Collaborative Workflow for Crowdsourcing Translation

Vamshi Ambati, Stephan Vogel, Jaime Carbonell
Language Technologies Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{vamshi,vogel,jgc}@cs.cmu.edu

ACM Classification Keywords
K.4.3 Organizational Impacts:
Computer-supported collaborative work]

General Terms
Design; Human Factors.

Author Keywords
Language translation, Collaborative workflow, Crowdsourcing, Amazon Mechanical Turk

ABSTRACT
In this paper, we explore the challenges involved in crowdsourcing the task of translation over the web, where remotely located translators work on providing translations independent of each other. We then, propose a collaborative workflow for crowdsourcing translation to address some of these challenges. In our pipeline model, the translators are working in phases where output from earlier phases can be enhanced in the subsequent phases. We also highlight some of the novel contributions of the pipeline model like assistive translation and translation synthesis that can leverage monolingual and bilingual speakers alike. We evaluate our approach by eliciting translations for both a minority-to-majority language-pair and a minority-to-minority language-pair. We observe that in both scenarios, our workflow produces better quality translations in a cost-effective manner, when compared to the traditional crowdsourcing workflow.

INTRODUCTION
Large-scale parallel data generation for new language-pairs requires intensive human effort and availability of experts. For most language-pairs, the paucity of expert translators or lack of access to even bilingual speakers makes it immensely difficult and expensive to build statistical machine translation systems that use large amounts of parallel data to train mathematical models for language translation. Therefore, only a small fraction of the world languages have automatic translation systems. If we can reliably use crowdsourcing for obtaining translations for low-resource language pairs, the cost of building translation systems for most languages can be significantly reduced.

Crowdsourcing, a term that has been popularized recently, is the process of farming out tasks to a large user population on the Internet. These tasks broadly belong to the language or vision community, where for a number of tasks it is either impossible or challenging and time-consuming for computers to complete them, whereas only requires a few seconds for a human to complete. For example, identifying a person in a photograph, tagging a video for a particular event, flagging an email as spam, identifying the sentiment of a written text, spotting characters in an image are still some of the challenging research problems to computers. With the advent of online market places like Amazon's Mechanical Turk[1] and systematic micro-payment mechanisms, crowdsourcing is becoming feasible and easy to conduct.

Recent work has seen a rapid adoption of crowdsourcing and Mechanical Turk for research tasks, especially in the language processing community [9]. In this paper, we conduct all our crowdsourcing experiments on Amazon's Mechanical Turk. Crowdsourcing translation involves posting tasks which are typically one or more sentences in the source language to be translated to a target language. We provide detailed instructions for both completion of the task and its evaluation. Mechanical Turk also has a provision to seek annotations from qualified workers from a specific location with a specific success rate in their past tasks. We set the worker qualification threshold to 85%. Similar to Callison-Burch [4], we post the input sentences as images in order to discourage workers (also called turkers) from copy-pasting the input into existing translation services. In the rest of the paper, we refer to this workflow as a *traditional crowdsourcing workflow*.

However, as the nature of the task grows in complexity it is to be understood that finding large number of users that are skilled to complete the task becomes difficult in the crowd. Language translation is one such example of a cognitive task that spans skill levels in at least two different languages. While most tasks on Mechanical Turk require monolingual speakers, the requirement of bi-linguality drastically reduces the potential size of suitable turkers on a crowdsourcing platform. In this paper, we discuss the problems with the traditional crowdsourcing approach and propose a collaborative workflow that is amenable to crowdsourcing for translation. Our main motivation for the workflow is to involve both bilingual and monolingual speakers to cost-effectively elicit quality data. In particular, we focus on involving 'weak' bilingual

[1] http://www.mturk.com/mturk/

CSCW'12, February 11–15, 2012, Seattle, Washington, USA.

speakers; those that are non-experts and require assistance in completing a full sentence translation. The rest of the paper is organized as follows: We discuss related work in crowdsourcing and translation in Section 2. In Section 3, we discuss in detail some of the challenges with crowdsourcing for translation. Section 4 details our collaborative workflow for translation in the crowd. We conclude with experiments and future work.

