Opportunities or Risks to Reduce Labor in Crowdsourcing Translation? Characterizing Cost Versus Quality via a PageRank-HITS Hybrid Model

Rui Yan*, **Yiping Song**†, **Cheng-Te Li**‡, **Ming Zhang**†, **and Xiaohua Hu**♮
*Dept. of Computer and Information Science, University of Pennsylvania, USA
†Dept. of Computer Science, Peking University, China
‡Academia Sinica, Taiwan
♮College of Computing and Informatics, Drexel University, USA
ruiyan@seas.upenn.edu, syp@pku.edu.cn,
mzhang_cs@pku.edu.cn, ctli@citi.sinica.edu.tw, xh29@drexel.edu

Abstract

Crowdsourcing machine translation shows advantages of lower expense in money to collect the translated data. Yet, when compared with translation by trained professionals, results collected from non-professional translators might yield low-quality outputs. A general solution for crowdsourcing practitioners is to employ a large amount of labor force to gather enough redundant data and then solicit from it. Actually we can further save money by avoid collecting bad translations. We propose to score Turkers by their authorities during observation, and then stop hiring the unqualified Turkers. In this way, we bring both opportunities and risks in crowdsourced translation: we can make it *cheaper* than *cheaper* while we might suffer from quality loss. In this paper, we propose a graph-based PageRank-HITS Hybrid model to distinguish authoritative workers from unreliable ones. The algorithm captures the intuition that good translation and good workers are mutually reinforced iteratively in the proposed frame. We demonstrate the algorithm will keep the performance while reduce work force and hence cut cost. We run experiments on the NIST 2009 Urdu-to-English evaluation set with Mechanical Turk, and quantitatively evaluate the performance in terms of BLEU score, Pearson correlation and real money.

1 Introduction

Nowadays, globalization brings frequent and closer international connections but automatic solutions to come over the barrier between different languages remains to be a problem, which stimulates a myriad of different researches. In recent Natural Language Processing studies, automatic translations are generally based on training data using statistical machine translation (SMT), where systems are trained using bilingual sentence-aligned parallel corpora. A perfect SMT instance could be ascribed to data like the Canadian Hansards (which by law must be published in both French and English), but the real prosperous existence of SMT owes to sufficient parallel linguistic data available on the Internet, e.g., multiple versions of news reports about an event described in various languages. Theoretically, SMT could be addressed for language pairs with ample data, which actually produces the state-of-art results for language pairs in this case such as English-Chinese, French-English, etc. However, SMT would get stuck in a severe bottleneck when facing with many relatively low-resource languages with insufficient annotated data: not enough bilingual parallel corpora is available.

In this situation, to collect more parallel corpora becomes a necessity for the success of SMT to process minor languages. There are various options to create new training resources for new language pairs, which 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]. However, without human supervision, such data collected from the Internet has a large probability to be inappropriately aligned, which would lead to a fatal failure when applying SMT techniques. Another intuitive way is to simply hire human translators to create enough high quality parallel data, which is conducted mostly by Linguistic Data Consortium (LDC). As well, this method receives relatively little favorable consideration for two reasons: the task requires professionally trained annotators and moreover, hiring these professionals would seem to be prohibitively expensive. Germann estimated the cost of hiring professional translators to create a Tamil-English corpus at $0.36/word [Germann, 2001]. At that rate, translating data to build even a small parallel corpus like 1.5 million words would exceed half a million dollars (which is a lot of money)!

A worthy effort would be seeking for high quality translators at a low cost. We notice that Amazon Mechanical Turk (AMT) provides a labor force platform at a cheap price for labor units, and hence we are able to hire a large group of translators, non-professional or maybe professional, at a similarly tempting low price. In the way of crowdsoucing, we

*Dr. Rui Yan now works in Baidu Research.

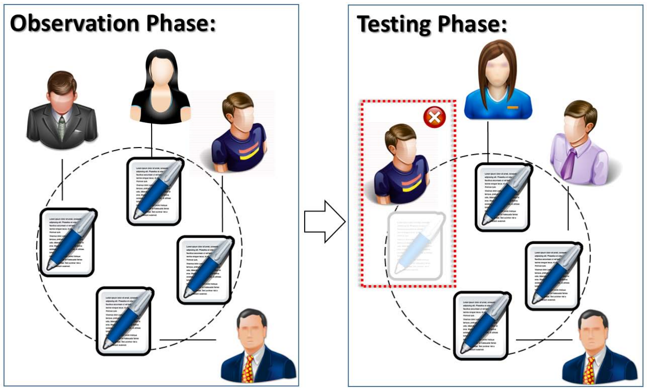

Figure 1: When identify unreliable workers, we will stop hiring them, and then test if the performance gets hurt.

