Two Heads are Better than One: Crowdsourcing Translation via a Two-Step Non-Professional Collaboration towards Professionals

Rui Yan
Dept. of Computer and Information Science
University of Pennsylvania
Philadelphia, USA
ruiyan@seas.upenn.edu

Ellie Pavlick
Dept. of Computer and Information Science
University of Pennsylvania
Philadelphia, USA
epavlick@seas.upenn.edu

Chris Callison-Burch
Dept. of Computer and Information Science
University of Pennsylvania
Philadelphia, USA
ccb@cis.upenn.edu

Abstract

Crowdsourcing based techniques have emerged to been the rising star for Machine Translation, with prominent advantages of low cost in money and turnaround time to collect the processed data. However, when compared with translation by trained professionals, naive collected results from non-professional translators yields low-quality outputs, which is insufficient. To this end, supervised quality control is a necessity when crowdsourcing the task for Natural Language Processing. In this paper, we propose a two-step collaboration schema, i.e., translation and post-editing, by non-professionals via graph-based ranking model and we demonstrate the schema will increase the translation quality to closer professionals. Specifically, we solicit multiple versions of translations and edits, and automatically select the best output. With appropriate quality control, we are able to distinguish acceptable translation from bad ones. We recreate the NIST 2009 Urdu-to-English evaluation set with Mechanical Turk, and quantitatively show that our models are able to achieve better quality in terms of BLEU score than previous methods with even less costs.

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 to study, which are driven by a myriad of different applications. In recent Natural Language Processing research, automatical 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., the 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 gets stuck in a severe bottleneck when facing with many relatively minority languages with significantly insufficient annotated data: not enough bilingual parallel corpora exist.

In this sense, to collect more parallel corpora for minor languages becomes a necessity for the success of SMT. There are various options for creating 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 inappropriate aligned, which could 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 professional trained annotators and moreover, hiring these professionals woudl seem to be prohibitively expensive. Germann (2001) estimated the cost of hiring professional translators to create a Tamil-English corpus at $0.36/word. 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 higher quality translations at a lower cost. We notice that Amazons Mechanical Turk (AMT) provides a la-

bor force platform at a cheap price for labor units, and hence we are able to hire a large group of translators, non-professional or even professional, at a similarly tempting low price. In the way of crowdsoucing, we 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. We have proposed a random walk of mutual reinforced ranking model based on a heterogenous graph consisting of translations and the workers on AMT, named MTurkers. In particularly, we design a schema of collaborative crowdsourcing: translation followed by post-editing. We also include linguistic quality scoring as the ranking prior during the selection process. After these two steps, the collaboration of two non-professional generate a translated sentence. The model discriminates acceptable translations from those that are not (and analogously competent Turkers from those who are not).

According to an objective and quantitative comparison with the professionally-produced reference translations as a weights set, we examine the idea of enabling the 2-step of "*collaborations*" between both roles of non-professional translators and post-editors, and using structural information from the collaboration relationships. We show that it is possible to get near professional high quality translations in aggregate by soliciting multiple translations to select the best of the bunch (and the cost can be even lower to identifying competent MTurkers). The overall performance is improved by the 2-step collaboration and the information hidden in the net structure: in other words, **TWO** heads are indeed better than **ONE**.

We start by introducing the crowdsourcing platform and data collection. In Section 3 we formulate a reinforced ranking model combined with random walk, and then describe experimental results in Section 4. We review related work in Section 5 and draw conclusions in Section 6.

2 Crowdsourcing Platform and Data Collection

To collect crowdsourced translations, we deploy our platform based on Amazons Mechanical Turk, an online market-place designed to pay people small sums of money to complete *Human Intelligence Tasks* (or HIT) tasks that are difficult for computers but easy for people. Example HIT 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 HIT or work on HIT that were submitted by others. Workers are referred to as "Turkers", and designers of HIT 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 HIT 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 hands, the Turkers are completely anonymous without any personal profile other than a Turker ID (eg., A143AWKU99STC9). Hence it is difficult to determine if a non-professional is qualified to fulfill the task before the Turker submits HIT.

