Characterizing Cost Reduction v.s. Translation Quality in Balance

Yucong Li, Alyssa Mensch, Chunxiao Mu, and Rui Yan
School of Engineering and Applied Science,
University of Pennsylvania, Philadelphia, PA 19104, USA

1 Introduction

Crowdsourcing based techniques have emerged to been the rising star for Machine Translation, with prominent advantages of lower cost in money expenditure to collect the processed data. However, when compared with translation by trained professionals, naive collected results from non-professional translators possibly yields low-quality outputs. On one hand, we could design effective methods to select the candidate translations with acceptable quality out of multiple redundant crowdsourced translations (Yan et al., 2014; Zaidan and Callison-Burch, 2011). On the other hand, we could define the target as selecting reliable non-professional translators so that their translations could be generally deemed as decent ones. Another merit for narrowing down the pool of candidate translator is that we could reduce the cost for crowdsourcing translation: we do not need to pay for unacceptable efforts of translations.

To investigate the potential benefits of reducing the labor force of crowdsouring workers, we investigate a series of worker ranking strategies to determine the quality of workers, and moreover, the tradeoff between cost reduction by shortlisting workers and the overall translation quality. To be more specific, given the source sentences to translate and the corresponding candidate translations authored by crowdsourcing workers, we rank the workers according to their translation abilities (i.e., the quality of their translations). Then we propose to keep a smaller subset from the whole worker list, trying to 1) maintain acceptable translation quality, 2) further reduce the costs and 3) characterizing the trade-off between cost reduction and translation quality.

2 Methods

2.1 Random Ranking

We first randomly rank all involved crowdsourcing workers and use the random list as the baseline for other methods to compare with. We run random ranking for 10 times and evaluate the performance of the theoretical lower bound.

2.2 Referential Ranking

The second method used in our experiment is actually based on the ground truth references. Given a worker t and all the candidate translations by t which could be denoted as $S^t = \{s_i^t | s_i^t$ authored by $t\}$. The quality score for a particular worker t is calculated as:

$$score(t) = \sum_{s_i^t \in S^t} (avgTER(s_i^t, \{ref(s_i^t)\})) \frac{1}{|S^t|} \quad (1)$$

where $ref(s)$ denotes the reference set for the sentence s, and $avgTER$ denotes the average TER score for this candidate translation compared with the set of references. When taking all candidate translations by this worker, we calculate the average TER score for all the translations authored by this particular worker, and then rank all workers according to this score. Intuitively, the less TER score means better translation quality, and hence the corresponding worker should be ranked higher.

Basically, the referential ranking could be regarded as the gold ranking for crowdsourcing workers.

2.3 Single Layer Random Walk

We frame the candidate sentences into a graph, modeling the relationships between sentences, i.e., semantic similarity. Let G denote the graph with nodes V and edges E, and $G = (V,E)$, which is a weighted undirected graph representing the candidate translations and their correlations (represented by the textual similarity between the translated sentences). Note that we only establish linkage between the candidate translation groups for the same source sentences. In other words, we do not establish textual linkage between candidate translations for different source sentences to trans-

late. The reason is that we do not want influence from other sentence groups will have biased impact on the quality judgement for a particular candidate translation group.

In this sense, we have a series of isolated subgraphs, each represents a source sentence with linkage established inside the sub-graph. We run random walks on these subgraphs. For every subgraph, a random walk on a graph is a Markov chain, its states being the vertices of the graph. It can be described by a square matrix, where the dimension is the number of vertices in the graph. The stochastic matrix prescribing the transition probabilities from one vertex to the next. Within each sub-graph, the score of candidate translations are calculated as follows:

$$\mathbf{c} = (1-\mu)M^T\mathbf{c} + \mu\frac{1}{|V_c|} \tag{2}$$

After we get the random walk scores for all the sentences, we attribute the scores of the candidate sentences to their corresponding authors:

$$\mathbf{t} = \frac{1}{|c_t|}\sum_{c_t}\mathbf{c}_{t_i} \tag{3}$$

In the end, we rank the workers according to their average translation scores by random walk.