could manage to create a large parallel corpus at a fraction of the cost of professional translators. With affordable human computation achieved, we aim at soliciting high quality translations out of the redundant, perhaps low quality, disfluent generations. Interestingly, we observe that sloppy translations are in general produced by unreliable or unauthorized workers. Hence we propose to distinguish good workers from bad workers, and then stop hiring some workers in future assignments (illustrated in Figure 1). To reduce labor force in crowdsourcing brings opportunities to save money but also leads potential risks to hurt the performance. Therefore, we ought to characterize the balance between costs and quality.

We have proposed a PageRank-HITS (Hyperlink-Induced Topic Search) Hybrid model based on a heterogenous graph consisting of translations and the workers (namely Turkers) on AMT. The model is a mutual reinforced ranking algorithm to capture the general intuition that good workers produce good translations. In this way, we could get the list of workers ranked by their authority and then we know who to hire or not. In particular, we design a PageRank frame [Page *et al.*, 1999] to rank candidate translations and then group redundant translations for the same source language into a document cluster. The document cluster forms a meta-node and we connect workers to the meta-nodes as a HITS frame [Kleinberg, 1999]. We update the scores from both frames iteratively to obtain the final ranking scores for the workers. The model discriminates acceptable Turkers from unreliable ones.

According to an objective and quantitative comparison with the professionally-produced reference translations, we show that it is possible to maintain the quality of crowdsourcing translation (95%) even when we cut nearly half of the labor force. In other words, these redundant translations by unreliable workers are not quite helpful for us to solicit high quality translations in aggregate and the money spent on unauthorized workers can actually be saved. To reduce the work force can bring opportunities to spend money on more worthy workers while at the same time avoid risks.

We start by reviewing related work in Section 2. We then introduce the crowdsourcing platform and data collection. In Section 4 we formulate a reinforced ranking model, namely PageRank-HITS Hybrid, and we describe experimental results in Section 5. We draw conclusions in Section 6.

2 Related Work

Extraneous data source could always be a supplement to improve machines translation models so that they are better suited to the low resource setting [Al-Onaizan *et al.*, 2002; Nießen and Ney, 2004]. Snow *et al.* were among the first to use Mechanical Turk (MTurk) to obtain data for several NLP tasks, such as textual entailment and word sense disambiguation [2008]. Their approach, based on majority voting, had a component for annotator bias correction. They showed that the platform was a viable way of collecting data for a wide variety of NLP tasks at low cost and in large volumes. Also, they further showed that non-expert annotations are similar to expert annotations.

For years, people have sought ways to solicit high quality outputs from non-expert workers [Ambati *et al.*, 2010]. Dawid and Skene investigated filtering annotations using the EM algorithm, estimating annotator error rates in patient medical records [1979]. Whitehill *et al.* proposed a probabilistic model to filter labels from non-experts for an image labeling task [2009]. Callison-Burch proposed several ways to evaluate MT output on MTurk [2009]. It showed the possibility of obtaining high-quality translations from non-professionals. As a follow-up, researchers solicited a single translation of the NIST Urdu-English dataset [Bloodgood and Callison-Burch, 2010; Zaidan and Callison-Burch, 2011; Yan *et al.*, 2014].

MTurk has subsequently been widely adopted by the NLP community and used for an extensive range of language applications [Callison-Burch and Dredze, 2010]. There is an underlying assumption that all participants are cooperative. Our setup uses anonymous crowd workers hired on Mechanical Turk, but Turkers can be unreliable or overly zealous [Bernstein *et al.*, 2010]. Different Turkers finish tasks with different quality. The basic idea behind crowdsourcing tasks is raising redundancy in large volumes at low costs to select good outputs. Our target is to control the quality of source, i.e., workers, so that we can reduce redundancy while maintain the performance. The idea can save the costs, bring more opportunities and avoid risks. To the best of our knowledge, we are the first study to control Turker quality in machine translation, and characterize costs and quality in balance.

3 Crowdsourcing Platform and Task

To collect crowdsourced translations, we deploy our platform based on Amazon Mechanical Turk, an online market-place designed to pay people small sums of money to complete Human Intelligence Tasks, which are difficult for computers but easy for people. Example task ranges from labeling images or annotating texts and semantics to providing feedback on relevance of results for search queries [Zaidan and Callison-Burch, 2011]. Anyone with an Amazon account can either submit a task or work on a task that were submitted by others. Workers are referred to as “Turkers”, and designers of a task as “Requesters”. A Requester specifies the reward to be paid for each completed item. The relationship between Turkers and Requesters is designed to be a mutual selection:

Turkers are free to select whichever task interests them, and requesters can choose not to pay for unsatisfied results.

