 4		his birthday has been celebrated as Gandhi Jayanthi in India	0.0
		His birth day being celebrated as Gandhi Jayathi	0.0
	✓	In India, his birthday is celebrated as Gandhi Jeyanthi.	0.62
		His birth day is celebrated as Gandhi Jayanthi in India	0.15

Table 5.2: Examples of translation reducing method where model selections agree with the proposed labeling metric.

Examples	System Selection	Candidates	BLEU Score

Example 1	✓	Today 40,000 Policemen are serving in this.	0.26
		today 40,000 police are in service	0.39
		Today 40,000 Police mans are servicing in this Service.	0.04
		NOW 40000 POLICE FORCE WERE SERVICING	0.06
Example 2		He got a best new face award for this movie	0.14
		He got award for a new face in this film	0.08
	✓	He collected good new face award in this film.	0.17
		He has been awared in this film as new actor.	3.8

Table 5.3: Examples of translation reducing method where model selections don't agree with the proposed labeling metric.

Chapter 6

Conclusion

In this thesis, we extend Zaidan and Callison-Burch (2011)'s crowdsourcing translation framework using different machine learning models. We show that a supervised learning framework performs well for performing quality control on crowdsourcing translation, for a variety of different machine learning models. In addition, we propose two novel mechanisms to optimize cost (Gao et al., 2015): a translation reducing method and a translator reducing method. Based on our experiments, the translator reducing method works well on Urdu data while the translation reducing method works well on the Urdu data and Tamil data. These two mechanisms 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.

Bibliography

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.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees*. CRC press.

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.

Mingkun Gao, Wei Xu, and Chris Callison-Burch. 2015. Cost optimization for crowdsourcing translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2015)*.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

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 2014 AAAI Conference on Human Computation and Crowdsourcing*.

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(ACL)*, pages 160–167.

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 (ACL)*, pages 311–318.

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.

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.

Michael JD Powell. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162.

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.

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 (EMNLP)*, pages 254–263.

Document of SPSS, 2011. *CART Algorithm*. Available at `ftp://ftp.boulder.ibm.com/software/analytics/spss/support/Stats/Docs/Statistics/Algorithms/14.0/TREE-CART.pdf`.

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 (ACL-HLT)*, 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 (NAACL-HLT)*, pages 49–59.

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 *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 612–616.