Crowdsourcing Translation: Professional Quality from Non-Professionals

Omar F. Zaidan and **Chris Callison-Burch**
Dept. of Computer Science, Johns Hopkins University
Baltimore, MD 21218, USA
{ozaidan,ccb}@cs.jhu.edu

Abstract

Naively collecting translations by crowdsourcing the task to non-professional translators yields disfluent, low-quality results if no quality control is exercised. We demonstrate a variety of mechanisms that increase the translation quality to near professional levels. Specifically, we solicit redundant translations and edits to them, and automatically select the best output among them. We propose a set of features that model both the translations and the translators, such as country of residence, LM perplexity of the translation, edit rate from the other translations, and (optionally) calibration against professional translators. Using these features to score the collected translations, we are able to discriminate between acceptable and unacceptable translations. We recreate the NIST 2009 Urdu-to-English evaluation set with Mechanical Turk, and quantitatively show that our models are able to select translations within the range of quality that we expect from professional translators. The total cost is more than an order of magnitude lower than professional translation.

1 Introduction

In natural language processing research, translations are most often used in statistical machine translation (SMT), where systems are trained using bilingual sentence-aligned parallel corpora. SMT owes its existence to data like the Canadian Hansards (which by law must be published in both French and English). SMT can be applied to any language pair for which there is sufficient data, and it has been shown to produce state-of-the-art results for language pairs like Arabic–English, where there is ample data. However, large bilingual parallel corpora exist for relatively few languages pairs.

There are various options for creating new training resources for new language pairs. These include harvesting the web for translations or comparable corpora (Resnik and Smith, 2003; Munteanu and Marcu, 2005; Smith et al., 2010; Uszkoreit et al., 2010), improving SMT models so that they are better suited to the low resource setting (Al-Onaizan et al., 2002; Probst et al., 2002; Oard et al., 2003; Niessen and Ney, 2004), or designing models that are capable of learning translations from monolingual corpora (Rapp, 1995; Fung and Yee, 1998; Schafer and Yarowsky, 2002; Haghighi et al., 2008). Relatively little consideration is given to the idea of simply hiring translators to create parallel data, because it would seem to be prohibitively expensive. For example, Germann (2001) estimated the cost of hiring professional translators to create a Tamil-English corpus at $0.36/word. At that rate, translating enough data to build even a small parallel corpus like the LDC's 1.5 million word Urdu–English corpus would exceed half a million dollars.

In this paper we examine the idea of creating low cost translations via crowdscouring. We use Amazon's Mechanical Turk to hire a large group of non-professional translators, and have them recreate an Urdu–English evaluation set at a fraction of the cost of professional translators. The original dataset already has professionally-produced reference translations, which allows us to objectively and quantitatively compare the quality of professional and non-professional translations. Although many of the individual non-expert translators produce low-quality, disfluent translations, we show that it is possible to

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1220–1229,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

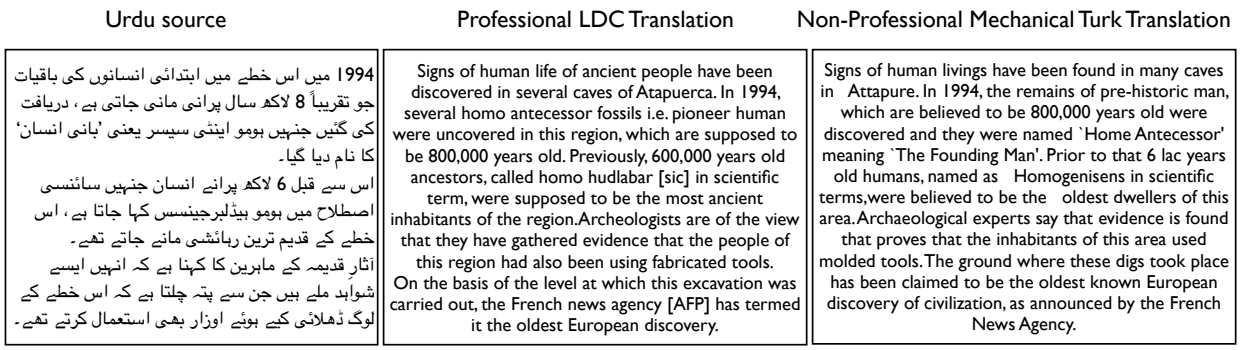

Figure 1: A comparison of professional translations provided by the LDC to non-professional translations created on Mechanical Turk.

get high quality translations in aggregate by soliciting multiple translations, redundantly editing them, and then selecting the best of the bunch.