The advantages of Mechanical Turk are obvious: zero overhead for hiring workers with a large, low-cost labor force and the task can be completed in a naturally parallel pattern by vast individuals so that the turnaround time is short. For Natural Language Processing, it is easier to access to foreign markets with native speakers of many rare languages. On the other hand, the Turkers are completely anonymous without any personal profile other than a Turker ID (e.g., A143AWKU99STC9). Hence it is difficult to determine if a non-professional is qualified to fulfill the task before the Turker submits the task.

Soliciting translations from anonymous non-professionals carries a risk of poor translation quality. To improve the accuracy of noisy translations from non-experts, a natural quality control solution would be to quantitatively rank the reliability of workers and hire only trustworthy wisdom of the crowds.

Our translation task involves showing the worker a sequence of sentences in source language (i.e., Urdu in this paper), 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. We solicit four translations per Urdu sentence (from distinct translators). We instead split the data set into groups of 10 sentences per task. We keep some of the strategies used in other crowdsourcing systems in designing interfaces [Zaidan and Callison-Burch, 2011; Yan *et al.*, 2014]. For instance, we converted the Urdu sentences into images so that Turkers cannot cheat by copying-and-pasting the Urdu text into an online commercial MT system such as Google translation. In general, the tasks are done conscientiously (in spite of the relatively small payment). To sum up, we collect redundant data: each original Urdu sentence is translated four times.

4 PageRank-HITS Hybrid Model

Since we aim at soliciting the most reliable Turkers to participate the crowdsoucing translation process, we ought to score their authority and then stop hiring the unreliable Turkers afterwards. There is a reinforcement relationship between workers and their works, i.e., translated texts. Intuitively, a mutual reinforcement is developed to model the following assumptions behind the workers and translations:

> *A translation is reliable if it associates to other reliable translations, or it is translated by the Turkers with high authority. Analogously, a Turker will be believed to have authority if they write translations with high reliability.*

We group the four versions of translations as a *document cluster* for a particular Urdu sentence. It is quite straightforward to compute the reliability for each translation against other translations within the same document cluster using a PageRank frame, as shown in each slot in Figure 2. However, it is not straightforward to propagate the quality of each translation from one document cluster to other document clusters, since the texts are not directly comparable. As shown in Figure 2, the score cannot be propagated across slots *directly*. Hence, we incorporate a HITS (Hyperlink-Induced Topic Search) frame. Here, each document cluster is regarded as a meta-node. In HITS algorithm, the hub scores and authority scores are computed in a reinforcement way, just as the document clusters and Turkers. We consider the document clusters, i.e., meta-nodes, as hubs and Turkers as authorities, which is a bipartite graph representation. In this way, the scores within each document cluster can be propagated to Turkers, and the Turker authority can also be propagated to document clusters and the corresponding translations. To sum up, we propose an iterative reinforcement framework based on a PageRank-HITS Hybrid model. The model is described in Algorithm 1, and then we introduce the PageRank frame and HITS frame one by one.

Algorithm 1: PageRank-HITS Hybrid Model

```
Input: Translations, Turkers, and graphs $G_t$ and $G$.
Output: Ranking list of Turkers by authority.
begin
    \\ Global iteration in PageRank-HITS
    repeat
        for each translation $\in$ DocCluster do
            \\ Local iteration in PageRank
            Update PageRank Score
        Update HITS link structure using PageRank
        scores assigned to translations
        for each Turker and DocCluster (meta-node) do
            \\ Local iteration in HITS
            Update authority and hub Score
        Update transition matrix with prior for PageRank
    until Convergence;
```

4.1 PageRank Frame

We deploy the PageRank frame to score every 4 translations for the same Urdu sentence. Within each document cluster, let G_t denote a weighted graph without directions to represent the relationships among translations. G_t=(V_t, E_t) where $V_t = \{t_i | t_i \in V_t\}$ denotes a collection of texts t translated by Turkers. E_t is the set of linkage representing the adjacency between the translations (adjacency established in Section 4.3). Based on the adjacency links, we can establish the transition matrix M, and score the authorities for the translations using the general PageRank paradigm [Page *et al.*, 1999]. Fix some damping factor μ (usually μ=0.15) and say that at each time step with probability (1-μ) we stick to random walking and with probability μ we do not make a usual random walk step, but instead jump to any vertex, chosen uniformly at random. A random walk on a graph is a Markov chain:

$$\mathbf{t} = (1-\mu)\mathrm{M}^{\mathrm{T}}\mathbf{t} + \frac{\mu}{|V_t|}\mathbf{1}\mathbf{1}^{\mathrm{T}} \qquad (1)$$