2.4 Two-Layer Random Walk

Finally, we include the post-editors and the translator-editor collaboration into the graph framework. We propose our method operates over a heterogeneous network that includes workers (both translators and post-editors) and translated sentences. We frame both components into graphs, using relationships to connect these parts as a ranking paradigm (Yan et al., 2012). Let G denote the heterogeneous graph with nodes V and edges E, and G = (V,E) = $(V_C, V_T, E_C, E_T, E_{CT})$. G is divided into three subgraphs, G_T, G_C, and G_{CT}. G_C = (V_C, E_C) is a weighted undirected graph representing the candidate translations and their relationships. Let V_C = $\{c_i|c_i \in V_C\}$ denote a collection of $|V_C|$ translated and edited sentences, and E_C the set of linkage representing affinity between them, established by textual similarity between the translated sentences. G_T = (V_T, E_T) is a weighted undirected graph representing the collaborative ties among Turkers. V_T = $\{t_i|t_i \in V_T\}$ is the set of working pairs with size $|V_T|$. Links E_T among Turkers are established by their shared *translation* and *post-editing* collaborations. Each collaboration would produce an output translation. $G_{CT} = (V_{CT}, E_{CT})$ is an unweighted bipartite graph that ties G_T and G_C together and represents "authorship". The graph G consists of nodes V_{CT} = $V_T \cup V_C$ and edges E_{CT} connecting each candidate with its generators. Typically, a candidate is generated by the collaboration of a translator and a post-editor.

• We use adjacency matrix $[M]_{|\mathbf{c}|\times|\mathbf{c}|}$ to describe the homogeneous affinity between candidates and $[N]_{|\mathbf{t}|\times|\mathbf{t}|}$ to describe the affinity between Turkers.

$$\mathbf{c} \propto M^T\mathbf{c}, \qquad \mathbf{t} \propto N^T\mathbf{t} \tag{4}$$

where $c = |V_C|$ is the number of vertices in the candidate graph and $t = |V_T|$ is the number of vertices in the Turker graph. The adjacency matrix $[M]$ denotes the transition probabilities between candidates, and analogously matrix $[N]$ denotes the affinity between Turker collaboration pairs.

• We use an adjacency matrix $[\hat{W}]_{|\mathbf{c}|\times|\mathbf{t}|}$ and $[\bar{W}]_{|\mathbf{t}|\times|\mathbf{c}|}$ to describe the authorship between the output candidate and the producer Turker pair from both of the candidate-to-Turker and Turker-to-candidate perspectives.

$$\mathbf{c} \propto \hat{W}^T\mathbf{t}, \qquad \mathbf{t} \propto \bar{W}^T\mathbf{c} \tag{5}$$

Step 1: compute the saliency scores of candidates, and then normalize using ℓ-1 norm.

$$\begin{aligned}\mathbf{c}^{(n)} &= (1-\lambda)M^T\mathbf{c}^{(n-1)} + \lambda\hat{W}\mathbf{t}^{(n-1)}\\ \mathbf{c}^{(n)} &= \mathbf{c}^{(n)}/||\mathbf{c}^{(n)}||_1\end{aligned} \tag{6}$$

Step 2: compute the saliency scores of Turker pairs, and then normalize using ℓ-1 norm.

$$\begin{aligned}\mathbf{t}^{(n)} &= (1-\lambda)N^T\mathbf{t}^{(n-1)} + \lambda\bar{W}\mathbf{c}^{(n-1)}\\ \mathbf{t}^{(n)} &= \mathbf{t}^{(n)}/||\mathbf{t}^{(n)}||_1\end{aligned} \tag{7}$$

where λ specifies the relative contributions to the saliency score trade-off between the homogeneous affinity and the heterogeneous affinity. In order to guarantee the convergence of the iterative form, we must force the transition matrix to be stochastic and irreducible. To this end, we must make the $\mathbf{c}$ and $\mathbf{t}$ *column stochastic* (Langville and Meyer, 2004). $\mathbf{c}$ and $\mathbf{t}$ are therefore normalized after each iteration of equations.