To select the best translation, we use a machine-learning-inspired approach that assigns a score to each translation we collect. The scores discriminate acceptable translations from those that are not (and competent translators from those who are not). The scoring is based on a set of informative, intuitive, and easy-to-compute features. These include country of residence, number of years speaking English, LM perplexity of the translation, edit rate from the other translations, and (optionally) calibration against professional translators, with the weights set using a small set of gold standard data from professional translators.

2 Crowdsourcing Translation to Non-Professionals

To collect crowdsourced translations, we use Amazon's Mechanical Turk (MTurk), an online marketplace designed to pay people small sums of money to complete *Human Intelligence Tasks* (or HITs) – tasks that are difficult for computers but easy for people. Example HITs range from labeling images to moderating blog comments to providing feedback on relevance of results for search queries. Anyone with an Amazon account can either submit HITs or work on HITs that were submitted by others. Workers are referred to as "Turkers", and designers of HITs as "Requesters." A Requester specifies the reward to be paid for each completed item, sometimes as low as $0.01. Turkers are free to select whichever HITs interest them, and to bypass HITs they find uninteresting or which they deem pay too little.

The advantages of Mechanical Turk include:

- zero overhead for hiring workers
- a large, low-cost labor force
- easy micropayment system
- short turnaround time, as tasks get completed in parallel by many individuals
- access to foreign markets with native speakers of many rare languages

One downside is that Amazon does not provide any personal information about Turkers. (Each Turker is identifiable only through an anonymous ID like A23KO2TP7I4KK2.) In particular, no information is available about a worker's educational background, skills, or even native language(s). This makes it difficult to determine if a Turker is qualified to complete a translation task.

Therefore, soliciting translations from anonymous non-professionals carries a significant risk of poor translation quality. Whereas hiring a professional translator ensures a degree of quality and care, it is not very difficult to find bad translations provided by Turkers. One Urdu headline, professionally translated as *Barack Obama: America Will Adopt a New Iran Strategy*, was rendered disfluently by a Turker as *Barak* ***Obam*** *will* ***do a new policy*** *with Iran*. Another translated it with snarky sarcasm: *Barak Obama and America* ***weave new evil strategies*** *against Iran*. Figure 1 gives more typical translation examples. The translations often reflect non-native English, but are generally done conscientiously (in spite of the relatively small payment).

To improve the accuracy of noisy labels from non-experts, most existing quality control mechanisms

employ some form of voting, assuming a discrete set of possible labels. This is not the case for translations, where the 'labels' are full sentences. When dealing with such a structured output, the space of possible outputs is diverse and complex. We therefore need a different approach for quality control. That is precisely the focus of this work: to propose, and evaluate, such quality control mechanisms.

In the next section, we discuss reproducing the Urdu-to-English 2009 NIST evaluation set. We then describe a principled approach to discriminate good translations from bad ones, given a set of redundant translations for the same source sentence.

3 Datasets

3.1 The Urdu-to-English 2009 NIST Evaluation Set

We translated the Urdu side of the Urdu–English test set of the 2009 NIST MT Evaluation Workshop. The set consists of 1,792 Urdu sentences from a variety of news and online sources. The set includes four different reference translations for each source sentence, produced by professional translation agencies. NIST contracted the LDC to oversee the translation process and perform quality control.

This particular dataset, with its multiple reference translations, is very useful because we can measure the quality range for professional translators, which gives us an idea of whether or not the crowdsourced translations approach the quality of a professional translator.

3.2 Translation HIT design

We solicited English translations for the Urdu sentences in the NIST dataset. Amazon has enabled payments in rupees, which has attracted a large demographic of workers from India (Ipeirotis, 2010). Although it does not yet have s direct payment in Pakistan's local currency, we found that a large contingent of our workers are located in Pakistan.

Our HIT involved showing the worker a sequence of Urdu sentences, and asking them to provide an English translation for each one. The screen also included a brief set of instructions, and a short questionnaire section. The reward was set at $0.10 per translation, or roughly $0.005 per word.

