TransDoop: A Map-Reduce based Crowdsourced Translation for Complex Domains

Anoop Kunchukuttan*, Rajen Chatterjee*, Shourya Roy†, Abhijit Mishra*, Pushpak Bhattacharyya*

* Department of Computer Science and Engineering, IIT Bombay,
{anoopk,abhijitmishra,pb}@cse.iitb.ac.in, rajen.k.chatterjee@gmail.com

† Xerox India Research Centre,
Shourya.Roy@xerox.com

Abstract

Large amount of parallel corpora is required for building Statistical Machine Translation (SMT) systems. We describe the *TransDoop* system for gathering translations to create parallel corpora from online crowd workforce who have familiarity with multiple languages but are not expert translators. Our system uses a Map-Reduce-like approach to translation crowdsourcing where sentence translation is decomposed into the following smaller tasks: (a) translation of constituent phrases of the sentence; (b) validation of quality of the phrase translations; and (c) composition of complete sentence translations from phrase translations. *TransDoop* incorporates quality control mechanisms and easy-to-use worker user interfaces designed to address issues with translation crowdsourcing. We have evaluated the crowd's output using the METEOR metric. For a complex domain like judicial proceedings, the higher scores obtained by the map-reduce based approach compared to complete sentence translation establishes the efficacy of our work.

1 Introduction

Crowdsourcing is no longer a new term in the domain of Computational Linguistics and Machine Translation research (Callison-Burch and Dredze, 2010; Snow et al., 2008; Callison-Burch, 2009). Crowdsourcing - basically where task outsourcing is delegated to a largely unknown Internet audience - is emerging as a new paradigm of *human in the loop* approaches for developing sophisticated techniques for understanding and generating natural language content. *Amazon Mechanical Turk(AMT)* and *CrowdFlower* [1] are representative general purpose crowdsourcing platforms where as *Lingotek* and *Gengo*[2] are companies targeted at localization and translation of content typically leveraging freelancers.

Our interest is towards developing a crowdsourcing based system to enable general, non-expert crowd-workers generate natural language content equivalent in quality to that of expert linguists. Realization of the potential of attaining great scalability and cost-benefit of crowdsourcing for natural language tasks is limited by the ability of novice multi-lingual workers generate high quality translations. We have specific interest in Indian languages due to the large linguistic diversity as well as the scarcity of linguistic resources in these languages when compared to European languages. Crowdsourcing is a promising approach as many Indian languages are spoken by hundreds of Millions of people (approximately, Hindi-Urdu by 500M, Bangla by 200M, Punjabi by over 100M [3]) coupled with the fact that representation of Indian workers in online crowdsourcing platforms is very high (close to 40% in Amazon Mechanical Turk (AMT)).

However, this is a non-trivial task owing to lack of expertise of novice crowd workers in translation of content. It is well understood that familiarity with multiple languages might not be good enough for people to generate high quality translations. This is compounded by lack of sincerity and in certain cases, dishonest intention of earning rewards disproportionate to the effort and time spent for online tasks. Common techniques for quality control like *gold data based validation* and *worker reputation* are not effective for a subjective task

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

[2] http://www.lingotek.com, http:///www.gengo.com

[3] http://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers

like translation which does not have any task specific measurements. Having expert linguists manually validate crowd generated content defies the purpose of deploying crowdsourcing on a large scale.

In this work, we propose a technique, based on the *Divide-and-Conquer* principle. The technique can be considered similar to a Map-Reduce task run on *crowd processors*, where the translation task is split into simpler tasks distributed to the crowd (the *map* stage) and the results are later combined in a *reduce* stage to generate complete translations. The attempt is to make translation tasks easy and intuitive for novice crowd-workers by providing translations aids to help them generate high quality of translations. Our contribution in this work is a end-to-end, crowdsourcing-platform-independent, translation crowdsourcing system that completely automates the translation crowdsourcing task by (i) managing the translation pipeline through software components and the crowd; (ii) performing quality control on workers' output; and (iii) interfacing with crowdsourcing service providers. The multi-stage, Map-reduce approach simplifies the translation task for crowd workers, while novel design of user interface makes the task convenient for the worker and discourages spamming. The system thus offers the potential to generate high quality parallel corpora on a large scale.