2.4.1 Intra-Graph Ranking

The standard PageRank algorithm starts from an arbitrary node and randomly selects to either follow a random out-going edge (considering the weighted transition matrix) or to jump to a random node (treating all nodes with equal probability).

In a simple random walk, it is assumed that all nodes in the transitional matrix are equi-probable before the walk starts. Then **c** and **t** are calculated as:

$$\mathbf{c} = \mu M^T \mathbf{c} + (1-\mu)\frac{\mathbf{1}}{|V_C|} \tag{8}$$

and

$$\mathbf{t} = \mu N^T \mathbf{t} + (1-\mu)\frac{\mathbf{1}}{|V_T|} \tag{9}$$

where **1** is a vector with all elements equaling to 1 and the size is correspondent to the size of V_C or V_T. μ is the damping factor usually set to 0.85, as in the PageRank algorithm.

2.4.2 Affinity Matrix Establishment

We introduce the affinity matrix calculation, including homogeneous affinity (i.e., M, N) and heterogeneous affinity (i.e., $\hat{W}, \bar{W}$).

As discussed, we model the collection of candidates as a weighted undirected graph, G_C, in which nodes in the graph represent candidate sentences and edges represent lexical relatedness. We define an edge's weight to be the cosine similarity between the candidates represented by the nodes that it connects. The adjacency matrix M describes such a graph, with each entry corresponding to the weight of an edge.

$$\begin{aligned} \mathcal{F}(c_i, c_j) &= \frac{c_i \cdot c_j}{||c_i||||c_j||} \\ M_{ij} &= \frac{\mathcal{F}(c_i, c_j)}{\sum_k \mathcal{F}(c_i, c_k)} \end{aligned} \tag{10}$$

where $\mathcal{F}(.)$ is the cosine similarity and c is a term vector corresponding to a candidate. We treat a candidate as a short document and weight each term with *tf.idf* (Manning et al., 2008), where *tf* is the term frequency and *idf* is the inverse document frequency.

The Turker graph, G_T, is an undirected graph whose edges represent "collaboration." Formally, let t_i and t_j be two translator/editor pairs; we say that pair t_i "collaborates with" pair t_j (and therefore, there is an edge between t_i and t_j) if t_i and t_j share either a translator or an editor (or share both a translator and an editor). Let the function $\mathcal{I}(t_i, t_j)$ denote the number of "collaborations" ($\#col$) between t_i and t_j.

$$\mathcal{I}(t_i, t_j) = \begin{cases} \#col & (e_{ij} \in E_T) \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

Then the adjacency matrix N is then defined as

$$N_{ij} = \frac{\mathcal{I}(t_i, t_j)}{\sum_k \mathcal{I}(t_i, t_k)} \tag{12}$$

In the bipartite candidate-Turker graph G_{TC}, the entry $E_{TC}(i, j)$ is an indicator function denoting whether the candidate c_i is generated by t_j:

$$\mathcal{A}(c_i, t_j) = \begin{cases} 1 & (e_{ij} \in E_{TC}) \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Through E_{TC} we define the weight matrices $\bar{W}_{ij}$ and $\hat{W}_{ij}$, containing the conditional probabilities of transitions from c_i to t_j and vice versa:

$$\begin{aligned} \bar{W}_{ij} &= \frac{\mathcal{A}(c_i, t_j)}{\sum_k \mathcal{A}(c_i, t_k)}, \\ \hat{W}_{ij} &= \frac{\mathcal{A}(c_i, t_j)}{\sum_k \mathcal{A}(c_k, t_j)} \end{aligned} \tag{14}$$

After the co-ranking process, we could get the ranking score for each collaboration pairs, and we attribute the pair score to the translator from the working pair, since in this study, we mainly focus on choosing the qualified translators.

3 Experiments

3.1 Evaluation Metric

In the first place, we propose to keep the subset of workers who could finish all the source sentences, and examine the proportion of the remained workers. Besides, we remove the workers from bottom-up, one at a time, and then we examine the performance variations in correspondence.