In this sense, soliciting translations from anonymous non-professionals carries a significant risk of poor translation quality, but it is not very difficult to find bad translations provided by Turkers using simple linear regression methods (Zaidan and Callison-Burch, 2011). To improve the accuracy of noisy translations from non-experts, a natural quality control solution would be employing a graph-based ranking model to quantitatively measure multiple outputs based on "votes" or "recommendations" between each other, namely the wisdom of crowds.

3 Crowdsourcing Translation

Our HIT involves showing the worker a sequence of sentences in source language (i.e., *Urdu* in this work), 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 three translations per Urdu sentence (from three distinct translators). we instead split the data set into groups of 10 sentences per HIT. We keep some of the strategies used in other crowdsourcing systems in designing interfaces. 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.

by Urdu translators:
According to the territory's people the pamphlets from the Taaliban had been read in the announcements in all the mosques of the Northern Wazeerastan.
by English post-editors:
According to locals, the pamphlet released by the Taliban was read out on the loudspeakers of all the mosques in North Waziristan.
by LDC professionals:
According to the local people, the Taliban's pamphlet was read over the loudspeakers of all mosques in North Waziristan.

Table 1: Different versions of translations.

There might be a potential concern that the competent translators are native Urdu speakers, who can understand the source sentences well but might not be able to express the original meanings in English as natural as in Urdu. It is feasible to include native English speakers to post-edit the translated English sentences into more adequate, standard and error-free ones, i.e., professional English. We aim to show that the collaboration design of two heads, non-professional Urdu speakers and non-professional English speakers, would yield better outputs than either one working in isolation, and can better approximate the results from professional trained translators. To this end, in addition to collecting multi-version of translations per source sentence, we also post another MTurk task where we asked US-based Turkers (who are likely native English speakers) to edit the translations into more fluent and grammatical sentences as the post-edited versions of the original translations. Table 1 gives an example of the whole process of translation: the unedited translations that we collected in the translation pass typically contain many simple mistakes like mis-spellings, typos, and awkward word choices. The translations often reflect non-native English, but are generally done conscientiously (in spite of the relatively small payment).

We collect redundant annotations in both tasks. Each original 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 3×3=9 post-edited translations for each source sentence in Urdu.

3.1 Problem Formulation

The problem definition of the crowdsourcing translation task is quite obvious: given the source sentences to translate and the Turkers (i.e., translators and post-editors) on AMT, we choose the best translated and post-edited HIT as output.

Our method operates over a heterogeneous network that includes translators, post-editors and translated sentences. We frame both components of HIT and Turkers into graphs, using relationships (i.e., semantic similarity, Turker collaboration and authorship correspondingly) to connect these parts as a co-ranking paradigm (Yan et al., 2012a; Yan et al., 2012b). Let G denote the heterogeneous graph with nodes V and edges E, and G = (V,E) = ($V_T, V_H, E_T, E_H, E_{TH}$). G is divided into three subgraphs, G_T, G_H, and G_{TH}. G_H = (V_H, E_H) is a weighted undirected graph representing the HIT and their relationships. Let V_H = $\{h_i|h_i \in V_H\}$ denote a collection of $|V_H|$ translated and edited sentences, and E_H the set of linkage representing affinity between them, established by textual similarity between the translated sentences (see Section 3.4 for details). 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 *translation* and *post-editing* collaboration. Each collaboration produces an output HIT. $G_{TH} = (V_{TH}, E_{TH})$ is an unweighted bipartite graph that ties G_T and G_H together and represents "authorship". The graph G consists of nodes V_{TH} = $V_T \cup V_H$ and edges E_{TH} connecting each HIT with its generators. Typically, an HIT is generated by the collaboration of a translator and a post-editor. The three sub-networks are illustrated in Figure 1.

3.2 Inter-Graph Ranking

The framework includes three random walks, one on G_T, one on G_H and the other one on G_{TH}. 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. The mutual reinforcement framework couples the two random walks on G_T and G_H that rank HIT and Turkers in isolation. The ranking method allows to obtain a more global ranking by taking into account the intra-/inter-component dependencies. In the following sections we first describe how we obtain the rankings on G_T and G_H, and then move on to discuss how the two are coupled.