We discuss related work in Section 2 and the multi-staged approach which is central to our system in Section 3. Section 4 describes the system architecture and workflow, while Section 5 presents important aspects of the user interfaces in the system. We present our preliminary experiments and observations in Section 6. Section 7 concludes the paper, pointing to future directions.

2 Related Work

Lately, crowdsourcing has been explored as a source for generating data for NLP tasks (Snow et al., 2008; Callison-Burch and Dredze, 2010). Specifically, it has been explored as a channel for collecting different resources for SMT - evaluations of MT output (Callison-Burch, 2009), word alignments in parallel sentences (Gao et al., 2010) and post-edited versions of MT output (Aikawa et al., 2012). Ambati and Vogel (2010), Kunchukuttan et al. (2012) have shown the feasibility of crowdsourcing for collecting parallel corpora and pointed out that quality assurance is a major issue for successful translation crowdsourcing.

The most popular methods for quality control of crowdsourced tasks are based on sampling and redundancy. For translation crowdsourcing, Ambati et al. (2010) use inter-translator agreement for selection of a good translation from multiple, redundant worker translations. Zaidan and Callison-Burch (2011) score translations using a feature based model comprising sentence level, worker level and crowd ranking based features. However, automatic evaluation of translation quality is difficult, such automatic methods being either inaccurate or expensive. Post et al. (2012) have collected Indic language corpora data utilizing the crowd for collecting translations as well as validations. The quality of the validations is ensured using gold-standard sentence translations. Our approach to quality control is similar to Post et al. (2012), but we work at the level of phrases.

While most crowdsourcing activities for data gathering has been concerned with collecting simple annotations like relevance judgments, there has been work to explore the use of crowdsourcing for more complex tasks, of which translation is a good example. Little et al. (2010) propose that many complex tasks can be modeled either as iterative workflows (where workers iteratively build on each other's works) or as parallel workflows (where workers solve the tasks in parallel, with the best result voted upon later). Kittur et al. (2011) suggest a map-and-reduce approach to solve complex problems, where a problem is decomposed into smaller problems, which are solved in the *map* stage and the results are combined in the *reduce* stage. Our method can be seen as an instance of the map-reduce approach applied to translation crowdsourcing, with two map stages (phrase translation and translation validation) and one reduce stage (sentence combination).

3 Multi-Stage Crowdsourcing Pipeline

Our system is based on a multi-stage pipeline, whose central idea is to simplify the translation task into smaller tasks. The high level block diagram of the system is shown in Figure 1. Source language documents are sentencified using standard NLP tokenizers and sentence splitters. Extracted sentences are then split into phrases using a standard chunker and rule-based merging of small chunks. This step creates small phrases

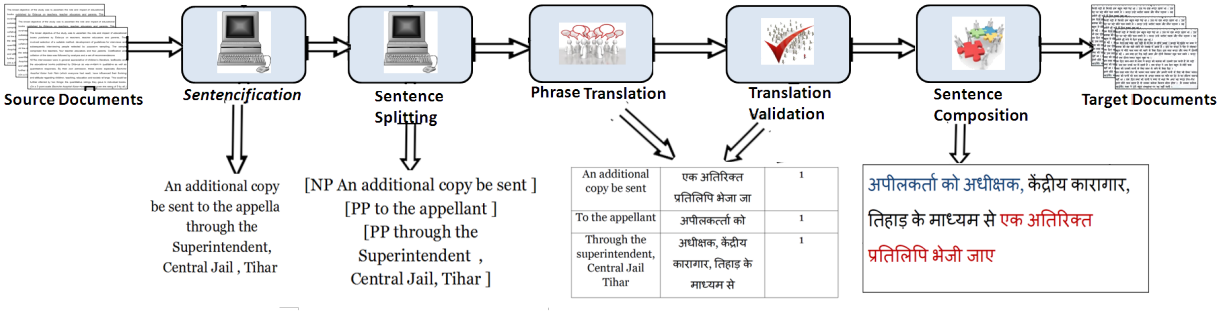


Figure 1: Multistage crowdsourced translation

from complex sentences which can be easily and independently translated. This leads to a crowdsourcing pipeline, with three stages of tasks for the crowd: Phrase Translation (PT), Phrase Translation Validation (PV), Sentence Composition (SC). A group of crowd workers translate source language phrases, the translations are validated by a different group of workers and finally a third group of workers put the phrase translation together to create target language sentences. The validation is done by workers by providing ratings on a k-point scale. This kind of divide and conquer approach helps to tackle the complexity of crowdsourcing translations since: (1) the tasks are simpler for workers; (2) uniformity of smaller tasks brings about efficiency as in any industrial assembly line; (3) pricing can be controlled for each stage depending on the complexity; and (4) quality control can be performed better for smaller tasks.