• The most intuitive evaluation metric is to measure how many workers are kept after the ranking and selection process: we desire to keep as few workers as possible to reduce the costs to hire workers.

• Since we have four professional translation sets, we can calculate the Bilingual Evaluation Understudy (BLEU) score (Papineni et al., 2002) for one professional translator (P1) using the other three (P2,3,4) 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.

Evaluation	# Remained	Ratio	BLEU
Random	41	0.839	41.64
One-Layer	34	0.667	**47.24**
Two-Layer	27	**0.529**	45.34
Referential	24	0.471	46.74
Full Set	51	1.000	48.44

Table 1: Performance comparison for 4 methods.

We evaluate each of our methods by calculating BLEU scores against the same four sets of three reference translations. Therefore, each number reported in our experimental results is an average of four numbers, corresponding to the four possible ways of choosing 3 of the 4 reference sets. This allows us to compare the BLEU score achieved by our methods against the BLEU scores achievable by professional translators. Intuitively, for BLEU scores, we hope to achieve high BLEU scores as possible.

3.2 Performance and Analysis

The results are listed as follows in Table 1. From the table, we could see that the one-layer random walk based method achieved the highest BLEU performance with the score of 47.24 achieved while 34% workers could be removed, as to the two-layer co-ranking method, the performance is not as ideal as single layer rank but the number of workers could be further reduced (about 50% workers could be removed), while the performance would be 45.34. The random method is as expected the worst of all four methods. For the referential method, it could achieve balanced performance as well but in general, we would not have references during testing in practice.

The removal of workers in a bottom-up fashion for all methods are shown in Figure 1-4. From the figures, we could see in general, as to the performance in BLEU score, one-layer method achieves the best results (and steady results as well)! The reason for the less promising results generated from the co-ranking paradigm might be that we have included the post-editors and pair the translator with post-editors. In this sense, the quality judgment should actually be credited to the pair rather than the translator. The translation generated by a high quality pair might result from the reliable post-editor working with an unreliable translator, and hence the credits to translators within the pair might be unfair.

References

Amy N Langville and Carl D Meyer. 2004. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

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 '02, pages 311–318. Association for Computational Linguistics.

Rui Yan, Mirella Lapata, and Xiaoming Li. 2012. Tweet recommendation with graph co-ranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 516–525, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rui Yan, Mingkun Gao, Ellie Pavlick, and Chris Callison-Burch. 2014. Are two heads better than one? crowdsourced translation via a two-step collaboration of non-professional translators and editors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL'14, Stroudsburg, PA, USA. 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 - Volume 1*, HLT '11, pages 1220–1229, Stroudsburg, PA, USA. Association for Computational Linguistics.

