Instructions for ACL2012 Proceedings

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

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

Abstract

This document contains the instructions for preparing a camera-ready manuscript for the proceedings of ACL2012. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. These instructions should be used for both papers submitted for review and for final versions of accepted papers. Authors are asked to conform to all the directions reported in this document.

1 Introduction

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

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

In this paper we focus on a different aspect of crowdsourcing than Zaidan and Callison-Burch (2011). We attempt to achieve the same high quality while **minimizing the associated costs**. We reduce costs using two complementary methods: (1) We quickly identify and filter out workers who produce low quality translations. The goal is to reduce the number of worker we hire, and retain only high quality translators. (2) Instead of soliciting a fixed number of translations for each foreign sentence, we stop soliciting translations after we get an acceptable one. We do so by building models to distinguish between acceptable translations and unacceptable ones. The goal is to reduces the number of independent translations that we solicit for each source sentence. Our work stands in contrast with Zaidan and Callison-Burch (2011) who had no model of annotator quality, and who always solicited and paid for a fixed number of translations of each source segment.

In this paper we demonstrate that

- Workers can be ranked by quality with high correlation against a gold standard ranking (ρ of 0.XXX), using logistic regression and a variety of features, or initially testing them using a small amount of calibration data with known professional translations.

- This ranking can be established after observing very small amounts of data (reaching ρ of 0.XXX after seeing only 10 translations from each worker), so bad workers can be filtered out quickly.

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

- We can achieve a similar BLEU score as Zaidan and Callison-Burch (2011) at $\frac{1}{X}$ of the cost.

2 Previous work

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

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

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

[1]LDC Catalog number XXX

2.1 Professional quality from non-professionals

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

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

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

- A Web *n*-gram log probability feature using Microsoft Web N-Gram Corpus, up to 5-grams.

- Geometric averages of Web *n*-grams.

- Sentence length features: we use the ratios of the length of the Urdu source sentence to the length of its English translation, and vice versa.

- Edit rate to other Turkers' translations of that sentence.

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

We introduce a new bilingual feature. We use the IBM Model 1 to construct the word alignment with probabilities between Urdu and English. For each foreign sentence, we calculate the word alignment feature by averaging the alignment probabilities of all words in Urdu sentence.

2.2 Cost

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

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

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

3 Data Analysis

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

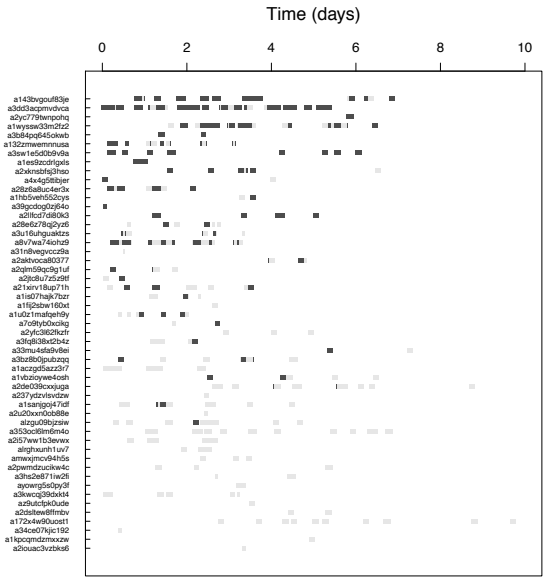

Figure 1: A time-series plot of all of the translations produced by Turkers (identified by their WorkerID serial number). Turkers are sorted based on the gold-standard ranking against professionals, with the best translators on top. Each tick represent a single translation HIT, and depicts the HIT's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

4 Automatically Ranking Translators

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

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

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

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

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

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

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

We get a surprisingly strong correlation with the gold standard ranking of workers, without using a model at all. Instead, we can use a small amount of calibration data (where gold standard translations are known). If we rank the worker based solely on their first HITs' BLEU score, comparing their translations of the 10 sentences in that HIT against the reference translations, then we do well at predicting their BLEU score for all of their translations. The Spearman Correlation is 0.84 when comparing this first-HIT ranking with gold standard ranking.

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

4.1 Experimental Setup

In the first approach ranking workers by model prediction, we evaluate models through five-fold cross validation. We divided the data into 3 parts: 20% data for worker aggregation feature calculation and 40% data for ranking workers and the remaining 40% data for testing. In the 20% portion, we use half of the data (10% of the full data set) to calculate the calibration feature and train models.