4 System Architecture

Figure 2 shows the architecture of *TransDoop*, which implements the 3-stage pipeline. The major design considerations were: (i) translation crowdsourcing pipeline should be independent of specific crowdsourcing platforms; (ii) support multiple crowdsourcing platforms; (iii) customize job parameters like pricing, quality control method and task design; and (iv) support multiple languages and domains.

The core component in the system is the **Crowdsourcing Engine**. The engine manages the execution of the crowdsourcing pipeline, lifecycle of jobs and quality control of submitted tasks. The Engine exposes its capabilities through the **Requester API**, which can be used by clients for setting up, customizing and monitoring translation crowdsourcing jobs and controlling their execution. These capabilities are made available to requesters via the **Requester Portal**. In order to make the crowdsourcing engine independent of any specific crowdsourcing platform, platform specific **Connectors** are developed. The Crowdsourcing system makes the tasks to be crowdsourced available through the **Connector API**. The connectors are responsible for polling the engine for tasks to be crowdsourced, pushing the tasks to crowdsourcing platforms, hosting worker interfaces for the tasks and pushing the results back to the engine after they have been completed by workers on the crowdsourcing platform. Currently the system supports the AMT crowdsourcing platform.

Figure 3 depicts the lifecycle of a translation crowdsourcing job. The requester initiates a translation job for a document (a set of sentences). The Crowdsourcing Engine schedules the job for execution. It first splits each sentence into phrases. For the job, PT tasks are created and made available through the Connector API. The connector for the specified platform periodically polls the Crowdsourcing Engine via the Connector API. Once the connector has new PT tasks for crowdsourcing, it interacts with the crowdsourcing platform to request crowdsourcing services. The connector monitors the progress of the tasks and on completion provides the results and execution status to the Crowdsourcing Engine. Once all the PT tasks for the job are completed, the crowdsourcing Engine initiates the PV task to obtain validations for the translations. The Quality Control system kicks in when all the PV tasks for the job have been completed.

The quality control (QC) relies on a combination of sampling and redundancy. Each PV task has a few gold-standard phrase translation pairs, which is used to ensure that the validators are honestly doing their tasks. The judgments from the

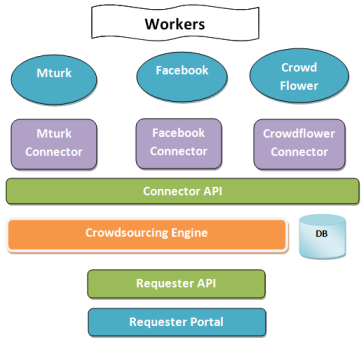


Figure 2: Architecture of *TransDoop*

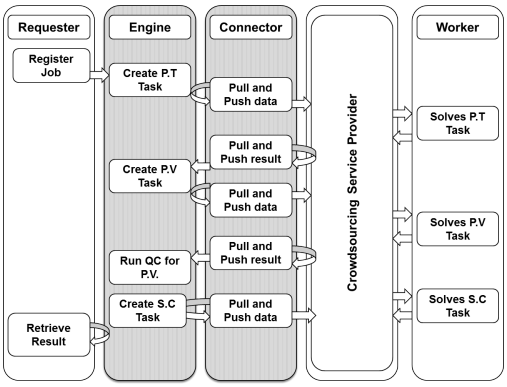


Figure 3: Lifecycle of a Translation Job

good validators are used to determine the quality of the phrase translation, based on majority voting, average rating, etc. using multiple judgments collected for each phrase translation. If any phrase validations or translations are incorrect, then the corresponding phrases/translations are again sent to the PT/PV stage as the case may be. This will continue until all phrase translations in the job are correctly translated or a pre-configured number of iterations are done.

Once phrase translations are obtained for all phrases in a sentence, the Crowdsourcing Engine creates SC tasks, where the workers are asked to compose a single correct, coherent translation from the phrase translation obtained in the previous stages.

