Cost Optimization for Crowdsourcing Translation

Mingkun Gao, Wei Xu and **Chris Callison-Burch**
Computer and Information Science Department
University of Pennsylvania, Philadelphia, PA, USA
{gmingkun, xwe, ccb}@seas.upenn.edu

Abstract

Crowdsourcing makes it possible to create translations at much lower cost than hiring professional translators. However, it is still expensive to obtain the millions of translations that are needed to train statistical machine translation systems. We propose two mechanisms to reduce the cost of crowdsourcing while maintaining high translation quality. First, we develop a method to reduce redundant translations. 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 for 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 method to reduce the pool of translators 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.

1 Introduction

Crowdsourcing is a promising new mechanism for collecting large volumes of annotated data at low cost. Many NLP researchers have started creating speech and language data through crowdsourcing (for example, Snow et al. (2008), Callison-Burch and Dredze (2010) and others). One NLP application that has been the focus of crowdsourced data collection is statistical machine translation (SMT) which requires large bilingual sentence-aligned parallel corpora to train translation models. Crowdsourcing's low cost has made it possible to hire people to create sufficient volumes of translation in order to train SMT systems (for example, Ambati and Vogel (2010), Zbib et al. (2012), Post et al. (2012), Zbib et al. (2013)).

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

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

In this paper we demonstrate that:

- 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 even when observing only small amounts of data. Compared with a gold standard ranking, we achieve a correlation of 0.94 after seeing the translations of only 20 sentences from each worker. Therefore, bad workers can be filtered out quickly.
- We can achieve a similar BLEU score as Zaidan and Callison-Burch (2011) at half the cost using our cost optimizing methods.

2 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 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 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 the 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).

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. In comparison, the average Turker translations score 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 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. 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.

3 Estimating Translation Quality

We use a linear regression model[2] to predict a quality score ($score(t) \in \mathbb{R}$) 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

[1] LDC Catalog number LDC2010T23

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

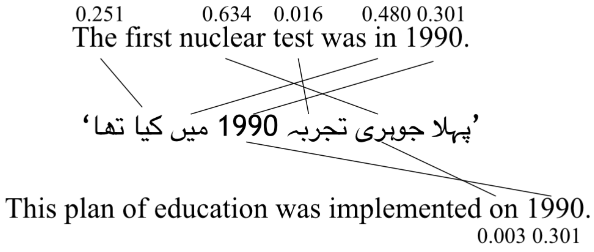
0.251 0.634 0.016 0.480 0.301
The first nuclear test was in 1990.
‘پہلا جوہری تجربہ 1990 میں کیا تھا’
This plan of education was implemented on 1990.
0.003 0.301

Figure 1: Example bilingual features for two crowd-sourced 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.

language model, sentence length and 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 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 is the average of its alignment probabilities between words in the source sentence and words in the Turker's translation. In Figure 1, we show how the bilingual feature allows us to distinguish between a valid translation (top) and an invalid/spammy translation (bottom).

4 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 from paying for another redundant translation.