The ranking chain shown in Figure 1 captures the following intuitions behind. An HIT is important if 1) it is "voted" by many of the other generated HIT; 2) it is authored by better qualified translators and/or post-editors. Analogously,

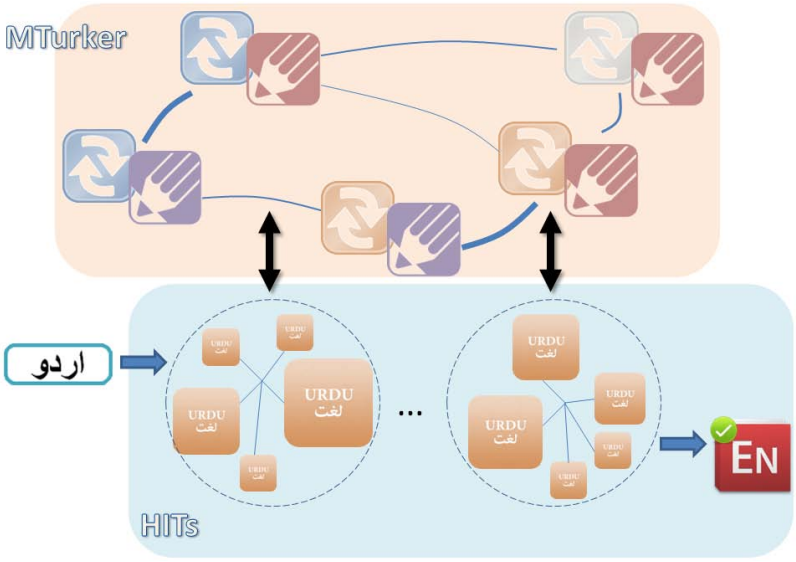

Figure 1: 2-step collaborative crowdsourcing translation model based on graph ranking framework including three sub-networks. The undirected links between users denotes *translation-editing* collaboration. The undirected links between HIT indicate semantic affinity. A bipartite graph (whose edges are shown with the larger arrows) ties HIT and Turker networks together by authorship. A dashed circle indicates the group of candidates for a source sentence with different weighting scores.

a Turker pair is believed to be better qualified if 1) the editor is collaborating with a good translator and vice versa; 2) the working pair has authored highly voted HIT. This ranking schema is actually a reinforced process across the heterogeneous graphs. We use two vectors $\mathbf{h} = [\pi(h)]_{1\times|h|}$ and $\mathbf{t} = [\pi(t)]_{1\times|t|}$ to denote the saliency scores $\pi(.)$ of HIT and Turkers. The mentioned two intuitions could be formulated as follows:

• **Homogeneity.** We use an adjacency matrix $[M]_{|\mathbf{h}|\times|\mathbf{h}|}$ to describe the homogeneous affinity between HIT, and use $[N]_{|\mathbf{t}|\times|\mathbf{t}|}$ between Turkers.

$$\mathbf{h} \propto M^T\mathbf{h}, \qquad \mathbf{t} \propto N^T\mathbf{t} \tag{1}$$

where $h = |V_H|$ is the number of vertices in the HIT graph and $t = |V_T|$ is the number of vertices in the Turker graph. The adjacency matrix $[M]$ denotes the transition probabilities among HIT, and analogously matrix $[N]$ to describe the affinity between Turker collaboration pairs.

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

$$\mathbf{h} \propto \hat{W}^T\mathbf{t}, \qquad \mathbf{t} \propto \bar{W}^T\mathbf{h} \tag{2}$$

All affinity matrices will be defined in the next section. By fusing the above equations, we can have the following iterative calculation in matrix forms. For numerical computation of the saliency scores, the initial scores of all sentences and images are set to 1 and the following two steps are alternated until convergence to select the best HIT. The ranking is formulated as:

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

$$\begin{aligned}\mathbf{h}^{(n)} &= (1-\lambda)M^T\mathbf{h}^{(n-1)} + \lambda\hat{W}^T\mathbf{t}^{(n-1)} \\ \mathbf{h}^{(n)} &= \mathbf{h}^{(n)}/||\mathbf{h}^{(n)}||_1\end{aligned} \tag{3}$$

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

$$\begin{aligned}\mathbf{t}^{(n)} &= (1-\lambda)N^T\mathbf{t}^{(n-1)} + \lambda\bar{W}^T\mathbf{h}^{(n-1)} \\ \mathbf{t}^{(n)} &= \mathbf{t}^{(n)}/||\mathbf{t}^{(n)}||_1\end{aligned} \tag{4}$$

where λ specify the relative contributions to the saliency scores trade-off from 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 $\mathbf{h}$ and $\mathbf{t}$ *column stochastic* to force transition matrix stochastic (Langville and Meyer, 2004). $\mathbf{h}$ and $\mathbf{t}$ are therefore normalized after each iteration in Equation (3) and (4).