5 User Interfaces

5.1 Worker User Interfaces

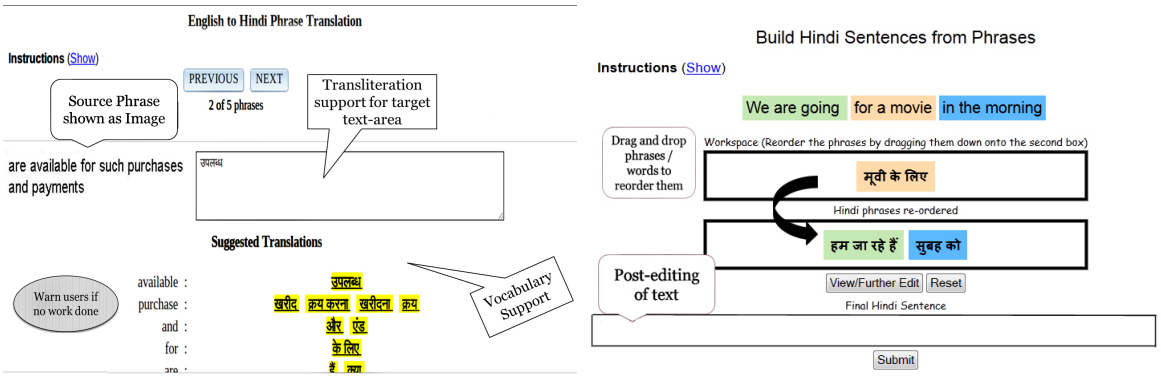
This section describes the worker user interfaces for each stage in the pipeline. These are managed by the **Connector** and have been designed to make the task convenient for the worker and prevent spam submissions. In the rest of the section, we describe the salient features of the PT and SC UI's. PV UI is similar to k-scale voting tasks commonly found in crowdsourcing platforms.

- **Translation UI:** Figure 4a shows the translation UI for the PT stage. The user interface *discourages spamming* by: (a) displaying source text as images; and (b) alerting workers if they don't provide a translation or spend very little time on a task. The UI also provides *transliteration support* for non-Latin scripts (especially helpful for Indic scripts). A *Vocabulary Support*, which shows translation suggestions for word sequences appearing in the source phrase, is also available. Suggested translations can be copied to the input area with ease and speed.

- **Sentence Translation Composition UI:** The sentence translation composition UI (shown in Figure 4b) facilitates composition of sentence translations from phrase translations. First, the worker can drag and rearrange the translated phrases into the right order, followed by reordering of individual words. This is important because many Indian languages have different constituent order (S-O-V) with respect to English (S-V-O). Finally, the synthesized language sentence can be post-edited to correct spelling, case marking, inflectional errors, etc. The system also captures the reordering performed by the worker, an important byproduct, which can be used for training reordering models for SMT.

5.2 Requester UI

The system provides a Requester Portal through which the requester can create, control and monitor jobs and retrieve results. The portal allows the requester to customize the job during creation by configuring various parameters: (a) domain and language pair (b) entire sentence vs multi-stage translation (c) price for task at each stage (d) task design (number of tasks in a task group, etc.) (e) translation redundancy (f) validation quality parameters. *Translation redundancy* refers to the number of translations requested for a source phrase. *Validation redundancy* refers to the number of validations collected for each phrase translation pair and the redundancy based acceptance criteria for phrase translations (majority, consensus, threshold, etc.)



(a) Phrase Translation UI

(b) Sentence Composition UI

Figure 4: Worker User Interfaces

6 Experiments and Observations

Using *TransDoop*, we conducted a set of small-scale, preliminary translation experiments. We obtained translations for English-Hindi and English-Marathi language pairs for the Judicial and Tourism domains. For each experiment, 15 sentences were given as input to the pipeline. For evaluation, we chose METEOR, a well-known translation evaluation metric (Banerjee and Lavie, 2005). We compared the results obtained from the crowdsourcing system with a expert human translation and the output of Google Translate. We also compared two expert translations using METEOR to establish a skyline for the translation accuracy. Table 1 summarizes the results of our experiments.

The translations with Quality Control and multistage pipeline are better than Google translations and translations obtained from the crowd without any quality control, as evaluated by METEOR. Multi-stage translation yields better than complete sentence translation. Moreover, the translation quality is comparable to that of expert human translation. This behavior is observed across the two language pairs and domains. This can be seen in some examples of crowdsourced translations obtained through the system which are shown in Table 2.