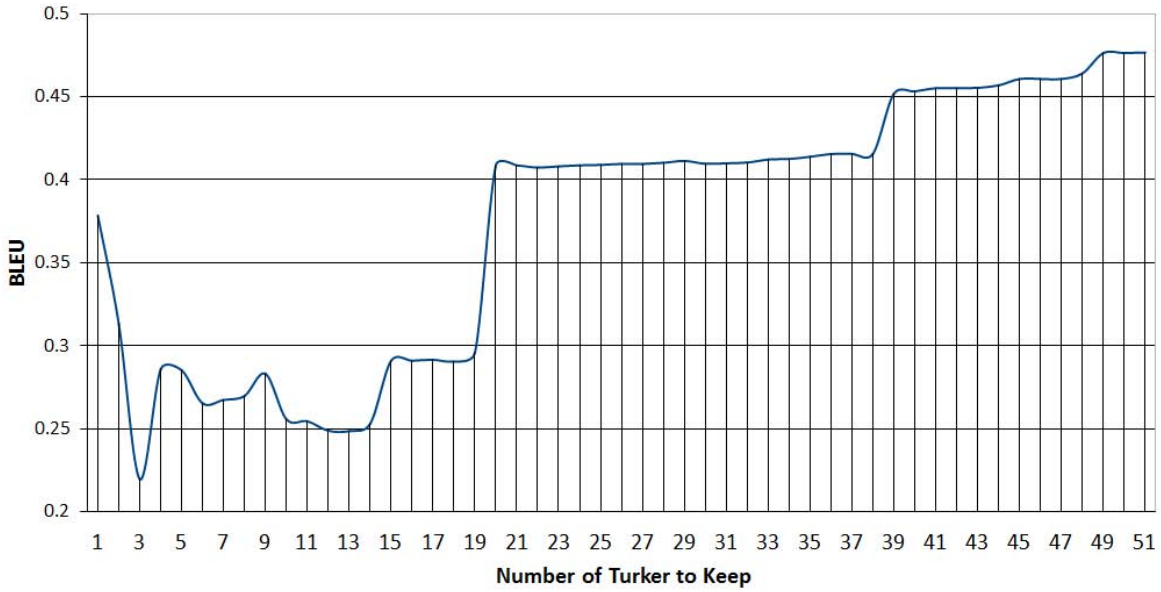

Figure 1: Worker-BLEU balance for random method.

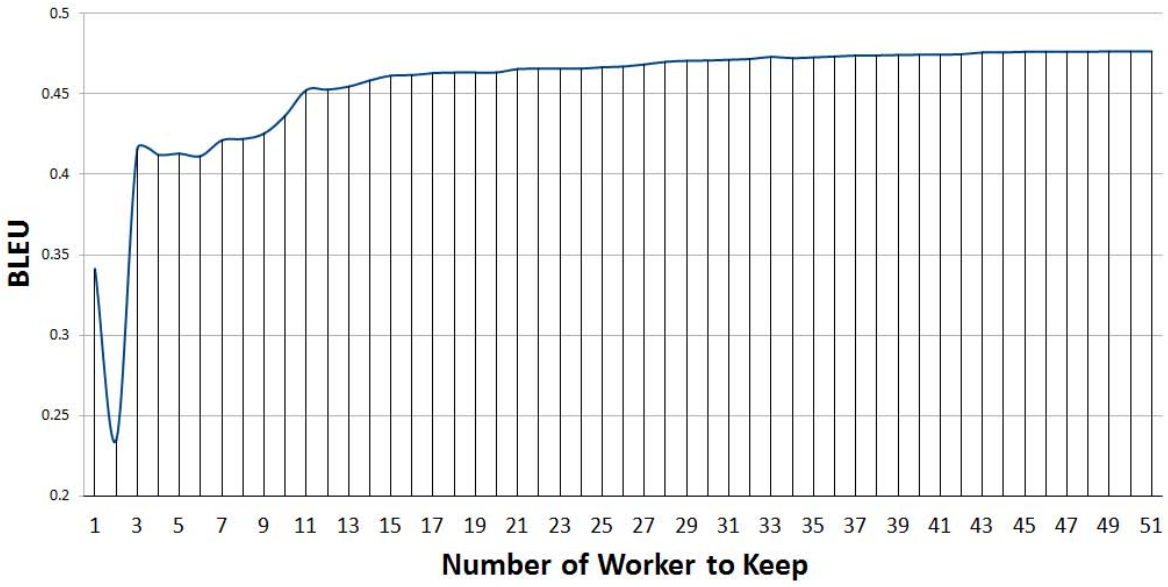

Figure 2: Worker-BLEU balance for one-layer random walk method.