Here, vector $\mathbf{t}$ contains the ranking scores for the vertices in G_t. The fact that there exists a unique solution to (1) fol-

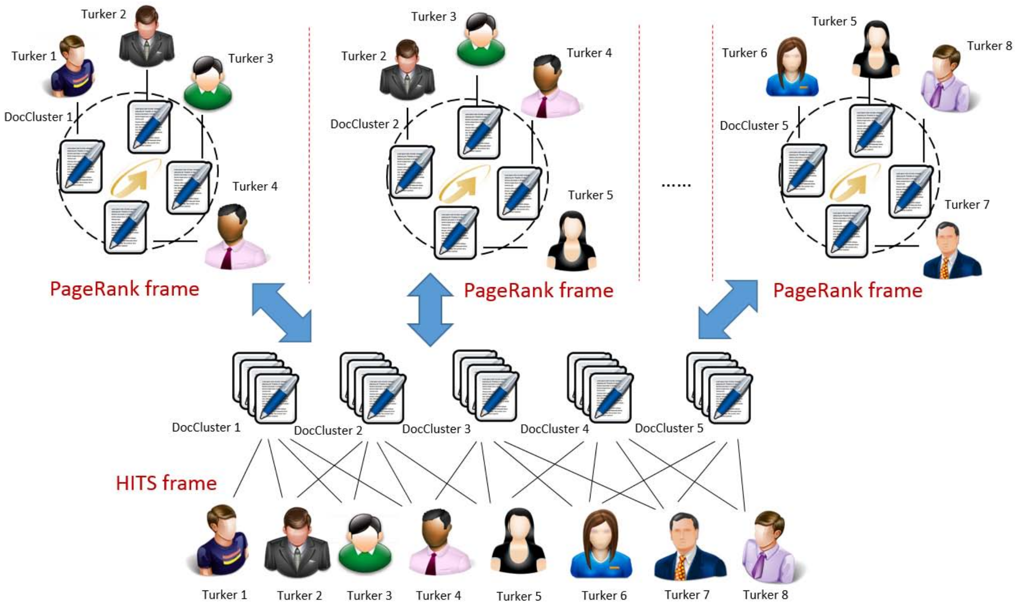

Figure 2: PageRank-HITS Hybrid: the top part shows the PageRank frame and the bottom part indicates the HITS frame.

lows from the random walk $\mathbf{M}$ being ergodic (μ >0 guarantees irreducibility, because we can jump to any vertex). $\mathbf{M}^{\mathrm{T}}$ is the transpose of $\mathbf{M}$. $\mathbf{1}$ is the vector of $|V_t|$ entries, each being equal to one. Let t∈ $\mathbf{R}$, $||\mathbf{t}||_1 = 1$ be the only solution.

The ranking chain shown above captures the following intuitions behind. A translation is important if it is “voted” by many of the other generated translations. There is one more intuition to capture as indicated before: if a translation is written by a Turker with high authority, it is still likely to be more reliable and vice versa. Hence, every time before we move to a new local PageRank iteration within the document cluster (see Algorithm 1), we incorporate priors into random walks. The standard PageRank starts from any node *equally*, and randomly selects a link from that node to follow considering the transition matrix, or jumps to a random node with *equal* probability. In contrast, since the Turkers can be judged to be of different authority after the calculation in the HITS frame and the authority will affect the “initial score” in every local PageRank iteration because translations are NOT equal.

We incorporate prior information for translations into random walk: it is natural to assume that translation by high authority Turker would be scored higher. Given the normalized Turker authority score vector $\mathbf{a}$, let Diag($\mathbf{a}$) denote a diagonal matrix whose eigenvalue is the vector $\mathbf{a}$. Then $\mathbf{t}$ becomes:

$$\mathbf{t} = (1-\mu)[\mathrm{Diag}(\mathbf{a})\mathrm{M}]^{\mathrm{T}}\mathbf{t} + \mu\mathbf{a} \tag{2}$$

In the beginning, we set Turker authority as equal scores. When we calculate Turker authority in the HITS framework, $\mathbf{a}$ is changed during each global iteration until convergence.