In our first collection effort, we solicited only one translation per Urdu sentence. After confirming that the task is feasible due to the large pool of workers willing and able to provide translations, we carried out a second collection effort, this time soliciting *three* translations per Urdu sentence (from three distinct translators). The interface was also slightly modified, in the following ways:

- Instead of asking Turkers to translate a full document (as in our first pass), we instead split the data set into groups of 10 sentences per HIT.
- We converted the Urdu sentences into images so that Turkers could not cheat by copying-and-pasting the Urdu text into an MT system.
- We collected information about each worker's geographic location, using a JavaScript plugin.

The translations from the first pass were of noticeably low quality, most likely due to Turkers using automatic translation systems. That is why we used images instead of text in our second pass, which yielded significant improvements. That said, we do not discard the translations from the first pass, and we do include them in our experiments.

3.3 Post-editing and Ranking HITs

In addition to collecting four translations per source sentence, we also collected **post-edited** versions of the translations, as well as **ranking judgments** about their quality.

Figure 2 gives examples of the unedited translations that we collected in the translation pass. These typically contain many simple mistakes like misspellings, typos, and awkward word choice. We posted another MTurk task where we asked workers to edit the translations into more fluent and grammatical sentences. We restrict the task to US-based workers to increase the likelihood that they would be native English speakers.

We also asked US-based Turkers to rank the translations. We presented the translations in groups of four, and the annotator's task was to rank the sentences by fluency, from best to worst (allowing ties).

We collected redundant annotations in these two tasks as well. Each translation is edited three times (by three distinct editors). We solicited only one edit per translation from our first pass translation effort. So, in total, we had 10 post-edited translations for

Avoiding dieting to prevent from flu	abstention from dieting in order to avoid Flu	Abstain from decrease eating in order to escape from flue	In order to be safer from flu quit dieting
This research of American scientists came in front after experimenting on mice.	This research from the American Scientists have come up after the experiments on rats.	This research of American scientists was shown after many experiments on mouses.	According to the American Scientist this research has come out after much experimentations on rats.
Experiments proved that mice on a lower calorie diet had comparatively less ability to fight the flu virus.	in has been proven from experiments that rats put on diet with less calories had less ability to resist the Flu virus.	It was proved by experiments the low calories eaters mouses had low defending power for flue in ratio.	Experimentaions have proved that those rats on less calories diet have developed a tendency of not overcoming the flu virus.
research has proven this old myth wrong that its better to fast during fever.	Research disproved the old axiom that " It is better to fast during fever"	The research proved this old talk that decrease eating is useful in fever.	This Research has proved the very old saying wrong that it is good to starve while in fever.

Figure 2: We redundantly translate each source sentence by soliciting multiple translations from different Turkers. These translations are put through a subsequent editing set, where multiple edited versions are produced. We select the best translation from the set using features that predict the quality of each translation and each translator.

each source sentence (plus the four original translations). In the ranking task, we collected judgments from five distinct workers for each translation group.

3.4 Data Collection Cost

We paid a reward of \$0.10 to translate a sentence, \$0.25 to edit a set of ten sentences, and \$0.06 to rank a set of four translation groups. Therefore, we had the following costs:

- Translation cost: \$716.80
- Editing cost: \$447.50
- Ranking cost: \$134.40

(If not done redundantly, those values would be \$179.20, \$44.75, and \$26.88, respectively.)

Adding Amazon's 10% fee, this brings the grand total to under \$1,500, spent to collect 7,000+ translations, 17,000+ edited translations, and 35,000+ rank labels.[1] We also use about 10% of the existing professional references in most of our experiments (see 4.2 and 4.3). If we estimate the cost at \$0.30/word, that would roughly be an additional \$1,000.

3.5 MTurk Participation

52 different Turkers took part in the translation task, each translating 138 sentences on average. In the editing task, 320 Turkers participated, averaging 56 sentences each. In the ranking task, 245 Turkers participated, averaging 9.1 HITs each, or 146 rank labels (since each ranking HIT involved judging 16 translations, in groups of four).

[1]Data URL: `www.cs.jhu.edu/~ozaidan/RCLMT`.

4 Quality Control Model

Our approach to building a translation set from the available data is to select, for each Urdu sentence, the one translation that our model believes to be the best out of the available translations. We evaluate various selection techniques by comparing the selected Turker translations against existing professionally-produced translations. The more the selected translations resemble the professional translations, the higher the quality.