RELATED WORK

Crowdsourcing has thus far been explored in the context of monolingual tasks that involve eliciting a discrete annotation label. Less has been explored to date for analyzing the feasibility of Mechanical Turk for eliciting structural annotations; for example, a summary of an article, translation of a sentence into another language, sequential labeling, syntactic analysis of a sentence, etc. Some recent research has looked at obtaining translations via crowdsourcing, in particular for low-resource languages [1]. Callison-Burch [4] has explored feasibility studies for evaluating the translation output. In both these studies, authors have assumed availability of bilingual speakers for translating a sentence and follow the workflow of eliciting multiple sentence translations independently from users. Recently, researchers have also looked at protocols to involve monolingual speakers collaborating with a translation system to produce translations [5]. Their protocol, however, requires the availability of an intermediary translation system for both the languages. Truly low-resource language pairs with no seed data may not be candidates for such a workflow.

Aside translation, there is general work in the area of designing collaborative workflows for effective crowdsourcing. Bernstein et al.[3] discuss a word processor that can crowdsource and involve humans for completing writing tasks. They introduce an interaction pattern called "find-fix-verify" for integrating crowdsourced human contributions directly into user interfaces. Little et al.[6] discuss a tool to perform iterative human computation algorithms on Mechanical Turk. Crowdforge is a framework for performing complex tasks in a crowdsourced manner [2].

CHALLENGES IN CROWDSOURCING TRANSLATION

In particular, the following three aspects make the task of translation using the crowd more challenging.

- **Large label space:** The input, a sentence in a source language, can be translated into some finite number of meaning-preserving sentences in the target language and a very large number of invalid translations under the target language vocabulary. With such an indefinitely large output space of possible annotations for any given input, it becomes difficult to evaluate the translation quality. Even with the availability of a set of gold standard translations, it is difficult to evaluate the quality of an annotator due to natural variations in the style of their translations which could result in a lexically different, yet valid translation. As an example, consider the translations into English provided by three different turkers which look different at the word-level, but preserve the meaning.

  Spanish: Me alegra mucho que hayas podido venir

  - I am so glad you could come
  - I 'm very happy you were able to come
  - I am glad you could make it

- **Availability:** The availability of bilingual speakers for translation is relatively more difficult when the languages are low-resource or have low-presence on the Web (e.g. Urdu, Thai, etc) than it is to find for high-resource languages (e.g. Chinese, Arabic, Spanish, etc). This also makes it difficult to seek multiple translations, for purposes of quality assurance.

- **Low quality:** Available translators for most languages are non-linguist bilingual speakers and therefore the quality of translation is typically low. Unlike tagging images or flagging spam, translation is a complex cognitive task that requires specific language skills like reading, understanding and writing in multiple languages.

- **Cost:** One of the approaches to getting quality data in crowdsourcing is repeated labeling. In the case of translation, repeatedly obtaining translation for an entire sentence in the wake of high costs and unavailability of translators becomes difficult and less appealing.

COLLABORATIVE WORKFLOW

In this section, we discuss the methodology and the three phases of our collaborative workflow. The desired characteristics of our collaborative workflow are three-fold:

- **Verifiable**: We want to improve the verifiability of crowdsourcing for complex outputs like in the case of translation, by breaking down the complex task into meaningful subtasks. For example, while it is difficult for multiple translators to agree upon a sentence translation, consensus can be reached upon when translating at word level, which may in turn be used to check validity of sentence translations.

- **Diversity**: Along with bilingual speakers, we want users of monolingual nature also to be part of our workflow, as it is relatively easier to find the latter. For example, while there are more than a billion Chinese speakers, only a very small portion of them may be able to translate from English into Chinese.

- **on-experts**: We want our workflow to not only be robust to low quality inputs, but also be able to assist non-expert translators in providing better translations. Bilinguals, efficient in translation of the entire sentences are few in number, but a major portion of speakers can translate individual words with high accuracy.

Figure 1 shows our collaborative translation pipeline. We conduct our workflow on Mechanical Turk as well, but unlike unlike the traditional workflow for crowdsourcing translation, here the translators are working in phases where the output from earlier phases is enhanced in the subsequent phases.