3.3 Intra-Graph Ranking

Prior. Before we move to the homogeneous intra-graph random walks, we propose to incorporate

priors into standard random walks. The standard PageRank algorithm starts from any node, then randomly selects a link from that node to follow considering the weighted transition matrix, or jumps to a random node with equal probability. As the generated HIT could be pre-judged to be of different linguistics qualities (Louis and Nenkova, 2013; Zaidan and Callison-Burch, 2011), we can incorporate textual quality as the transitional prior and it is natural to assume an HIT with better quality will be more likely to be chosen.

As linguistics quality is not the focus of this work, we apply the general textual quality judgement described in (Zaidan and Callison-Burch, 2011). We use the following criteria as indicators of the quality of the generations:

Sentence-Level. The first set of features attempt to discriminate good English sentences from bad ones before the graph ranking process.

• Language model features: each sentence is assigned with 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.

Turker-Level. We add Turker-level features that evaluate a translation based on the generators who provided the translation and post-editing.

• Aggregate features: for each sentence-level feature above, we have a corresponding feature computed over all of that workers translations. The score for the working pair is calculated as the 1) average, 2) maximum and 3) minimum score of both Turkers.

• Activity features: we also investigate the total number of productions and the ratio of the high quality productions from the working pair.

We could also use the features such as location information or language ability information. Since currently these information is not well certified, we temporarily leave it out of the Turker-Level feature now.

After calculation of all features, we sort out an overall score to measure the linguistics and Turker quality based on regression when apply the linear combination measurement in (Zaidan and Callison-Burch, 2011), the quality scores are denoted as $\mathbf{h}_0$ for HIT and $\mathbf{t}_0$ for Turker pairs.

In a simple random walk, it is assumed that all nodes in the transitional matrix are equi-probable before the walk starts. In contrast, we use the linguistics/Turker quality as a prior on the affinity M and N. Let Diag(.) denote a diagonal matrix whose eigenvalue is the pre-calculated score of $\mathbf{h}_0$ and $\mathbf{t}_0$. Then $\mathbf{h}$ and $\mathbf{t}$ becomes:

$$\mathbf{h} = (1-\mu)[\text{Diag}(\mathbf{h}_0)M]^T\mathbf{h} + \mu\mathbf{h}_0 \qquad (5)$$

and

$$\mathbf{t} = (1-\mu)[\text{Diag}(\mathbf{t}_0)N]^T\mathbf{t} + \mu\mathbf{t}_0 \qquad (6)$$

The modified formula indicates HIT and Turkers with higher quality will be more likely to be transited to within both the new random starts and the random walking part.

3.4 Affinity Matrix Establishment

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

The HIT collection can be modeled as a weighted undirected graph. Nodes in the graph represent sentences, edges represent intersentential relatedness, and their weights are computed via cosine similarity. The adjacency matrix M describes such a graph with each entry corresponding to the weight of an edge.

$$\mathcal{F}(h_i, h_j) = \frac{h_i \cdot h_j}{||h_i||||h_j||}, \qquad M_{ij} = \frac{\mathcal{F}(h_i, h_j)}{\sum_k \mathcal{F}(h_i, h_k)} \qquad (7)$$

where $\mathcal{F}(.)$ is the cosine similarity and h is a term vector corresponding to an HIT. We treat an HIT 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 is a undirected graph based on the collaboration linkage. When t_i and t_j have a shared "collaboration", we add an edge between t_i and t_j. There are 3 different schema to define collaborations, e.g., pairs with the same translator and/or post-editor, which will be discussed in details in the experiment section. Let the function $\mathcal{I}(t_i, t_j)$ denote the times of "collaborations" (#c)

when there is an edge between t_i and t_j. The adjacency matrix N is then defined as:

$$\mathcal{I}(t_i, t_j) = \begin{cases} \#c & (e_{ij} \in E_T) \\ 0 & \text{otherwise} \end{cases}, N_{ij} = \frac{\mathcal{I}(t_i, t_j)}{\sum_k \mathcal{I}(t_i, t_k)} \tag{8}$$