4.2 HITS Frame

After the PageRank frame, we obtain the normalized ranking score for the candidate translations. We attribute the scores to their corresponding workers. Formally, we represent the bipartite graph as $G = (V, E)$ where $V = \{V_a \cup V_c\}$, where V_a for Turkers and V_c for DocClusters (meta-nodes). Let W denote the adjacency matrix for the HITS frame. Not all links are of the same importance in determining authoritative and hub scores. In this case, the adjacency matrix is a weighted matrix. As long as the matrix W stays non-negative, the convergence property of HITS is preserved [Kleinberg, 1999]. The different weight of the linkage is decided by the ranking score from the PageRank frame. Note that during each local HITS iteration process (see Algorithm 1), the weight of linkage actually varies and hence the weighted adjacency matrix is *dynamic*: in other words, the linkage structure changes between two local HITS iterations until convergence.

The mutual reinforcing relationship of authorities and hub scores can be expressed in matrix representation as follows:

$$\begin{aligned}\mathbf{c}^{(i+1,k)} &= [W^{(0,k)}]\mathbf{a}^{(i,k)}\\ \mathbf{c}^{(i+1,k)} &= \mathbf{c}^{(i+1,k)}/||\mathbf{c}^{(i+1,k)}||_1\end{aligned} \tag{3}$$

and

$$\begin{aligned}\mathbf{a}^{(i+1,k)} &= [W^{(0,k)}]^{\mathrm{T}}\mathbf{c}^{(i,k)}\\ \mathbf{a}^{(i+1,k)} &= \mathbf{a}^{(i+1,k)}/||\mathbf{a}^{(i+1,k)}||_1\end{aligned} \tag{4}$$

For the *superscripts*, the first one indicates local HITS iteration and the second one indicates global iteration. Note that within each local iteration, the adjacency matrix W is stable. When we finish a local HITS iteration and then go to the local PageRank iteration, the link structure in the HITS frame could change after the PageRank iteration: the hub score is mostly represented by the good translations and hence the good Turker who gives the good translation. In this way, the adjacency matrix updates from $W^{(i,k)}$ to $W^{(i+1,k+1)}$.

In order to guarantee the convergence of the iterative form, we must force the transition matrix to be stochastic and irre-

Table 1: Performance comparisons. "Decr." denotes performance decrease compared with the full Turker set.

RatioKept	Metrics	**Random**	**WLoad**	**Regress.**	**AvgPR**	**HITS**	**PRHITS**
$\sim$@10%	BLEU	30.89	30.37	31.63	32.39	32.83	34.12
	Decr.	18.70%	19.99%	16.68%	14.67%	13.51%	10.11%
$\sim$@20%	BLEU	29.39	31.69	31.79	34.11	31.85	34.10
	Decr.	22.58%	16.51%	16.25%	10.14%	16.10%	10.29%
$\sim$@30%	BLEU	28.38	32.13	32.34	34.27	33.96	34.93
	Decr.	25.24%	15.35%	14.81%	9.72%	10.58%	9.40%
$\sim$@40%	BLEU	30.25	34.52	33.46	34.73	34.15	35.31
	Decr.	20.31%	9.06%	11.85%	8.52%	10.04%	6.98%
$\sim$@50%	BLEU	31.29	32.73	33.70	33.86	34.23	35.88
	Decr.	17.57%	13.78%	11.22%	10.80%	9.82%	5.48%
$\sim$@60%	BLEU	30.96	34.85	34.21	33.95	34.50	36.15
	Decr.	18.44%	8.19%	9.88%	10.56%	9.11%	4.77%
$\sim$@70%	BLEU	32.17	33.29	34.58	30.21	35.16	36.73
	Decr.	15.25%	12.30%	8.90%	7.24%	7.38%	3.24%
$\sim$@80%	BLEU	31.85	34.38	35.67	35.82	36.25	37.83
	Decr.	16.10%	9.43%	6.03%	5.64%	4.50%	0.34%
$\sim$@90%	BLEU	33.94	34.17	36.18	36.74	36.39	37.89
	Decr.	10.59%	9.99%	4.69%	3.21%	4.13%	0.18%
	Pearson	0.016	0.375	0.415	0.510	0.773	0.944
$\sim$@100%	BLEU	37.96	37.96	37.96	37.96	37.96	37.96
	Decr.	–	–	–	–	–	–

ducible. To this end, we must make the **t**, **c** and **h** column stochastic to force transition matrix stochastic [Langville and Meyer, 2004]. **t** and **c** and **h** are therefore normalized after each iteration in Equation (3) and (4). Usually the convergence of the iteration algorithm is achieved when the difference between the scores computed at two successive iterations for any translation or Turker falls below a threshold ϵ (0.001 in this study).