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

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

[4]Linear Regression

[5]Regression Tree

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

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

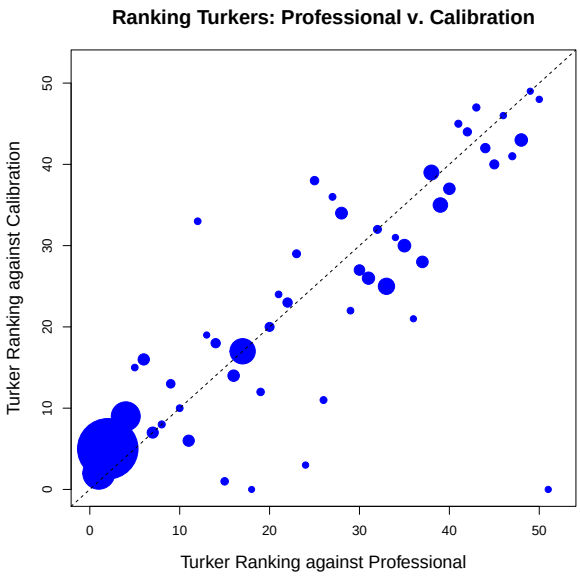

Figure 2: Correlation between gold standard ranking and calibration ranking. We use 10% training data as calibration data to rank workers. The corresponding Spearman's Correlation is 0.79. Each bubble represents a worker with his/her rank in gold standard ranking on x-axis and rank in calibration ranking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

In the second approach, we compared workers' first k sentences with references and ranked workers by their accumulated performance on these sentences . We choose the top 25% workers in the ranking list as the ones we keep hiring. To evaluate this method, for each source sentence in testing set, we selected the translation with the highest rank from translating candidates provided only by the workers we kept. We define the test set for first k sentences as T_k and for each source sentence $t \in T$:

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

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

4.2 Cost Savings

"TODO: Here please write an analysis of how much money we could save if we choose some threshold for discarding Turkers. Also, give an estimate of how much worse the translation quality is compared with keeping all of the workers."

In the first approach, we quickly evaluate workers and rank them for filtering out workers with low rankings. We train linear regression models using

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

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

a variety of features to score each translation and evaluate workers by averaging the scores of his/her translations.

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

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

Since the calibration data represents a substantial portion of the costs involved with collecting our translations via crowdsourcing, we can attempt to reduce it. The Linear Regression model reaches a correlation of 0.80 if we omit the calibration data entirely. Rather than using 10% of each HIT as calibration data, we experimented with using the first-K translation sentences provided by each Turker. Table 3 shows the results of Spearman's Correlation. We achieve very strong correlation when calibrating the workers based on the translations of their first 40 sentences. Even we only use the first 10 sentences to evaluate and rank workers, the correlation (ρ) is higher than 0.80. Consequently, we can decide whether to continue to hire a worker in a very short time after analyzing the first 10 or less 10 sentences provided by each worker. If we use the first 20 sentences, which is still only a small part of data compared with the whole data set, rho is higher than 0.90, almost perfect match with the gold standard ranking.

Table 4 shows the BLEU score when we select top 25% workers from the ranking list based on the performance of first k sentences. As a comparison, we also listed the BLEU scores for random selection and selection on the gold standard ranking. If we use first 10 sentence,

5 Get Another Translation?

In the second approach, we train a model to decide whether a translation is 'good enough,' in which case we don't need to pay for another redundant translation of the source sentence.

In the second approach, we train a model to decide whether a translation is 'good enough,' in which case we don't need to pay for another redundant translation of the source sentence. To perform this experiment, we divide the data into 3 parts: 10% of the data as a training set, 10% of the data as a validation set and the remaining 80% of the data as a test set. We train the model to score each translation we've got already, to use this score to evaluate whether to get another translation. The challenge is how to set the threshold to separate acceptable translations and unacceptable ones.

In our design, we set the threshold empirically us-

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

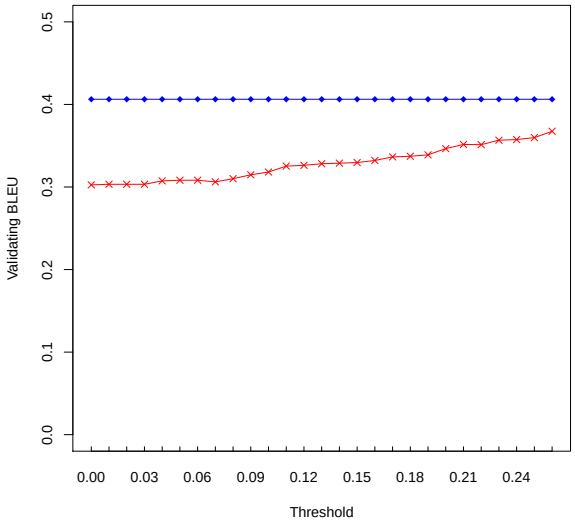

Figure 3: The Process to Sweep the Threshold

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

To evaluate the performance of the model running with different thresholds, we first compute an upper bound by selecting the best translation among all 4 candidates for each foreign sentence of the validation set according to our model. We call this set S_{upper}. S_{upper} is the highest BLEU score we can get by choosing translation using the model, since it has access to all of the available translations.

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

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

5.1 Cost Savings

"TODO:Here please write an analysis of how much money we could save if we choose some threshold for when we ask for a new translation. Also, give an estimate of how much worse the translation quality is compared with keeping all of the workers."

6 Related Work

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

fect quality. As long as the integrated quality (the probability of the integrated labeling being correct) is higher than 0.5, repeated labeling benefits model training. More closer the quality to 0.5, the more benefits obtained in model prediction.

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

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

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

7 Discussion

8 Conclusion

Acknowledgments

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

References

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

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

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

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

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language

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

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