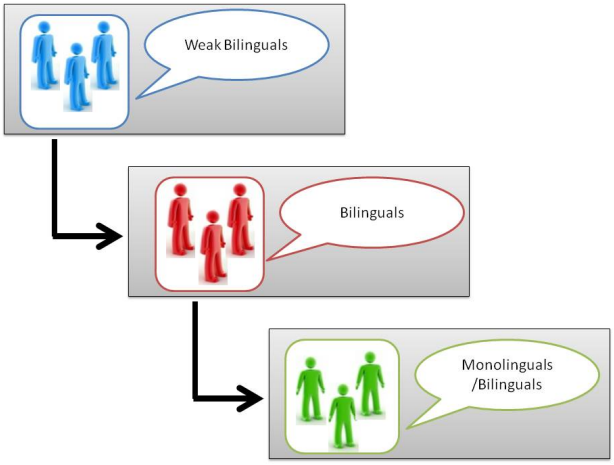

Figure 1. Our three phased collaborative workflow for translating in the crowd - 1. Lexical translation , 2. Assistive translation, 3. Monolingual synthesis

Phase 1: Context-sensitive Lexical Translation

In the first phase, we focus on translations at the word/phrase level. We first, identify content words in the input sentence and crowdsource for collecting their translations. This task can be repeated a large number of times as it is cheaper to translate a single word than an entire sentence. Unlike translation at the sentence level, this can also be verified easily by a simple lexical comparison.

We designed the task so that the user can also see the sentence that the word is in and translate the word in its context. We also observe that turkers converge on a translation much faster when required to translate within the context of an input sentence than when asked to translate out-of-context.

Phase 2: Assistive Translation by Weak Bilinguals

In the second phase, we now collect complete sentence level translations. The turker is required to translate the entire sentence into a target language by preserving the meaning. However, in this phase, we require the translators to use vocabulary gathered from Phase 1 in order to translate certain words in the sentence. As part of a post-verification process, we ensure that the turkers indeed use one of the potential translations for the words in the sentence. As translating a word is not an expensive task when compared to entire sentence translation, we can repeat the word translation task more number of times until reach a sufficient level of inter-translator agreement and only then, proceed to Phase 2.

We observed that breaking the translation task into two different phases enables us to not only control spammers and increase verifiability of the task, but also engage non-expert translators, who may require some guidance in completing a translation.

Phase 3: Target Synthesis by Monolingual Speakers

In the final phase we do not require bilingual speakers, but only monolingual speakers of the target language. The task in this phase is to construct a new translation by synthesizing a translation from among the multiple translations produced in Phase 2. We also allow for post-editing of the translation for spelling and grammar errors.

For example, consider the multiple translations for the Spanish sentence below. We observe that typically there are missing words, mis-spelt words, non-translated words and overall incorrect grammar. We also notice that while there is no evidently better translation among the multiple translations, a meaningful and a complete translation can be synthesized from them. This is similar in spirit to multi-engine machine translation [7].

Spanish: lo tomar desde la parada de taxis

- i'll climb it from the taxi stop
- i'll take it from the taxi rank
- i will have it from the taxi rank

In this phase, a turker is only shown the multiple translations from Phase 2 and is required to guess and select the correct translation. Redundancy among the translations gives sufficient evidence to even a monolingual speaker to guess the correct intent of the source sentence, although he does not speak the source language. Once the intent and context of the sentence is understood, the turker can either select the best suitable translation or synthesize a new translation from the alternatives provided. One can also obtain multiple translations in this phase, although we observe that usually a single solution is sufficient as the monolingual speakers do a good job of constructing the right sentence from the alternatives.

EVALUATION

Baseline

Our baseline is the traditional setup for crowdsourcing translation as described in the introduction section, where sentences were translated by independently working turkers and a majority agreement was conducted to select the best translation. For the baseline, we obtained translations from 5 different turkers and, similar to Ambati et al.[1], use a fuzzy matching algorithm for comparing two sentences and computing majority agreement. The fuzzy matcher has an internal aligner that matches words in the sentences given and scores them separately based on whether the match was supported by the exact match or the fuzzy match. The scores are then combined to provide a global matching score. If the score is above a threshold, we treat the sentences to be equivalent translations of the source sentence.