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

$$\mathcal{A}(h_i, t_j) = \begin{cases} 1 & (e_{ij} \in E_{TH}) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

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

$$\bar{W}_{ij} = \frac{\mathcal{A}(h_i, t_j)}{\sum_k \mathcal{A}(h_i, t_k)}, \hat{W}_{ij} = \frac{\mathcal{A}(h_i, t_j)}{\sum_k \mathcal{A}(h_k, t_j)} \tag{10}$$

4 Experiments and Evaluation

4.1 Data

We translated the Urdu side of the UrduEnglish test set of the 2009 NIST MT Evaluation Workshop, used in (Zaidan and Callison-Burch, 2011). 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 the crowdsourced translations could better approach to the quality of a professional translator.

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.

4.2 Evaluation

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) 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. We can examine the results from different translation methods compares to this range by calculating the BLEU scores against the same four sets of three reference translations. We will evaluate different strategies for selecting different translation sets, and see how much each improves on the BLEU score, compared to randomly picking from among the Turker translations.

We also evaluate translation quality by using reference sets to score various submissions to the NIST MT evaluation. Specifically, we measure the correlation (using Pearson 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.

4.3 Comparison Methods

We first include an intuitive method of random selection, picking HIT out of all generations at random, which could be estimated as a lower bound. As mentioned, we evaluate the reference sets against each other, in order to quantify the concept of "professional quality". We establish the performance of professional translators, calculate the upper bounds for Turkers' translation quality, and then carry out a set of experiments that demonstrate the effectiveness of our model. Each number reported in the experimental results is an average of four numbers, corresponding to the four possible ways of choosing 3 of the 4 reference sets.

For the second group, we apply the state-of-art machine translation system of Joshua system and we also 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 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 translations the one provided by the worker whose translations (over all sentences) score the highest.

We also examine other two voting-inspired methods, since the majority vote usually works

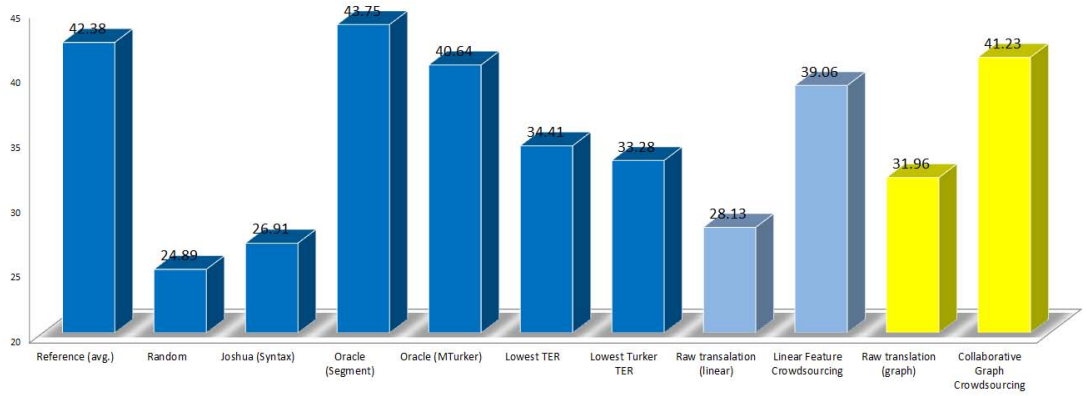

Figure 2: Overall BLEU performance for all methods. The highlighted yellow bars indicate methods based on both HIT and Turker information, with and without post-editing.

well in real world NLP problems. The first selects the translation with the minimum average TER (Snover et al., 2006) against the other translations, since that would be a "consensus" translation. The second method selects the translation generated by the Turkers who provide translations with the minimum average TER.

The last group of methods are based on the utilization of both HIT and Turker information, either based on linear combination or on the graph based random walking. We report the results from the previous crowdsourcing translation system using raw translations and edited translation with all kinds of additional features (Zaidan and Callison-Burch, 2011). Also, we include our proposed collaborative crowdsourcing translation method, based on the co-ranking of HIT and Turker collaboration pairs. For a full comparison, we also include variations of different ranking modelings based on raw translations and edited translations in further discussions.