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: $\theta$ and model parameter $\vec{w}$
```

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 so-

$\delta(\%)$	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 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*).

licit translations. Algorithm 1 details the process of model training and searching for θ.

4.1 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.

4.1.1 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 best translation for each source segment from full sets of candidates. It reaches a BLEU score of 40.13.

4.1.2 Translation reducing method

Table 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.

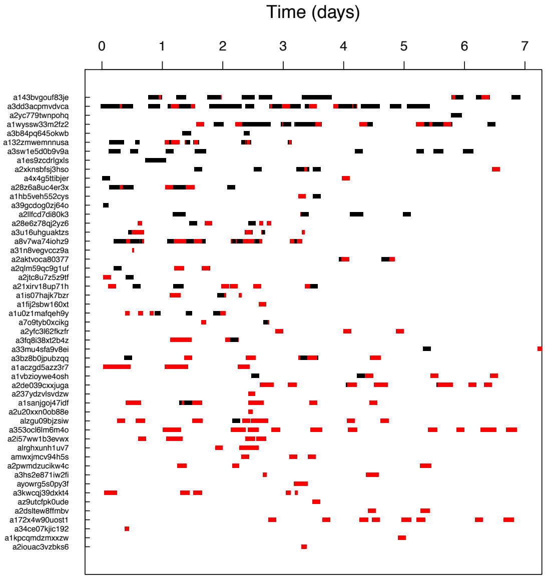

Figure 2: A time-series plot of all of the translations produced by Turkers (identified by their WorkerID serial number). Turkers are sorted with the best translator at the top of the y-axis. Each tick represent a single translation and black means better than average quality.

5 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.

5.1 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 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 red otherwise. Good translators tend to produce consistently good translations and bad translators rarely produce good translations.

5.2 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 weighted Pearson correlation ρ can be calculated as:

$$\rho(x, y; w) = \frac{cov(x, y; w)}{\sqrt{cov(x, x; w)cov(y, y; w)}} \quad (1)$$

where cov is weighted covariance:

$$cov(x, y; w) = \frac{\sum_i w_i(x_i - m(x; w))(y_i - m(y; w))}{\sum_i w_i} \quad (2)$$

and m is weighted mean:

$$m(x; w) = \frac{\sum_i w_i x_i}{\sum_i w_i} \quad (3)$$

5.3 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 a 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:

$$performance(w) = \frac{\sum_{t \in T_w} score(t)}{|T_w|} \quad (4)$$

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

5.4 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 were used to rank the workers, or which did not have any translations from the top workers according to our rankings.

5.4.1 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.

5.4.2 Ranking workers using their first k translations

Without using any model, we rank workers using their first k translations. We select best translation of each source sentence from the top ranked worker who translated that sentence.

Table 2 shows the results of Pearson correlations for different value of k. As k increases, our rankings

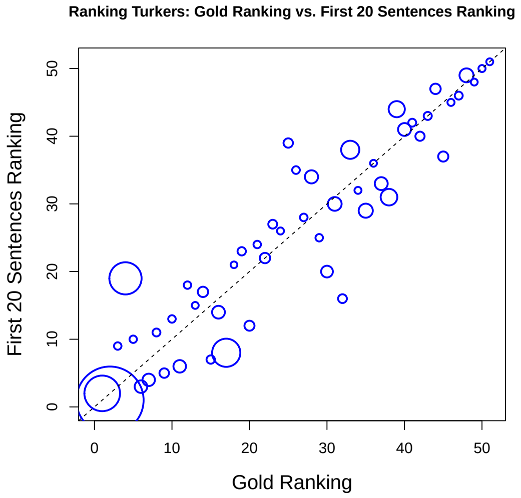

Figure 3: Correlation between gold standard ranking and ranking computed using the first 20 sentences as calibration. Each bubble represents a worker. The radius of each bubble shows the relative volume of translations completed by the worker. The weighted correlation is 0.94.

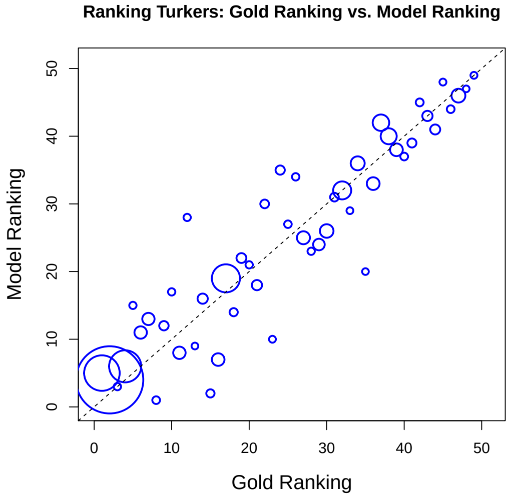

Figure 4: Correlation between gold standard ranking and our model's ranking. The corresponding weighted correlation is 0.95.

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 3 shows the correlation of gold ranking and the ranking based on workers' first 20 sentences.

5.4.3 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 3 shows that the model trained on all features achieves a very high correlation with the gold standard ranking (Pearson's ρ = 0.95), and a BLEU score of 39.80.

Figure 4 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.

5.5 Filtering out bad workers

Ranking translators would allow us to reduce costs by only re-hiring top workers. Table 4 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.

6 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

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

		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**
Baseline (MERT)	0.73	-	38.99

Table 3: 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.

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.

- 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.

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: 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 3) 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.

7 Related Work

Sheng et al. (2008) focused on training a machine learning model from noisy labels. We cannot always get high-quality labeled data from crowdsourcing, but we can still ensure that a model trained on the data is accurate by redundantly labeling the data. 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 accuracy is improved even if labels in its training data are noisy and imperfect. 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 avoid spam workers. They measured the performance of worker by comparing worker's 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 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.

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.

8 Conclusion

In this paper, we propose two mechanisms to optimize cost: a translation reducing method and a 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 must reach a certain threshold. This model is most effective when reasonable amounts of pre-existing professional translations are available for setting the model's 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 is available.

Acknowledgements

This material is based on research sponsored by a DARPA Computer Science Study Panel phase 3 award entitled "Crowdsourcing Translation" (contract D12PC00368). The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements by DARPA or the U.S. Government. This research was supported by the Johns Hopkins University Human Language Technology Center of Excellence and through gifts from Microsoft, Google and Facebook.

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.

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.

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*.

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 (NAACL-HLT)*, pages 207–215.

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.

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.

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.