4.1 Features Used to Select Best Translations

Our model selects one of the 14 English options generated by Turkers. For a source sentence s_i, our model assigns a score to each sentence in the set of available translations $\{t_{i,1}, ...t_{i,14}\}$. The chosen translation is the highest scoring translation:

$$tr(s_i) = tr_{i,j^*} \text{ s.t. } j^* = \underset{j}{\operatorname{argmax}}\, score(t_{i,j}) \quad (1)$$

where $score(.)$ is the dot product:

$$score(t_{i,j}) \stackrel{\text{def}}{=} \vec{w} \cdot \vec{f}(t_{i,j}) \quad (2)$$

Here, $\vec{w}$ is the model's weight vector (tuned as described below in 4.2), and $\vec{f}$ is a translation's corresponding feature vector. Each feature is a function computed from the English sentence string, the Urdu sentence string, the workers (translators, editors, and rankers), and/or the rank labels. We use 21 features, categorized into the following three sets.

Sentence-level (6 features). Most of the Turkers performing our task were native Urdu speakers whose second language was English, and they do not always produce natural-sounding English sentences. Therefore, the first set of features attempt to discriminate good English sentences from bad ones.

- Language model features: each sentence is assigned a log probability and per-word perplexity score, using a 5-gram language model trained on the English Gigaword corpus.
- Sentence length features: a good translation tends to be comparable in length to the source sentence, whereas an overly short or long translation is probably bad. We add two features that are the ratios of the two lengths (one penalizes short sentences and one penalizes long ones).
- Web n-gram match percentage: we assign a score to each sentence based on the percentage of the n-grams (up to length 5) in the translation that exist in the Google N-Gram Database.
- Web n-gram geometric average: we calculate the average over the different n-gram match percentages (similar to the way BLEU is computed). We add three features corresponding to max n-gram lengths of 3, 4, and 5.
- Edit rate to other translations: a bad translation is likely not to be very similar to other translations, since there are many more ways a translation can be bad than for it to be good. So, we compute the average edit rate distance from the other translations (using the TER metric).

Worker-level (12 features). We add *worker*-level features that evaluate a translation based on *who* provided it.

- Aggregate features: for each sentence-level feature above, we have a corresponding feature computed over *all* of that worker's translations.
- Language abilities: we ask workers to provide information about their language abilities. We have a binary feature indicating whether Urdu is their native language, and a feature for how long they have spoken it. We add a pair of equivalent features for English.
- Worker location: two binary features reflect a worker's location, one to indicate if they are located in Pakistan, and one to indicate if they are located in India.

Ranking (3 features). The third set of features is based on the ranking labels we collected (see 3.3).

- Average rank: the average of the five rank labels provided for this translation.
- Is-Best percentage: how often the translation was top-ranked among the four translations.
- Is-Better percentage: how often the translation was judged as the better translation, over all pairwise comparisons extracted from the ranks.

Other features (not investigated here) could include source-target information, such as translation model scores or the number of source words translated correctly according to a bilingual dictionary.

4.2 Parameter Tuning

Once features are computed for the sentences, we must set the model's weight vector $\vec{w}$. Naturally, the weights should be chosen so that good translations get high scores, and bad translations get low scores. We optimize translation quality against a small subset (10%) of reference (professional) translations.

To tune the weight vector, we use the linear search method of Och (2003), which is the basis of Minimum Error Rate Training (MERT). MERT is an iterative algorithm used to tune parameters of an MT system, which operates by iteratively generating new candidate translations and adjusting the weights to give good translations a high score, then regenerating new candidates based on the updated weights, etc. In our work, the set of candidate translations is *fixed* (the 14 English sentences for each source sentence), and therefore iterating the procedure is not applicable. We use the Z-MERT software package (Zaidan, 2009) to perform the search.

4.3 The *Worker Calibration* Feature

Since we use a small portion of the reference translations to perform weight tuning, we can also use that data to compute another worker-specific feature. Namely, we can evaluate the competency of each worker by scoring their translations against the reference translations. We then use that feature for every translation given by that worker. The intuition

is that workers known to produce good translations are likely to continue to produce good translations, and the opposite is likely true as well.