4.3 Adjacency Matrices

We introduce the adjacency matrix calculation, including the matrix M used in the PageRank frame and matrix W in the HITS frame. For M, the translation collection within each document cluster can be modeled as an undirected graph with weighted linkage. Each link between two translation t_i and t_j by the cosine similarity metric $\phi(.)$ [Manning *et al.*, 2008]:

$$\phi(t_i, t_j) = \frac{t_i \cdot t_j}{||t_i||||t_j||} \qquad (5)$$

t_i and t_j are the corresponding term vectors for the translations, and the weights associated with the terms are calculated with tf-isf formula [Neto *et al.*, 2000; 2002], where *tf* is the term frequency and *isf* is the inverse sentence frequency. The matrix is normalized to make the sum of each row equal to 1:

$$M_{i,j} = \frac{\phi(t_i, t_j)}{\sum_{t'} \phi(t_i, t')} \qquad (6)$$

For the HITS frame, we need to differentiate the link weights connecting a particular Turker to the same metanode. The hub score for each DocCluster is more represented by the high quality translation and hence the Turker who translates it. Given the normalized ranking score for all translations obtained from the PageRank frame, denoted as **t**, for each $t_{a_i} \in c_j$ where c_j is a DocCluster, we establish the relationship between the worker and the translation, using the score $\mathbf{t}_{a_i}$:

$$W_{i,j} = \frac{\mathbf{t}_{a_i}}{||\mathbf{t}||_1} \qquad (a_i \text{ translate } t_{a_i} \in c_j) \qquad (7)$$

The translations with higher scores are more likely to represent the document cluster (meta-node). Till now, we have modeled the intuitions mentioned in Section 4, addressed in the PageRank frame and the HITS frame in a hybrid.

5 Experiments and Evaluations

5.1 Data

We translated the Urdu side of the Urdu-English test set of the 2009 NIST MT Evaluation Workshop, used in [Zaidan and Callison-Burch, 2011; Yan *et al.*, 2014]. 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 of Turkers compared against professional translators, which gives us an idea whether to keep a worker or not. 51 different Turkers took part in the translation task, each translating 138 sentences on average. The translation data was collected from Jan. 1, 2009 to Jan. 31, 2009. We use nearly half data (from Jan. 1 to Jan. 16) for observation (training), and test on the other half of the data (from Jan.17 to Jan. 31).

5.2 Evaluation Metric

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 Bilingual Evaluation Understudy (BLEU) score [Papineni *et al.*, 2002]. We can examine how the performance changes when we start to reduce the lower ranked labor force in crowdsourcing. For fairness, we apply the same state-of-art linear soliciting strategies in [Zaidan and Callison-Burch, 2011; Yan *et al.*, 2014]. We will evaluate different Turker ranking algorithms and see their performance comparison against PageRank-HITS Hybrid model.

We also calculate the ground truth ranking based on the professional references, and then measure the correlation between the automatically generated Turker rankings and the ground truth ranking using Pearson correlation coefficient r [Pearson, 1895]. The ground truth Turker ranking is generated as follows: given all translations by a particular Turker, we calculate the Translation Error Rate (TER) between each translation against all references and average all TER scores for this Turker [Snover *et al.*, 2006]. We rank Turkers based on the average TER score for all their translations. Intuitively, the less average TER score means better translation quality, and the corresponding worker should be ranked higher.

5.3 Comparison Methods

We establish the ground truth Turker ranking by average TER as well as other ranking strategies. We carry out a set of experiments that demonstrate our model can reduce the cost and avoid the risk of severe performance loss: we can bring new opportunities with the saved money.

We first include an intuitive method of random selection (**Random**), picking Turkers out of all participants at random, which could be estimated as a lower bound. A second baseline ranks Turkers according to their working load (**WLoad**). The assumption is that hard-working labor force are likely to be people who treat the work more seriously. It is also intuitive to rank Turkers using simple linear regression (**Regress.**) methods [Zaidan and Callison-Burch, 2011]. We also make comparisons with straightforward PageRank method (**AvgPR**) and HITS method (**HITS**). In PageRank, we rank translations within the document cluster, normalize the scores, and average all scores for a specific Turker. In HITS, we do not distinguish the weights of link structure and assume the links to each document cluster are equal. Actually, both PageRank and HITS are components of our proposed PageRank-HITS Hybrid model (**PRHITS**). We test the BLEU scores after removing the lower ranked Turkers and their translations and also show the correlation coefficient with the ground truth ranking.