4.4 Results and Analysis

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 compared with other baselines.

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 fit regressions, compute feature values and tune parameters, and 20% used for evaluation. From the results, we have the following observations:

As expected, random selection yields the worst performance with a BLEU score of 24.89.

Methods	Pearson's r^2
Reference	0.81±0.07
Random	0.63±0.10
Joshua	0.80±0.09
Oracle (Segment)	0.81±0.09
Oracle (Turker)	0.79±0.10
Lowest TER	0.50±0.26
Lowest TER (Turker)	0.48±0.11
Raw Translation (Linear)	0.60±0.17
Linear Combination	0.77±0.11
Raw Translation (Graph)	0.69±0.13
Graph Collaboration	0.80±0.09

Table 2: Correlation (±std. dev.) for different selection methods, compared against the references.

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. On average, evaluating one reference set against the other three gives a BLEU score of 42.38. 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. 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 from the Turkers with the minimum average TER based on all their translations. These approaches achieve BLEU scores of 34.41 and 33.28, respectively.

A raw set of translations without post-editing

scores 28.13 on average based on linear-based combination and 32.96 from on the graph-based ranking, which highlights the loss in quality when collecting translations from amateurs. The linear combination of all features for the crowdsourcing translation achieves a score of 39.06. While the structure information is incorporated into the graph-based ranking framework, the performance achieves a score of 41.23, which verifies the hidden collaboration networks between HIT and Turkers are indeed useful.

Besides, 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 2, 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.

4.5 Analysis

4.5.1 Cost Reduction

The most prominent advantage of crowdsoucing translation would be the low cost to spend. We paid a reward of $0.10 to translate a sentence, $0.25 to edit a set of ten sentences. Therefore, we had the following costs:

- Translation cost: $716.80
- Editing cost: $447.50

(If not done redundantly, those values would be $179.20, $44.75, respectively.) Adding Amazons 10% fee, this brings the grand total to under $1,500, spent to collect 7,000+ translations, 17,000+ edited translations. While the combined cost of our data collection effort is quite low considering the amount of collected data, it is more attractive when the cost could be reduced further without losing much in translation quality by finding more professional non-experts in the Turker graph, and decreasing the amount of translated/edited translations. We indeed improved the performance over the linear combination (regression) based method and reduce costs by ranking Turker graph and HIT graph correspondingly.

4.5.2 Parameter Tuning

The professional translations are used in our approach for measuring the performance against the ground truth and for tuning the weights of the parameters.

There are two parameters in our experimental setups: μ controls the probability to start a new random walk and λ deals with coupling between the HIT and Turker sub-graphs. We set the damping factor μ to 0.15 following the standard PageRank paradigm. We opted for more or less generic parameter values as we did not want to tune our framework to the specific dataset at hand. We examined the parameter λ which controls the balance of the HIT-Turker graph in more detail. We experimented with values ranging from 0 to 0.9, with a step size of 0.1. Small λ values place little emphasis on the coupling part, whereas larger values rely more heavily on the co-ranking. Overall, we observed better performance with values within the range of 0.1-0.4. This suggests that both sources of information for HIT and their authors are important for the crowdsourcing translation task. All our experiments used the same λ value which was set to 0.3.

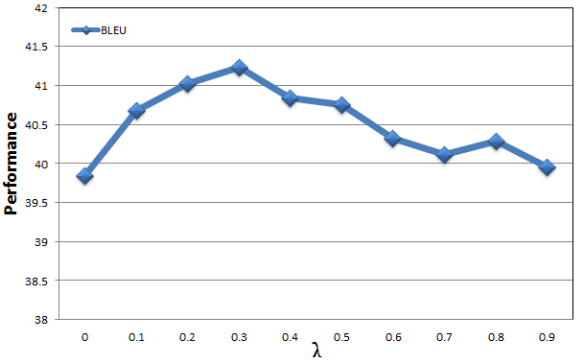

Figure 3: Effect of HIT-Turker coupling by λ.