4.4 Evaluation Strategy

To measure the quality of the translations, we make use of the existing professional translations. Since we have *four* professional translation sets, we can calculate the BLEU score (Papineni et al., 2002) for one professional translator P_1 using the other three $P_{2,3,4}$ as a reference set. We repeat the process four times, scoring each professional translator against the others, to calculate the expected range of professional quality translation. We can see how a translation set T (chosen by our model) compares to this range by calculating T's BLEU scores against the same four sets of three reference translations. We will evaluate different strategies for selecting such a set T, and see how much each improves on the BLEU score, compared to randomly picking from among the Turker translations.

We also evaluate Turker translation quality by using them as reference sets to score various submissions to the NIST MT evaluation. Specifically, we measure the correlation (using Pearson's r) between BLEU scores of MT systems measured against non-professional translations, and BLEU scores measured against professional translations. Since the main purpose of the NIST dataset was to compare MT systems against each other, this is a more direct fitness-for-task measure. We chose the middle 6 systems (in terms of performance) submitted to the NIST evaluation, out of 12, as those systems were fairly close to each other, with less than 2 BLEU points separating them.[2]

5 Experimental Results

We establish the performance of professional translators, calculate oracle upper bounds on Turker translation quality, and carry out a set of experiments that demonstrate the effectiveness of our model and that determine which features are most helpful.

Each number reported in this section is an average of four numbers, corresponding to the four possible ways of choosing 3 of the 4 reference sets. Furthermore, each of those 4 numbers is itself based on a five-fold cross validation, where 80% of the data is used to compute feature values, and 20% used for evaluation. The 80% portion is used to compute the aggregate worker-level features. For the worker calibration feature, we utilize the references for 10% of the data (which is within the 80% portion).

5.1 Translation Quality: BLEU Scores Compared to Professionals

We first evaluated the reference sets against each other, in order to quantify the concept of "professional quality". On average, evaluating one reference set against the other three gives a BLEU score of 42.38 (Figure 3). A Turker set of translations scores 28.13 on average, which highlights the loss in quality when collecting translations from amateurs. To make the gap clearer, the output of a state-of-the-art machine translation system (the syntax-based variant of Joshua; Li et al. (2010)) achieves a score of 26.91, a mere 1.22 worse than the Turkers.

We perform two oracle experiments to determine if there exist high-quality Turker translations in the first place. The first oracle operates on the segment level: for each source segment, choose from the four translations the one that scores highest against the reference sentence. The second oracle operates on the worker level: for each source segment, choose from the four translations the one provided by the worker whose translations (over all sentences) score the highest. The two oracles achieve BLEU scores of 43.75 and 40.64, respectively – well within the range of professional translators.

We examined two voting-inspired methods, since taking a majority vote usually works well when dealing with MTurk data. The first selects the translation with the minimum average TER (Snover et al., 2006) against the other three translations, since that would be a 'consensus' translation. The second method selects the translation that received the best average rank, using the rank labels assigned by other Turkers (see 3.3). These approaches achieve BLEU scores of 34.41 and 36.64, respectively.

The main set of experiments evaluated the features from 4.1 and 4.3. We applied our approach using each of the four feature types: sentence features, Turker features, rank features, and the cali-

[2]Using all 12 systems artificially inflates correlation, due to the vast differences between the systems. For instance, the top system outperforms the bottom system by **15** BLEU points!

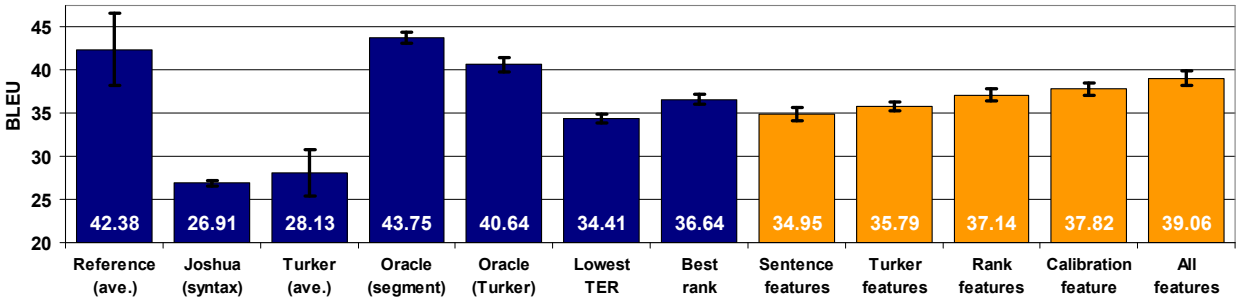

Figure 3: BLEU scores for different selection methods, measured against the reference sets. Each score is an average of four BLEU scores, each calculated against three LDC reference translations. The five right-most bars are colored in orange to indicate selection over a set that includes both original translations as well as edited versions of them.

bration feature. That yielded BLEU scores ranging from 34.95 to 37.82. With all features combined, we achieve a higher score of 39.06, which is within the range of scores for the professional translators.