5.4 Results

Our results are summarized in Tables 1, which reports how the BLEU performance changes along with the reduction of hired Turkers in different percentages (from 10% to 90%). The performance is evaluated compared with the full set of Turkers against professional translations. Also, we report the Pearson correlation between different ranked lists against the ground truth Turker rankings.

From Table 1, we could see that the random selection has the worst performance as expected, without taking the quality of translations or Turkers into account at all. The random ranking captures no intuition. Yet, the assumption that people who work more will work better is proved to be not quite helpful. The ranking by work load has less positive correlation with the ground truth ranking by average TER. To remove the lower ranked Turkers in the ranking list given by random and work load brings significant risks to hurt the performance. The linear regression, PageRank and HITS method generally provide much better ranking list compared with the ground truth ranking. It is understandable that all 3 methods have addressed translation quality or Turker quality. Neither of these 3 algorithms fully explore the mutual reinforcement via the link structures among the translations and workers. Our proposed PageRank-HITS Hybrid model formulates the relationship under an iterative reinforced framework. It is natural to see our PRHITS model generates the most correlative ranking list to the ground truth and minimize the performance loss when reduce the unauthorized labor force.

For a full comparison, we need include correlations with ground truth ranking. Due to space limits, we only visualize the results of PageRank, HITS and PRHITS because PageRank and HITS also rank Turkers using structure information and they are literally components of PRHITS model. The visualizations are shown in Figure 3∼5. If we aim at reducing 50% of Turkers, people in Zone 1 and Zone 2 are false positives and false negatives: we might remove authorized Turkers or keep unreliable Turkers by PageRank and HITS, while in PRHITS, we generally keep the right personnel!

5.5 Risks or Opportunities?

The most prominent advantage of ranking Turkers by authority would be the money to save when we decide not to hire unreliable workers. In the translation task, we paid a reward of $0.10 to translate a sentence. Therefore, we had the total translation costs at $716.80. If we do not collect redundant translations, the cost would be $179.20, 25% of the original cost. Hence there is large room to spare the money. With appropriate quality control in rankings, we do not bother to collect redundant translations from bad Turkers. We add some comparison cases illustrated in Table 2: 1) we demonstrate the BLEU performance when we aim at saving 50% or 75% money; 2) we show how much money can be saved when we aim at maintaining at least 95% or 98% BLEU score. It is a trade-off between cost and quality and we can see our proposed PRHITS model balances best.

6 Conclusion

We have proposed a PageRank-HITS Hybrid Model to rank the authority of workers on the MTurk platform. The model is established on a heterogeneous graph between Turkers and translation texts. We demonstrate that we can reduce costs by stop hiring lower ranked Turkers, while avoid the risk of performance loss.

We believe that crowdsourcing can play a pivotal role in future efforts for Natural Language Processing. As crowdsourcing deems to be cheap, it seems to be unnecessary to reduce

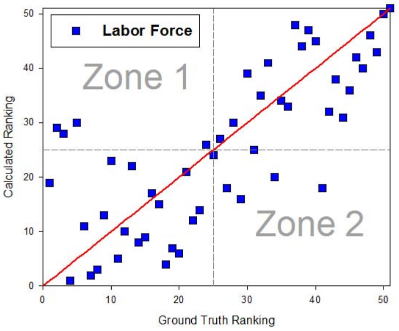

Figure 3: PageRank correlation.

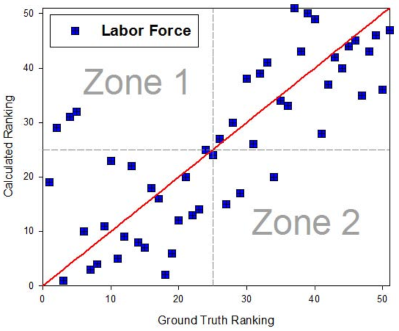

Figure 4: HITS correlation.

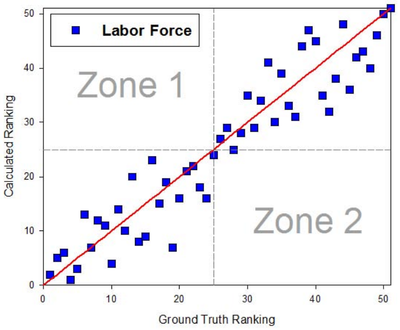

Figure 5: PRHITS correlation.

Table 2: 1) The table on the left hand-side indicates the BLEU scores when we spent only 50% (or 75%) of money of the original costs; 2) the table on the right hand-side denotes the saved money against all expenses (measured in percentage) when we kept 95% (or 98%) of BLEU scores.