4.5.3 Component Strategy

We next examine the relative contribution of different strategies when modeling our proposed ranking framework. Specifically, we first examine the standard random walk on the HITs subgraph to see the effect of voting among translated sentences, denoted as *plain ranking*. Then we incorporate the Turker graph to make utilization of the structural information but without any collaboration included, i.e., no edges between Turkers in the Turker graph. Every Turker is associated with the HIT (either translation or post-editing) with authorship while each HIT is associated with two Turkers: the translator and the post-editor. The co-ranking paradigm is exactly the same with the framework mentioned in Section 3.4 with simplified structures.

In the following steps, we examine the two-step collaboration based HIT-Turker graph with several variations of edge establishment: the nodes are the translator/post-editor working pairs. The edges are connected when 1) two nodes share a translator

	w/o prior	w/ prior
Plain ranking	37.88	39.85
w/o collaboration	38.09	40.37
Shared translator	38.95	**41.23**
Shared post-editor	37.62	40.04
Shared Turker	38.59	41.09

Table 3: Variations of all component settings.

only, 2) two nodes share a post-editor only, and 3) two nodes share either a translator or a post-editor. For all of the ranking paradigm, we include the comparison of two versions with and without linguistics quality prior.

As shown in Table 3, we can see consistently incorporating priors to ranking paradigm yield better results than those without linguistic quality priors. An interesting observation is that when modeling the linkage between the collaboration pairs, we note that to establish linkage when the Turker pairs share the translator will achieve the best performance, while to establish linkage when they share the post-editor will lead to a worst result than without the collaborative linkage. It is intuitive to understand that post-editing based on a good raw translation will be more likely to a good selection: the saliency is propagated via the good translator.

5 Related Work

Extraneous data source could always be a supplement to improve MT models so that they are better suited to the low resource setting (Al-Onaizan et al., 2002; Nießen and Ney, 2004). Transfer learning based models, or designing models that are capable of learning translations from monolingual corpora (Schafer and Yarowsky, 2002; Haghighi et al., 2008).

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.

6 Conclusion

We have proposed a two-step non-professional collaboration based co-ranking model to select the best crowdsourced translation on the heterogeneous HIT-Turker graph, and we have demonstrated that compared with a series of MT methods, it is possible to obtain improved performance near professional quality and even less costs from non-professional collaborations.

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. There are possible ways to further improve the crowdsourcing translation and reduce costs by: 1) build ground truth against good Turkers only, instead of professionals, once they have been identified and 2) predict when it is unnecessary to solicit more translations after a certain threshold.

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):1–17.

Vamshi Ambati, Stephan Vogel, and Jaime G Carbonell. 2010. Active learning and crowd-sourcing for machine translation. In *LREC*, volume 11, pages 2169–2174. Citeseer.

Michael Bloodgood and Chris Callison-Burch. 2010. 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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 1–12, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chris Callison-Burch. 2009. 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, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Ulrich Germann. 2001. 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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *ACL*, volume 2008, pages 771–779.

Ann Irvine and Alexandre Klementiev. 2010. Using mechanical turk to annotate lexicons for less commonly used languages. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 108–113, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Ann Irvine, Sanjeev Khudanpur, Lane Schwartz, Wren N. G. Thornton, Ziyuan Wang, Jonathan Weese, and Omar F. 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*, WMT '10, pages 133–137, Stroudsburg, PA, USA. Association for Computational Linguistics.

Annie Louis and Ani Nenkova. 2013. What makes writing great? first experiments on article quality prediction in the science journalism domain. *Transactions of Association for Computational Linguistics*.

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

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

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

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, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Philip Resnik, Olivia Buzek, Chang Hu, Yakov Kronrod, Alex Quinn, and Benjamin B. Bederson. 2010. Improving translation via targeted paraphrasing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 127–137, Stroudsburg, PA, USA. Association for Computational Linguistics.

Charles Schafer and David Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

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, HLT '10, pages 403–411, Stroudsburg, PA, USA. 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*, pages 223–231.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 254–263, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

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

Rui Yan, Mirella Lapata, and Xiaoming Li. 2012a. 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, Xiaojun Wan, Mirella Lapata, Wayne Xin Zhao, Pu-Jen Cheng, and Xiaoming Li. 2012b. Visualizing timelines: Evolutionary summarization via iterative reinforcement between text and image streams. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 275–284, New York, NY, USA. ACM.

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.