5.2 Fitness for a Task: Correlation With Professionals When Ranking MT Systems

We evaluated the selection methods by measuring correlation with the references, in terms of BLEU scores assigned to outputs of MT systems. The results, in Table 1, tell a fairly similar story as evaluating with BLEU: references and oracles naturally perform very well, and the loss in quality when selecting arbitrary Turker translations is largely eliminated using our selection strategy.

Interestingly, when using the Joshua output as a reference set, the performance is quite abysmal. Even though its BLEU score is comparable to the Turker translations, it cannot be used to distinguish closely matched MT systems from each other.[3]

6 Analysis

The oracles indicate that there is usually an acceptable translation from the Turkers for any given sentence. Since the oracles select from a small group of only 4 translations per source segment, they are not overly optimistic, and rather reflect the true potential of the collected translations.

The results indicate that, although some features are more useful than others, much of the benefit from combining all the features can be obtained from any one set of features, with the benefit of adding more features being somewhat orthogonal.

[3]It should be noted that the Joshua system was not one of the six MT systems we scored in the correlation experiments.

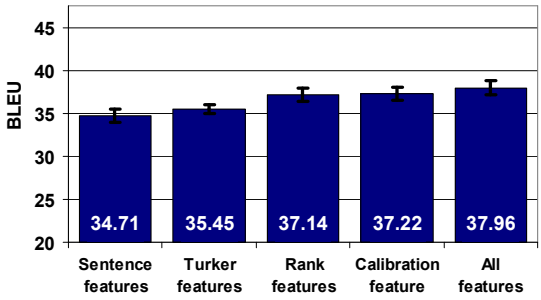

Figure 4: BLEU scores for the five right-most setups from Figure 3, constrained over the original translations.

Finally, we performed a series of experiments exploring the calibration feature, varying the amount of gold-standard references from 10% all the way up to 80%. As expected, the performance improved as more references were used to calibrate the translators (Figure 5). What's particularly important about this experiment is that it shows the added benefit of the other features: We would have to use 30%–40% of the references to get the same benefit obtained from combining the non-calibration features and only 10% for the calibration feature (dashed line in the Figure; BLEU = 39.06).

6.1 Cost Reduction

While the combined cost of our data collection effort ($2,500; see 3.4) is quite low considering the amount of collected data, it would be more attractive if the cost could be reduced further without losing much in translation quality. To that end, we investigated lowering cost along two dimensions: eliminating the need for professional translations, and decreasing the amount of edited translations.

Selection Method	Pearson's r^2
Reference (ave.)	0.81 ± 0.07
Joshua (syntax)	0.08 ± 0.09
Turker (ave.)	0.60 ± 0.17
Oracle (segment)	0.81 ± 0.09
Oracle (Turker)	0.79 ± 0.10
Lowest TER	0.50 ± 0.26
Best rank	0.74 ± 0.17
Sentence features	0.56 ± 0.21
Turker features	0.59 ± 0.19
Rank features	0.75 ± 0.14
Calibration feature	0.76 ± 0.13
All features	0.77 ± 0.11

Table 1: Correlation ($\pm$ std. dev.) for different selection methods, compared against the reference sets.

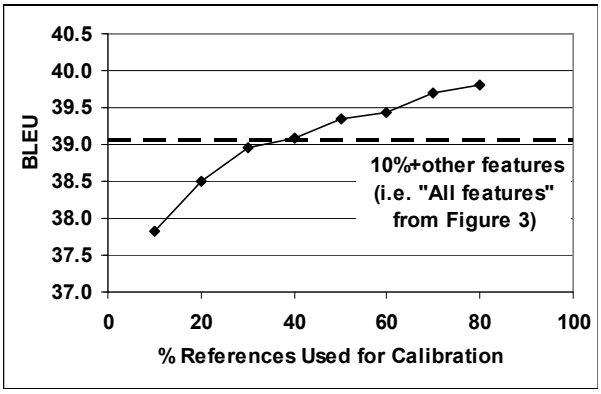

Figure 5: The effect of varying the amount of calibration data (and using only the calibration feature). The 10% point (BLEU = 37.82) and the dashed line (BLEU = 39.06) correspond to the two right-most bars of Figure 3.