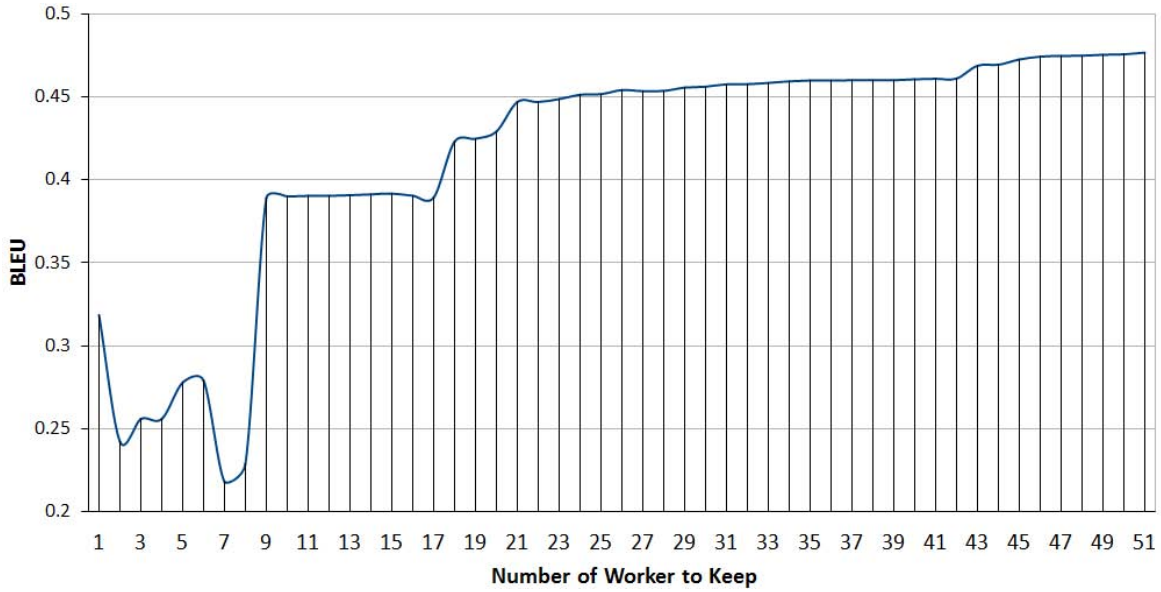

Figure 3: Worker-BLEU balance for two-layer co-ranking method.

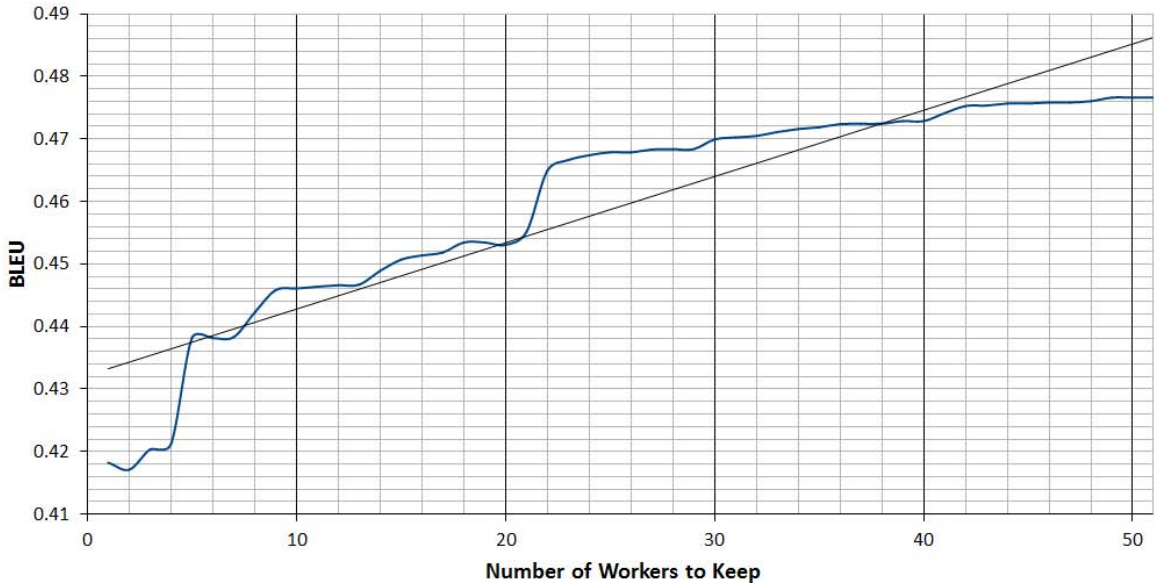

Figure 4: Worker-BLEU balance for referential method.