COSTS	50%	75%
Random	29.96	31.19
WLoad	32.59	33.17
Regress.	33.86	34.76
AvgPR	33.42	35.29
HITS	34.50	35.16
PRHITS	35.83	37.05

BLEU	95%	98%
Random	0.00%	0.00%
WLoad	0.00%	0.00%
Regress.	13.23%	0.00%
AvgPR	11.98%	0.00%
HITS	18.39%	0.00%
PRHITS	45.25%	17.36%

costs for crowdsourcing tasks. Actually cost saving is always a big concern when tasks are launched in large volumes: many a little saving makes a mickle savings. We also find that different Turkers have different turnaround time: some Turkers submit results very quickly while some have huge lags. In the future, we will formulate turnaround time into cost saving measurement since time is "money" as well.

Acknowledgments

This paper is partially supported by the National Natural Science Foundation of China (NSFC Grant Number 61472006), the Doctoral Program of Higher Education of China (Grant No. 20130001110032) and the National Basic Research Program (973 Program No. 2014CB340405).

References

[Al-Onaizan *et al.*, 2002] Yaser Al-Onaizan, Ulrich Germann, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Daniel Marcu, and Kenji Yamada. Translation with scarce bilingual resources. *Machine translation*, 17(1):1–17, 2002.

[Ambati *et al.*, 2010] Vamshi Ambati, Stephan Vogel, and Jaime G Carbonell. Active learning and crowd-sourcing for machine translation. In *LREC*, volume 1, page 2, 2010.

[Bernstein *et al.*, 2010] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. Soylent: A word processor with a crowd inside. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 313–322, 2010.

[Bloodgood and Callison-Burch, 2010] Michael Bloodgood and Chris Callison-Burch. Using mechanical turk to build machine translation evaluation sets. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 208–211, 2010.

[Callison-Burch and Dredze, 2010] Chris Callison-Burch and Mark Dredze. 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*, CSLDAMT '10, pages 1–12, 2010.

[Callison-Burch, 2009] Chris Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using amazon's mechanical turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 286–295, 2009.

[Dawid and Skene, 1979] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[Germann, 2001] Ulrich Germann. Building a statistical machine translation system from scratch: How much bang for the buck can we expect? In *Proceedings of the Workshop on Data-driven Methods in Machine Translation - Volume 14*, DMMT '01, pages 1–8, 2001.

[Kleinberg, 1999] Jon M. Kleinberg. Hubs, authorities, and communities. *ACM Comput. Surv.*, 31(4es), December 1999.

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

[Manning *et al.*, 2008] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. 2008.

[Munteanu and Marcu, 2005] Dragos Stefan Munteanu and Daniel Marcu. Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.*, 31(4):477–504, December 2005.

[Neto *et al.*, 2000] Joel Larocca Neto, Alexandre D Santos, Celso AA Kaestner, Neto Alexandre, D Santos, et al. Document clustering and text summarization. 2000.

[Neto *et al.*, 2002] Joel Larocca Neto, Alex A Freitas, and Celso AA Kaestner. Automatic text summarization using a machine learning approach. In *Advances in Artificial Intelligence*, pages 205–215. Springer, 2002.

[Nießen and Ney, 2004] Sonja Nießen and Hermann Ney. Statistical machine translation with scarce resources using morpho-syntactic information. *Computational linguistics*, 30(2):181–204, 2004.

[Page *et al.*, 1999] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

[Pearson, 1895] Karl Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242, 1895.

[Resnik and Smith, 2003] Philip Resnik and Noah A. Smith. The web as a parallel corpus. *Comput. Linguist.*, 29(3):349–380, September 2003.

[Smith *et al.*, 2010] Jason R. Smith, Chris Quirk, and Kristina Toutanova. 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*, HLT '10, pages 403–411, 2010.

[Snover *et al.*, 2006] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231, 2006.

[Snow *et al.*, 2008] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 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 '08, pages 254–263, 2008.

[Uszkoreit *et al.*, 2010] Jakob Uszkoreit, Jay M. Ponte, Ashok C. Popat, and Moshe Dubiner. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1101–1109, 2010.

[Whitehill *et al.*, 2009] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

[Yan *et al.*, 2014] Rui Yan, Mingkun Gao, Ellie Pavlick, and Chris Callison-Burch. 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*, pages 1134–1144, 2014.

[Zaidan and Callison-Burch, 2011] Omar F. Zaidan and Chris Callison-Burch. 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, 2011.