The professional translations are used in our approach for computing the worker calibration feature (subsection 4.3) and for tuning the weights of the other features. We use a relatively small amount for this purpose, but we investigate a different setup whereby no professional translations are used at all. This eliminates the worker calibration feature, but, perhaps more critically, the feature weights must be set in a different fashion, since we cannot optimize BLEU on reference data anymore. Instead, we use the rank labels (from 3.3) as a proxy for BLEU, and set the weights so that better ranked translations receive higher scores.

Note that the rank features will also be excluded in this setup, since they are perfect predictors of rank labels. On the one hand, this means no rank labels need to be collected, other than for a small set used for weight tuning, further reducing the cost of data collection. However, this leads to a significant drop in performance, yielding a BLEU score of 34.86.

Another alternative for cost reduction would be to reduce the number of collected edited translations. To that end, we first investigate completely eliminating the editing phase, and considering only unedited translations. In other words, the selection will be over a group of four English sentences rather than 14 sentences. Completely eliminating the edited translations has an adverse effect, as expected (Figure 4). Another option, rather than eliminating the editing phase altogether, would be to consider the edited translations of only the translation receiving the best rank labels. This would reflect a data collection process whereby the editing task is delayed until after the rank labels are collected, with the rank labels used to determine which translations are most promising to post-edit (in addition to using the rank labels for the ranking features). Using this approach enables us to greatly reduce the number of edited translations collected, while maintaining good performance, obtaining a BLEU score of 38.67.

It is therefore our recommendation that crowdsourced translation efforts adhere to the following pipeline: collect multiple translations for each source sentence, collect rank labels for the translations, and finally collect edited versions of the top ranked translations.

7 Related Work

Dawid and Skene (1979) investigated filtering annotations using the EM algorithm, estimating annotator-specific error rates in the context of patient medical records. Snow et al. (2008) were among the first to use MTurk to obtain data for several NLP tasks, such as textual entailment and word sense disambiguation. Their approach, based on majority voting, had a component for annotator bias correction. They showed that for such tasks, a few non-expert labels usually suffice.

Whitehill et al. (2009) proposed a probabilistic model to filter labels from non-experts, in the context of an image labeling task. Their system generatively models image difficulty, as well as noisy, even

adversarial, annotators. They apply their method to simulated labels rather than real-life labels.

Callison-Burch (2009) proposed several ways to evaluate MT output on MTurk. One such method was to collect reference translations to score MT output. It was only a pilot study (50 sentences in each of several languages), but it showed the possibility of obtaining high-quality translations from non-professionals. As a followup, Bloodgood and Callison-Burch (2010) solicited a single translation of the NIST Urdu-to-English dataset we used. Their evaluation was similar to our correlation experiments, examining how well the collected translations agreed with the professional translations when evaluating three MT systems.

That paper appeared in a NAACL 2010 workshop organized by Callison-Burch and Dredze (2010), focusing on MTurk as a source of data for speech and language tasks. Two relevant papers from that workshop were by Ambati and Vogel (2010), focusing on the design of the translation HIT, and by Irvine and Klementiev (2010), who created translation lexicons between English and 42 rare languages.

Resnik et al. (2010) explore a very interesting way of creating translations on MTurk, relying only on *monolingual* speakers. Speakers of the target language iteratively identified problems in machine translation output, and speakers of the source language paraphrased the corresponding source portion. The paraphrased source would then be retranslated to produce a different translation, hopefully more coherent than the original.

8 Conclusion and Future Work

We have demonstrated that it is possible to obtain high-quality translations from non-professional translators, and that the cost is an order of magnitude cheaper than professional translation. We believe that crowdsourcing can play a pivotal role in future efforts to create parallel translation datasets. Beyond the cost and scalability, crowdsourcing provides access to languages that currently fall outside the scope of statistical machine translation research. We have begun an ongoing effort to collect translations for several low resource languages, including Tamil, Yoruba, and dialectal Arabic. We plan to:

- Investigate improvements from system combination techniques to the redundant translations.
- Modify our editing step to collect an annotated corpus of English as a second language errors.
- Calibrate against good Turkers, instead of professionals, once they have been identified.
- Predict whether it is necessary to solicit another translation instead of collecting a fixed number.
- Analyze how much quality matters if our goal is to train a statistical translation system.