Incorrect splitting of sentences can cause difficulties in translation for the worker. For instance, discontinuous phrases will not be available to the worker as a single translation unit. In the English interrogative sentence, the noun phrase splits the verb phrase, therefore the auxiliary and main verb could be in different translation units. *e.g.*

*Why **did** you **buy** the book?*

In addition, the phrase structures of the source and target languages may not map, making translation difficult. For instance, the *vaala* modifier in Hindi translates to a clause in English. It does not contain any tense information, therefore the tense of the English clause cannot be determined by the worker. *e.g.*

***Lucknow vaalaa** ladkaa*

could translate to any one of:

*the boy **who lives/lived/is living in Lucknow***

We rely on the worker in sentence composition stage to correct mistakes due to these inadequacies and compose a good translation. In addition, the worker in the PT stage could be provided with the sentence context for translation. However, there is a tradeoff between the cognitive load of context processing versus uncertainty in translation. More elaborately, to what extent can the cognitive load be reduced before uncertainty of translation sets in? Similarly, how much of context can be shown before the cognitive load becomes pressing?

7 Conclusions

In this system demonstration, we present *TransDoop* as a translation crowdsourcing system which has the potential to harness the strength of the crowd to collect high quality human translations on a large scale. It simplifies the tedious translation tasks by decomposing them into several "easy-to-solve" subtasks while ensuring quality. Our evaluation on small scale data shows that the multistage approach performs better than complete sentence translation. We would like to extensively use this platform for large scale experiments on more language pairs and complex domains like Health, Parliamentary Proceedings, Technical and Scientific literature etc. to establish the utility of

Language Pair	Domain	Google Translate	No QC	Translation with QC		Reference Human
				single stage	*multi stage*	
en-mr	Tourism	0.227*	0.30	0.368	0.372	0.48
en-hi	Tourism	0.292	0.363	0.387	0.422	0.51
en-hi	Judicial	0.252	0.30	0.388	0.436	0.49

Table 1: Experimental Results: Comparison of METEOR scores for different techniques, language pairs and domains
*Translated by an internal Moses-based SMT system

Accordingly the penalty imposed by AO is not justified and the same is cancelled.
इसके अनुसार ए औ द्वारा लगाये गये दंड उचित नहीं है और एक ही रद्द कर दिया है
Accordingly A O by imposed penalty justified not is and one also cancel did
तदानुसार ए ओ द्वारा लगाया गया दंड जायज नहीं है और उसे रद्द कर दिया है
Accordingly A O by imposed penalty justified not is and that cancel did

(a) English-Hindi Judicial Translation

A crowd of devotees engulf Haridwar during the time of daily prayer in the evening
शाम में दैनिक प्रार्थना के समय के दौरान भक्तों को अपनी चपेट में ले हरिद्वार की भीड
evening in daily prayer of time during devotees its engulf in take Haridwar of crowd
श्रद्धालुओं की भीड शाम में दैनिक प्रार्थना के समय हरिद्वार को अपनी चपेट में लेती है
devotees of crowd evening in daily prayer of time haridwar its engulf in take

(b) English-Hindi Tourism Translation

Table 2: Examples of translation from Google and three staged pipeline for source sentence (2^{nd}, 3^{rd} and 1^{st} rows of each table respectively). Domains and languages are indicated above.

the method for collection of parallel corpora on a large scale.

References

Takako Aikawa, Kentaro Yamamoto, and Hitoshi Isahara. 2012. The impact of crowdsourcing post-editing with the collaborative translation framework. In *Advances in Natural Language Processing*. Springer Berlin Heidelberg.

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

Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. 2010. Active learning and crowd-sourcing for machine translation. *Language Resources and Evaluation LREC*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

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*.

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*.

Qin Gao, Nguyen Bach, and Stephan Vogel. 2010. A semi-supervised word alignment algorithm with partial manual alignments. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*.

Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*.

Anoop Kunchukuttan, Shourya Roy, Pratik Patel, Kushal Ladha, Somya Gupta, Mitesh Khapra, and Pushpak Bhattacharyya. 2012. Experiences in resource generation for machine translation through crowdsourcing. *Language Resources and Evaluation LREC*.

Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. 2010. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD workshop on human computation*.

Matt Post, Chris Callison-Burch, and Miles Osborne. 2012. Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*.

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*.

Omar Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.