Preliminary Results

We obtain translations for two language pairs using our workflow and compare it with the traditional crowdsourcing workflow. Firstly we pick Telugu-English language pair, where the source language is a minority and the target language is a majority language. This represents a scenario where availability of bilingual speakers is scarce, but obtaining users for target language is quite easier. We also try a new language pair Telugu-Hindi, where it is extremely difficult to find experts that speak both the languages, although it is relatively

Method	Language	BLEU
Baseline	Telugu-English	21.22
Collaborative	Telugu-English	27.82
Baseline	Telugu-Hindi	18.91
Collaborative	Telugu-Hindi	20.9

Table 1. Evaluation of quality under different workflows

easier to find weak bilingual speakers, who, given enough word level assistance, can perform a decent job of translation. The target language, Hindi, is linguistically a richer language than English, with greater scope for grammatical errors due to gender, number inflections on nouns and verbs. For both language-pairs we translated 100 sentences each using both workflows and compared with the available gold-standard translations that were obtained from experts. As shown in Table 1, we compare the gold-standard match using automatic translation evaluation metric: BLEU [8].

From Table 1, we can see that the collaborative workflow proposed in this paper performs much better for obtaining translations in a crowdsourcing paradigm, when compared to a traditional crowdsourcing setup of farming the task to multiple independent turkers. We also observe that our collaborative workflow enables quicker turn-around time for translations as it fosters participation of weak-bilinguals and monolingual speakers, which is a greater portion of the population than pure bilingual speakers.

Cost Factor

When working with crowd data we would like to extract annotations that are of highest possible quality using non-experts. This is typically achieved by requesting annotations from multiple turkers and computing agreement statistics, which helps in accounting for natural variability of translators and reduces human error. However, repeating a task multiple times comes at a cost, and in this section we argue that our collaborative workflow while providing better quality and faster turn-around times, is still cost-effective.

For instance the cost of obtaining translations from the crowd using baseline approach for 100 sentences was $100 * (5 * 0.10) = 50$USD, where each sentence was translated by five different turkers for 10 cents each. The cost for our three staged process was $100 * (5 * 5 * 0.01 + 3 * 0.05 + 3 * 0.02) = 41$USD , where the first phase of word translations was done by five turkers, sentence translation by three and monolingual synthesis by three turkers at prices of 1 cent per word (avg 5 words per sentence), 5 cents per sentence, 2 cents per monolingual repair respectively. We see a 20% savings in cost when using our workflow as compared to the baseline workflow.

CONCLUSION

Crowdsourcing is increasingly being adopted by researchers in order to elicit annotations over the Internet. In this paper we proposed a novel collaborative workflow for language translation. Our three-phase workflow involves breaking the atomic task of sentence translations into three stages - word translation, assisted sentence translation and translation synthesis. We showed that collecting translations using our collaborative workflow has several advantages over the traditional crowdsourcing approach of independently obtaining multiple translations. We evaluated our approach on two language pairs and showed that the overall quality of translations has improved at lowered costs.

Acknowledgement

This material is based upon work supported by, or in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract number W911NF-10-1-0533.

REFERENCES

1. V. Ambati, S. Vogel, and J. Carbonell. Active learning and crowd-sourcing for machine translation. In *Proceedings of the LREC 2010*, Malta, May 2010.
2. R. E. K. Aniket Kittur, Boris Smus. Crowdforge: Crowdsourcing complex work. Technical report, Human-Computer Interaction Institute, SCS, Carnegie Mellon University, January 2011.
3. M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23rd UIST*, UIST '10, pages 313–322. ACM, 2010.
4. C. Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *EMNLP 2009*, pages 286–295, Singapore, August 2009. Association for Computational Linguistics.
5. C. Hu, B. B. Bederson, and P. Resnik. Translation by iterative collaboration between monolingual users. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 54–55. ACM, 2010.
6. G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Turkit: tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09, pages 29–30, New York, NY, USA, 2009. ACM.
7. S. Nirenburg and R. Frederking. Toward multi-engine machine translation. In *HLT '94: Proceedings of Human Language Technology*, pages 147–151, Morristown, NJ, USA, 1994. ACL.
8. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*, pages 311–318, Morristown, NJ, USA, 2002.
9. R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the EMNLP 2008*, pages 254–263, Honolulu, Hawaii, October 2008.