Acknowledgments

This research was supported by the Human Language Technology Center of Excellence, by gifts from Google and Microsoft, and by the DARPA GALE program under Contract No. HR0011-06-2-0001. The views and findings are the authors' alone.

We would like to thank Ben Bederson, Philip Resnik, and Alain Désilets for organizing workshops focused on crowdsourcing translation (Bederson and Resnik, 2010; Désilets, 2010). We are grateful for the feedback of workshop participants, which helped shape this research.

References

Yaser Al-Onaizan, Ulrich Germann, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Daniel Marcu, and Kenji Yamada. 2002. Translation with scarce bilingual resources. *Machine Translation*, 17(1), March.

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

Ben Bederson and Philip Resnik. 2010. Workshop on crowdsourcing and translation. `http://www.cs.umd.edu/hcil/monotrans/workshop/`.

Michael Bloodgood and Chris Callison-Burch. 2010. Using Mechanical Turk to build machine translation evaluation sets. In *Proceedings of the NAACL HLT Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*, pages 208–211.

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

Chris Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Me-

chanical Turk. In *Proceedings of EMNLP*, pages 286–295.

A. P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28.

Alain Désilets. 2010. AMTA 2010 workshop on collaborative translation: technology, crowdsourcing, and the translator perspective. `http://bit.ly/gPnqR2`.

Pascale Fung and Lo Yuen Yee. 1998. An ir approach for translating new words from nonparallel, comparable texts. In *Proceedings of ACL/CoLing*.

Ulrich Germann. 2001. Building a statistical machine translation system from scratch: How much bang for the buck can we expect? In *ACL 2001 Workshop on Data-Driven Machine Translation*, Toulouse, France.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL/HLT*.

Panos Ipeirotis. 2010. New demographics of Mechanical Turk. `http://behind-the-enemy-lines.blogspot.com/2010/03/new-demographics-of-mechanical-turk.html`.

Ann Irvine and Alexandre Klementiev. 2010. Using Mechanical Turk to annotate lexicons for less commonly used languages. In *Proceedings of the NAACL HLT Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*, pages 108–113.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Ann Irvine, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Ziyuan Wang, Jonathan Weese, and Omar Zaidan. 2010. Joshua 2.0: A toolkit for parsing-based machine translation with syntax, semirings, discriminative training and other goodies. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 133–137.

Dragos Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting comparable corpora. *Computational Linguistics*, 31(4):477–504, December.

Sonja Niessen and Hermann Ney. 2004. Statistical machine translation with scarce resources using morpho-syntatic analysis. *Computational Linguistics*, 30(2):181–204.

Doug Oard, David Doermann, Bonnie Dorr, Daqing He, Phillip Resnik, William Byrne, Sanjeeve Khudanpur, David Yarowsky, Anton Leuski, Philipp Koehn, and Kevin Knight. 2003. Desperately seeking Cebuano. In *Proceedings of HLT/NAACL*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, pages 160–167.

Kishore Papineni, Salim Poukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Katharina Probst, Lori Levin, Erik Peterson, Alon Lavie, and Jamie Carbonell. 2002. MT for minority languages using elicitation-based learning of syntactic transfer rules. *Machine Translation*, 17(4).

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of ACL*.

Philip Resnik and Noah Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380, September.

Philip Resnik, Olivia Buzek, Chang Hu, Yakov Kronrod, Alex Quinn, and Benjamin Bederson. 2010. Improving translation via targeted paraphrasing. In *Proceedings of EMNLP*, pages 127–137.

Charles Schafer and David Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. In *Conference on Natural Language Learning-2002*, pages 146–152.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 403–411, Los Angeles, California, June. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas (AMTA)*.

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 EMNLP*, pages 254–263.

Jakob Uszkoreit, Jay M. Ponte, Ashok C. Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proc. of the International Conference on Computational Linguistics (COLING)*.

Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Proceedings of NIPS*, pages 2035–2043.

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.